



POLITECNICO

MILANO 1863

POLITECNICO DI MILANO
Master of Computer Science and Engineering
Department of Electronics, Information and Bioengineering

I.DRIVE: Correlation analysis of driver, vehicle, and environment interactions

AIR Lab
Artificial Intelligence & Robotics Lab
Politecnico di Milano

Advisor: Prof. Matteo Matteucci
Co-advisor: Alessandro Gabrielli

Master Thesis:
Gabriel Esteban Manzoni
Matricola 863415

Academic Year 2019-2020

A mia figlia Sofia

Contents

Contents	V
Abstract	XVII
Sommario	XIX
Acknowledgments	XXI
1 Introduction	2
1.1 Motivation	2
1.2 Contributions	4
1.3 Thesis outline	6
2 State of art	7
2.1 Autonomous driving	8
2.1.1 Brief History of Autonomous Cars	9
2.1.2 Contemporary progresses	12
2.1.3 Do people trust autonomous cars?	13
2.1.4 Autonomous driving: driving styles	16
2.2 Physiological indexes of stress	20
2.2.1 What is exactly the stress?	21
2.2.2 What happens to human body in a stressful situation?	22
2.2.3 Heart Beat Rate	22
2.2.4 Galvanic Skin Response	25
2.2.5 Analysis of Physiological indexes	26
2.3 Correlation algorithms	27
3 I.DRIVE Hardware	29
3.1 I.DRIVE Vehicle	29
3.2 Environmental sensors	30
3.2.1 LiDAR	30

3.2.2	GPS receiver	31
3.2.3	IMU	32
3.2.4	External cameras	33
3.3	Physiological sensors	33
3.3.1	Procomp Infiniti	33
3.3.2	Eye tracker	34
3.4	On-board Computer	35
4	Software Architecture	36
4.1	Integration and acquisition framework (ROS)	37
4.2	Big Data storage (MongoDB)	38
4.3	Scripting and processing	40
4.4	Synchronization	40
4.5	System performance	42
5	Processing pipeline	45
5.1	Pipeline	45
5.2	Data preparation	47
5.3	Data processing	52
5.3.1	Extraction of Event Sequence	54
5.4	Algorithms definition	57
5.4.1	Pearson Correlation Coefficient	59
5.4.2	ARMA Model	60
5.4.3	Nearest Neighbour Statistical Based Method	61
5.4.4	Energy Test	62
5.5	Algorithms comparison	63
5.5.1	Accuracy	63
5.5.2	Computational complexity	65
5.6	Correlation extraction	66
6	Results	75
6.1	Experimental setup	75
6.2	Acquisition	77
6.3	Time Characteristics Analysis	78
6.3.1	Low Frequency	78
6.3.2	High Frequency	79
6.3.3	Low Frequency High Frequency Ratio	80
6.3.4	HRV	80
6.3.5	Tonic Component	82
6.3.6	Phasic Component	82

6.4	Physiological Indexes Analysis	84
6.4.1	Phasic Component	84
6.4.2	HRV	85
6.5	Conclusions	88
7	Conclusions and future works	91
7.1	Future works	92
	Bibliography	93
A	Software installation	100
A.1	Requirements	100
A.2	Python modules	100
A.3	Download scripts	101
A.4	MongoDB	101
A.5	ROS	101
B	Programming documentation	102
B.1	PCC	102
B.2	ET	102
B.3	ARMA	103
B.4	NNSBM	104
C	Manual	105
C.1	Params.py	105
C.2	Extract_mongo_collections.py	107
C.3	Extract_velodyne.py	108
C.4	Extraction_features.py	109
C.5	Correlation_extraction.py	109
D	Questionnaire	111

List of Figures

1.1	Tazzari Zero used for I.DRIVE project	5
2.1	First AC prototypes: (a) ALV (b) VaMP (c) Vita-2 (d) ARGO	10
2.2	Modern AC: (a) Google car prototitpe (b) Ford AC (c) Lexus model in Google car project (d) Uber AC	11
2.3	Key literature and factors influencing the adoption of ACs[22]	14
2.4	Mean Defensiveness score across participants in Basu’s exper- iment[31]	18
2.5	Lever used by passenger throughout autonomous drive to in- dicate the level of comfort in Scherer’s study[39]	20
2.6	General structure of the transformation of manual driving parameters into optimized automated driving parameters[39]	20
2.7	Illustration of a possible combination of sympathetic and parasym- pathetic activity of LFHF[43]	24
2.8	Skin conductance measurement locations: 1)fingers, 2) distal wrist, 3) central wrist, 4) vertical wrist, 5) chest, 6) foot (in- step), 7) calf, 8) forehead, 9) neck, 10) shoulders,11) back, 12) buttock, 13) abdomen, 14) armpit, 15) upper arm, and 16) thighbone. [45]	25
2.9	Results for each of 16 measurement locations [45]	26
2.10	SC as sum of PC (green) and TC (yellow) [46]	27
2.11	Example of temporal order between Time-Serie CPU Usage and Temporal Event System Event [49]	27
2.12	Example of front and rear sub-series [49]	28
3.1	Tazzari Zero without external sensors	30
3.2	An Example of Point cloud(b) of Velodyne HDL-32E(a)	30
3.3	Internal sensors of Velodyne HDL-32E: IMU(a) and GPS(b) .	31
3.4	GPS Yuan10 with Skytraq S1315F-RAW receiver	32
3.5	IMU Xsens MTi with fixed reference system	32
3.6	Prosilica GC1020/650 (a) with Theia wide-angle optics (b) .	33

3.7	Procomp Infiniti (a) with a set of available sensors (b) top from left: EMG, ECG, EEG and SC. Bottom from left: body temperature, breathing and HBR	34
3.8	Pupil Labs [®] eye tracker	35
3.9	Shuttle Slim	35
4.1	Pipeline of information flow from Sensors to MongoDB Database	36
4.2	Example of RVIZ window with 4 cameras, on the left, and point cloud	38
4.3	MongoDB data structure organization	39
4.4	MongoDB document example	39
4.5	Dependencies graph of the order of the scripts called by Bash <i>data_extraction_for_correlation.sh</i> : dashed arrows indicate calls that do not prevent the flow if fail or absent, solid lines indicate calls that would block the flow if a failure happens, blank boxes represent data's file and coloured boxes represent scripts.	41
4.6	Hierarchical structure of NTP	42
4.7	Architecture of PTP	43
4.8	Increase of occupied space over time with linear interpolation: $y = x * 2.2 + 3.7$	44
5.1	Pipeline of processes from MongoDB Database to DataFrame with extracted data	45
5.2	Diagram of all steps from Environmental and Physiological Data to Extracted probabilities	46
5.3	Structure of SECOND detector [57]	47
5.4	Axis position	48
5.5	Relative position of identified objects, defining front, back, right and left position of acquisition on 2018/12/13	49
5.6	Distance of nearest identified object for category of acquisition on 2018/12/13	50
5.7	Number of identified objects in each frame for category of acquisition on 2018/12/13	51
5.8	Frequency analysis of Xsens IMU	51
5.9	Plot of acceleration from IMU Data with found events at 99 percentile	56
5.10	Event Sequence (red vertical lines) on Phasic Component (PC) Time-Serie (TS)	56

5.11	Example of rules on time characteristics: A 0s Latency, 0s Shift 5s Duration; B 0s Latency, +1s Shift 5s Duration; C 1s Latency, 0s Shift 5s Duration; D 0s Latency, 0s Shift 10s Duration. Front sub-sequence is defined by background color red, rear sub-sequence by background color green	57
5.12	Comparison of two different sub-sequences with ARMA Model. On the left time-series has an abrupt change, on the right time-series has no abrupt change	61
5.13	Comparison of two different sub-sequences with NNSBM Algorithm. On the left time-series has an abrupt change, on the right time-series has no abrupt change	62
5.14	ROC curves with relative AUC for PCC and ET algorithms	64
5.15	ROC curve with relative AUC for ARMA algorithms	64
5.16	AUC value for every k-NNSBM	65
5.17	ROC curve with relative AUC for 4-NNSBM	65
5.18	Results with Algorithm NNSBM with some TSs	67
5.19	Example of defining the limit for average correlation, green background indicates the CI	67
5.20	Example of defining the limit for average increasing and decreasing probabilities, red background indicates the CI for Percentage Decreasing, blue background indicates the CI for Percentage Increasing	70
5.21	Comparing of Average with CI and Exact value calculated over all population for average correlation, increasing and decreasing probabilities	70
5.22	Example of defining increasing and decreasing events extracted from ES linear acceleration on X axis over Phasic Component of Skin Conductance	71
5.23	Comparing Increasing probabilities between Random one and ESs with their CI at specific percentile and using Phasic Component time-serie	71
5.24	Probability of ES increasing probabilities being higher than Random extracted at specific percentile and using Phasic Component time-serie	72
5.25	Graph of probability of correlation for percentiles from 0th to 100th for Jerk x with respect to Phasic Component of Skin Conductance	72
5.26	Detected curve from 0th to 100th percentile with comparison to full range stress-relevant and no-stress-relevant curves.	73

5.27	Detected curve from 50th to 100th percentile with comparison to high values range stress-relevant and no-stress-relevant curves.	73
6.1	RMSE no-stress relevant/stress relevant from LF with different durations	79
6.2	RMSE no-stress relevant/stress relevant from HF with different durations	79
6.3	RMSE no-stress relevant/stress relevant from LFHF with different durations	80
6.4	RMSE no-stress relevant/stress relevant from HRV with different durations	80
6.5	RMSE no-stress relevant/stress relevant from HRV with different latencies	81
6.6	RMSE no-stress relevant/stress relevant from HRV with different shifts	81
6.7	RMSE no-stress relevant/stress relevant from TC with different durations	82
6.8	RMSE no-stress relevant/stress relevant from PC with different durations	82
6.9	RMSE no-stress relevant/stress relevant from PC with different latencies	83
6.10	RMSE no-stress relevant/stress relevant from PC with different shifts	83
6.11	Jerk X RMSE no-stress relevant/stress relevant from PC at high percentiles	85
6.12	Linear velocity X RMSE no-stress relevant/stress relevant from PC at high percentiles	85
6.13	Angular velocity Y RMSE no-stress relevant/stress relevant from PC at low percentiles	86
6.14	Jerk X RMSE no-stress relevant/stress relevant from HRV at low percentiles	87
6.15	Linear velocity X RMSE no-stress relevant/stress relevant from HRV at low percentiles	88
6.16	Jerk Z RMSE no-stress relevant/stress relevant from HRV at low percentiles	89
6.17	Angular velocity Y RMSE no-stress relevant/stress relevant from HRV at low percentiles	89

List of Tables

2.1	Confirmed and predicted features on ACs	13
2.2	Driving features for style classification	17
2.3	Results obtained in the urban environment showing comparisons between the driving styles comfortable and dynamic driving	19
2.4	Stress definitions in Literature	21
2.5	HRV time-domain measures	23
2.6	HRV frequency-domain measures	23
2.7	Comparing HR and HRV across different conditions	24
4.1	Table of static IP Addresses	37
4.2	Amount of data needed for the system	43
5.1	3D Detection Performance comparison [57]	47
5.2	Accuracy classification by AUC	63
5.3	AUC results for ARMA(p,q)	64
6.1	Ratio of driving style's features in correlation with PC index at high percentiles	84
6.2	Ratio of driving style's features in correlation with PC index at low percentiles	86
6.3	Ratio of driving style's features in correlation with HRV index at high percentiles	87
6.4	Ratio of driving style's features in correlation with HRV index at low percentiles	88

List of Algorithms

1	Transform TS to ES	55
2	Apply percentiles and time characteristics	58
3	Extraction of average level of correlation	68
4	Extraction of average probability of events increasing and decreasing TS	69

Glossary

AC Autonomous car. VIII, XII, 2–4, 6–20, 92

ACC Adaptive Cruise Control. 12

ANS Autonomic nervous system. 22

ARMA Auto-Regressive Moving Average. X, XII, 28, 59–61, 63–66, 100, 103, 107, 110

AUC Area Under ROC Curve. X, XII, 63–65

CI Confidence Interval. X, 67, 68, 70, 71, 74, 87, 106

CLT Central Limit Theorem. 66

CNN Convolutional Neural Network. 47

ECG Electrocardiogram. IX, 4, 26, 34, 40, 45, 53, 76–78, 80, 83, 89

EMG Electromyogram. IX, 26, 34

ES Event Sequence. IX, X, XIII, 54–57, 70–72, 106, 109

ET Energy Test. X, 28, 59, 63–66, 102, 110

FN False Negative. 63

FP False Positive. 63

FPR False Positive Rate. 63

GPS Global Positioning System. VIII, XVII, XIX, 4, 31, 32, 40–42, 44

GSR Galvanic Skin Response. 4, 5, 25, 26

HBR Heart Beat Rate. IX, 4, 22, 34

HF High Frequency. XI, 5, 23–25, 45, 53, 59, 77, 79, 89

HR Heart Rate. XII, 24, 26

HRV Heart Rate Variability. XI, XII, 4, 5, 22–24, 45, 49, 53, 78, 80, 81, 86–91

I.DRIVE Interaction between Driver, Road Infrastructure, Vehicle and Environment. I, VIII, 2, 5, 7, 15, 19, 26, 29, 36

IBI Interbeat Intervals. 22

IDL Interface Definition Language. 37

IMU Inertial Measurement Unit. VIII, IX, XVII, XIX, 4, 5, 31–33, 51–53, 56, 77

KNN k-Nearest Neighbour. 61

LF Low Frequency. XI, 5, 23–25, 45, 53, 77–79

LFHF Ratio Low Frequency High Frequency. VIII, XI, 4, 5, 23–25, 45, 53, 77, 79, 80, 89

LiDAR Light Detection and Ranging. XVII, XIX, 3–5, 30

ML Machine Learning. 2, 6

NaN Not a Number. 50, 51

NN Neural Network. 12, 22, 46

NNSBM Nearest Neighbor Statistic Based Method. X, 28, 59, 62–67, 100, 104, 106, 110

NTP Network Time Protocol. IX, 41, 42

PC Phasic Component. VIII–XII, 25–27, 45, 46, 53, 56, 71, 72, 78, 80, 82–86, 89–91

PCC Pearson Correlation Coefficient. X, 28, 59, 60, 63–66, 100, 102, 110

PNS Parasympathetic Nervous System. 22, 24, 78, 81

PTP Precision Time Protocol. IX, 41–43

RMSE Root Mean Square Error. XI, 60, 61, 66, 73, 74, 78–83, 85–89

ROC Receiver Operating Characteristic. X, 63–65

ROS Robot Operating System. 31, 37–39, 42, 51, 100, 101

SC Skin Conductance. VIII–X, 4, 5, 25–27, 34, 40, 45, 53, 56, 71, 72, 76–78, 80, 82, 83, 89, 91

SECOND Sparsely Embedded Convolutional Detection. IX, 47, 108

SNS Sympathetic Nervous System. 22, 24, 26, 59, 79, 81

SR Skin Resistance. 25

TC Tonic Component. VIII, XI, 25–27, 46, 53, 78, 80, 82

TN True Negative. 63

TP True Positive. 63

TPR True Positive Rate. 63

TS Time-Serie. IX, X, XIII, 54–57, 66–70, 103, 110

Abstract

With progress in enabling autonomous cars to drive safely on the road, a new concern is emerging about how they should be driving. A common thought is that they should be adopting their users' driving style, but new studies suggest that this is not the best solution. The goal of this work is to search for a correlation between environmental and vehicle's data from car sensors and level of stress for the driver obtained from physiological sensors. To reach this goal a modified car, on which we have installed cameras and Inertial Measurement Unit (IMU), magnetometer, Global Positioning System (GPS) and Light Detection and Ranging (LiDAR) sensors, has been driven through an urban environment while measuring sweating and heart beat rate of the driver. Once data has been collected, a software system has processed the data and it has obtained indicators of stress of the driver and through these the correlation between the level of stress for the driver and the data from environment and car.

Sommario

Grazie ai progressi ottenuti sulla questione sicurezza per le autovetture autonome, un nuovo quesito sta sorgendo a proposito di che stile di guida dovrebbero avere. Un pensiero comune è che dovrebbero adottare lo stile di guida del conducente, ma recenti studi hanno dimostrato che questa non è la soluzione migliore. L'obiettivo di questa tesi è di trovare una correlazione fra i dati ambientali e del veicolo ottenuti dai sensori installati sulla vettura e il livello di stress del conducente ottenuto da alcuni sensori fisiologici. Per raggiungere questo obiettivo è stata utilizzata una autovettura modificata, sulla quale sono state installate diverse telecamere e alcuni sensori quali Inertial Measurement Unit (IMU), magnetometro, Global Positioning System (GPS) e Light Detection and Ranging (LiDAR). L'esperimento è stato condotto in un ambiente urbano, ed indici fisiologici come sudorazione e battito cardiaco sono stati misurati nel conducente. I dati sono stati quindi raccolti e analizzati da un sistema software dal quale sono stati estratti alcuni indicatori di stress del conducente e attraverso questi il livello di correlazione fra lo stress e i dati dell'ambiente e dell'autovettura.

Acknowledgments

I would like to truly thank Giulia, who supported and helped me throughout my university career and my 1 year old daughter Sofia that was present for all the writing of this thesis.

Thanks to my parents, Fernando and Daniela, and my brother, Matteo, who always believed in me.

Deeply thanks to Alessandro that helped and guided me from the beginning of the thesis and to professor Matteucci that allowed me to be part of this stimulating project.

Last but not least thanks to all professors through these unforgettable years for their dedication and passion that enlightened my interest.

Chapter 1

Introduction

“In a properly automated and educated world, then, machines may prove to be the true humanizing influence. It may be that machines will do the work that makes life possible and that human beings will do all the other things that make life pleasant and worthwhile ”

Isaac Asimov - Robot visions

The I.DRIVE project is an interdisciplinary Politecnico di Milano project which goal is to acquire, analyse, and model the interaction between driver, vehicle and environment. The thesis work described in this paper was carried out as part of the project. Indeed, we have collected, cleaned and analysed data from a car with full sensors' equipment then various Machine Learning (ML) algorithms have been compared to extract the correlation between environmental data and physiological data in order to find which condition, between environmental and mechanical, has the highest probability to increase the level of stress of the driver.

1.1 Motivation

In last decades the Autonomous cars (AC) have been obtaining an emergent role [1], it is possible to find early examples since 1920, while the first semi-automated car was developed in 1977 by Japan's Tsukuba Mechanical Engineering Laboratory. Since then there have been many projects that improved the autonomous driving such as the Autonomous Land driven Vehicle (AVL) funded by United States Department of Defense's agency DARPA, that made use of new technologies developed by the University of Maryland, Carnegie Mellon University, the Environmental Research Institute of Michigan, Martin Marietta and SRI International achieved a milestone for

autonomous vehicles using sensors such as LiDAR, computer vision and autonomous robotic control to direct the vehicle up to 31 km/h [2]. In 1987, HRL Laboratories (formerly Hughes Research Labs) demonstrated the first sensor-based autonomous drive on the ALV when the vehicle traveled for 610 m on complex terrain. By 1989 Carnegie Mellon University used Neural Networks (NN) to steer and control autonomous vehicles. In 1995, Carnegie Mellon University's Navlab project completed a 5.000 km cross-country journey, the AC used a NN to control the steering wheel, while throttle and brakes were human-controlled mainly for safety reasons. In 1996, Professor Alberto Broggi of the University of Parma launched the ARGO project, which allowed a modified Lancia to follow the normal painted lines of an unmodified highway. In the 2000s DARPA organized some Grand Challenges in which offered a prize of 1\$ million prize to any team that would succeed in creating an AC that would be able to complete a 150-mile course in the Mojave Desert. In 2005, the second Grand Challenged, five vehicles completed the course. In last years many automotive manufactures, including General Motors, Ford, Mercedes Benz, Volkswagen, Audi, Nissan, Toyota, BMW and Volvo are in the process of testing driverless car systems. On May 27 2014, Google announced to unveil 100 AC prototypes and on October 2014 Tesla announced its first version of AutoPilot with Tesla Model S which, based on signals image recognition, is capable of lane control with autonomous steering, braking and speed limit adjustment.

Interest in AC shifted from academic and Defense projects to automotive manufactures for customers cars, the every day application brought to light a trust issue between the driver and the AC's driving style [3]. Many different approaches have been studied, such as which style should adopt the AC [4], how different kind of drivers would impact on human-machine-iterations [5] and comparing different AC's driving styles, focusing on different features [6]. All these studies show that the AC's driving style impact on the level of stress of the driver and that finding the right style to decrease the level of stress is not an easy task as each user has different demands and the requested AC's driving style could also be different from user's personal driving style.

Correlation between environmental data and physiological driver's data has been analyzed because in a fully AC, the disengagement of the driver implies the loss of control on the car and this condition, as shown later in this work, causes insecurity and stress. So, in order to make people comfortable while driving AC we have to detect which situations maximize the level of stress of the driver in order to change the driving style of the AC to decrease the level of stress and make more comfortable the driver in future

driving sections. The main goal of this work is to develop an algorithm that automatically extracts data from driving sections and gives as result the correlation between style of driving and level of stress of the driver.

1.2 Contributions

To achieve this goal, we need to retrieve as much information as possible in order to obtain the level of stress of the driver and to detect all the possible external conditions that could cause a rise in driver's level of stress. To do this we used some sensors attached to the driver monitoring some physiological signals such as the Electrocardiogram (ECG), from the Heart Beat Rate (HBR), from which we obtained signals of Ratio Low Frequency High Frequency (LFHF) and Heart Rate Variability (HRV) and the Skin Conductance (SC) from sweating from which we obtained phasic and tonic signals of Galvanic Skin Response (GSR). We used a properly modified car on which we have installed cameras, Inertial Measurement Unit (IMU), magnetometer, Global Positioning System (GPS) and Light Detection and Ranging(LiDAR). From these sensors we obtained information about speed, acceleration and jerk of the car in the three axes, number and category of obstacles detected, their position and their time headway with respect to the leading car, among many others. The results obtained show that for each user there are from three to five external situations that maximize the level of stress, while these situations reflect the ones that have been selected from previous studies, they differ from user to user. These results have been obtained analyzing more driving sections for each user and finding a confidence interval of the probability that each condition could have increased the driver's stress level. For future work a fully-AC could be employed and a live analysis could bring to minimize the level of stress of the driver while driving and new algorithms to detect stressing situations could be employed. With the increase of number of driving sections and the employment of surveys, to obtain a driver's consideration about stressing conditions, training and testing of new techniques to extract the level of stress could be found.

The level of stress has been obtained using sensors attached to user's body and through questionnaires which objective was to obtain a gap between emotive reactions before and after the driving [7]. In order to obtain an indicator of user's stress level through all the driving some sensors have been attached to user's body, main physiological data that has been analyzed in studies are:

- Heart Beat Rate (HBR) from which we obtained Heart Rate Vari-



Figure 1.1: Tazzari Zero used for I.DRIVE project

ability (HRV), High Frequency (HF) signal and Low Frequency (LF) signal and ratio between LF and HF (LFHF).

- Galvanic Skin Response (GSR) from which we obtained Skin Conductance (SC) with Phasic and Tonic signal.
- Pupil dilation from which we obtained the pupil diameter trend.

In order to analyze the driving style some factors of the driving have been analyzed:

- Speed, Acceleration and Jerk in XYZ axes of the vehicle thanks to IMU sensor.
- Distance, number and category of obstacles from the vehicle thanks to cameras and LiDAR.
- Time Headway of obstacles thanks to both IMU sensor and LiDAR.

All these physiological and environmental data have been collected in a MongoDB Database during drivings conducted in an urban area in daylight thanks to the system mounted on Tazzari car (Figure 1.1), just before the start of driving it was requested to the user to stay still with closed eyes for 3 minutes in order to normalize the level of stress and consider this as the ground zero, then the length of drivings would be between 15 and 30 minutes. The collected data has been then cleaned and analyzed through

automatic software that, through different ML algorithms, checked for correlation between physiological and environmental data giving as output the characteristics of the driving style that increased the level of stress in the user. During the work of this project it was possible also to verify the results of state of art's documents considered for analysis of physiological data in order to obtain user's level of stress.

1.3 Thesis outline

The structure thesis is the following:

- **Chapter 2:** lists all the progress made with AC's technology and how human-machine interactions changed through time until recent times. Lists main studies conducted concerning physiological stress indexes and analysis of driving styles in correlation to level of stress detected. Then lists ML algorithms employed for this thesis.
- **Chapter 3:** describes briefly the experimental car used for this work thesis, describing all sensors mounted on the car and sensors attached to the user and the computer hardware that made possible to collect data through the drivings.
- **Chapter 4:** describes the software architecture used for the work.
- **Chapter 5:** focuses on data preparation, data processing, description of applied algorithms and correlation extraction.
- **Chapter 6:** describes the experiments conducted with description of used protocol and final results.
- **Chapter 7:** lists conclusion of this work and possible future works.
- **Appendix A:** provides software requirements to execute the scripts.
- **Appendix B:** provides Programming Documentation for main algorithms.
- **Appendix C:** provides a Manual for the user.
- **Appendix D:** provides the proposed questionnaire.

Chapter 2

State of art

In this chapter we see the state of art concerning autonomous driving, as one possible application of I.DRIVE project is to develop an AC that cares about the passengers analyzing their mood and changing the driving style in order to decrease the level of stress. We give an explanation of what exactly means autonomous driving and AC, then we see a brief history of ACs through the years until recent years and show the importance of the work we are conducting and how interest shifted from academic and government to every days as automotive manufactures become more and more interested in this area. We talk about definition of stress and studies concerning how to measure and which sensors to use, we see which are the driving style factors that in normal driving sections could increase the level of stress and consider these factors in autonomous driving to see which change and which keep the same and how to measure these factors. Finally we talk about the state of art in algorithms used to check for correlation in this work area and others work areas but that could be useful for our work.

2.1 Autonomous driving

Before describing existing projects we give a definition of AC and autonomous driving. In 2014, SAE International published a formal classification system with six levels from fully manual to fully autonomous [8] in contrast to the classification published in 2013 from National Highway Traffic Safety Administration (NHTS) [9] that was abandoned in 2016 in favor of the SAE International’s new standard J3016. The six level identify how much control has the car over the driver:

- **Level 0:** No automation, full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems.
- **Level 1:** Driver assistance, driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task.
- **Level 2:** Partial automation, driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task.
- **Level 3:** Conditional automation, driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene.
- **Level 4:** High automation, driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene.
- **Level 5:** Full automation, full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver.

Some key definitions in SAE International standard J3016 include:

- **Dynamic driving task** includes the operational (steering, braking, accelerating, monitoring the vehicle and roadway) and tactical (responding to events, determining when to change lanes, turn, use signals, etc.) aspects of the driving task, but not the strategic (determining destinations and waypoints) aspect of the driving task.
- **Driving mode** is a type of driving scenario with characteristic dynamic driving task requirements (e.g., expressway merging, high speed cruising, low speed traffic jam, closed-campus operations, etc. . .).
- **Request to intervene** is notification by the automated driving system to a human driver that s/he should promptly begin or resume performance of the dynamic driving task.

These six level of automatism identify how much control has the car over the driver, a relevant change happens between level 2 and 3 when the human driver does not need anymore to monitor the environment. In Washington DC's district code the legal definition of an AC is:

“Autonomous vehicle” means a vehicle capable of navigating District roadways and interpreting traffic-control devices without a driver actively operating any of the vehicle’s control systems. The term “autonomous vehicle” excludes a motor vehicle enabled with active safety systems or driver- assistance systems, including systems to provide electronic blind-spot assistance, crash avoidance, emergency braking, parking assistance, adaptive cruise control, lane-keep assistance, lane-departure warning, or traffic-jam and queuing assistance, unless the system alone or in combination with other systems enables the vehicle on which the technology is installed to drive without active control or monitoring by a human operator.

Washington, DC's district code [10]

In 2020 there are no cars available to customers that provide full automation, it is possible on the other hand to find ACs for research purpose, but looking at the history of ACs it is possible to see how the interest of automotive manufactures has shifted toward this new technology.

2.1.1 Brief History of Autonomous Cars

The history of ACs starts from 1925 with the Linriccan Wonder, it was a radio controlled car built by Houdina Radio Control through New York City streets [11]. The car was a 1926 Chandler equipped with an antenna on the tonneau that received radio signal by a person in another car that



(a)



(b)



(c)



(d)

Figure 2.1: First AC prototypes: (a) ALV (b) VaMP (c) Vita-2 (d) ARGO

was following it. The radio impulses gave signal to small electric motors that directed every movement of the car. There were other demonstrations of this radio controlled car over the years. RCA Labs made significantly improves for the ACs as in the 1953 built a miniature car that, through wires laid in a pattern on a laboratory floor, was able to move automatically [1]. Leland Hancock, a traffic engineer in Nebraska, and L.N. Ress, a state engineer extended this idea to a greater scale experimenting the system in an actual highway. The experiment was conducted in 1958 and took place in a 121 m strip of highway outside of Lincoln city, Nebraska. A series of detector circuits buried in the pavement and a series of light along the edge of the street were able to send impulses to guide the car. Also General Motors collaborated in the experiment, and paired two standard models equipped with radio receivers, audible and visible warnings that were able to simulate brake, steering and acceleration control.

In 1959 and throughout the 1960s some advanced models of previous self-driving cars were tested and showcased in Motorama, an auto show by General Motors. Firebird was a series of experimental cars which had an electronic guide system that allowed the car to rush over an automatic



(a)



(b)



(c)



(d)

Figure 2.2: Modern AC: (a) Google car prototitpe (b) Ford AC (c) Lexus model in Google car project (d) Uber AC

highway without any control from the driver. These experiments led in 1966 to Ohio State University's Communication and Control Systems Laboratory to develop a project of a self-driving car activated by electronic devices embedded in the highway. In same years United Kingdom's Transport and Road Research Laboratory tested a Citroen DS that through interaction with magnetic cables embedded in the road was able to drive at 130 km/h through a track with no deviation in speed or direction in any weather condition [12][13].

In 1980s, Ernst Dickmanns and his team at the Bundeswehr University Munich, Germany, designed a vision-guided driverless Mercedes-Benz van that was able to drive at speed of 63 km/h on streets without traffic. In same years Defense Advanced Research Projects Agency (DARPA) showed interest in this emerging technology by funding Autonomous Land driven Vehicle (ALV) [2] project using new technologies developed by the University of Maryland, Carnegie Mellon University, the Environmental Research Institute of Michigan, Martin Marietta and SRI International. The self-driving car used LiDAR and computer vision as sensors and autonomous robotic control to achieve a speed of 31 km/h [14]. HRL (formerly Hughes Research

Labs) using the ALV demonstrated the first off-road map and sensor-based autonomous navigation. The vehicle traveled through complex terrain with natural obstacles at 3.1 km/h for 610 m [15].

The progress allowed the newer AC to be more and more efficient with time. Daimler-Benz and Ernst Dickmanns of Bundeswehr University Munich in 1991 tested twin robot vehicles VaMP and Vita-2 that drove more than 1.000 km on a Paris three-lane highway in standard traffic at max speed of 130 km/h semi-autonomously with human intervention [16]. In 1995, Dickmann's autonomous S-Class Mercedes-Benz drove from Munich, Germany, to Copenhagen, Denmark, covering a 1.590 journey and using computer vision and microprocessors with integral memory designed for parallel processing to react in real time. Results achieved are: speed exceeding 175 km/h on the German Autobahn, with 95% autonomous driving and one human intervention in average every 9 km. The AC drove in traffic, executed various maneuvers to pass other cars [17][18]. In 1995, a semi-AC that used a NN to control the steering wheel while throttle and brakes were human controlled was created by Carnegie Mellon University. The car achieved 98,2% of autonomous driving over a 5.000 km cross-country journey [19]. In 1996, an AC project was presented by Alberto Broggi of the University of Parma. The project was called ARGO and was an autonomous vehicle that could follow the normal painted marks on the highway. The car succeeded in driving for 1.900 km in Northern Italy's highways over 6 days with an average speed of 90 km/h, the car drove in automatic mode for 94% of the journey, covering up to 55 km/h without human interaction [20]. With new millennium dawn the interest in AC shifted to various sectors which comprehend autonomous public transportation and private cars.

2.1.2 Contemporary progresses

Modern companies keep coming up with new features for ACs. Main features include: self-parking, lane-keeping, collision avoidance, Adaptive Cruise Control (ACC) and power assisting steering. In Table 2.1 main present and future features are summarized. All these features of autonomous driving make it possible in a near future to use everyday an AC, while this technology keeps improving and new ACs will be available on market a new issue must be considered. Will the passenger willingly choose ACs over non-autonomous vehicles? And how will passenger feel throughout the ride? In our work we consider these questions and try to answer considering how the passenger feels comfortable while driving ACs.

Organization	Confirmed and predicted product introduction	Predictions of readiness for autonomous vehicles
Audi/VW	2016 – Piloted Driving	Full AV by 2021
BMW	2014 – traffic jam assist 2014 – automated parking	Available by 2021
Bosch	2017 – Integrated Highway Assist 2020 – Highway Pilot	Auto Pilot by 2025
Continental		Available by 2020
Daimler-Benz	2014 – Intelligent Drive	Available by 2020
Ford	2015 – fully assisted parking	To mass produce AV in 2021
GM	2017 – Super cruise	
Google	2015 – Driverless Pod prototype	Available by 2018
Honda		Available by 2020
Hyundai		Available by 2030
Mobile Eye	2016 – technology ready for OEMs	
Nissan	2016 – traffic jam pilot 2018 – multiple lane control	Available by 2020
Tesla	2015 – Lane Assist + ACC 2016 – highly autonomous	Self-driving 2020-2025
Toyota	Mid 2010s – highly autonomous	
Volvo	2015 – traffic jam assist 2017 – Drive Me FOT in Sweden	Zero fatality cars by 2020

Table 2.1: Confirmed and predicted features on ACs

2.1.3 Do people trust autonomous cars?

The question if people trust ACs has increased importance in recent years as shown in recent studies. In 2018, Kaur and Rampersad [3] used a questionnaire to obtain the level of trust in ACs. Pertinent technology adoption theories include TAM [21] (Technology Adoption Model) and UTAUT [22] (Unified Theory of Acceptance and Use of Technology), from these theories some relevant factors have been extracted such as reliability, performance expectancy, trust, security and privacy. TAM theory was developed in 1989 by Davis F.D. in which acceptance was the decision of the user to use technology. Two main factors would decide the level of acceptance and they would be:

- Perceived usefulness: defined as “the degree to which a person believes that using a particular system would enhance his or her job performance”, which is how the user perceives the new technology useful or not for the job that he wants to do;
- Perceived ease-of-use (PEOU): defined as “the degree to which a person believes that using a particular system would be free from effort”, which is how the user perceives how easy is the adoption of the new technology.

However, with later studies, it was believe that these two only factors would not be enough to define the level of acceptance by the user of technology. In 2003 Venkatesh V. developed the UTAUT theory, by this theory the reasons for a user to use a technology can be found in main 4 factors:

- Performance expectancy;
- Effort expectancy;

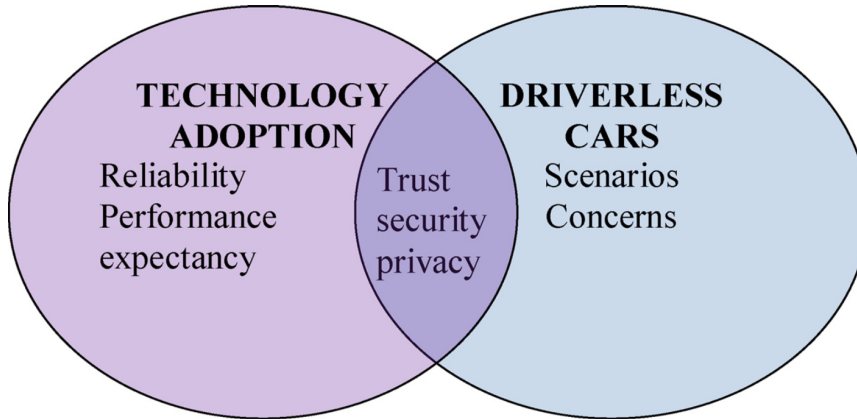


Figure 2.3: Key literature and factors influencing the adoption of ACs[22]

- Social influence;
- Facilitating conditions.

Considering these two theories the application on ACs focuses on some main factors, as shown in Figure 2.3. Kaur and Rampersad[3] made 5 hypothesis on these factors:

- Performance expectancy positively influences the adoption of driverless cars: ACs' performance must outperform non-ACs' performance both in terms of security but also in terms of passengers' performance as an AC would allow much more free time at user's hands;
- Reliability positively influences the trust in driverless cars: the reliability of the user in AC is fundamental, it can be achieved with many demonstrations in different conditions in order to increase trust in how the ACs would handle unlikely events;
- Security positively increases trust in driverless cars: not only security in terms of driving but also security from car hacking, remote access, remote control of the vehicle, computer virus's malwares and spoofing must be taken in account. Higher level in security would increase the trust of the user;
- Privacy positively influences trust in driverless cars: the loss of autonomy extends to the loss of privacy. The user might feel a loss in privacy due to the car choices of road, constant control over location and choice of future travels, some privacy protection must be taken into account in order to increase the level of trust of the user;

- Trust positively influences the adoption of driverless cars: all these hypothesis together aim to increase the level of trust of the user and with the increase of trust increases also the probability of adoption of ACs.

One of I.DRIVE project goals is to increase the level of trust in ACs by decreasing the level of stress during drivings which can be considered in both performance expectancy and reliability. Some studies were conducted in order to increase the level of trust by changing the human-machine interactions, an example is the study conducted by Helldin T. in 2013 [23]. The study shows how using an User Interface (UI) that keeps monitored the level of uncertainty of ACs' sensors increase the level of trust of passengers. While the amount of time the passenger is focused on the road decreases, which leads to increase performance expectancy, the passenger is ready to react faster having under control the level of uncertainty. The experiment was carried out at the Human Machine Interaction (HMI) laboratory at Volvo Car Corporation in Sweden, the participants drove the car through a snowed two-lane country-side road with sharp turns and no other traffic.

Other studies focused not only on giving to the passenger the information about the level of uncertainty but giving to the user information on what the car is going to perform. In 2014 Jeamin Koo [24] conducted an experiment to analyze the importance in giving feedback to passengers of ACs on what is going to happen. The experiment consisted in 4 scenarios:

- No information given to passengers.
- “How” information: internal activity of what the car is going to perform.
- “Why” information: information of external environment.
- Both “How” and “Why” information: internal and external information referring to automation.

The “How” is the information given by the car to the passenger of actions that is going to perform, a voice with no particular mood would advise the passenger of various actions like “Car is braking”, on the other hand “Why” information refers to external environment, the car advise the passenger of the reason in taking action like “Obstacle ahead”. Giving both “How” and “Why” information leads to both internal and external activity, for the given example the car would advise the passenger “Car is braking due to obstacle ahead”. Sixty-four students were recruited to participate in the

study and the results showed that highest level of acceptance was when only “Why” information was given, worst results when only “How” was given. The reason behind this is that both “How+Why” was considered too much of information and the passenger felt anxious when the car advised him, only “How” information obtained worst level of acceptance because left the user with a question of why the car would had to take that particular action. “Why” information would allow the user to know what the car was sensing without at the same time giving too much information.

Many other studies showed the importance of human-machine interaction and how a right interaction leads to increase of trust in the ACs and consequently in increasing the level of acceptance. A fundamental factor is the ACs driving style, in the next section we discuss different features of driving styles and how these change how the passenger feels during the driving which could lead to different level of acceptance by the user.

2.1.4 Autonomous driving: driving styles

There are different definitions of driving style as the one given by Lajunen and Ozkan in 2011 [25] similar to Elander, West and French’s definition in 1993 [26] which focuses on observable behaviour. Murphey, Milton and Kiliaris 2009 [27] definition differs considerably and its definition is too general to be useful. Deery 1999[28] definition focuses on decision making, while Ishibashi, Okuwa, Doi and Akamatsu 2007 [29] definition on ways of thinking. Although all these, and many more, definitions there are some factors common to most of them that have been summarize by Sagberg F. in 2015 study [30]:

- Driving styles differ across individuals or between groups of individuals;
- A driving style is a habitual way of driving, which means that it represents a relatively stable aspect of driving behavior;
- Driving styles reflect conscious choices made by the driver.

Therefore the driving style can be defined as “habitual way of driving, which is characteristic for a driver or a group of drivers” and by “habitual way of driving” we mean the driving behaviours that tend to occur while driving in both cases in a conscious way or automatized way by the driver. Driving behaviours change with respect to road, traffic and driving conditions as can be weather, light conditions and so on. Considering that each individual has its own driving style it could be an idea to have the AC to learn passenger

Features	Definitions
Mean distance to lead car	During car following (with 200 meters distance) the average distance between middle of the driver car and the lead car.
Mean Time Headway	During car following (with 200 meters distance) average time headway, defined as ratio of Distance headway and speed of the driver car.
Time Headway during Lane change	Distance headway divided by the speed of the driver car during lane change.
Distance Headway during Lane change	Distance between the middle of the driver car and the lead car during lane change
Distance Headway Merge Back	This is the same as Distance Headway during lane change except measured in between driver car and the following car in the destination lane.
Braking Distance from the Intersection	Braking distance divided by the speed of the car right before brake is applied.
Maximum Turn Speed	Maximum speed of the driver car over a time window during a left turn or a right turn.
Speed at the Intersection	Instantaneous speed at the intersection.
Average Speed for 20 meters before Intersection	This is the speed of the driver car averaged over a distance range of 20 meters from the intersection.

Table 2.2: Driving features for style classification

driving style and emulate it following the hypothesis that the passenger wants the AC to drive with the same driving style of its own.

Basu C. in 2007 study [31] confuted this hypothesis. The experiment was conducted using a driving simulator and it was divided in two parts. In the first one the data of participants driving styles was collected by letting them drive in the simulation environment, in the second part the participants experienced their own driving style. The driving styles were divided in 4 categories: aggressive, defensive, own style and distractor style (driving style of different participant). In order to learn the driving style some driving features have been considered as summarized in Table 2.2. The first part of the experiment was a 15-20 minutes driving on a 9.6 mile long test track

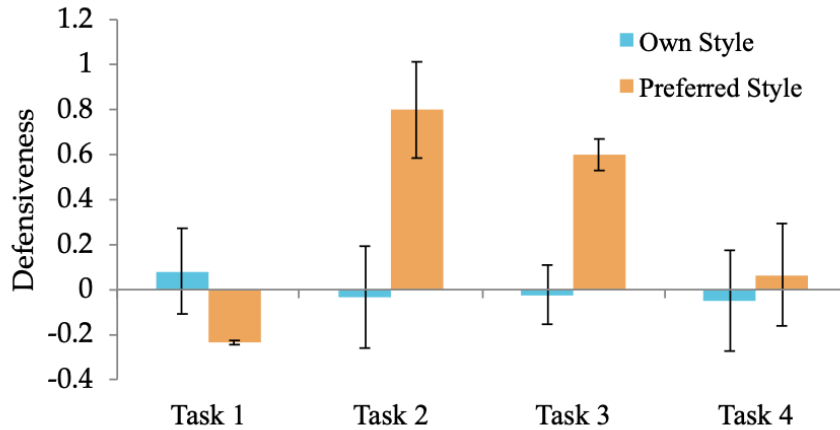


Figure 2.4: Mean Defensiveness score across participants in Basu's experiment[31]

with 14 different driving tasks, each driving task consisted in a sequence of driving manoeuvres in response to traffic conditions. In the second part the participants experienced 6 of these 14 tasks performed by AC of four different styles. To simplify the 6 tasks have been combined in 4 tasks. In Figure 2.4 it is possible to see how the participants preferred a style different from their own and in particular they preferred a much more defensive style. From Basu study it is possible to see how the driving style perceived by passenger differs from their own, which leads to better understand features of driving styles in order to automatically learn which driving style makes more comfortable the passenger.

Another study on driving styles has been done by Bellem in 2016 [32] comparing results from rural, urban and suburban drivings with maximum speed of 100 km/h and results from highway drivings with maximum speed of 120 km/h. Bellem considered maneuvers that are common used in both rural and highway drivings and considered the ones with maximum frequency as stated in literature's studies of Manstetten in 2014 [33], Toledo, Musicant and Loatin in 2018 [34] and Wu, Yeh and Chen in 2014 [35] extracting these main 4 different maneuvers:

- Decelerating to a moving target: approx 16% of maneuvers, the vehicle decelerates from steady velocity upon closing in on another vehicle;
- Accelerating from non-zero speed: approx 18% of maneuvers, the vehicle accelerates from non-zero speed to a goal speed without a leading vehicle;

		Overall effect		Everyday – comfortable		Everyday – dynamic		Comfortable – dynamic	
		<i>F</i>	<i>p</i>	<i>t</i>	<i>p</i>	<i>t</i>	<i>p</i>	<i>t</i>	<i>p</i>
Accelerating from non-zero speed	Acceleration	48.49	<.001***	5.06	.001**	-5.30	<.001***	-9.34	<.001***
	Jerk pressing gas pedal	17.64	<.001***	3.90	.010**	-2.69	.009**	-5.67	<.001***
	Jerk releasing gas pedal	34.87	<.001***	-4.29	.315	5.46	<.001***	6.48	<.001***
	Quickness	26.71	<.001***	3.40	.028*	-4.25	<.001***	-6.61	<.001***
Lane change	Acceleration	19.77	<.001***	4.35	.009**	-3.02	.002**	-5.81	<.001***
	Jerk	10.53	<.001***	2.94	.046	-2.03	.071	-4.60	<.001***
	Quickness	14.32	<.001***	2.46	.071	-2.95	.005**	-5.59	<.001***
Following	Headway distance in seconds	31.47	<.001***	-2.51	.012*	5.07	<.001***	8.72	<.001***
	Standard lane deviation	3.10	.049*	0.65	1.000	-1.67	.313	-2.57	.052

* $p < .05$.
** $p < .01$.
*** $p < .001$.

Table 2.3: Results obtained in the urban environment showing comparisons between the driving styles comfortable and dynamic driving

- Lane change: approx 20% of maneuvers, the vehicle changes lanes;
- Following at a non-varying speed: approx 27% of maneuvers, the vehicle follows a leading vehicle, both without change velocity.

For each maneuver a variety of data was collected, the study focused in metrics that can be perceived by the vestibular system as it plays a key role in development of nausea or motion sickness as showed in Reason’s 1978 study [36] and for the perception of driving in general as showed in Lange’s 2014 [37] and Muller’s 2013 [38] studies. Humans’ vestibular system is able to perceive acceleration both linear, as happens in decelerating to a moving target and accelerating from non-zero speed maneuvers, and angular, as happens when changing lane, but is not able to perceive speed itself. However, humans are more sensitive to rapid changes in acceleration, which is the jerk and can be calculated as the first derivative of acceleration. In the study both acceleration and jerk have been considered.

Bellem considered three different driving style: comfortable, dynamic and every day trip. The perceived comfort should be maximal for comfortable driving and minimal for dynamic driving. The participants were randomly asked to choose one of the three driving styles. We show results for the first part of the study, in urban streets, as for I.DRIVE project we are working on same environment. In Table 2.3 it is possible to view the considered results. It is possible to see how, considering acceleration from non-zero speed, everyday and comfortable driving are much more closer than everyday and dynamic driving. Overall the only proposed metric which was not able to discriminate between driving styles is standard lane deviation. The study demonstrates how acceleration and jerk are important parameters and that a maneuvers-based study can help in better understanding driving styles. We have seen how the AC’s driving style affects passenger’s



Figure 2.5: Lever used by passenger throughout autonomous drive to indicate the level of comfort in Scherer's study[39]

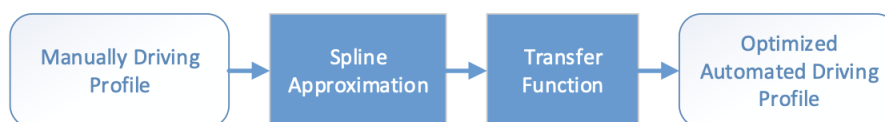


Figure 2.6: General structure of the transformation of manual driving parameters into optimized automated driving parameters[39]

comfort, we question now if it is possible to customize AC's driving style.

In 2015, Scherer S. tried to answer this question [39]. The study was divided in two sessions, in the first the participants drove manually through different scenarios and applying different main maneuvers. After a short break the participants experienced their own driving style as highly automated drive with no possibility in intervention. Throughout the ride the passenger pressed a level, as can see in Figure 2.5, from which it was possible to obtain the level of comfort indicating from 0 (very comfortable) to 100 (very uncomfortable). The more the level was pressed by participant, the more uncomfortable situations he felt. In second session the participant would experience three different driving styles in random order. One would be their own driving style, while the others two would be random picked from other participants' driving style. Scherer used information from the lever, questionnaires after the ride and data from the AC to extract an optimized automated driving profile. The general structure of transformation from manually driving to automated driving profile is shown in Figure 2.6. The study shows how a custom driving profile can be achieved but proof of increase in passenger's comfort was left for future studies.

2.2 Physiological indexes of stress

In shown studies it has been tried to find by questionnaires, giving possibility to passengers to indicate level of comfort throughout the ride or by just considering AC's data how the passenger felt as experiencing autonomous

Author(s)	Stress Definition
Shalev, Yehuda, & McFarlane, (2000) (11)	Stress is a normal psychophysiological response to events which result in the sense of threat, sadness, Dysphoria, and imbalance in people.
American Psychiatric Association (2014) (12)	Stress is described as a sense of being overwhelmed, worry, destruction, press, exhaustion, and lethargy. Therefore, stress can influence people in every age, sex, race, and situation and can result in both physical and psychological health
McEwen (2007) (13)	(In medicine and biology, stress is called to any physical, psychological, and/or emotional factor which results in physical, and/or psychological tension
Behnoudi (2005) (14)	Stress is a situation in which individual is forced to act, and cannot bear the received mental tension. In other words, stress means readjustment of individual with new situations and conditions. Whenever a change occur in life, individual is confronting with stress.
Kumari, et al. (2009) (15)	Stress is bodily response to any demand. Stress could be caused by either good or bad experiences.
Falsetti, Monier, & Resnick (2005) (16)	Stress is any unpleasant emotional experience which is accompanied with predictable biochemical, physiological, and behavioral changes
Silverman, et al. (2010) (17)	Stress is a bodily reaction to a change which needs response, regulation, and/or physical, psychological, and or emotional adaptation. Stress could derive from any situation, condition, thought, and/or state; just need to cause frustration, anger, nervousness, and or anxiety.
Sarafino (2002) (18)	Stress is considered as a situation which is the result of interactions of individuals and their surrounding environments and causes disharmony between situational demands and biopsychosocial resources.
Lazarus, & Folkman (1984) (19); Lazarus (1990) (20)	Stress is an exclusive relation between person and her/his surrounding environment which she/he perceives as taxing, or is gone far beyond her/his coping resources and threatening her/his health.
McEwen (2004; 1999; 1998) (21, 22, 23)	Stress is equivalent to allostasis. Allostasis is the process and capability of gaining stability in the moment of change. Whenever body stress systems are activated in response to high levels of stress, allostasis is evacuated and body exposes to harm.
Shahsavarani, et al. (2013) (5)	Any effect of change in surrounding environment on living being which results in disruption of homeostasis (internal balance) of that living being is called stress.

Table 2.4: Stress definitions in Literature

drive. In our work we want to understand the level of stress of the passenger throughout the ride, the objective is to better understand what parameters make one driving style more comfortable than others and which feature increases the level of stress. To do so we need first to understand what is stress and which physiological information we can retrieve from passenger.

2.2.1 What is exactly the stress?

In 2015, Shahsavarani published on International Journal of Medical Reviews a review [40] of stress trying to cover all facts and theories. In psychological sciences stress can be defined as a feeling of mental press and tension. Many formal definitions have been proposed in recent years, some of these definitions can be found in Table 2.4. We use Shahsavarani’s 2013 definition “*Any effect of change in surrounding environment on living being which results in disruption of homeostasis (internal balance) of that living being is called stress.*” Stress has a positive form that can improve bio psychological health and facilitate performance, but high level of stress could result in biological, psychological and social problems. Stress can be external with environmental sources, as could be in our work, or caused by internal perceptions of the individual. Other factor that influences the level of stress is

the exposure time of stressors that could be acute or short-term, chronic or long-term, for our work we consider only short-term.

2.2.2 What happens to human body in a stressful situation?

Autonomic nervous system (ANS) has the central role in mechanism of stress. ANS is composed of Parasympathetic Nervous System (PNS) and Sympathetic Nervous System (SNS), two branches both tonically active with opposing activities. SNS is activated as primary response to stress in what is called “fight or flight” response so that the living being can adapt to environment as faster as possible, sympathetic nerves exert their effects in more than 5 seconds. In the process of stress response it involves mydriasis (pupil dilatation), increased heart rate and force contraction, vasoconstriction, bronchodilation, sweating and many more. PNS is then activated as secondary response in what is called “rest and digest” and involves return to maintaining homeostasis, parasympathetic nerves exert their effects in less than 1 second. The hypothalamic-pituitary-adrenal axis (HPA axis) is a neuroendocrine system that mediates a stress response between SNS and PNS activity. From this information the changes in physiological indexes that could be useful to determine the level of stress are mydriasis, change in heart rate and sweating. Mydriasis, or pupil dilatation, has been studied by Pedrotti in 2014 [41] in response of external stimuli like auditory, tactile, gustatory, olfactory or noxious. Given this knowledge he tried to extract level of stress of participants to experiment in a driving simulation. Maintaining a constant level of light he succeeded in measuring the pupil dilatation when stressful events occurred, building a NN he was able to classify stress moments from pupil diameter. For our work it is impossible to maintain a constant level of light, this factor introduces noise in measurements and make it harder to find a valid correlation between stress events and pupil diameter. For our work we use pupil eye tracker, which allows to analyse driver gaze and we have also information about pupil diameter throughout the ride.

2.2.3 Heart Beat Rate

Heart Beat Rate (HBR), the number of heartbeats per minute, is another physiological index that could help in detecting the level of stress. It has been studied [42] that HBR has to be analyzed in order to extract information and the first index to be considered is Heart Rate Variability (HRV) which consists of changes in time interval between consecutive heartbeats called Interbeat Intervals (IBI). HRV has an impact in 24h-term, short term

Parameter	Unit	Description
SDNN	ms	Standard deviation of NN intervals
SDRR	ms	Standard deviation of RR intervals
SDANN	ms	Standard deviation of the average NN intervals for each 5 min segment of a 24 h HRV recording
SDNN index (SDNNI)	ms	Mean of the standard deviations of all the NN intervals for each 5 min segment of a 24 h HRV recording
pNN50	%	Percentage of successive RR intervals that differ by more than 50 ms
HR Max – HR Min	bpm	Average difference between the highest and lowest heart rates during each respiratory cycle
RMSSD	ms	Root mean square of successive RR interval differences
HRV triangular index		Integral of the density of the RR interval histogram divided by its height
TINN	ms	Baseline width of the RR interval histogram

Table 2.5: HRV time-domain measures

Parameter	Unit	Description
ULF power	ms ²	Absolute power of the ultra-low-frequency band (≤ 0.003 Hz)
VLF power	ms ²	Absolute power of the very-low-frequency band (0.0033–0.04 Hz)
LF peak	Hz	Peak frequency of the low-frequency band (0.04–0.15 Hz)
LF power	ms ²	Absolute power of the low-frequency band (0.04–0.15 Hz)
LF power	nu	Relative power of the low-frequency band (0.04–0.15 Hz) in normal units
LF power	%	Relative power of the low-frequency band (0.04–0.15 Hz)
HF peak	Hz	Peak frequency of the high-frequency band (0.15–0.4 Hz)
HF power	ms ²	Absolute power of the high-frequency band (0.15–0.4 Hz)
HF power	nu	Relative power of the high-frequency band (0.15–0.4 Hz) in normal units
HF power	%	Relative power of the high-frequency band (0.15–0.4 Hz)
LF/HF	%	Ratio of LF-to-HF power

Table 2.6: HRV frequency-domain measures

(~ 5 minutes) and ultra-short term (< 5 minutes). For our work we are more interested in ultra-short term in order to identify the singular driving feature that causes stress. We need to analyze HRV in time-domain and frequency-domain. In Table 2.5 it is possible to see main measures for time-domain. In Table 2.6 it is possible to see main measures for frequency-domain, heart rate oscillations have been divided in ultra-low-frequency (ULF), very-low-frequency (VLF), Low Frequency (LF), and High Frequency (HF) bands. LF band is 0.04-0.15 Hz which means that is comprised of rhythms from 7 to 25 seconds. HF band is 0.15-0.40 Hz which means that is comprised of rhythms from 2.5 to 7 seconds. The ratio of LF and HF power (LFHF) may

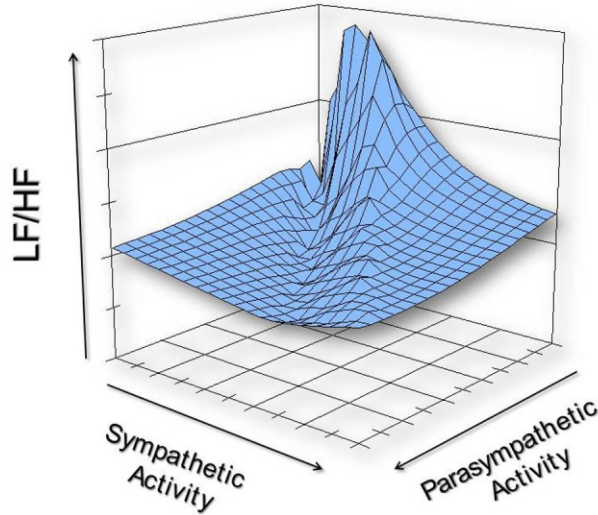


Figure 2.7: Illustration of a possible combination of sympathetic and parasympathetic activity of LFHF[43]

	Rest	Mental task
Mean RR (ms)	0.816 (± 0.13)	0.790 (± 0.13)*
Mean SD	0.0496 (± 0.0174)	0.0452 (± 0.0134)
Mean pNN50 (%)	18,6 (± 14.8)	14.2(± 12.6)*
Mean LF/HF	2.55(± 1.94)	3.14(± 2.68)

Table 2.7: Comparing HR and HRV across different conditions

estimate the ratio between SNS and PNS under controlled conditions. As demonstrated by Billman in 2013 [43] the effects of varying cardiac sympathetic and cardiac parasympathetic nerve activity on LFHF are non-linear as shown in Figure 2.7, this leads to an accurate analysis of LFHF in order to obtain valid information, on the other hand it has been demonstrated that cardiac parasympathetic is exclusively responsible for HF peak, while cardiac sympathetic nerve activity is only responsible for a major part in LF peak. Taelman in 2009 [44] summarized how various parameters change from condition of rest to mental task with linear methods, results can be seen in table 2.7. In our study we have information about HRV with its

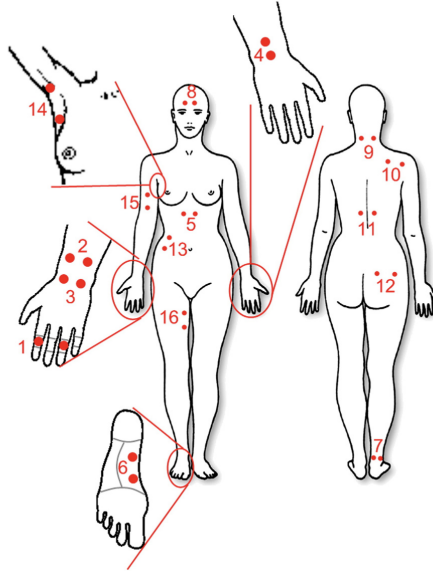


Figure 2.8: Skin conductance measurement locations: 1) fingers, 2) distal wrist, 3) central wrist, 4) vertical wrist, 5) chest, 6) foot (instep), 7) calf, 8) forehead, 9) neck, 10) shoulders, 11) back, 12) buttock, 13) abdomen, 14) armpit, 15) upper arm, and 16) thighbone. [45]

parameters in time and frequency domain such as HF, LF and consequently of ratio LFHF.

2.2.4 Galvanic Skin Response

Another stress index is GSR, which is the measurement of skin resistance and skin conductance. As we have seen that stress response includes sweating and altering cutaneous blood flow and these in turn change GSR. To extract Skin Resistance (SR) it is enough to place two electrodes on skin and record the electrical resistance between them. SC is the inverse of SR. For best positioning we can look at Dooren's 2012 study [45], he tried 16 different measurement locations on body to explore which could have best results. On Figure 2.8 and of Figure 2.9 it is possible to see results from each location. Best results are obtained on forehead, foot, finger and shoulders. Considering our work we need to choose best location with less movements that could cause noise between these 4. Villarejo in 2012 described in their work how to build a galvanometer using ZigBee and which were the data collected in rest and high effort conditions. In 2016, Matteucci, Fontana and Galimberti [46] considered SC as a sum of two components: a Phasic Component (PC) and a Tonic Component (TC). PC is the result of convo-

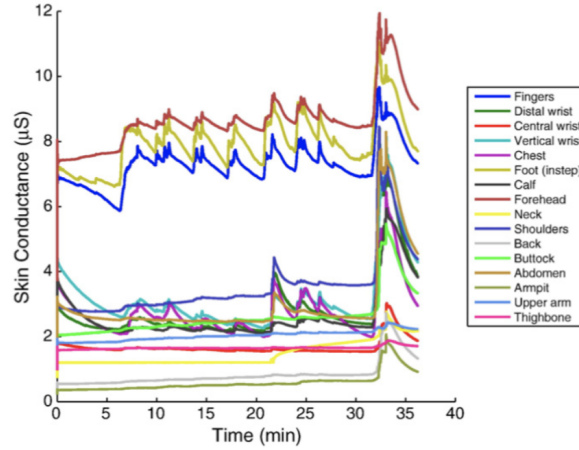


Figure 2.9: Results for each of 16 measurement locations [45]

lution of a signal defined driver, this signal indicates impulses sent through nerves, in response to a stressful event, to sweat glands that causes increase in sweating, which concentration is defined by Bateman function. PC indicates the measurement of conductance of this phenomena. The driver signal represents the quantity of sent impulses, so PC has to be deconvoluted to obtain the driver signal and convoluted with a Gaussian filter to remove high-frequency noise. Then they obtained PC by extracting local maxima, sections not included in local maxima have no PC component, thus they represent TC. Interpolating data in these sections reconstructing TC signal where also PC was present it is possible to obtain the complete TC. By subtracting it to the original to obtain PC which is more informative as it is related with SNS activity. In Figure 2.10 it is possible to see the SC as the sum of the two components.

2.2.5 Analysis of Physiological indexes

In 2005, Healey and Picard [47] used all these physiological measurements to study stress level by collecting data of GSR, ECG, EMG and chest expansion. The study has been conducted in real-world and not in a simulation as previous works, they used a questionnaire to validate information extracted from physiological indexes. Results showed that HR and SC have the highest correlation with driver's continuous stress level. In our work we have information about all these physiological indexes, from pupil dilatation, ECG and GSR and we compare obtained results analyzing the correlation with environmental and vehicle data. In 2019, Matteucci, Gabrielli and Sesto[48] have studied stress indexes for I.DRIVE project, results of work

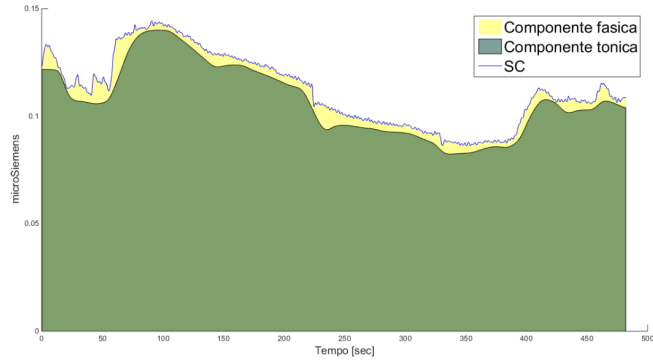


Figure 2.10: SC as sum of PC (green) and TC (yellow) [46]

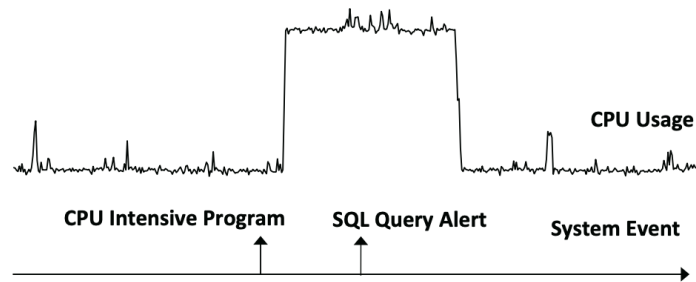


Figure 2.11: Example of temporal order between Time-Series CPU Usage and Temporal Event System Event [49]

are explained in Chapter 6.

2.3 Correlation algorithms

For our work we have information about physiological indexes and environmental data, we need to find the level of correlation between these two samples. This is a problem of two-sample testing but as during acquisition we have multiple features we need to find a way to isolate the feature we are considering. Similar studies have been conducted in different areas, in 2014 Luo [49] studied correlation between time-series and temporal event data in incident investigation of online services. Luo analyzed a two-sample testing problem between CPU usage and System Event as shown in Figure 2.11. The used approach is to analyze sub-series of Time-Series considering front and rear sub-series of same length, as shown in Figure 2.12, when an Event, defined by System Event serie, happens. When front and rear are statistically different this might mean that the Event brought a change in

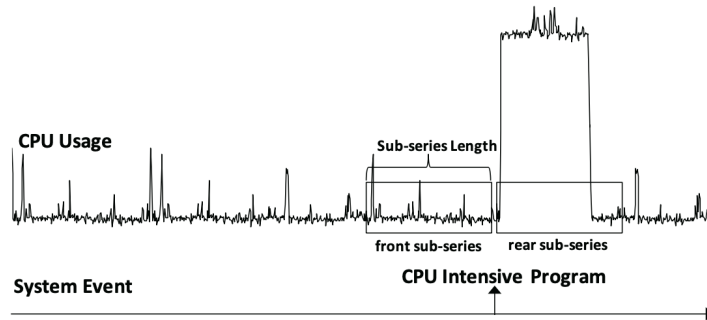


Figure 2.12: Example of front and rear sub-series [49]

the Time-Serie, otherwise the Event has not cause any change in the Time-Serie. This operation must be repeated for every event and then a statistical analysis defines most emergent Events that caused a change in the Time-Serie.

Luo used different algorithms used to compare front and rear sub-series, comparing baseline algorithms such as PCC (Pearson Correlation Coefficient), as defined in *Statistical Power Analysis for the Behavioral Sciences* of Jacob Cohen[50] and ARMA models (Auto-Regressive Moving Average) as defined in *Time series analysis : forecasting and control*[51], and two main specific algorithms such as NNSBM (Nearest Neighbor Statistic Based Method)[52] and ET (Energy Test)[53]. For our work we consider these main algorithms and compare results obtained. In Chapter 5 we look in detail to each of these algorithms, and analyze their complexities. From Basseville and Nikiforov “Detection of Abrupt Changes: Theory and Application” book [54] we extracted definitions of abrupt changes in time analysis, applying these definitions for our project and considering which characteristic fits better for our model, we see in detail this part in Chapter 5.

Chapter 3

I.DRIVE Hardware

To achieve our goal, we need to obtain information of vehicle, environment and driver's physiological data. In 2015 Matteucci and Gabrielli [55] described the structure of sensor systems, communication protocols, power consumption and acquisition and storing methods. In this chapter we give a description of all the hardware used for I.DRIVE system, which comprehends of the vehicle and all equipped sensors.

3.1 I.DRIVE Vehicle

The vehicle is a modified *Tazzari Zero* (Figure 3.1) which has been used for every I.DRIVE project so far, it is a 2 seats electric citycar with an asynchronous brushless engine that supplies 15 KW power and max torque of 150 Nm, the car can reach a max speed of 100 km/h and has autonomy of 140 km given by lithium ion battery with nominal voltage of 80 V and capacity of 160 Ah. Maximum autonomy of battery is 9 hours but considering that all sensors are attached to battery this results in a limited autonomy, for our work this is not a problem as drivings last up to 30 minutes. The vehicle supports 4 driving styles:

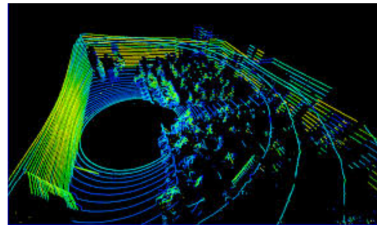
- **Standard:** preferable for everyday use in urban environment
- **Economy:** limits acceleration and max speed but increases autonomy
- **Rain:** does not differ much from standard style but with limits to brake assist and acceleration in order to increase grip in case of rainy weather
- **Race:** maximum performance in terms of acceleration and speed at the expense of autonomy



Figure 3.1: Tazzari Zero without external sensors



(a)



(b)

Figure 3.2: An Example of Point cloud(b) of Velodyne HDL-32E(a)

3.2 Environmental sensors

In this section we go through all equipped environmental sensors.

3.2.1 LiDAR

LiDAR (Light Detection and Ranging) is a technology that measures distance by lighting a target with a laser and analyzing the reflected signal. For our work we have installed *Velodyne HDL-32E* (Figure 3.2a), that thanks to 32 laser layers allows to reconstruct the surrounding environment with a point cloud (Figure 3.2b) with vertical field view between $+10.67^\circ$ and -30.67° and horizontal field view of 360° , with a rotating head at frequency from 5Hz to 20Hz, user can choose custom frequency while standard one is 10 Hz, it reaches a 700.000 samples/sec. Velodyne HDL 32-E gets

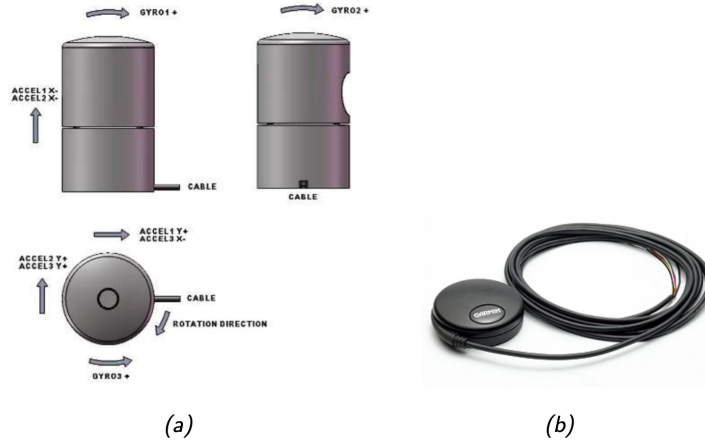


Figure 3.3: Internal sensors of Velodyne HDL-32E: IMU(a) and GPS(b)

information of environment from 1 m to 70 m with an accuracy of $\pm 2\text{cm}$.

Velodyne HDL 32-E has an internal IMU consisting in 3 gyroscopes, 3 accelerometers and 2 axes positioned as in Figure 3.3a. Accelerometers allow to retrieve information of acceleration in 3 axes and gyroscopes to retrieve information of angular velocity in 3 axes. Velodyne HDL 32-E has also an external GPS Garmin 18LV (Figure 3.3b) to retrieve information of position and is used to synchronized data generated from Velodyne with precision of impulses given by GPS. If GPS is not connected an internal clock is used, which is much less precise than GPS. GPS Garmin 18LV generates data according to standard \$GPRMC NMEA with frequency of 1Hz and is linked to Velodyne through a serial port RS-232.

3.2.2 GPS receiver

Beyond GPS Garmin 18LV another external GPS Yuan10 (Figure 3.4) has been used. The use of a second GPS is due to problems of synchronicity, through ROS, between PC and GPS. GPS Yuan10 data is used only to synchronize data to PC and no other information is stored. GPS Yuan10 has an internal receiver Skytraq S1315F-RAW, this receiver has a sensitivity when tracking of -161 dBm , which allows to retrieve information of position in almost every condition. This device can retrieve information at max frequency of 20 Hz but for our work we use it with a frequency of 1 Hz with NMEA standard. Using RTKLIB, an open source library, is possible to obtain precise position through GNSS (Global Navigation Satellite System), which allows to use modality DGPS (Differential Global Positioning System)

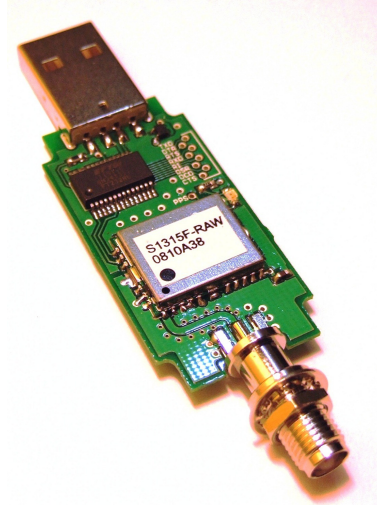


Figure 3.4: GPS Yuan10 with Skytraq S1315F-RAW receiver

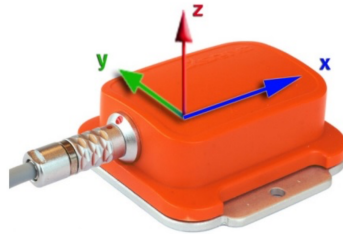


Figure 3.5: IMU Xsens MTi with fixed reference system

that uses two GPS one placed on top of the car and another on the ground in a fixed position. This modality reduces error from 3-5 m to 10 cm. GPS Yuan10 is connected to PC through a standard USB.

3.2.3 IMU

Used IMU is a Xsens MTi (Figure 3.5) that contains a 3D magnetometer that retrieves information of Earth's magnetic field, a 3D accelerometer that retrieves information of linear accelerations and a 3D gyroscope that retrieves information angular velocities. Xsens MTi maximum sampling rate is 100 Hz, for our work we keep this sampling rate. Positioning of 3 axes, as shown in Figure 3.5, has a fixed reference system which is independent from positioning of device and is defined as:

- **X**: positive in direction of magnetic North;
- **Y**: follows right-hand rule (magnetic West);



Figure 3.6: *Prosilica GC1020/650 (a) with Theia wide-angle optics (b)*

- **Z**: positive if upward.

IMU Xsens MTi is linked to PC through an USB adapter.

3.2.4 External cameras

On the roof of the car two cameras *Prosilica GC1020* have been mounted, one facing the street and one facing the right side of the vehicle. These cameras are used to acquire images of external environment. *Prosilica GC1020* (Figure 3.6a) is a network camera connected through an ethernet cable to PC, it works at 30 fps with resolution of 1024 x 768 and uses BayerRG8 to limit dimension of images. *Prosilica GC1020* facing street mounts a wide-angle lens *Theia MY125M* (Figure 3.6b) with a 125° horizontal field and 109° vertical field with very low distortion.

3.3 Physiological sensors

In this section we'll go through all equipped sensors to acquire information of the driver. Two different kind of sensors have been used to retrieve information of physiological index of the driver.

3.3.1 Procomp Infiniti

Procomp Infiniti card (Figure 3.7a) is 8 channels encoder used to acquire in real time, using different sensors (Figure 3.7b), physiological indexes of human body. 2 of 8 channels have sampling rate of 2048 Hz, remaining ones of 256 Hz. Procomp Infiniti samples input signals from sensors, encodes and sends to TT-USB interface. For transmission from encoder to interface a fiber optic cable is used, in order to have the best electrical insulation. The cable can reach up to 7.62 m without losing quality in signal. TT-



Figure 3.7: Procomp Infiniti (a) with a set of available sensors (b) top from left: EMG, ECG, EEG and SC. Bottom from left: body temperature, breathing and HBR

USB interface transform signal in USB format and sends it to PC through a standard USB. Standard Procomp Infiniti sensors are:

- Electromyogram
- Electrocardiogram
- Electroencephalography
- Skin Conductance
- Body temperature
- Breathing
- Heart Beat Rate

These sensors are not invasive. Sensors for SC are placed on participant's shoulder, as we have seen in Chapter 2 this solution is between the best 4 measuring locations. Each channel of Procomp Infiniti can acquire any sensor's data, the only sensor that requires 2048 Hz is EMG. Encoder is powered by batteries and has an autonomy of 20/30 hours, which is not limiting for our project.

3.3.2 Eye tracker

The eye tracker used for our work is the Pupil Labs[®] (Figure 3.8, Kassner in 2014 [56] explained in detail the device. The device has 3 cameras, 1 for world image and 2 for the eyes. World camera can acquire images at different resolution and frame rate: 1920x1080 @30fps, 1280x720 @60fps, 640x480 @120fps. For our work we have chosen 1280x720 @60 fps which is a good



Figure 3.8: Pupil Labs[®] eye tracker



Figure 3.9: Shuttle Slim

trade-off between resolution and sampling frequency. Eyes cameras have 3 different configurations: 1920x1080 @30fps, 1280x720 @60fps and 640x480 @120fps. For our work we have used 640x480 @120fps in order to have the maximum sampling rate. Along with the headset Pupil Labs[®] implemented a software that allows to change some parameters of acquisition.

3.4 On-board Computer

In the car has been installed an on-board mini-pc *Shuttle Slim* (Figure 3.9) that is equipped with an Intel Core i7-4790S with 4 cores in Hyperthreading, base clock frequency of 3,2 GHz, 8 GB of RAM and 64 GB SSD for OS and 750GB HDD for usage. Installed OS is Ubuntu-Linux v 14.04.

Chapter 4

Software Architecture

Matteucci and Gabrielli explained in detail in their work[55] the software architecture of I.DRIVE project. The pipeline of the system is illustrated in Figure 4.1. We now see every single module of the architecture with their components. Later we have a look to the form of data of some sensors. Then we finally see the dependencies of the scripts that have been used to extract data, with their flow and characteristics.

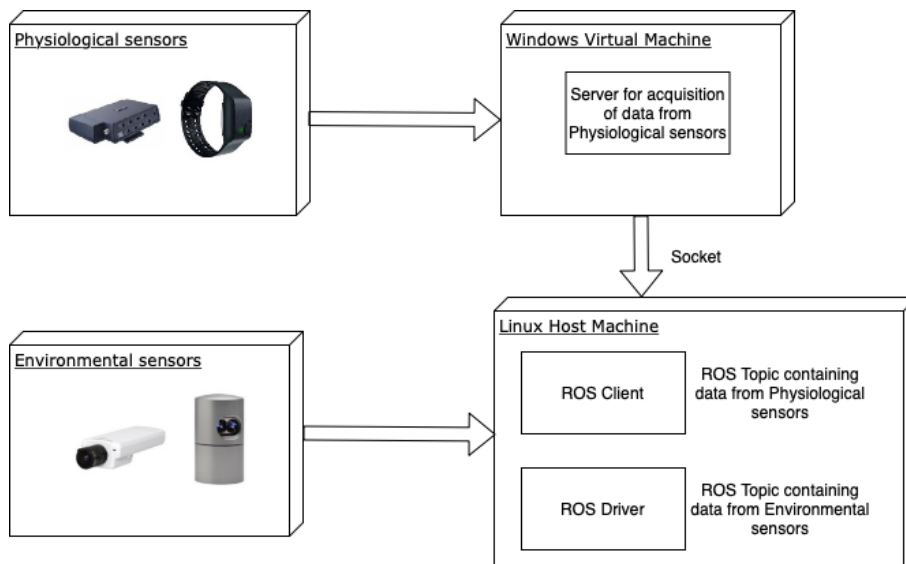


Figure 4.1: Pipeline of information flow from Sensors to MongoDB Database

Sensor	Static IP Address
Computer	192.168.1.3
Virtual Machine	192.168.1.10
Velodyne	192.168.1.201
Prosilica GC1010 1	192.168.1.121
Prosilica GC1010 2	192.168.1.122
Axis P1343 1	192.168.1.101
Axis P1343 2	192.168.1.103

Table 4.1: Table of static IP Addresses

4.1 Integration and acquisition framework (ROS)

To develop the system it has been used Robot Operating System (ROS), an open-source framework widely used for robotics. One of main characteristics of ROS is the modularity obtained by the communication between nodes, this characteristic is needed for our project as we could constantly change environment and physiological sensors attached to the system without the need to rewrite the code but by inserting or removing nodes. ROS nodes are contained in packets by creation of a folder containing a XML file that describes the content and the dependencies. Nodes can be written in four languages: C++, Python, Octave and LISP. Nodes written in different languages can still communicate between them, this allows the framework to be able to extract strengths from each language. To support the multi language feature, the sent messages between different nodes are defined by Interface Definition Language (IDL).

One node sends a message on a specific topic which is identified by a string that represents its name. Another node that is interested to a certain type of message subscribes to the appropriate topic. We can have multiple nodes sending messages at the same time and multiple nodes that subscribe to the same topic and a single node can send and subscribe to many topics. Peer-to-peer technology needs a mechanism to allow processes to find each other at run-time, the master has this role. The master node handles communication between node that is sending the message (Talker) and node that is subscribing to the message (Listener). Once nodes have found each other they communicate with a peer-to-peer communication. We installed Ubuntu Linux OS on Shuttle Slim in order to use ROS, which is compatible with almost every used sensors, the only exception is Procomp Infiniti that needed a Windows OS. To overcome this we used a Virtual Machine, using VirtualBox, running Windows 8.

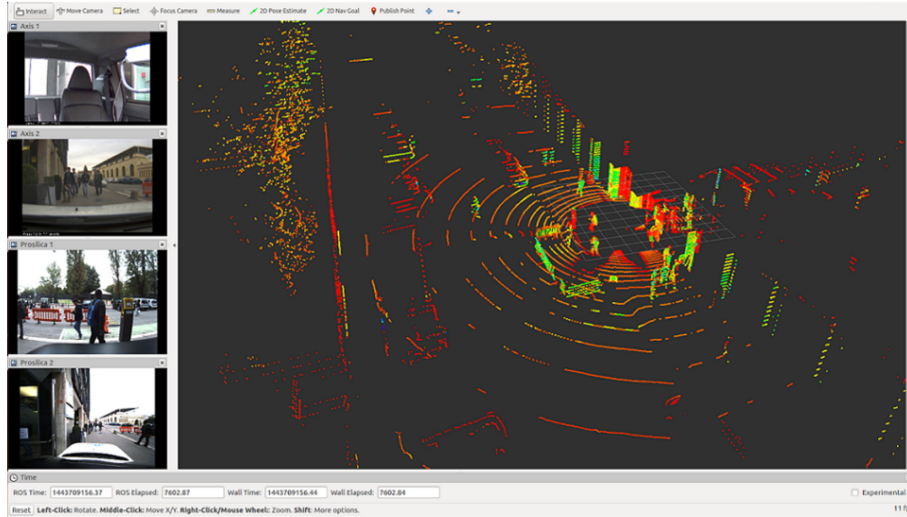


Figure 4.2: Example of RVIZ window with 4 cameras, on the left, and point cloud

Virtual Machine and Shuttle Slim have been provided of a static IP to be able to easily communicate over TCP socket, also sensors with Ethernet interface have been provided with a static IP as reported in Table 4.1

RVIZ is a ROS tool that allows to graphically visualize some data, in particular for our work we can visualize images from cameras Axis P1343, cameras Prosilica GC1020 and Velodyne point cloud as shown in Figure 4.2. It is very useful to visualize obstacles, such as cars and pedestrians, in four directions with different maximum distance.

4.2 Big Data storage (MongoDB)

For Big Data storage we have chosen a NoSQL database, which means that a database that does not use the relation model usually used for normal databases. The NoSQL database chosen is MongoDB, due to its wide popularity and high performances with respect to other NoSQL databases. Furthermore ROS contains a packet that handles saving to MongoDB.

MongoDB is a NoSQL open source document oriented, it uses JSON documents with a dynamic schema that are called BSON. BSON format is a data structure composed by value and field. The advantage to use dynamic documents is the possibility to change data structures easily, this allows to keep the modularity obtained by ROS. MongoDB data structure (Figure 4.3) is defined as follows:

- Database that contains one to many collections;

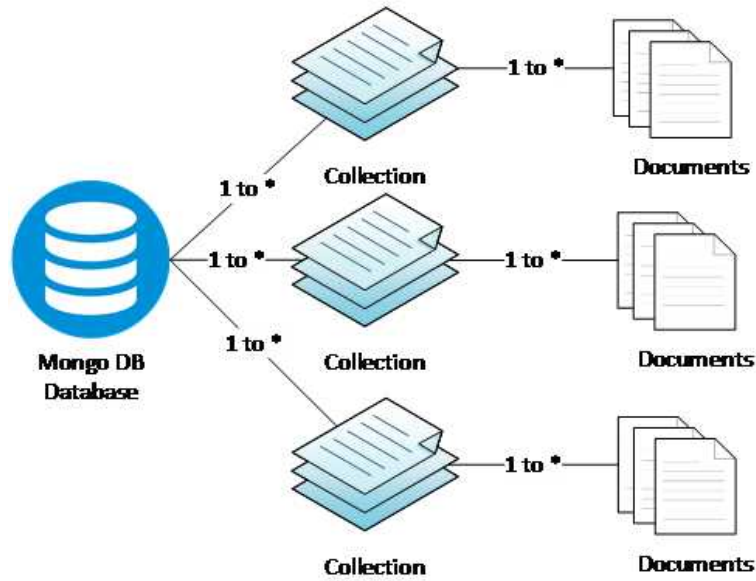


Figure 4.3: MongoDB data structure organization



Figure 4.4: MongoDB document example

- Collection contains one to many documents;
- Document contains a set of fields, each of these fields is a key-value pair where the key is a string and the value can be a value, a document or an array of values as is illustrated in the example in Figure 4.4.

ROS natively allows to interact with MongoDB through `mongodb_store` packet. On MongoDB we have a different collection for each sensor's data,

through C++ or Python scripts it is possible to retrieve all information from database. For our work we have used Python 3.6. The packet has two sub packets with different functions:

- **mongodb_store**: this packet interfaces with database and allows to save and to recovery data;
- **mongodb_log**: this packet that handles data from topic and to save them in the database. To saved document some metadata is added, such as date, name of topic and type of message used by the topic. This metadata is used later to retrieve saved data.

4.3 Scripting and processing

For the work we have used mainly Python language due to its flexibility. In Appendix C we see in detail requirements, input data and output for each script. The only exceptions are the main script that handles the calls to other scripts that has been written in Linux Bash in order to be more compatible with Linux System and scripts to extract Physiological indexes from ECG and SC data that have been written in Matlab[®]. The bash script calls various Matlab[®] and Python scripts in sequence, in Figure 4.5 is represented the Dependencies graph for the flow to obtain probabilities of correlation. The architecture has been projected, as the hardware component, to be able to add physiological indexes and environmental data without changing the flow by just adding new data between MongoDB and the script *feature_extraction.py*. In case of failure of one or more sensors the flow would not fail but would continue the execution with remaining data, obviously at least one physiological index and one environmental index must be present to obtain a result in correlation. The output of single acquisition is compared with output of previous acquisitions, if the acquisition considered is the first or previous acquisitions are not available the script is still able to give the results as output.

4.4 Synchronization

A fundamental part of the whole project is the synchronization of data from various sensors. Most used sensors do not have any protocol to ensure synchronization, the only sensor to have a synchronization protocol is Prosilica GC1020 while Velodyne is equipped with a GPS Garmin from which it ensures an updated acquisition time. To ensure synchronization we have to

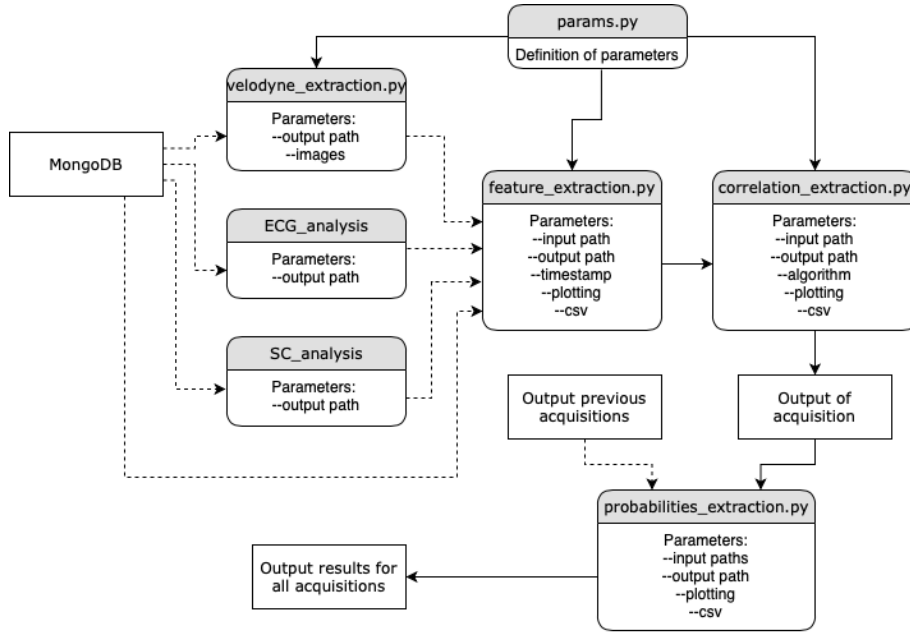


Figure 4.5: Dependencies graph of the order of the scripts called by Bash `data_extraction_for_correlation.sh`: dashed arrows indicate calls that do not prevent the flow if fail or absent, solid lines indicate calls that would block the flow if a failure happens, blank boxes represent data's file and coloured boxes represent scripts.

use a GPS, which using GPS satellites ensures a precise acquisition time. For PC synchronization we have used Network Time Protocol (NTP) which is a synchronization protocol used to synchronize computers in the same network. Functioning is based upon detection latency times of travelling packets on the network. In optimal conditions NTP can ensure an accuracy of $200 \mu\text{s}$. NTP server is organized on different levels (Figure 4.6) from initial stratum that is synchronized with an external time source, as can be GPS for our project, to next stratum that receives time data from previous stratum. Maximum number of stratum is 15, as 16th stratum is used to detect an asynchronous stratum. In order to use NTP in Linux is possible to call daemon `ntpd`. NTP server can be configured in configuration file `/etc/ntpd.conf`. This daemon can both keep the system time updated and estimate the error, correcting it in case of systematic error.

Prosilica GC1020 uses Precision Time Protocol (PTP), this protocol is used to synchronize clocks between computers in a network. PTP has a structure master-slave (Figure 4.7). In this architecture, a system PTP consists in one or more network segments and one or more clocks. An ordinary clock is a device with a singular network connection and either be

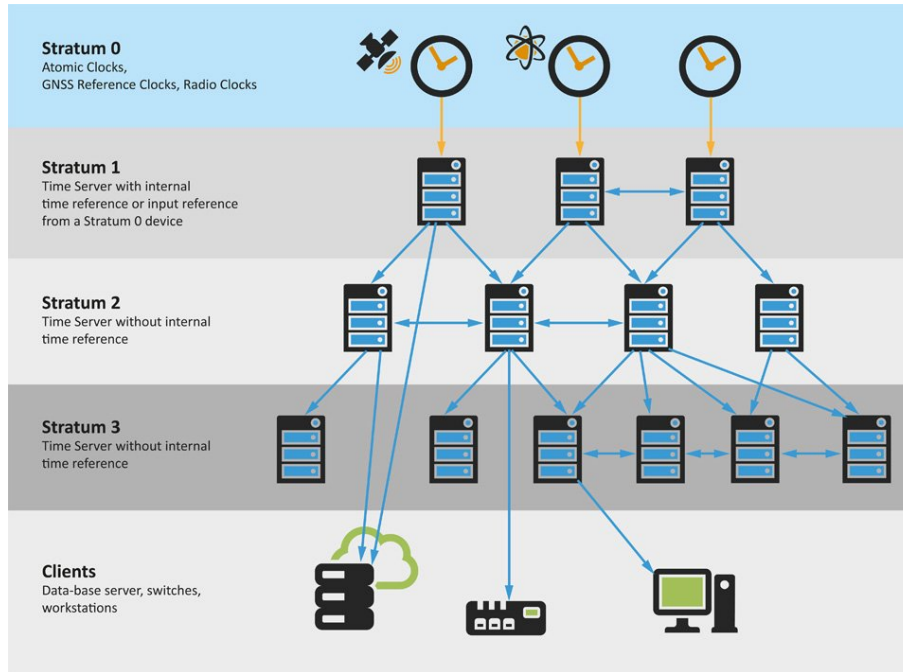


Figure 4.6: Hierarchical structure of NTP

the source (Master) or the recipient (Slave) of synchronization. A boundary clock has multiple network connections and can synchronize one network segment with another one. A simplified PTP system can be built without use of boundary clock, in this case a PTP Grandmaster is elected and all others clocks synchronize directly with it. For our project is possible to see a delay of 33 ms between PC time and Prosilica time, this delay is due to exposure time of camera as the timestamp indicates the starting of acquisition. Considering that Prosilica camera works at 30 fps, it is easy to see how subtracting the acquisition time the delay is almost 0.

Velodyne uses the GPS Garmin to synchronize, but all other sensors that do not have a synchronization protocol use as acquisition timestamp the time defined by PC when data arrives to ROS driver. To decrease error we have synchronized PC time using NTP protocol with a GPS.

4.5 System performance

We have tested the system to check for RAM usage and occupied space over time since each acquisitions lasts up to 20 minutes. In Table 4.2 and in Figure 4.8 are summarized the obtained results, the increase of occupied space fits perfectly a linear interpolation with an increase of 2.2GB for every

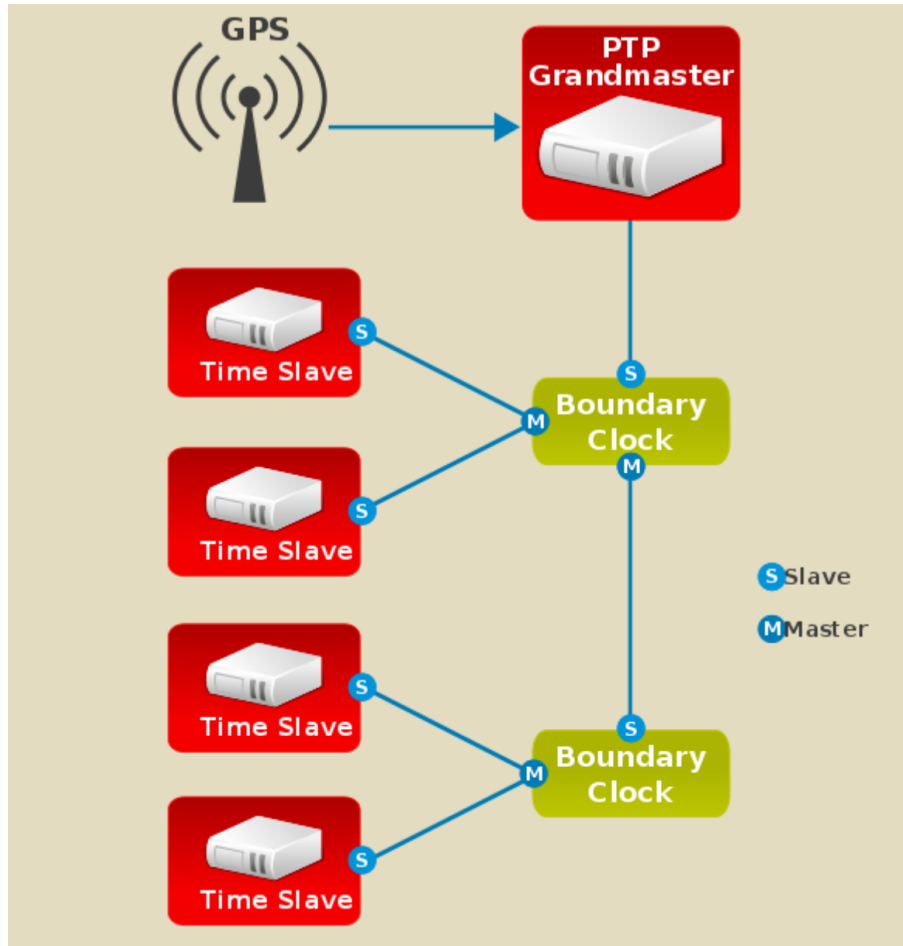


Figure 4.7: Architecture of PTP

Minutes	RAM usage	Occupied space
0 min	5.2 GB	0 GB
5 min	6.2 GB	17.1 GB
10 min	6.2 GB	27.9 GB
15 min	6.3 GB	39.8 GB
20 min	6.2 GB	47.2 GB
25 min	6.2 GB	57.9 GB

Table 4.2: Amount of data needed for the system

minute of the acquisition. The on-board mini-pc has a HDD dimension of 750GB, which means that it can record more than 5 hours of acquisitions. Almost 80% of available RAM (8 GB) has been used for the experiment, so for further acquisitions with more sensors this could be a critical issue.

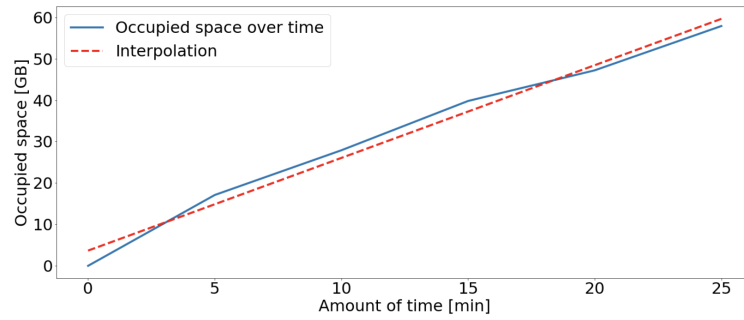


Figure 4.8: Increase of occupied space over time with linear interpolation:
 $y = x * 2.2 + 3.7$

CPU usage was between 35% and 55%, so far below the critical threshold. Detected offset between GPS Yuan10 time given by NTP server and PC time fluctuated between 3 ms and -3 ms with maximum offset of 6 ms.

Chapter 5

Processing pipeline

In this chapter we are going to see the pipeline of the flow of information after the acquisition. Data of acquisition is stored in MongoDB, we need to perform extraction of sensors' data and extraction of new features and physiological indexes from acquired data and filtering. We use different algorithms, performing comparison between them in term of accuracy and of computational complexity, to extract correlation between each feature and different physiological indexes. Once these results have been obtained we consider which of these results have obtained best correlation and compare these results with State of art from Chapter 2.

5.1 Pipeline

The pipeline can be split into two specific parts, the first part of the pipeline, Data preparation, (Figure 5.1) starts from acquisition data stored in MongoDB Database. We need to extract data from database, two different scripts have been used. For Physiological indexes two Matlab[®] scripts have been use to extract data from ECG obtaining time-series of HRV,LF,HF and LFHF, and from Skin Conductance (SC) obtaining time-series of Pha-

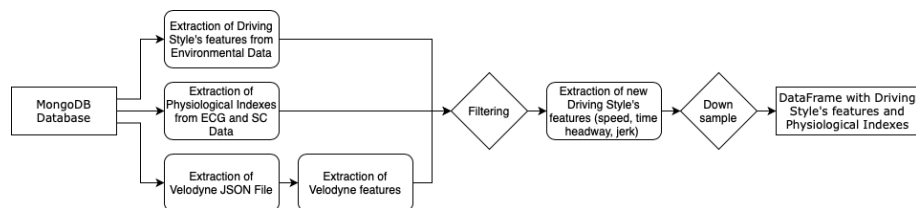


Figure 5.1: Pipeline of processes from MongoDB Database to DataFrame with extracted data

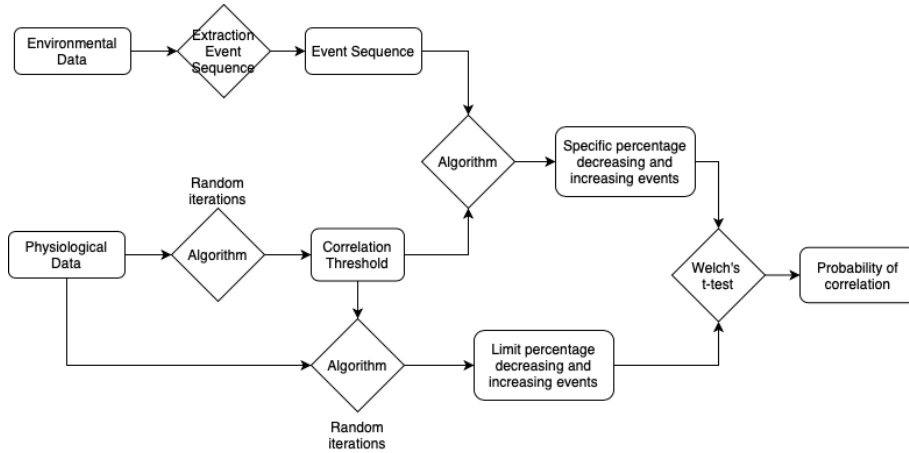


Figure 5.2: Diagram of all steps from Environmental and Physiological Data to Extracted probabilities

sic Component (PC) and Tonic Component (TC). For environmental data we have used one script to extract data from all sensors except Velodyne point cloud, for this we need to extract first of all a JSON file that contains information about identified objects, to do so a NN has been used. Once all these extractions have been performed, data has to be filter in order to remove noise and additional driving style’s features not directly obtainable from sensors can be extracted, such as time-headway, jerk and speed. Down-sampling is performed in order to be able to compare time-series with different sample rate, feature extraction is performed before the down-sample, and down-sample is performed without losing any useful information. The result is saved on a DataFrame, this DataFrame is saved in a CSV in order to be able to retrieve this information without the need to start the process from the beginning.

The second part of the pipeline, Data processing, (Figure 5.2) starts from DataFrame saved in previous part, in this part we perform segmentation of time-series from environmental data in order to transform a time-serie in an event sequence. For Physiological indexes we extract some correlation thresholds in order to understand, by random iterations, the random percentage of events that increase the physiological index. These thresholds are compared with event sequences in order to understand if the environmental data increases or not the percentage of events increasing stress. We obtain a DataFrame with probabilities of correlation between each pair Physiological index and Environmental data, these probabilities need to be compared between more acquisitions in order to detect a stronger correlation.

Method	Time (s)	Car			Pedestrian			Cyclist		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D	0.36	71.09	62.35	55.12	N/A	N/A	N/A	N/A	N/A	N/A
MV3D (LiDAR)	0.24	66.77	52.73	51.31	N/A	N/A	N/A	N/A	N/A	N/A
F-PointNet	0.17	81.20	70.39	62.19	51.21	44.89	40.23	71.96	56.77	50.39
AVOD	0.08	73.59	65.78	58.38	38.28	31.51	26.98	60.11	44.90	38.80
AVOD-FPN	0.1	81.94	71.88	66.38	46.35	39.00	36.58	59.97	46.12	42.36
VoxelNet (LiDAR)	0.23	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND	0.05	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90

Table 5.1: 3D Detection Performance comparison [57]

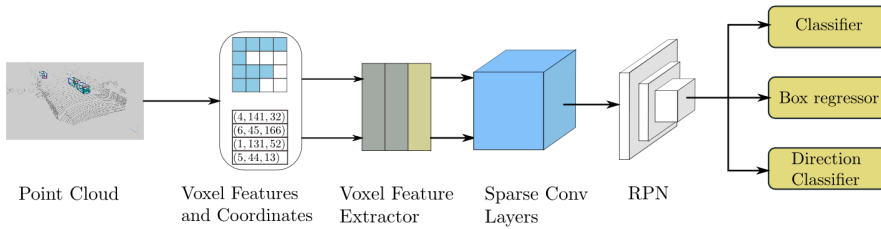


Figure 5.3: Structure of SECOND detector [57]

5.2 Data preparation

We have seen how data is stored on MongoDB, now we see how data has been prepared for processing. Velodyne point cloud's data needs to be extracted in order to be analyzed, a CNN (Convolutional Neural Network) named SECOND has been used as proposed in Yan 2018 [57] study. In this study it has been considered objects of cars, pedestrians and cyclists. The used CNN SECOND has been proven to be much faster than precedent methods as shown in Table 5.1. The structure of SECOND (Figure 5.3) is composed of a voxelwise feature extractor, a sparse convolutional middle layer and an region proposal network. The script `velodyne_extraction.py` extracts information from point cloud into a `json` file that is organized in frames, every frame contained information of each identified object seen as 3D box as follows:

- **box3d_lidar:** Array with information of position in xyz axes, size in xyz axes, rotation on z axis, in Figure 5.4 is possible to see how the axes are positioned.
- **label_preds:** Array with information of which object has been identified, defined by a number. The identified objects with their identification number are: 0 car, 1 bicycle, 2 bus, 3 construction vehicle, 4 motorcycle, 5 pedestrian, 6 traffic cone, 7 trailer, 8 truck and 9 barrier.
- **scores:** Array with information of probability that the identified object is correct.

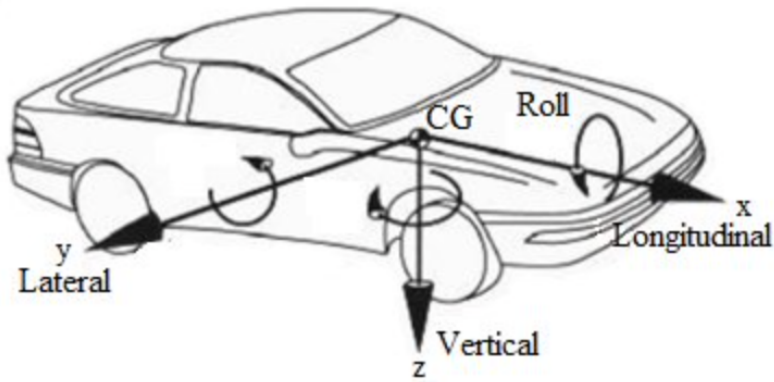


Figure 5.4: Axis position

- **pointcloud_id:** name of bin file from which information has been extracted. The name of bin file is a timestamp in nanoseconds, we extract information of timestamp directly from this name.

An example of a frame with two identified objects, a car and a pedestrian:

- **box3d_lidar:** [[8.977237701416016,10.217676162719727,-1.2461702823638916,1.887690782546997,4.4023356437683105,1.5308451652526855,1.556740164756775],[9.452740669250488,7.507191181182861,-1.010738492012024,1.9031622409820557,4.432695388793945,1.5714941024780273,4.6955342292785645]]
- **label_preds:** [0,5]
- **scores:** [0.8672887086868286, 0.7873099446296692]
- **pointcloud_id:** LIDAR_TOP/1544711841262000000.bin

From the acquisition we have extracted position of identified objects and distance from car. In Figure 5.5 it is possible to see every identified object with a score greater than 0.6 and maximum distance of 20 m. Figure 5.6 shows the nearest identified object with a score greater than 0.6 and maximum distance of 20 m for every timestamp. Figure 5.7 illustrates how many identified objects are present with a score greater than 0.6 and maximum distance of 20 m in each frame.

We need than to extract some features from available data:

- **Speed:** extracted from linear velocities in Velodyne Data as:

$$\sqrt{\text{linear}.x^2 + \text{linear}.y^2} \quad (5.1)$$

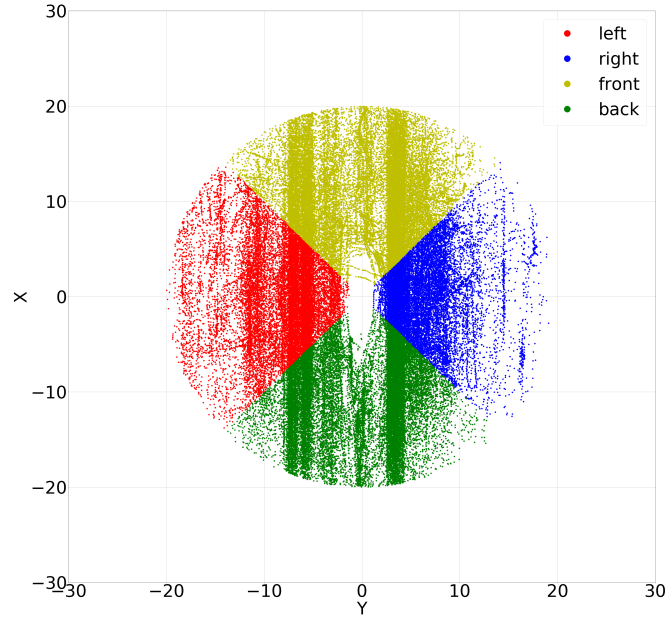


Figure 5.5: Relative position of identified objects, defining front, back, right and left position of acquisition on 2018/12/13

- **Time headway:** time difference between the nearest identified object and is calculated as:

$$\frac{\text{distance of nearest object}}{\text{speed of car}} \quad (5.2)$$

- **Jerk:** first derivative of acceleration, indicates the amount of sudden movements

For HRV extraction we have used Python *Neurokit2*[58] module, which is a widely used package used to extract physiological indexes. The used function is `nk.ecg_process(ecg_signal, sampling_rate = sample_rate)` with `sample_rate` being the sample rate used for the acquisition and `ecg_signal` the time serie obtained from acquisition.

We need to check the alignment of timestamp between all sensors to Unix timestamp as not all sensors are automatically aligned to it as an example the Pupil Eye Tracker. To do so we applied the formula to all timestamps

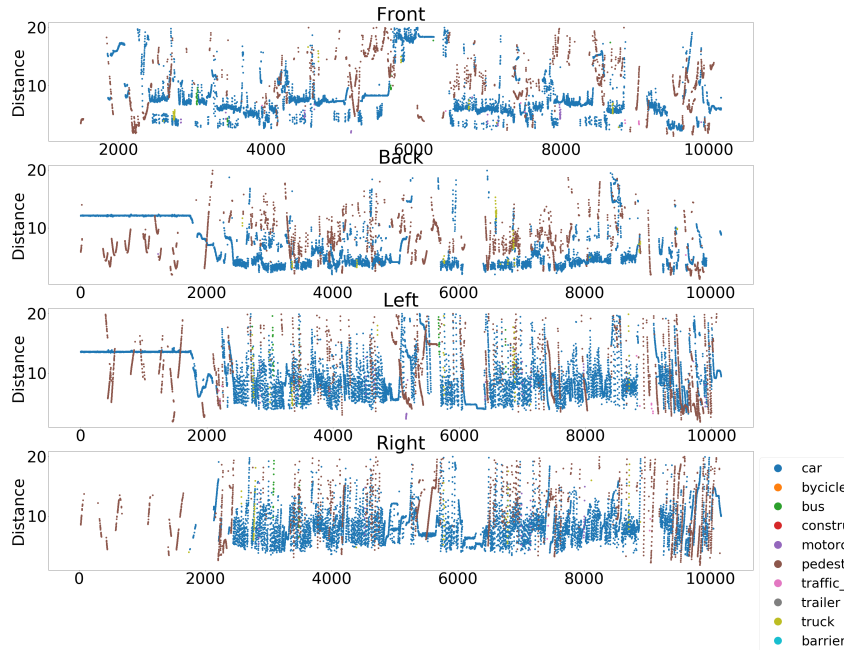


Figure 5.6: Distance of nearest identified object for category of acquisition on 2018/12/13

in acquisition as:

$$t_{aligned}[i] = t_{original}[i] + t_{Unix}[0] - t_{original}[0] \quad (5.3)$$

$$i \in 0 \dots T$$

As we have seen in Chapter 3 different sensors have different sampling frequency, in order to prepare data to correlation analysis we need every sensors' data to have same sampling frequency and this frequency to be fixed. Figure 5.8 shows that the frequency is not fixed but varies throughout the acquisition.

We merged all sensors data and down-sample this merge at 100 Hz, which is the most common frequency of sensors. The frequency of 100 Hz is much higher than the needed for physiological data, which changes with an average frequency of 1 Hz, this allows us not to lose any useful information about body sensors. The down-sample to 100Hz allows us not to lose any useful information in sensors with lower and higher sample rate. Due to down-sampling we interpolate data when there are missing values only if the missing is surrounded by valid data, missing information after interpolation is considered as NaN from the script.

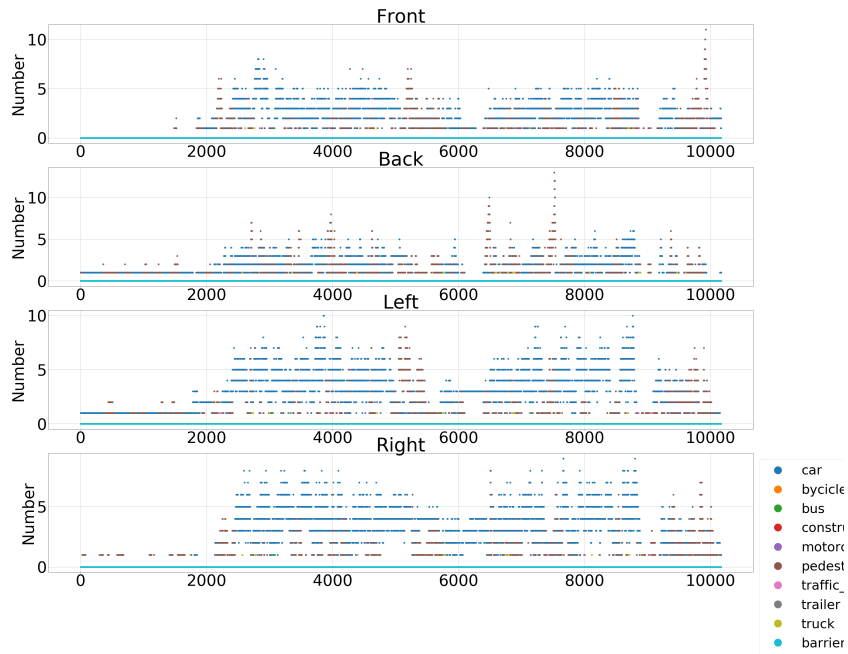


Figure 5.7: Number of identified objects in each frame for category of acquisition on 2018/12/13

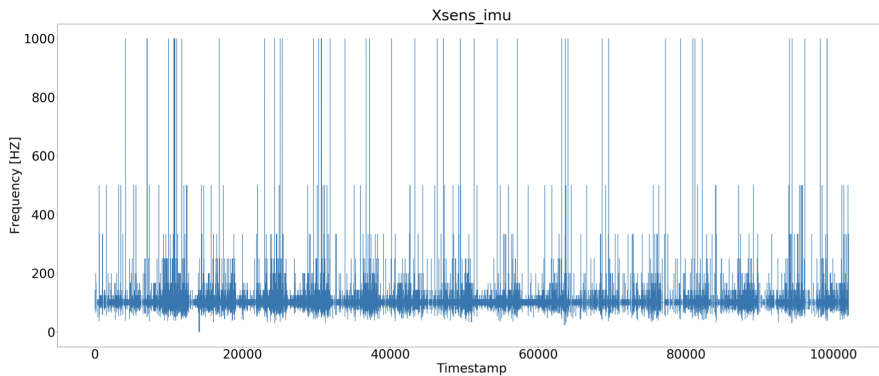


Figure 5.8: Frequency analysis of Xsens IMU

At the end of the acquisition ROS shuts down the nodes in random sequence, taking up to 5 seconds to shut down all nodes. In this condition the closed nodes information are NaN, while open nodes continue the acquisition. The script avoids this situation by considering the acquisition over the moment one of the nodes is shut down in shutting down operation.

5.3 Data processing

In order to find a solution to the problems explained in Chapter 2 we need firstly to understand the level of correlation between driver's physiological data and car's environmental data. Many of next definitions can be found in *An Introduction to Statistical Learning*[59] written by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani and *Pattern Recognition and Machine Learning*[60] written by Christopher M. Bishop.

The problem we are considering is an inference problem, we need to understand which, between available factors, might be the cause of increase in driver's stress. We do not know yet how the relationship between environmental and physiological data could be summarized but we expect the relationship to be much more complicated than a linear equation considering all physiological processes that happens in human body during a stressful situation.

We are dealing with a non-parametric model as we do not make any assumption about the relationship form, the disadvantage is that we may need much more acquisitions but on the other hand it has been proven in Chapter 2 that physiological process can be very complicated.

The learning problem for our work is unsupervised, which means that we do not use any control a priori to check for valid relationships. The reason for this choice is that we do not want to introduce a bias that could penalize our results as the only way to check for driver's stress is through questionnaires or using tools to allow driver to indicate the level of stress as shown in Chapter 2 and this is linked to driver's perception of stress that could differ from user to user. We still use two different questionnaires, one proposed before the driving and one after the driving in Appendix D we have inserted the proposed questionnaire.

The problem, at first step, can be considered as a Classification problem as we want to know which feature affect level of stress in driver and on second step as a Regression as we want to know which values of single feature affect and in which way the level of stress. Considering these assumptions and data saved on MongoDB Database these are the features we are considering:

- **Angular velocity:** obtained from Xsens IMU data on axes xyz;
- **Linear acceleration:** obtained from Xsens IMU data on axes xyz;
- **Jerk:** extracted from Xsens IMU data on axes xyz as first derivate of linear acceleration;
- **Linear velocity:** obtained from Velodyne data on axes xyz;

- **Angular velocity:** obtained from Velodyne data on axes xyz;
- **Linear acceleration:** obtained from Velodyne data on axes xyz;
- **Number of identified objects:** extracted by Velodyne point cloud, identified objects are cars, bicycles, buses, construction vehicles, motorcycles, pedestrians, traffic cones, trailers, trucks and barriers. Position of objects is divided in back, front, left and right;
- **Distance of nearest identified object:** extracted by Velodyne point cloud, identified objects are the same as above and position is divided in back, front, left and right;
- **Speed:** extracted by Velodyne data;
- **Timeheadway:** extracted from Velodyne point cloud and speed.

Some of these features are obtained from both Xsens IMU and Velodyne to check if data is consistent. Other features could be not considered as too few data in the acquisition, this is considered in the part of preparation. For physiological data these are the indexes that we are considering:

- **ECG:** not elaborated data of ECG;
- **SC:** not elaborated data of SC;
- **LF:** norm of LF extracted from ECG;
- **HF:** norm of HF extracted from ECG;
- **LFHF:** ratio LFHF extracted from ECG;
- **Tonic signal:** TC of SC;
- **Phasic signal:** Phasic Component of Skin Conductance;
- **SC signal:** sum of tonic and PCs of SC;
- **HRV:** extracted from ECG.

As we have seen in Chapter 2 not all these indexes are useful and each of this index must be considered in a different way, we are also able to check studies referenced in Chapter 2.

Once obtained physiological data from body sensors and environmental data from car sensors we need to find, if it exists, a correlation between these two samples. We consider N physiological indexes one by one and comparing them with M driving features one by one, we could obtain N x M level of correlations.

5.3.1 Extraction of Event Sequence

As described in book “Detection of Abrupt Changes: Theory and Application” [54], our problem is an Off-line Hypothesis Test. We need to consider two hypothesis “*with change*” and “*without change*” in time-serie at occurring of the event. Given a finite sample y_1, \dots, y_N , and p_θ the sequence parameters, we need to test between:

$$\begin{aligned}
 H_0 : & \text{for } 1 \leq k \leq N : \\
 & p_\theta(y_k|y_{k-1}, \dots, y_1) = p_{\theta_0}(y_k|y_{k-1}, \dots, y_1) \\
 H_1 : & \text{with } t_0 \in (1 \dots N) \text{ such that :} \\
 & \text{for } 1 \leq k \leq t_0 - 1 : p_\theta(y_k|y_{k-1}, \dots, y_1) = p_{\theta'_0}(y_k|y_{k-1}, \dots, y_1) \\
 & \text{for } t_0 \leq k \leq N : p_\theta(y_k|y_{k-1}, \dots, y_1) = p_{\theta_1}(y_k|y_{k-1}, \dots, y_1) \quad (5.4)
 \end{aligned}$$

In our problem t_0 is defined by the extraction of Event Sequence (ES). The criterion used in hypothesis testing is a trade-off between the ability to detect actual changes when they occur and the ability not to detect changes when no change occurs. We need to define the threshold of this trade-off. We need to consider five performances indexes for change detection algorithms:

- mean time between false alarms;
- probability of false detections;
- mean delay for detection;
- probability of nondetection;
- accuracy of the change time and magnitued estimates.

Each algorithm has also different robustness to noise and modelling errors, this is a characteristic that we must consider. As shown in Chapter 2 we considered Luo studies [49] to find the correlation between driver’s physiological indexes and environmental data. We need first of all to define the events in environmental data, as we do not have any kind of information of which thresholds to use to define these events we use percentiles with respect to all acquisition. The used algorithm to obtain sequence of events from Time-Serie (TS) is defined in Algorithm 1.

An example of an extracted ES can be found in Figure 5.9 as red vertical lines indicate the found events. In the example it was considered a 99th percentile, with steps of 1th percentile. The threshold is 11.17 m/s^2 . In *params.py* we have defined an array of all used percentiles, while 0th and 100th are not significant as they would only consider, respectively, the global

Algorithm 1 Transform TS to ES

```
1: return Events
2: require TS = time-serie with timestamps, Percentile, PercentileStep,
   min = minimum time between two events from 0 to T
3: define Events = [ ]
4: if Percentile > 50 then
5:   CurrentPercentile = 100
6:   while CurrentPercentile > Percentile do
7:     for i in 0..T do
8:       if TS[i] >= CurrentPercentile of TS then
9:         Events  $\leftarrow$  i
10:        i+ = min
11:       end if
12:     end for
13:     CurrentPercentile - = PercentileStep
14:   end while
15: else
16:   CurrentPercentile = 0
17:   while CurrentPercentile < Percentile do
18:     for i in 0..T do
19:       if TS[i] <= CurrentPercentile of TS then
20:         Events  $\leftarrow$  i
21:         i+ = min
22:       end if
23:     end for
24:     CurrentPercentile + = PercentileStep
25:   end while
26: end if
27: Drop Duplicates of Events
```

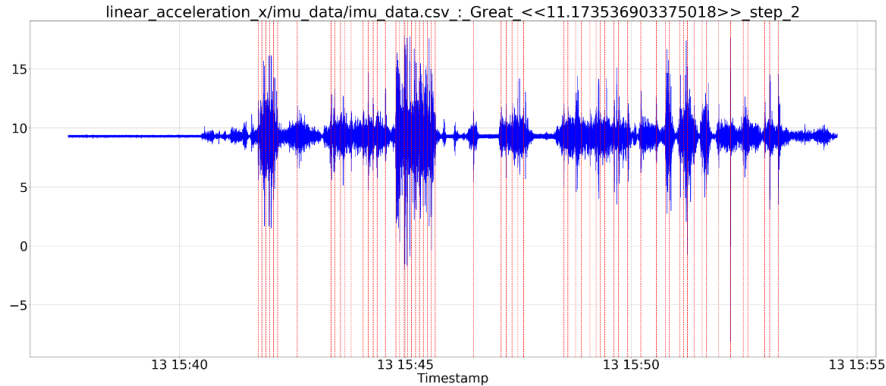


Figure 5.9: Plot of acceleration from IMU Data with found events at 99 percentile

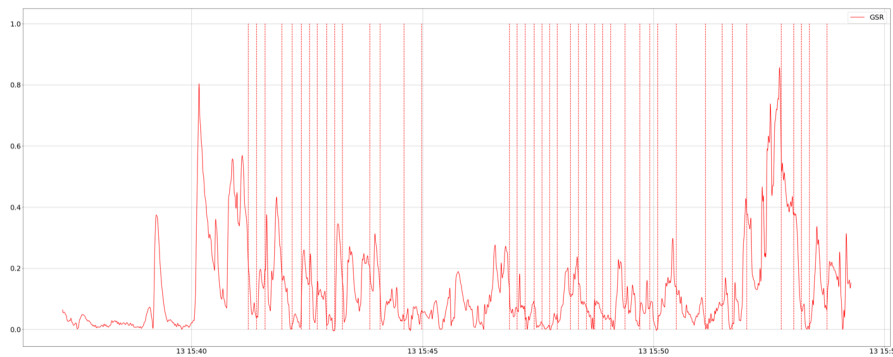


Figure 5.10: Event Sequence (red vertical lines) on Phasic Component (PC) Time-Series (TS)

minimum and maximum, we have used from 1st to 45th and from 99th to 55th percentiles. Values of *min* and *PercentileStep* are defined in *params.py* and not defined by user. In Figure 5.10 is possible to see the ES extracted from nearest car on front side with threshold of 5 m defining on TS of PC from SC physiological data.

As shown in Luo study [49] we compare the sub-sequence before and after the event occurred. As we have seen in Chapter 2 each physiological index has different time characteristics, we then apply some rules to identify rear and front sub-sequences to be compared:

- **Latency:** defined as the time between the end of event and the start of the effect on physiological index, if 0 it means that the change in physiological index is considered immediately when the event occurs. Latency cannot be negative for definition;
- **Duration:** defined as the time window needed for the physiological

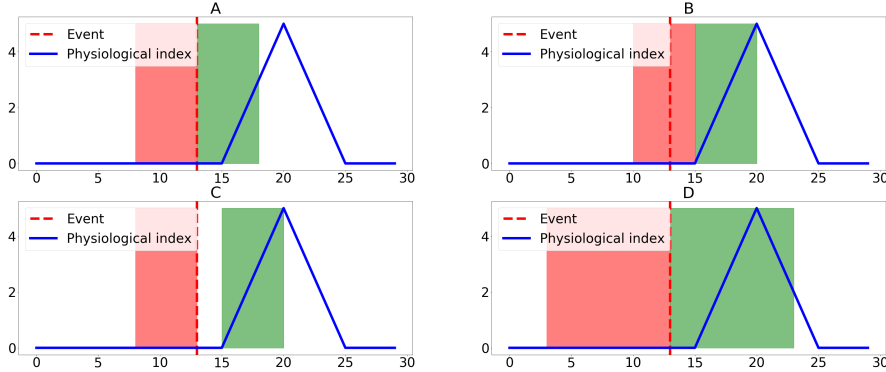


Figure 5.11: Example of rules on time characteristics: A 0s Latency, 0s Shift 5s Duration; B 0s Latency, +1s Shift 5s Duration; C 1s Latency, 0s Shift 5s Duration; D 0s Latency, 0s Shift 10s Duration. Front sub-sequence is defined by background color red, rear sub-sequence by background color green

index to change, rear and front sub-sequences must always have same size to be comparable;

- **Shift:** defined as a shift in time between when the event occurred and when it must be considered, it can be either negative which means that the physiological index changes before the event occurs as can happen when the body reacts before an event occurred (for example before braking) or positive which means that the physiological index changes after the event occurs as the physiological index changes after a certain amount of time.

In Figure 5.11 is shown an example of latency, duration and shift use. In `params.py` we have defined 3 arrays with values of latencies, delays and durations. After extraction of ESs we need to extract segmentation of the time-serie.

5.4 Algorithms definition

Once obtained the ES, as shown in Algorithm 1, we apply it to every TSs with different percentiles, latencies, shifts and durations applying Algorithm 2. Considering Equation 5.4 we need to consider different situations:

- θ_1 is known, in this case we have information of what implicates the change;
- few information of θ_1 is known, in this case we do not exactly know the value of θ_1 but we have some information about the change

Algorithm 2 Apply percentiles and time characteristics

```
1: result Correlations
2: require TS = time-serie with timestamps from 0 to T
3: define Arrays of percentiles[0..P],latencies[0..L],shifts[0..S] and durations[0..D], Correlations=[ ]
4: for percentile in percentiles do
5:   Events = Algorithm 1(TS,percentile)
6:   for latency in latencies do
7:     for shift in shifts do
8:       for duration in durations do
9:         FrontStart = -duration + shift
10:        FrontEnd = +shift
11:        RearStart = +shift + delay
12:        RearEnd = +duration + shift + delay
13:        for event in Events do
14:          Front = TS[event + FrontStart : event + FrontEnd]
15:          Rear = TS[event + RearStart : event + RearEnd]
16:          Correlations ← Algorithm(Front,Rear)
17:        end for
18:      end for
19:    end for
20:  end for
21: end for
```

- θ_1 is unknown, in this case no information is known and this complicates detection algorithms.

In our problem θ_1 is unknown, so we need to consider algorithms that allow us to detect a change without knowing what this change means exactly. We need also to consider two different situations:

- Additive changes, changes do not affect the whole system and they can be considered one at a time;
- Non Additive changes, changes do affect the whole system, this means that after a change the system does not return to initial state but to a changed one.

In general changes in our problem changes are Non Additive, as different kind of stresses can sum up, from short to long term. If we consider T_0 as initial state of body stress of driver, when a stressful event occurs, due to

short-term stress, the state changes in T_1 . After a certain amount of time, depending on the user and on which physiological index we are considering, the influence of short-term stress decreases but longer term stress could remain transforming body stress state to T_0 . In our problem we are just considering short-term stress and not the influence in longer term stress, this allows us to consider changes as Additive as long as the time between events is equal or greater to the duration of short-term stress.

In Additive changes we can define four cases:

- θ_0 and θ_1 are known;
- θ_0 is known and the magnitude of change is known but not its direction is known;
- θ_0 is known and the direction of the change is known but not its magnitude;
- θ_0 is known but nothing is known about θ_1 .

In our problem we can extrapolate from physiological characteristics of indexes the direction of the change. In Chapter 2 we have seen how a stressful event causes the activation of SNS and increase of HF, but we do not know, and most importantly it depends of each user, how much it changes. This means that our problem can be defined as we know the direction but not the magnitude of the change. Our algorithms need then to be able to consider each occurred event as a singular event, it must search for increase or decrease, depending on used physiological indexes, and give as an output the number of events that caused an abrupt change in time-series.

We look in detail the algorithms used to compare Rear and Front subsequences, which must always be of the same length, in Algorithm 2 are the ones defined in Chapter 2:

- **PCC**;
- **ARMA**;
- **NNSBM**;
- **ET**.

5.4.1 Pearson Correlation Coefficient

PCC is a statistic that measures correlation between two variables X and Y , for our work we consider the population PCC which is defined as follows:

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (5.5)$$

where

$$\text{cov}X, Y = \exp[(X - \mu_X)(Y - \mu_Y)] \quad (5.6)$$

- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y
- μ_X is the mean of X
- μ_Y is the mean of Y

The correlation coefficient ranges from -1 to 1, a value of 1 implies a linear correlation between X and Y, while a value of -1 implies a total negative linear correlation and a value of 0 implies no linear correlation. PCC is weak to outliers and might not be efficient if the population is not normal. Python already has a built-in function `pearsonr(x, y)` in `scipy.stats` module.

5.4.2 ARMA Model

ARMA Models are widely used in forecasting of time-series. For our problem we compare prediction, based on Front sub-sequence's data, and Rear sub-sequence's data. The value of error between the two indicates the probability of change in time-series. The AR (Auto-Regressive) part indicates that the evolving variable of interest is regressed on its own lagged values. The MA (Moving Average) part indicates that the regression error is actually a linear combination of error terms.

Given a time-series X_t an $ARMA(p, q)$ Model is defined as:

$$X_t - \alpha_1 X_{t-1} - \dots - \alpha_p = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_p \varepsilon_{t-q} \quad (5.7)$$

To compare prediction vs Rear sub-sequence we use RMSE, which is defined as:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(X_t - X_0)^2}{n}} \quad (5.8)$$

with X_t being the forecast observation (prediction) and X_0 the actual value (Rear sub-sequence). We need to define parameters of ARMA, in Subsection 5.6 we compare results using different parameters. In Figure 5.12 we can see two different situations, on the right graph the time-serie has an abrupt

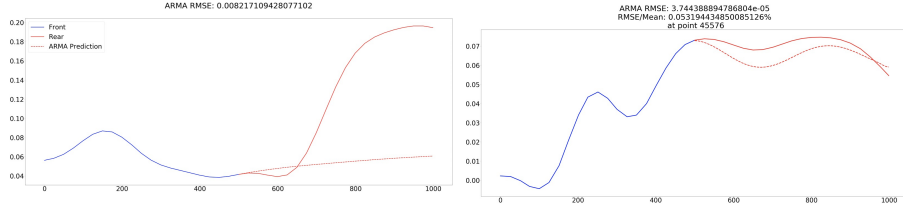


Figure 5.12: Comparison of two different sub-sequences with ARMA Model. On the left time-series has an abrupt change, on the right time-series has no abrupt change

change and it is possible to see how the prediction cannot fit the Rear sub-sequence, on the left graph there is no abrupt change in the time-series and prediction is able to fit much better the Rear sub-sequence. RMSE is 160 times higher on the left with respect to the example with no abrupt change. Python already has a built-in function $ARMA(array, (p, q))$ in *statsmodels* module.

5.4.3 Nearest Neighbour Statistical Based Method

There exist a substantial number of non-parametric methods based on KNN, the one presented in this section is a distribution-free test for the general multivariate two-sample problem.

Let X_1, \dots, X_{n_1} and Y_1, \dots, Y_{n_2} be independent random samples of \mathfrak{R}^d from unknown distributions $F(x)$ and $G(x)$, respectively, with corresponding densities $f(x)$ and $g(x)$. The Two-sample problem is the test between hypothesis $H_0 : F(X) = G(X)$ against the hypothesis $H_1 : F(x) \neq G(X)$. Take $n = n_1 + n_2$, $\Omega_1 = \{1, \dots, n_1\}$ and $\Omega_2 = \{n_1 + 1, \dots, n\}$ and label the pooled samples as Z_1, \dots, Z_n where:

$$\begin{aligned} Z_i &= X_i \quad i \in \Omega_1 \\ &= Y_{i-n_1} \quad i \in \Omega_2 \end{aligned}$$

$NN_i(r)$ represents the r th nearest neighbour to the sample point Z_i .

Define:

$$\begin{aligned} I_i(r) &= 1 \text{ if } NN_i(r) \text{ belongs to same sample as } Z_i \\ &= 0 \text{ otherwise} \end{aligned}$$

The statistic considered for testing H is the quantity:

$$T_{k,n} = \frac{1}{nk} \sum_{i=1}^n \sum_{r=1}^k I_i(r) \quad (5.9)$$

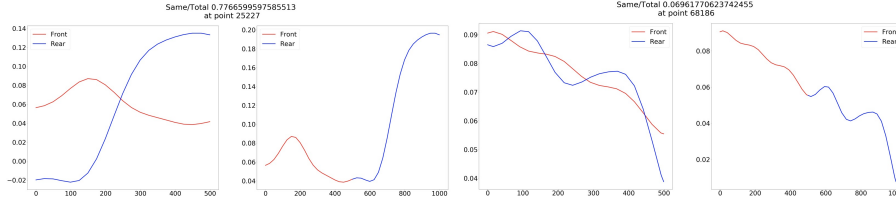


Figure 5.13: Comparison of two different sub-sequences with NNSBM Algorithm. On the left time-series has an abrupt change, on the right time-series has no abrupt change

which is the proportion of all k nearest neighbour comparisons in which a point and its neighbour are members of the same sample. $T_{k,n}$ is small under null hypothesis H_0 and large when the two sub-sequences are different.

In Figure 5.13 we can see Front and Rear sub-sequences compares using NNSBM Algorithm, on the right the two sub-sequences overlap much more then on the left, this can be seen also in the correlation coefficient which is 10 times higher on the left, when an abrupt change is present, as Nearest Neighbour is most probably in the same sample from the considered point.

Python has some Nearest Neighbour functions in *sklearn* that we have rewritten in order to optimize our NNSBM.

5.4.4 Energy Test

Energy statistics are functions of distances between statistical observations. The concept is based on Newton's gravitational potential energy. The idea of energy statistics is to consider statistical observations as bodies governed by a statistical potential energy, which is zero if and only if null hypothesis is true. For our problem we are going to consider the two-sample energy statistic $\varepsilon(X, Y)$ for independent random samples $X = X_1, \dots, X_{n_1}$ and $Y = Y_1, \dots, Y_{n_2}$:

$$\varepsilon_{n_1, n_2}(X, Y) = \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{m=1}^{n_2} |X_i - Y_m| - \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} |X_i - X_j| - \frac{1}{n_2^2} \sum_{l=1}^{n_2} \sum_{m=1}^{n_2} |X_l - X_m| \quad (5.10)$$

The statistic:

$$T_{n_1, n_2} = (n_1 n_2 / (n_1 + n_2)) \varepsilon_{n_1, n_2} \quad (5.11)$$

is applied to test equality of distributions X and Y , H_0 null hypothesis is rejected for large values of T_{n_1, n_2} .

AUC Range	Classification
$0.9 < AUC < 1.0$	Excellent
$0.8 < AUC < 0.9$	Good
$0.7 < AUC < 0.8$	Worthless
$0.6 < AUC < 0.7$	Not Good

Table 5.2: Accuracy classification by AUC

5.5 Algorithms comparison

Algorithms have been compared using a set of arrays extracted from acquisitions with and without abrupt changes. We compare accuracy and complexity.

5.5.1 Accuracy

We have then compared output of each algorithm with true positives and true negatives and building ROC Curves and calculating AUC [61]. In ROC we have calculated TPR (True Positive Rate) as:

$$TPR = TP / (TP + FN) \quad (5.12)$$

and FPR (False Positive Rate) as:

$$FPR = FP / (FP + TN) \quad (5.13)$$

while AUC is the area beneath the curve as:

$$AUC = \int_0^1 ROC(t) dt \quad (5.14)$$

The closer to 1 is AUC the more accurate is the test, while a value around 0.5 indicates an accuracy similar to a random test. In Table AUC we can see a classification for AUC accuracy. We do not have any parameters in PCC and ET algorithms. In Figure 5.14 we can see results for PCC algorithm, the resulting AUC is 0.77, which is not a sufficient result for the base algorithm. For ET the result is 0.91, which is an excellent result. For ARMA we need to define p and q parameters, we have tested from 0 to 5. In Table 5.3 we present AUC results for each pair (p,q). Best results have been obtained with ARMA(4,2) with a AUC value of 0.90, for our problem we use this model.

In Figure 5.15 we can see ROC Curve for ARMA(4,2). For NNSBM we need to define k, we have tested from 1 to 10. In Figure 5.16 we can see all AUC results, best result is obtained with k = 4 with a value of 0.889, for our problem we use k = 4 In Figure 5.17 we can see ROC Curve for 4-NNSBM.

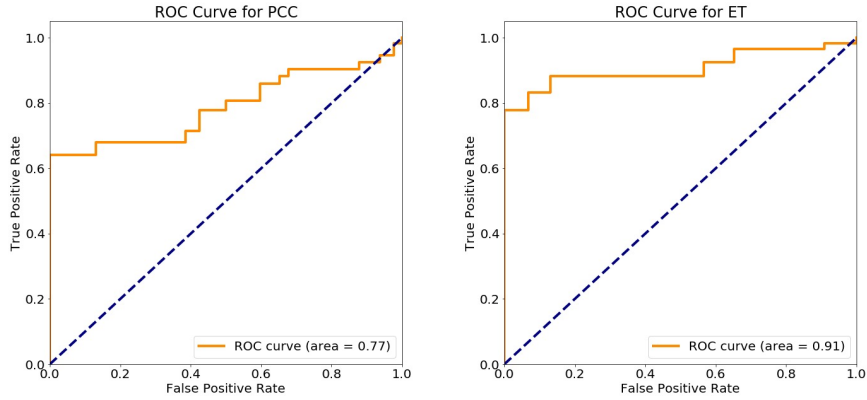


Figure 5.14: ROC curves with relative AUC for PCC and ET algorithms

$p \backslash q$	0	1	2	3	4	5
0	-	0.75	0.56	0.52	0.56	0.77
1	0.78	0.79	0.80	0.75	0.70	0.64
2	0.79	0.79	0.81	0.69	0.63	0.56
3	0.84	0.85	0.87	0.82	0.79	0.61
4	0.79	0.83	0.90	0.77	0.70	0.59
5	0.77	0.85	0.82	0.74	0.72	0.61

Table 5.3: AUC results for ARMA(p,q)

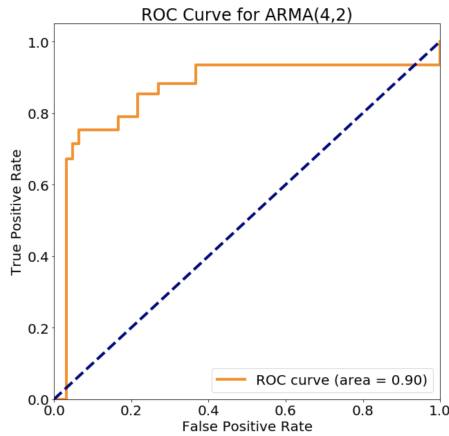


Figure 5.15: ROC curve with relative AUC for ARMA algorithms

Algorithms ET, NNSBM and ARMA have obtained similar results in term of accuracy, while PCC has obtained very low accuracy.

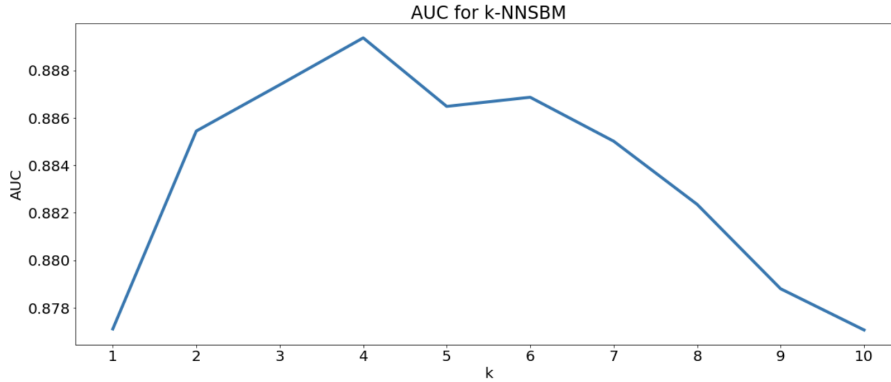


Figure 5.16: AUC value for every k-NNSBM

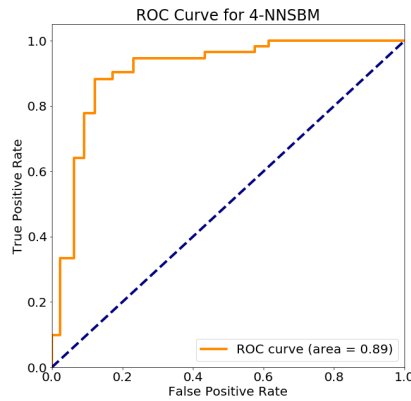


Figure 5.17: ROC curve with relative AUC for 4-NNSBM

5.5.2 Computational complexity

In this section we consider complexity applied to a single sequence of length n , comparing Front sub-sequence and Rear sub-sequence. In Chapter 6 we consider complexity applied to the acquisition.

- **PCC:** the complexity is linear $f(n) = O(n)$, as it grows linearly with the size of sample when calculating standard deviations and covariance;
- **ET:** the complexity is quadratic $f((\frac{2n}{2})^2) = O(n^2)$, as the equation 5.10 has two nested loops on $\frac{n}{2}$;
- **ARMA:** complexity of model depends on parameters (p, q) , the model needs to build the forecast $f(n) = (p + q) * (\frac{n}{2})^2$ and then compare

it with Rear sub-sequence with RMSE $f(n) = n$ for a complexity of $O((p + q)n^2)$;

- **NNSBM:** complexity of model depends on parameter k , we have succeeded in optimize it not to compare every point of the sequence but only a smaller subset that contains nearest neighbours. The algorithm checks every n point with the k nearest before and after in same sub-sequence and in the other sub-sequence $f(4k(\frac{n}{2}))$, then repeats the procedure with $k = k + 1$ from 1 until goal k is reached $f(\frac{n}{2} \frac{(4k)^2 + 4k}{2})$ which leads to $O(k^2n)$. If all pairs were to be compared the complexity would be $O(kn^2)$.

We have ran some tests to check complexity. The test is a run of 1000 randomly selected sequences with a length of $n = 1000$, Front and Rear sub-sequences of $n_{1,2} = 500$, these are the results in average for a single execution:

- **PCC:** $84 * 10^{-6}$ seconds
- **NNSBM:** $13 * 10^{-3}$ seconds
- **ET:** $124 * 10^{-3}$ seconds
- **ARMA:** $312 * 10^{-3}$ seconds

Results are in line with complexities. As we have seen in previous subsection the algorithm PCC had a very low accuracy, despite the lowest complexity we cannot consider it as the best algorithm. The remaining algorithms had same level of accuracy but in terms of complexity the algorithm NNSBM is the best one for one order, so it is the main used.

5.6 Correlation extraction

From considered algorithms we obtain a value of correlation between Rear and Front sub-sequences, but this alone might not be enough to understand if there is been a change in TSs after the event occurred. In order to understand this we need to sample whole acquisition and obtain an average level of correlation. In order to do so we use Algorithm 3 In Figure 5.18 an example of results using Algorithm NNSBM on different TSs, it is possible to see how each TSs has a different average correlation. We applied the CLT in Algorithm 3 as correlations of random Front and Rear sub-sequences are independent their sum tends toward a normal distribution. We defined a

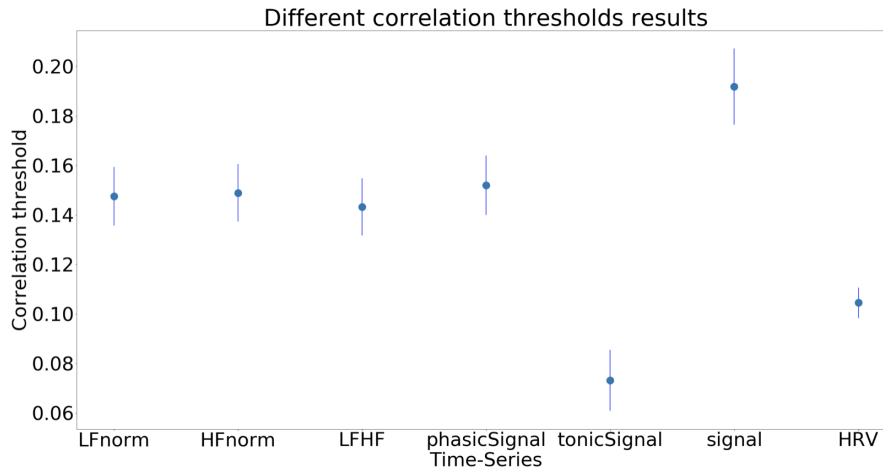


Figure 5.18: Results with Algorithm NNSBM with some TSs

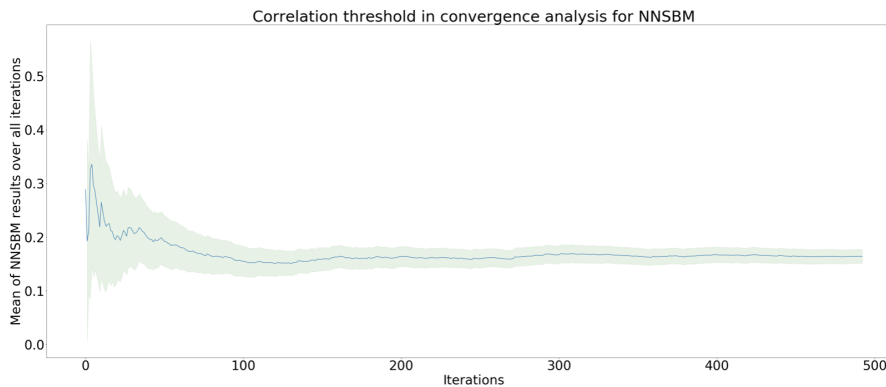


Figure 5.19: Example of defining the limit for average correlation, green background indicates the CI

minimum number of iterations with *minIterations*, which can be the standard 30, and a maximum number of iterations with *maxIterations*, which can be the 5% of all population. The obtained sample mean correlation is a good estimator of population mean. The algorithm stops when maximum number of iterations is reached or when the Confidence Interval (CI) is sufficient small. All these values are defined in *params.py*. In Figure 5.19 is possible to see an example of Algorithm 3 applied. Once obtained the average level of correlation of the population we need to understand on average if the TSs increases or decreases.

In order to extract this information we apply Algorithm 4 Algorithm 4 compares random Front and Rear sub-sequences, if level of correlation if below the average level it might mean that the event occurred cause a

Algorithm 3 Extraction of average level of correlation

```
1: result AverageCorrelation
2: require TS = time-serie with timestamps from 0 to T, Latency, Shift,
   Duration, maxIterations, minIterations, maxConfidence
3: define Correlations=[ ]
4: FrontStart = -duration + shift
5: FrontEnd = +shift
6: RearStart = +shift + delay
7: RearEnd = +duration + shift + delay
8: for iterations in [0...maxIterations] do
9:   r = Random i ∈ [0...T]
10:  Front = TS[r + FrontStart : r + FrontEnd]
11:  Rear = TS[r + RearStart : r + RearEnd]
12:  Correlations ← Algorithm(Front, Rear)
13:  if iterations > minIterations then
14:    if Confidence(Correlations) ≤ maxConfidence then
15:      break loop
16:    end if
17:  end if
18: end for
19: AverageCorrelation = average(Correlations)
```

change between Front and Rear. It then compares the two sub-sequences to check if it was present an increase or decrease and adds result to an array, to compare the sub-sequences a t_{score} has to be calculated using Equation 5.15, if $t_{score} < -\alpha$ with α predefined in *params.py* Event cause a decrease in TS, while if $t_{score} > \alpha$ Event cause an increase in TS, if $t_{score} < |\alpha|$ it is impossible to say if the Event cause an increase or decrease but this should not happen due to Algorithm used to check for correlation.

$$t_{score} = \frac{\mu_{front} - \mu_{rear}}{\sqrt{\frac{\sigma_{front}^2 + \sigma_{rear}^2}{n}}} \quad (5.15)$$

In Figure 5.20 is illustrated an example of application of Algorithm 4. In Figure 5.21 we have checked that the real value of average correlation and average increasing and decreasing probabilities calculated is between the obtained CI.

Once obtained average increasing and decreasing probabilities we need to obtain exact increasing and decreasing probability for each time-characteristic, this can be obtained using Algorithm 2 and Algorithm 4 where instead of

Algorithm 4 Extraction of average probability of events increasing and decreasing TS

```

1: result AverageIncreasing, AverageDecreasing
2: require TS = time-serie with timestamps from 0 to T, Latency, Shift,
   Duration, maxIterations, minIterations, maxConfidence
3: define Increasing = [ ], Decreasing = [ ]
4: AverageCorrelation = Algorithm 3(TS)
5: FrontStart =  $-duration + shift$  FrontEnd =  $+shift$ 
6: RearStart =  $+shift + delay$  RearEnd =  $+duration + shift + delay$ 
7: for iterations in  $[0 \dots maxIterations]$  do
8:    $r = Random\ i \in [0 \dots T]$ 
9:   Front =  $TS[r + FrontStart : r + FrontEnd]$ 
10:  Rear =  $TS[r + RearStart : r + RearEnd]$ 
11:  Correlation = Algorithm(Front, Rear)
12:  if Correlation < AverageCorrelation then
13:    if Front > Rear then
14:      Increasing  $\leftarrow 1$  Decreasing  $\leftarrow 0$ 
15:    else
16:      if Front < Rear then
17:        Increasing  $\leftarrow 0$  Decreasing  $\leftarrow 1$ 
18:      else
19:        Increasing  $\leftarrow 0$  Decreasing  $\leftarrow 0$ 
20:      end if
21:    end if
22:  else
23:    Increasing  $\leftarrow 0$  Decreasing  $\leftarrow 0$ 
24:  end if
25:  if iterations > minIterations then
26:    if Confidence(Increasing)  $\leq maxConfidence$  then
27:      if Confidence(Decreasing)  $\leq maxConfidence$  then
28:        break loop
29:      end if
30:    end if
31:  end if
32: end for
33: AverageIncreasing = average(Increasing)
34: AverageDecreasing = average(Decreasing)

```

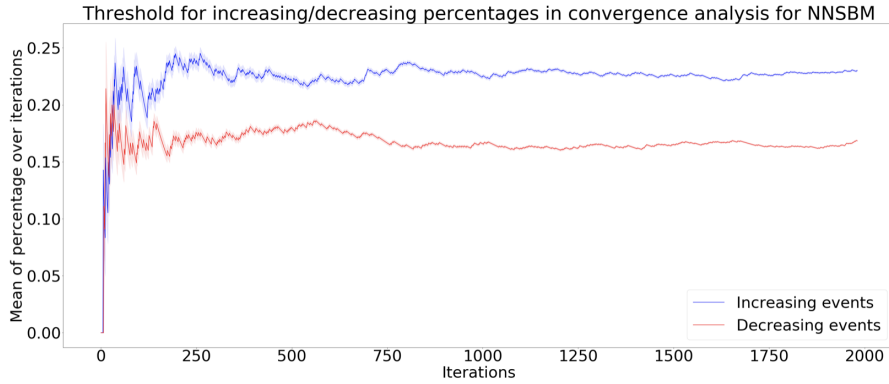


Figure 5.20: Example of defining the limit for average increasing and decreasing probabilities, red background indicates the CI for Percentage Decreasing, blue background indicates the CI for Percentage Increasing

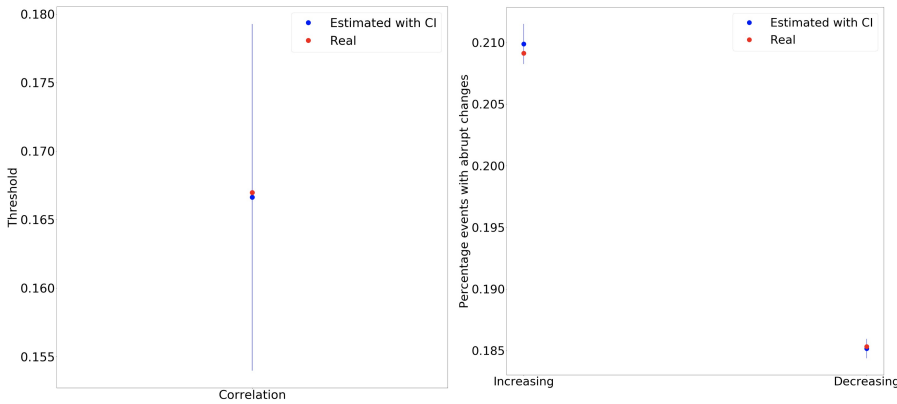


Figure 5.21: Comparing of Average with CI and Exact value calculated over all population for average correlation, increasing and decreasing probabilities

picking random points we pick each one of the event of the ES extracted with its CI, in order to have a significant CI we need to consider only ESs with at least 30 events. In Figure 5.22 we have an example of extracted events, red vertical lines indicate events that cause a decrease in TS while blue vertical lines indicate events that cause an increase. We then need to compare probabilities of increasing and decreasing events between random one and the extracted from ES. In Figure 5.23 it is possible to see difference in increasing probabilities between different ESs and random extractions, CI of random extraction is much smaller due to higher number of iterations.

Finally we need to compare these CI, as the samples have different size, unknown mean and unknown variance. In order to compare them we use

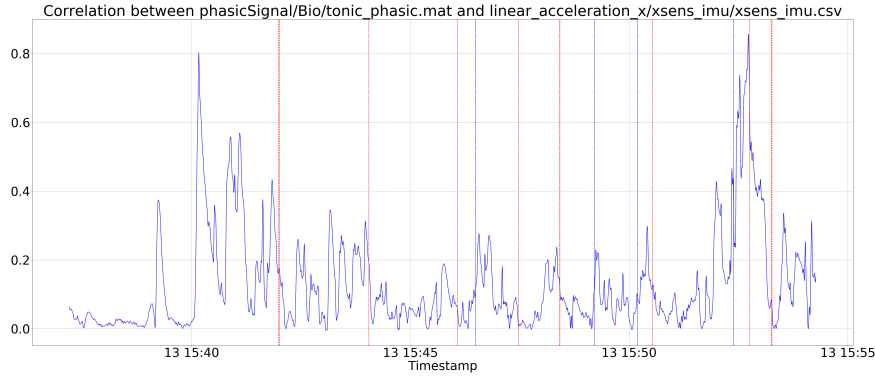


Figure 5.22: Example of defining increasing and decreasing events extracted from ES linear acceleration on X axis over Phasic Component of Skin Conductance

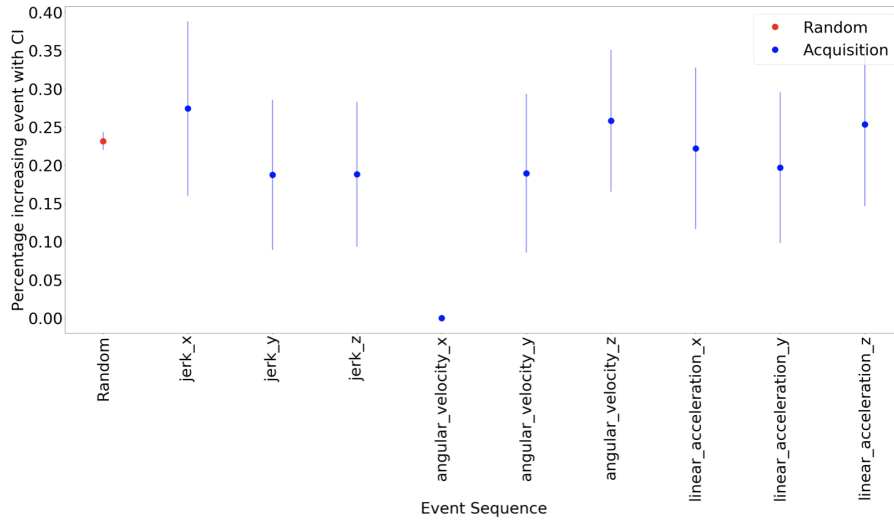


Figure 5.23: Comparing Increasing probabilities between Random one and ESs with their CI at specific percentile and using Phasic Component time-serie

Welch’s t-test[62] using formula:

$$t = \frac{\mu_X - \mu_Y}{\sqrt{\frac{\sigma_X^2}{n_1} + \frac{\sigma_Y^2}{n_2}}} \quad (5.16)$$

Using Welch’s t-test we obtain probability that increasing or decreasing probability of ES is higher than random extracted one. In Figure 5.24 we have an output of probabilities between ES and Random extracted one.

Results are plot with respect to percentiles and with relative probability of correlation, in this way it is much more clear to understand how different percentiles influence on probability of increasing or decreasing the time-serie.

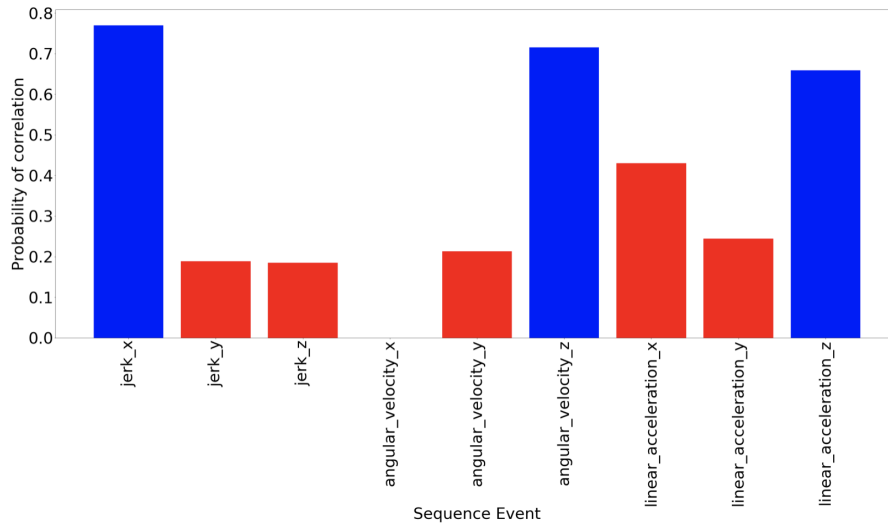


Figure 5.24: Probability of ES increasing probabilities being higher than Random extracted at specific percentile and using Phasic Component time-serie

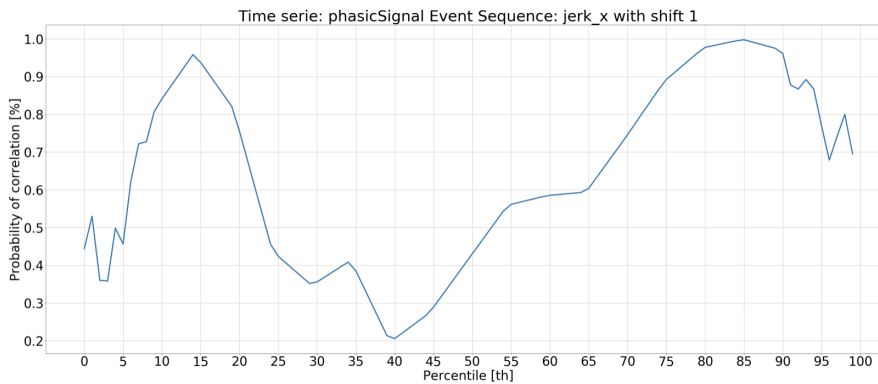


Figure 5.25: Graph of probability of correlation for percentiles from 0th to 100th for Jerk x with respect to Phasic Component of Skin Conductance

In Figure 5.25 we can see an example with Jerk on X axis considering PC time-serie with shift of 1 second and latency of 2 seconds. It is possible to see as we get near the 50th the probability gets around 50% as with the increase of the event considered the result is not much different from considering random ones, we have a peak at around 10th-20th percentile with a probability over 90%, which is not high enough to prove a correlation but indicates that at lower values of Jerk on X axis (decrease of Acceleration on X axis due to braking of the vehicle) the correlation increases, at 85th-90th percentile we have another peak, which coincides with high Jerk due to increase of acceleration. In Chapter 6 we see in detail the obtained results.

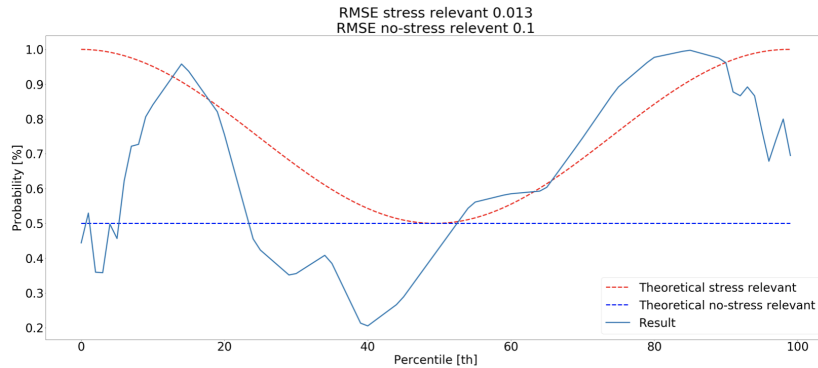


Figure 5.26: Detected curve from 0th to 100th percentile with comparison to full range stress-relevant and no-stress-relevant curves.

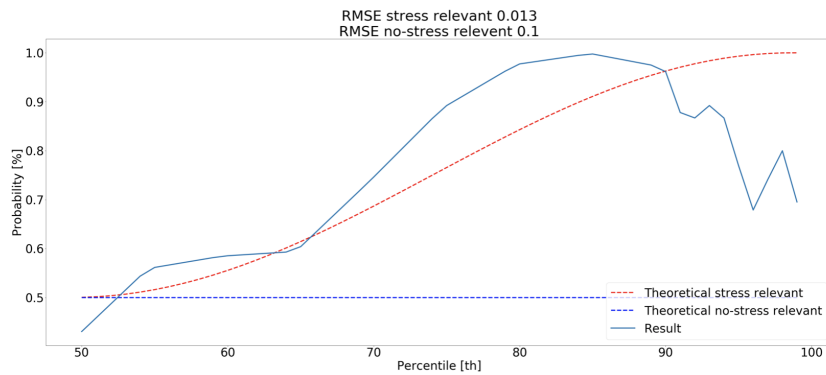


Figure 5.27: Detected curve from 50th to 100th percentile with comparison to high values range stress-relevant and no-stress-relevant curves.

In order to understand if a feature of driving style is relevant for stress detection or not we need to compare the curve obtained with percentiles with two different curves. One curve is defined as stress-relevant at high values and starts from 1.0 at the 100th percentile and decreases to 0.5 at 50th percentile, while the curve no-stress-relevant is a flat line at 0.5. For low values the stress-relevant curve starts from 1.0 at the 0th percentile and decreases to 0.5 at 50th percentile, while the curve no-stress-relevant is the same as before and is flat at 0.5 for all percentiles. We need, by RMSE, if our detected curve is nearer to stress-relevant or no stress-relevant curve. In Figure 5.26 we can see the curve of Figure 5.25 compared to theoretical stress-relevant and no-stress-relevant curves. In Figure 5.27 we can see the part for only high values from 50th to 100th percentile, the comparison has been obtained with RMSE and we can see how the detected curve is much near to stress-relevant, as its RMSE is almost 10 times smaller than

RMSE with no-stress-relevant. A good ratio must be at least of 2 in order to have a good confidence that the curve is much closer to stress relevant than no-stress relevant. This procedure must be applied for all acquisitions, an average with its CI must be done over multiple acquisitions in order to obtain a more significant result. With the increase of number of acquisitions the algorithm detects false positives decreasing their probability.

Chapter 6

Results

In this chapter we are going to see the experimental setup with its protocol, the acquisition of the driving we have obtained and listing all results, in term of correlation between driving style's feature and physiological indexes.

6.1 Experimental setup

The protocol of the experiment has been defined by Matteucci and Gabrielli[63]. Inclusion criterion for participants are:

- Age: 18+
- Holding valid driving licence B
- Phd students or professors of Politecnico di Milano

Participants must wear following sensors already described in Chapter 3:

- Pupil eye tracker;
- Physiological sensors of Procomp Infiniti.

The protocol includes some operations that must be done outside and inside of the vehicle. The experimenter follows the participant in every step. Operations carried out outside of the vehicle are the following:

- A short questionnaire pre-driving is proposed to participant, the questionnaire can be seen in Appendix D;
- In order to decrease interference with Procomp Infiniti, the participant is asked to remove any electronic device;

- Some electrodes are attached to participants chest and shoulder for reading of Electrocardiogram and Skin Conductance physiological indexes;
- The participant are asked to wear glasses for eye tracking. Eye tracker needs a short calibration that consists in looking at a specific marker moving the head while staying sit

Once these operations have been performed the participant takes place as driver in the car, Procomp Infiniti sensors are attached and some test is performed to check functioning of every sensor. Experimenter asks to participant not to talk from this point on, if not strictly necessary, in order not to distort physiological indexes as talking changes breathing and consequently the heart rate. Operations carried out inside of the vehicle are the following:

- The participant needs to relax with closed eyes for three minutes in order to have a base ground level of stress;
- The participant manually drives for two rounds of a pre-defined route;
- A short questionnaire is asked to the participant to describe the manual drive;
- The participant needs again to relax with closed eyes for three minutes;
- The participant automatically drives for two rounds of the same pre-defined route;
- Another short questionnaire is asked to the participant to describe the autonomous drive;
- The sensors are dis-attached from the participant, and the participant takes place as passenger and sensors are re-attached to the car system. The experimenter performs again test to check functionality of the sensors;
- The participant needs again to relax with closed eyes for three minutes;
- The participant assists to a manual drive on the same pre-defined route as a passenger;
- The participant has to answer to a final questionnaire about driving experience.

Once all these operations are performed the experimenter removes every sensor and ask to participant to remove electrodes.

6.2 Acquisition

The acquisition we are considering for the results is the acquisition on 13th December 2018. The participant of the experiment was Alessandro Gabrielli. The acquisition started at 15:37:12 and ended at 15:54:32 with a length of 17 minutes and 20 seconds. Collected data comprehends:

- Axis Cameras;
- Velodyne and its IMU;
- Xsens IMU;
- Prosilica Camera;
- ECG;
- SC;
- Pupil Diameter.

Pupil Diameter information has been discarded because, as shown in Matteucci, Gabrielli and Sesto work[48] in a open world the measurement is too noisy due to continues change of environmental light, the balancing of external light did not succeed in increasing quality of data.

Environmental indexes extracted from data are:

- Angular velocity on XYZ axes from Xsens IMU;
- Linear acceleration on XYZ axes from Xsens IMU;
- Linear velocity on XY axes from Velodyne IMU;
- Speed from Velodyne IMU;
- Jerk on XYZ axes from Xsens IMU;
- Timeheadway to nearest object detected in front position divided in categories from Velodyne.

Physiological indexes extracted from data are:

- LF from ECG;
- HF from ECG;
- LFHF from ECG;

- HRV from ECG;
- TC from SC;
- PC from SC.

In order to test all Environmental indexes with different Physiological indexes we need to understand which of Physiological indexes are more reliable. In order to do so we consider the main Environmental index that has always been considered as stress-relevant feature, such as Jerk on X axe, in Chapter 2. Considering that this feature is stress-relevant we test Physiological indexes using different time characteristics of duration, latency and shift and compare results with state of art as seen in Chapter 2. A good indicator of correlation is a ratio between RMSE no-stress relevant and RMSE stress relevant:

- Ratio above 2 means strong correlation as the curve is clearly better approximated to the stress relevant curve;
- Ratio between 1 and 2 means weak correlation as the curve does not approximate clearly enough to the stress relevant curve;
- Ratio below 1 means no correlation as the curve approximates better the no-stress relevant curve.

6.3 Time Characteristics Analysis

We now see in detail results obtained from each Physiological index when considering correlation with Jerk on X axe. We first test different durations in order to see if there is a value of duration that obtains a RMSE ratio equal or greater to 2 with values between 1s and 20s, then we test latencies with values between 0s and 10s and finally we test shifts with values from -5s and 5 s in order to obtain the best time characteristics for every single Physiological index.

6.3.1 Low Frequency

We have a peak at 4s of duration but below 1 of ratio, which means that LF has results closer to no-stress relevant to stress-relevant, after 10s of duration the ratio seems to converge at 0.645. Considering that no value of duration gives a results above 2 it means that we cannot use LF as a index to find stress relevant environmental indexes. This is in line with State of Art as in Chapter 2 as LF is connected to PNS activity to decrease stress

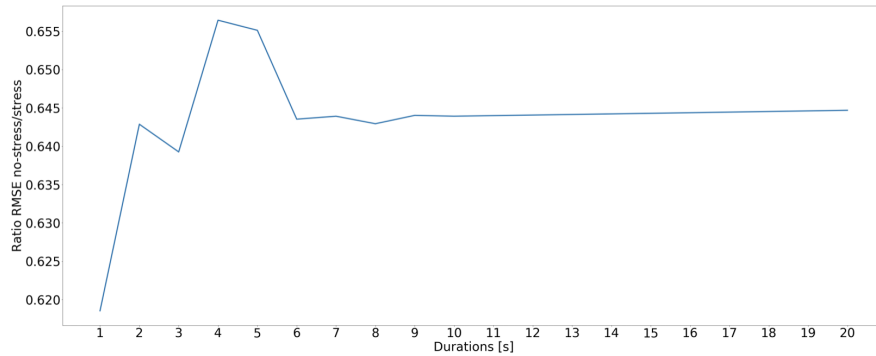


Figure 6.1: RMSE no-stress relevant/stress relevant from LF with different durations

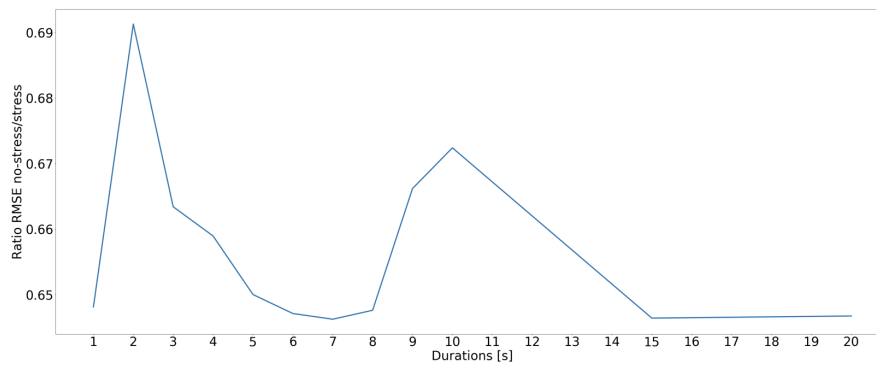


Figure 6.2: RMSE no-stress relevant/stress relevant from HF with different durations

and its behaviour is not linear. Durations of increase in short-term stress for LF in literature are above 10s, but in our study for this acquisition we did not find any increase in correlation from 10s to 20s. We could find more information in HF or LFHF.

6.3.2 High Frequency

For HF we have two peaks at 2s and 10s but both below 1 of ratio. From duration of 15s the curve seems to converge to a value of ratio of 0.65. Also for HF we do not have any value of ratio above 2, which makes also this Physiological index useless in order to find a correlation. In Chapter 2 we have found that HF is correlated to SNS activity and indicates the increase of stress, but still we have not found prove in our acquisition.

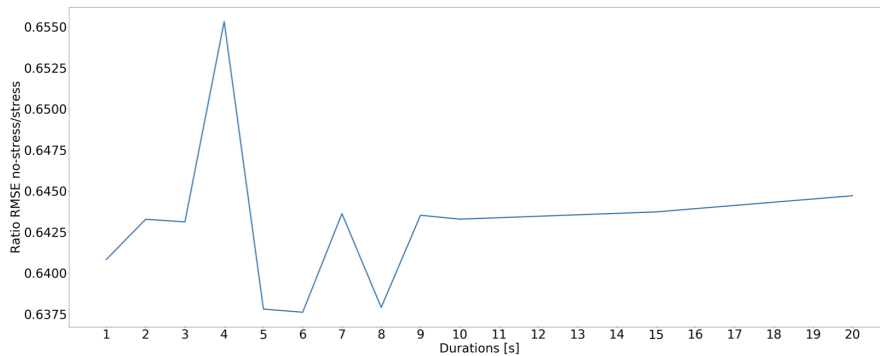


Figure 6.3: RMSE no-stress relevant/stress relevant from LFHF with different durations

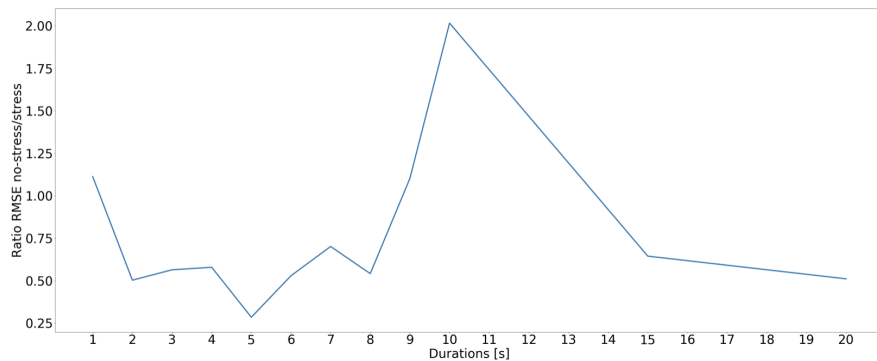


Figure 6.4: RMSE no-stress relevant/stress relevant from HRV with different durations

6.3.3 Low Frequency High Frequency Ratio

For LFHF we have a peak at 4s but below 1 of ratio. From duration of 10s the curve tends to increase slowly but stays steady below 1. We cannot find useful correlation for LFHF index. In Chapter 2 we have seen as LFHF has a non linear behaviour, future acquisitions might be able to detect useful time characteristics to obtain a useful correlation.

6.3.4 HRV

For HRV we have a peak at 10s with value of ratio above 2. Before and after this the ratio decreases with values under of 1. We consider 10s as the duration for HRV. As we have seen in Chapter 2 the duration for ECG are longer comparing to duration for SC so we expect duration of TC or PC to be shorter. We need now to extract the best value for latencies. In Figure 6.5 we can see how we have a peak at value of latency of 2s and a continuous decrease after this value, so we consider a value of 2s for HRV latency. Last

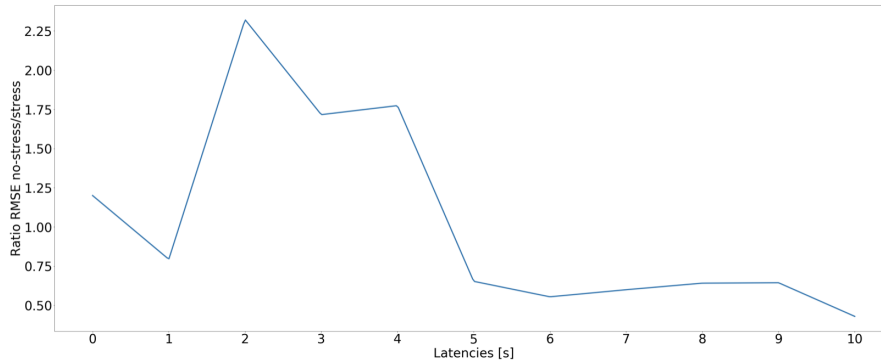


Figure 6.5: *RMSE no-stress relevant/stress relevant from HRV with different latencies*

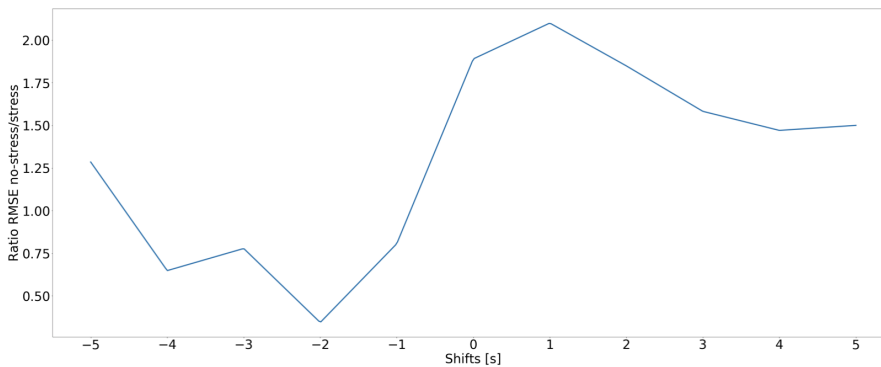


Figure 6.6: *RMSE no-stress relevant/stress relevant from HRV with different shifts*

time characteristic left is shift. In Figure 6.6 we can see how values between 0s and 3s have a ratio around 2, but the best results is obtained with a ratio of 1s.

For HRV we consider as time characteristics:

- Duration: 10s;
- Latency: 2s;
- Shift: 1s.

The sum between latency and shift is of 3s, which is plausible time for activation of SNS and reaction of PNS. The duration can be very variable, depending on situation and on user, but 10s is a plausible time. These values could certainly change when considering different users and must be re-calculated for each user. Considering same user small changes might happen between different acquisitions.

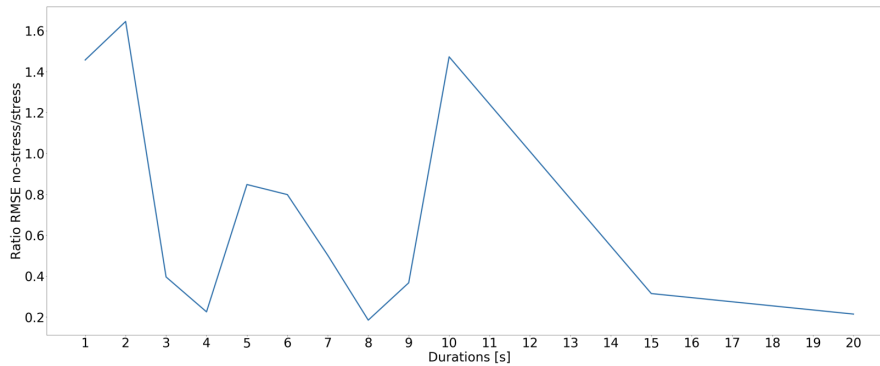


Figure 6.7: RMSE no-stress relevant/stress relevant from TC with different durations

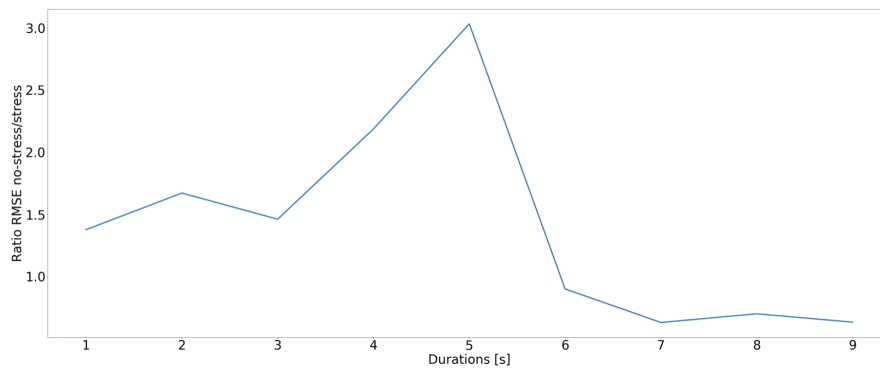


Figure 6.8: RMSE no-stress relevant/stress relevant from PC with different durations

6.3.5 Tonic Component

For TC of SC we have 3 peeks, at 2s, 5s and 10s but none of them are above 2, with maximum at 2s with a ratio of 1.6 which is not enough to distinguish between no-stress relevant and stress relevant. We cannot consider TC, in Chapter 2 we have seen how PC should hold more information than TC.

6.3.6 Phasic Component

In contrast to TC the curve of PC has a single peek at 5s with a ratio above 3, which indicates a strong difference between no-stress relevant and stress relevant.

In Figure 6.9 we can see a peek at value of latency equal to 3s and a constant decrease after this value. We consider then latency with a value of 3s.

In Figure 6.10 we can see that on value of 0s we have a peek with ratio

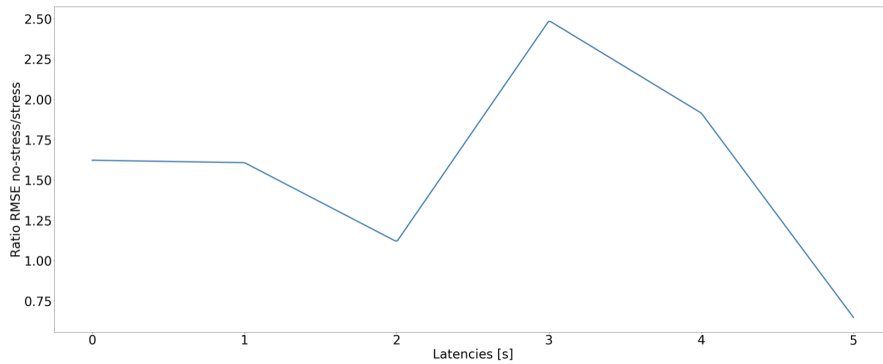


Figure 6.9: RMSE no-stress relevant/stress relevant from PC with different latencies

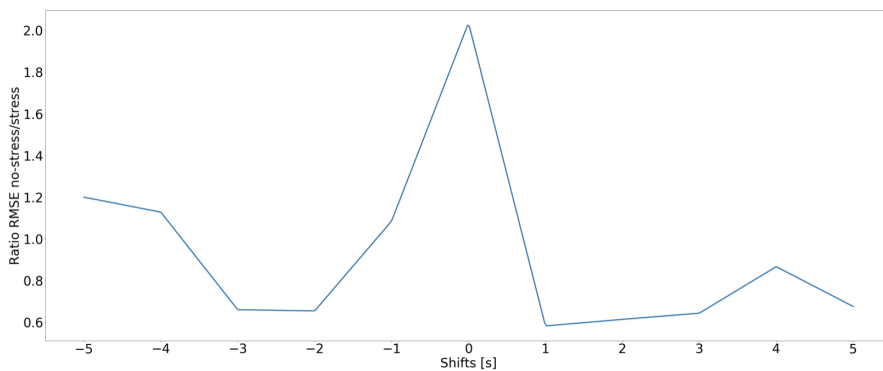


Figure 6.10: RMSE no-stress relevant/stress relevant from PC with different shifts

around 2, we consider this for PC which means that no shift has to be inserted.

For PC we consider as time characteristics:

- Duration: 5s;
- Latency: 3s;
- Shift: 0s.

These time characteristics are plausible values with respect to what we have seen in Chapter 2, as SC durations are shorter than ECG. PC has been proven also in Matteucci, Sesto and Gabrielli[48] work as been the best physiological index for stress detection.

Jerk X	2.75
Jerk Y	1.70
Jerk Z	1.59
Speed	1.27
Timeheadway pedestrians	—
Timeheadway cars	1.55
Angular velocity X	0.48
Angular velocity Y	1.47
Angular velocity Z	1.38
Linear acceleration X	1.64
Linear acceleration Y	0.54
Linear acceleration Z	1.63
Linear velocity X	2.88
Linear velocity Y	1.30

Table 6.1: Ratio of driving style's features in correlation with PC index at high percentiles

6.4 Physiological Indexes Analysis

We now extract which feature of driving style has highest stress relevant correlation with considered Physiological Indexes.

6.4.1 Phasic Component

Using time characteristics extracted in previous section we have obtained ratio results for high percentiles shown in Table 6.1.

In Table 6.1 we can see how each feature performed in correlation with PC for high percentiles, the only two feature that showed a ratio above 2 are Jerk X and Linear velocity X. Timeheadway for pedestrians is not available due to few events through acquisition. High value of Jerk X indicate sudden increase in acceleration, this happened mostly when starting from stationary vehicle. High linear velocity X indicates moment at highest speed, especially on straight segments of road. With this results we can conclude that for PC in this single acquisition these are the two feature that might have increased stress in driver, with multiple acquisitions we can obtain a higher level of confidence in this results. In Figure 6.11 and Figure 6.12 we can see curves from 50th to 100th percentiles. In Table 6.2 we considered results for low percentiles.

For low percentiles the unique feature with ratio above 2 is Angular velocity Y, in Figure 6.13 we can see the curve from 0th to 50th percentiles.

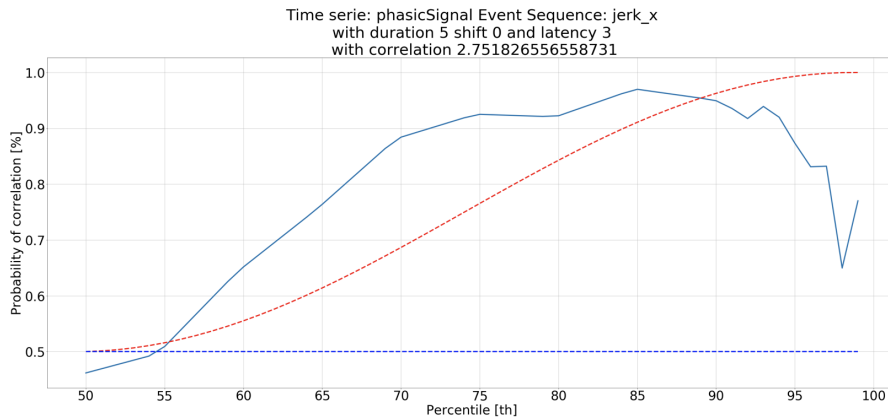


Figure 6.11: Jerk X RMSE no-stress relevant/stress relevant from PC at high percentiles

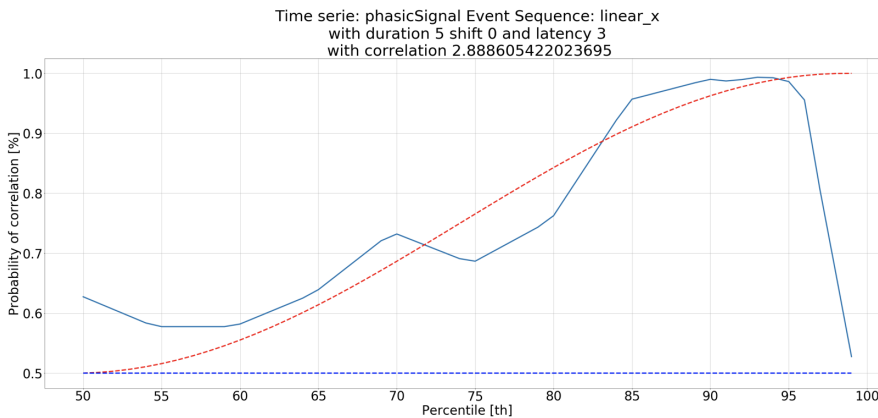


Figure 6.12: Linear velocity X RMSE no-stress relevant/stress relevant from PC at high percentiles

Low angular velocity Y indicates negative values of angular velocity, in particular high values of angular velocity in direction of passenger, as happens on curves. Considering that we do not have a strong correlation for high value of angular velocity Y we can conclude that in this acquisition signal of stress were detected only on negative values of angular velocity.

6.4.2 HRV

Using time characteristics extracted in previous section we have obtained ratio results for high percentiles shown in Table 6.3.

Features with ratio greater than 2 are Jerk x and Linear velocity X for percentiles from 50th and 100th. For high percentiles the stress relevant features are the same obtained using PC.

Jerk X	1.34
Jerk Y	1.72
Jerk Z	0.75
Speed	0.46
Timeheadway pedestrians	—
Timeheadway cars	1.68
Angular velocity X	0.64
Angular velocity Y	2.73
Angular velocity Z	1.59
Linear acceleration X	1.03
Linear acceleration Y	0.67
Linear acceleration Z	1.71
Linear velocity X	1.71
Linear velocity Y	1.82

Table 6.2: Ratio of driving style's features in correlation with PC index at low percentiles

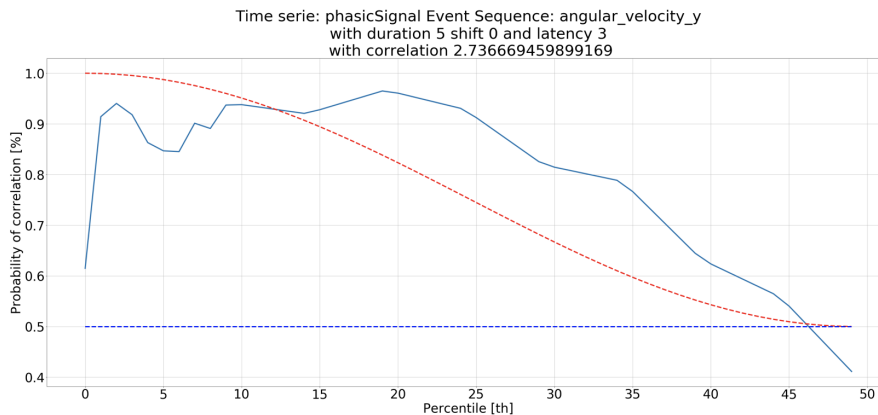


Figure 6.13: Angular velocity Y RMSE no-stress relevant/stress relevant from PC at low percentiles

In Figure 6.14 is shown the curve from 50th to 100th percentiles for Jerk x. In Figure 6.15 the curve from 50th to 100th percentiles for Linear velocity X. These features are the same detected with PC.

For low percentiles from 0th to 50th HRV considers Angular Velocity Y as stress relevant similarly to PC but considers also Jerk Z as stress relevant, while using PC index the ration was very low, this could be a false positive, with the increase of the number of acquisitions such false positives could be detected and not considered.

The results presented are from an unique acquisition, this might intro-

Jerk X	2.35
Jerk Y	1.50
Jerk Z	1.95
Speed	1.09
Timeheadway pedestrians	—
Timeheadway cars	1.62
Angular velocity X	0.91
Angular velocity Y	0.48
Angular velocity Z	0.40
Linear acceleration X	0.45
Linear acceleration Y	0.54
Linear acceleration Z	0.35
Linear velocity X	2.47
Linear velocity Y	1.51

Table 6.3: Ratio of driving style's features in correlation with HRV index at high percentiles

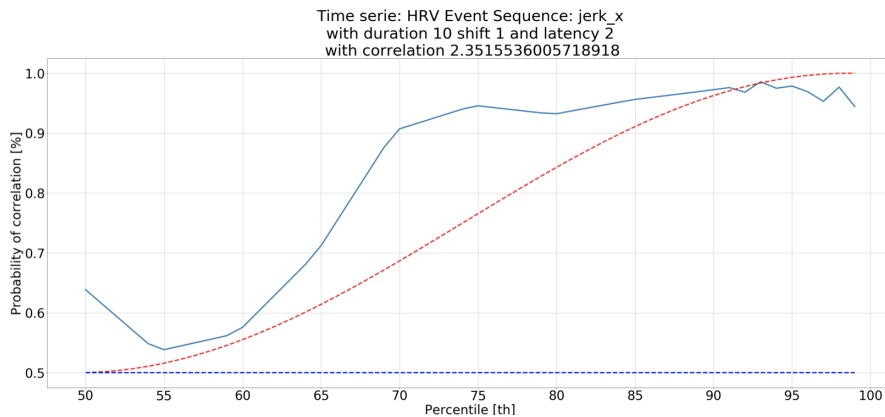


Figure 6.14: Jerk X RMSE no-stress relevant/stress relevant from HRV at low percentiles

duce some errors due to some event not considered that could have happened in the same moment as the feature's events considered. Considering more acquisitions would decrease the number of false positives due to external not considered events. False negatives are more rare but depend on each user, as same values of the feature (acceleration, speed, time-headway, etc...) could cause stress on a particular acquisition and not on another. Also for false negatives a higher number of acquisitions could decrease them. The curves would obtain a CI for each percentile giving more information of

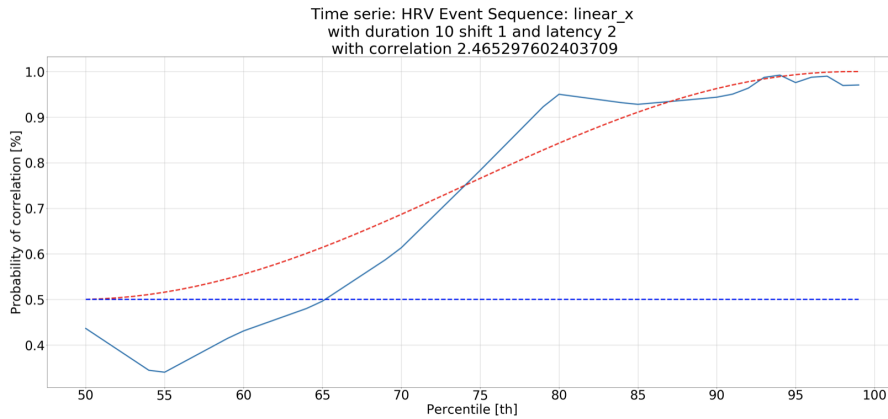


Figure 6.15: Linear velocity X RMSE no-stress relevant/stress relevant from HRV at low percentiles

Jerk X	1.57
Jerk Y	1.44
Jerk Z	2.36
Speed	0.83
Timeheadway pedestrians	—
Timeheadway cars	1.12
Angular velocity X	0.57
Angular velocity Y	2.16
Angular velocity Z	0.42
Linear acceleration X	1.16
Linear acceleration Y	1.84
Linear acceleration Z	0.58
Linear velocity X	1.56
Linear velocity Y	1.59

Table 6.4: Ratio of driving style's features in correlation with HRV index at low percentiles

stress relevant features.

6.5 Conclusions

The software has been developed to be as flexible as possible, new driving style's features and physiological indexes can be inserted or present indexes can be removed or not considered without any change in code. The software allows the user to choose freely value of duration, latency and shift, having

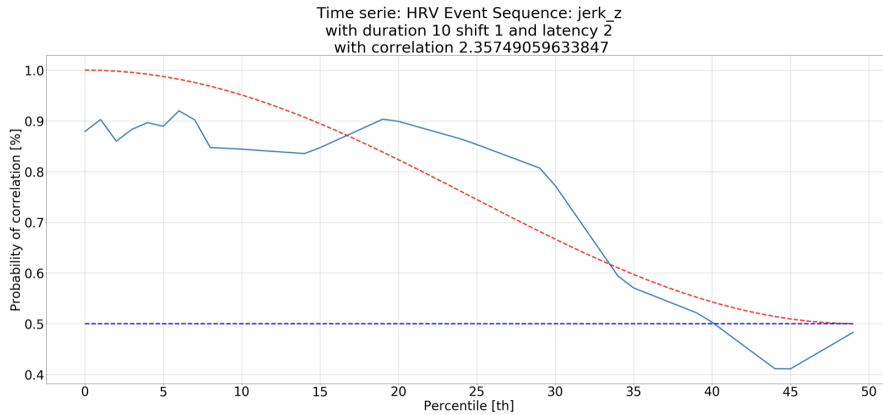


Figure 6.16: Jerk Z RMSE no-stress relevant/stress relevant from HRV at low percentiles

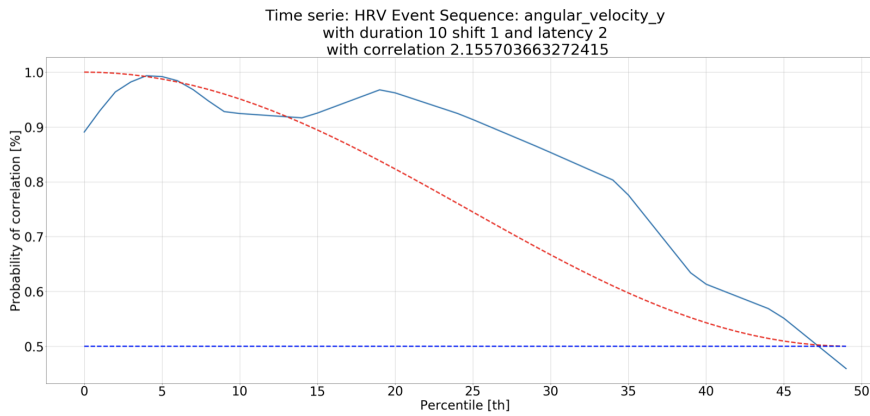


Figure 6.17: Angular velocity Y RMSE no-stress relevant/stress relevant from HRV at low percentiles

possibility to choose the same values as found in State of Art or, as has been done for our results, choose the best result by checking for correlation ratio and comparing it with State of Art.

Best results are obtained with the increase in number of acquisitions, the model tends to converge to the real value of correlation for each feature and can show how the correlation increases or decreases with the change in driving style. For our single acquisition we have found that just two Physiological Indexes have proven to have correlation and they are HRV extracted from ECG and PC extracted from SC, which, as we have seen in Chapter 2, were the most promising, while LFHF and HF did not obtain sufficient correlation ratio to be considered. Feature with most correlation were Jerk x and Linear velocity X for high percentiles, which brings in

the case of Jerk x to abrupt accelerations and Linear velocity x to high velocity but only on X axe, as speed that considers also Y axis did not obtain a sufficient correlation ratio. Angular velocity Y and Jerk z obtained a sufficient correlation ratio for low percentiles, while Angular velocity Y is due to velocity on curves and low percentiles does not mean 0 but negative (positive is on direction of passenger while negative on direction of driver), Jerk z compares only in PC and not in HRV which means that it could be a false positive, as the meaning of Jerk z is an abrupt change in deceleration on Z axe, perpendicular to floor, could be due to some difference in height of the floor or as mentioned a false positive. New acquisitions could answer these questions.

Each user could have different time characteristics for different Physiological Indexes and could also change in time, when getting more comfortable with autonomous driving. A constant study for each user with multiple acquisitions could lead to detection of Concept Drifts and detect in this way a decrease or increase level of stress in general. The goal is that no feature is stress relevant but flattens to no-stress relevant curve because this would mean that no driving style's feature cause stress in driver.

Chapter 7

Conclusions and future works

The goal of this work was to search for a correlation between environmental and vehicle's data from car sensors and level of stress for the driver obtained from physiological sensors. The approach of this work is different from previous approaches and concentrates on analysis of physiological indexes of the driver.

The developed software is able to detect the probability of correlation between the time-serie of a set of Physiological indexes and the event sequences extracted from Vehicle and Environmental's data, this correlation indicates an increase in level of the stress for the driver. Through many acquisitions is possible to detect, with higher confidence, the driving style's features that cause stress for the driver and consequently change the driving style of the autonomous car in order make more comfortable the driver, and through this increase trust of user in the autonomous car. The obtained driving style does not consider user's personal driving style, as this has been proven not to be conclusive by previous works, but considers only the stress extracted from physiological sensors. Between all Physiological indexes considered the ones that obtained higher correlation were Heart Rate Variability and Phasic Component of Skin Conductance, results obtained between the two indexes are much similar and this could prove that the analysis correctly detects stress for the driver. These results must be confirmed with many more acquisitions.

One of the characteristic of the developed software was to be modular, in order to be able to insert and remove sensors. This modularity has been obtained in both hardware and software part. Other characteristic is the ability to detect different behaviours for different users, this change is obtainable by extraction of different time characteristic for each user. The system can analyze acquisitions in order to obtain best time characteristic

for each Physiological index, and these time characteristic could differ from user to user.

7.1 Future works

Main future developments comprehend the use of a fully-autonomous car in order to delete eventual biases caused by the use of a not AC. Increasing the number of acquisitions could confirm the found correlations, new correlations could be found. Once correlation are confirmed the driving style of the AC could be modified in order to decrease the stress relevant driving style's feature, checking for correlation after this change in driving style it will be possible to check if correlation has changed transforming stress relevant feature into a no-stress relevant feature.

The work could be carried on introducing new driving style's feature and driving style's event such as driving conditions or obstacles, such as different road signs. Another step would be to change the urban environment to extra-urban, such as high roads, to introduce some other driving style's feature, such as change of line, and also in these environment the speed could obtain a different level of correlation from urban speeds' correlation. Analysis could be done online, especially for long acquisitions, in order to detect on the air stressful driving style's features and modifying them through the acquisition.

Bibliography

- [1] Keshav Bimbraw. “Autonomous Cars: Past, Present and Future”. In: *Mechanical Engineering Department, Thapar University, P.O. Box 32, Patiala, Punjab, India* (2015). DOI: 10.5220/0005540501910198.
- [2] DARPA AVL (Autonomous Land Vehicle) summary(U) Army Engineer topographic labs fort Belvoir VA R D Leighty Mar 86 ETL-R-085. DOI: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a167472.pdf>.
- [3] Giselle Rampersad Kanwaldeep Kaur. “Trust in driverless cars: Investigating key factors influencing the adoption of driverless cars”. In: *Journal of Engineering and Technology Management 48 (2018) 87-96* (). DOI: https://www.researchgate.net/publication/283757446_Autonomous_Cars_Past_Present_and_Future_-_A_Review_of_the_Developments_in_the_Last_Century_the_Present_Scenario_and_the_Expected_Future_of_Autonomous_Vehicle_Technology.
- [4] “Do You Want Your Autonomous Car To Drive Like You?” In: *HRI 17, March 06-09, 2017, Vienna, Austria* (). DOI: <http://dx.doi.org/10.1145/2909824.3020250>.
- [5] “Drive me comfortably”. In: *VDI-Berichte 2264* (2015).
- [6] “How the driver wants to be driven - Modelling driving styles in highly automated driving”. In: *Technische University Chemnitz Interdisciplinary Center for Driver Assistant Systems (I-FAS) Chemnitz* ().
- [7] “Detecting Stress During Real-World Driving Tasks Using Physiological Sensors”. In: *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 6, NO. 2, JUNE 2005* ().
- [8] *Automated Driving, Levels of Driving Automation are Defined in New SAE International Standard J3016*. 2014.
- [9] *U.S. Department of Transportation Releases Policy on Automated Vehicle Development*. 2013.

- [10] *Chapter 23A. Autonomous Vehicles*. 2013. URL: <https://code.dccouncil.us/dc/council/code/titles/50/chapters/23A/#>.
- [11] George Heinzelman. “Autonomous Vehicles, Ethics of Progress”. In: (2019). DOI: 10.13140/RG.2.2.28046.31048.
- [12] K. H. F. Cardew. “The Automatic Steering of Vehicles: An Experimental System Fitted to a DS 19 Citroen Car.” In: *RRL report* (1970).
- [13] John Pressnell. “Citroen DS: The complete story”. In: *Crowood* (1999).
- [14] “Vision-based navigation: A status report.” In: *In Proceedings of Image Understanding Workshop*, pp. 153-169 (1987).
- [15] “Low Speed Automation: technical feasibility of the driving sharing in urban areas.” In: *In 16th International IEEE Conference on Intelligent Transportation Systems* (2013).
- [16] “Autonomous road vehicle guidance from autobahnen to narrow curves.” In: *Robotics and Automation, IEEE Transactions on* 14, no. 5 (1998): 810-815 (1998).
- [17] Josef Wenger. “Automotive radar-status and perspectives.” In: *In Compound Semiconductor Integrated Circuit Symposium, 2005. CSIC'05. IEEE*, pp. 4-pp. IEEE (2005).
- [18] “Steps towards an intelligent vision system for driver assistance in urban traffic.” In: *In Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*, pp. 601-606. IEEE (1997).
- [19] “Toward autonomous driving: the cmu navlab”. In: *IEEE expert* 6, no. 4 (1991): 31-42. (1991).
- [20] “Architectural issues on vision-based automatic vehicle guidance: the experience of the ARGO project.” In: *Real-Time Imaging* 6, no. 4 (2000): 313-324. ().
- [21] F.D. Davis. “MIS Quarterly. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology.” In: (1989). DOI: pp.319-340.<http://dx.doi.org/10.2307/249008>..
- [22] “User acceptance of information technology: toward a unified view.” In: (2003). DOI: MISQuarterly425-478.<http://www.jstor.org/stable/30036540>..
- [23] “Presenting system uncertainty in automotive UIs for supporting trust calibration in autonomous driving”. In: (2013). DOI: DOI:10.1145/2516540.2516554.

- [24] “Why did my car just do that? Explaining semi-autonomous driving actions to improve driver understanding, trust, and performance”. In: (2014). DOI: [10.1007/s12008-014-0227-2](https://doi.org/10.1007/s12008-014-0227-2).
- [25] “Self-report instruments and methods.” In: *In Porter, B. E. (Ed.), Handbook of traffic psychology (pp. 43-59). London, UK: Elsevier.* (2011).
- [26] “Behavioral correlates of individual differences in road traffic crash risk: An examination of methods and findings.” In: *Psychological Bulletin, 113, 279-294.* (1993).
- [27] “Driver’s style classification using jerk analysis. In IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems”. In: *2009 (pp. 23-28). Piscataway, NJ: IEEE.* (2009).
- [28] H. A. Deery. “Hazard and risk perception among young novice drivers.” In: *Journal of Safety Research, 30, 225-236.* (1999).
- [29] “Indices for characterizing driving style and their relevance to car following behavior.” In: *In SICE Annual Conference 2007 (pp. 1132-1137). Piscataway, NJ: IEEE.* (2007).
- [30] “A Review of Research on Driving Styles and Road Safety”. In: (2005). DOI: <https://doi.org/10.1177/0018720815591313>.
- [31] “Do You Want Your Autonomous Car To Drive Like You?” In: (2007). DOI: <http://dx.doi.org/10.1145/2909824.3020250>.
- [32] “Objective metrics of comfort: Developing a driving style for highly automated vehicles.” In: *Transportation Research Part F: Traffic Psychology and Behaviour, 41:45-54, August 2016. ISSN 13698478.* (2016). DOI: [10.1016/j.trf.2016.05.005](https://doi.org/10.1016/j.trf.2016.05.005). URL: <https://linkinghub.elsevier.com/retrieve/pii/S136984781630064X>..
- [33] D. Manstetten. “Fahrerzustandserfassung für assistiertes und (hoch) automatisiertes Fahren.” In: *Stuttgart: Fraunhofer Institut für Arbeitswissenschaft und Organisation IAO, Vehicle Interaction Summit.* (2014).
- [34] “In-vehicle data recorders for monitoring and feedback on drivers’ behavior.” In: *Transportation Research Part C: Emerging Technologies, 16(3), 320-331.* (2008). DOI: <http://dx.doi.org/10.1016/j.trc.2008.01.001>..
- [35] “Driving behaviour-based event data recorder.” In: *IET Intelligent Transport Systems, 8(4), 361-367.* (2014). DOI: <http://dx.doi.org/10.1049/iet-its.2013.0009>..

- [36] J. T. Reason. "Motion sickness adaptation: A neural mismatch model." In: *Journal of the Royal Society of Medicine*, 71, 819-829. (1978).
- [37] "Automatisiertes Fahren - So komfortable wie möglich, so dynamisch wie nötig: Vestibulare Zustandsrückmeldung beim automatisierten Fahren." In: *In 30. VDI/VW Gemeinschaftstagung. Fahrerassistenz und Integrierte Sicherheit*. (2014).
- [38] "Can you feel the difference? The just noticeable difference of longitudinal acceleration." In: *Proceedings of the human factors and ergonomics society annual meeting (Vol. 57, No. 1, pp. 1219-1223)*. (2013). DOI: <http://dx.doi.org/10.1177/1541931213571271>..
- [39] "How the driver wants to be driven - Modelling driving styles in highly automated driving." In: *Technische Universität Chemnitz Interdisciplinary Center for Driver Assistant Systems (I-FAS) Chemnitz* (2015).
- [40] "Stress: Facts and Theories through Literature Review". In: *International Journal of Medical Reviews* (2015).
- [41] "Automatic Stress Classification With Pupil Diameter Analysis". In: *International Journal of Human-Computer Interaction* (2014). DOI: 10.1080/10447318.2013.848320.
- [42] "An Overview of Heart Rate variability Metrics and Norms". In: (2017). DOI: 10.3389/fpubh.2017.00258.
- [43] George E. Billman. "The LF/HF ratio does not accurately measure cardiac sympatho-vagal balance". In: (2013). DOI: 10.3389/fphys.2013.00026.
- [44] "Influence of mental stress on heart rate and heart rate variability." In: *Proceedings 4th European Conference of the International Federation for Medical and Biological Engineering (IFMBE '09);pp. 1366-1369* (2009).
- [45] "Emotional sweating across the body: Comparing 16 different skin conductance measurement locations". In: (2012). DOI: <http://dx.doi.org/10.1016/j.physbeh.2012.01.020>.
- [46] "I.DRIVE: Elaborazione e analisi di segnali fisiologici per la valutazione dello stress alla guida". MA thesis. Politecnico di Milano, 2015.
- [47] "Detecting Stress During Real-World Driving Tasks Using Physiological Sensors". In: *IEEE Transactions on Intelligent Transportation Systems* 6(2):156-166 (2015). DOI: 10.1109/TITS.2005.848368.
- [48] "I.DRIVE: A software system for driver, vehicle, and environment interaction data analytics". MA thesis. Politecnico di Milano, 2019.

- [49] “Correlating Events with Time Series for Incident Diagnosis”. In: (2014). DOI: <http://dx.doi.org/10.1145/2623330.2623374>..
- [50] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. 1988.
- [51] *Time series analysis : forecasting and control*. 2016.
- [52] M. F. Schilling. “Multivariate two-sample tests based on nearest neighbors.” In: *Journal of the American Statistical Association* (1986).
- [53] “Energy statistics: A class of statistics based on distances”. In: *Journal of Statistical Planning and Inference* 143 (2013) 1249–1272 (2013).
- [54] *Detection of Abrupt Changes: Theory and Application*.
- [55] “I.DRIVE: Progetto e sviluppo di un data logger multisensoriale per studi di interazione veicolo conducente”. MA thesis. Politecnico di Milano, 2015.
- [56] “Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction”. In: (2014). DOI: 10.1145/2638728.2641695.
- [57] Yuxing Mao Yan Yan and Bo Li. “SECOND: Sparsely Embedded Convolutional Detection”. In: (2018).
- [58] Dominique Makowski et al. *NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing*. 2020. DOI: 10.5281/ZENODO.3597887. URL: <https://github.com/neuropsychology/NeuroKit>.
- [59] *An Introduction to Statistical Learning*. 2017. DOI: 10.1007/978-1-4614-7138-7.
- [60] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006.
- [61] “Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS® Implementations”. In: *Health Care and Life Sciences* (2010).
- [62] Zhenqiu Laura Lu and Ke-Hai Yuan. “Welch’s t-test”. In: (2010). DOI: 10.13140/RG.2.1.3057.9607.
- [63] Matteo Matteucci and Alessandro Gabrielli. “Improving user acceptance in autonomous driving”. In: (2020).

Appendix A

Software installation

Below a list of requirements and third-party Python modules.

A.1 Requirements

Following requirements have to be satisfied:

- Ubuntu Linux operating system version 16.04 LTS or above;
- ROS framework Kinetic version;
- Python 2.7 version or above (compatible with ROS version).

A.2 Python modules

Below a list of non-standard Python modules that need to be installed using command

pip install [Python module]

- *Pymongo* for MongoDB elaborations;
- *Pandas* for DataFrame elaborations;
- *Numpy* that comes with useful array methods;
- *Matplotlib* for Matlab[®]-like plot methods;
- *Sklearn* used for NNSBM algorithm;
- *Statsmodels* used for ARMA algorithm;
- *Scipy* used for data filtering and PCC.

A.3 Download scripts

All scripts are stored online on bitbucket repository and can be downloaded after registration and login with command:

```
git clone
https://username@bitbucket.org/airlab-polimi/idrive_ros.git
```

Scripts of these thesis can be found at:

```
idrive_ros > script > correlation
```

Bash script can be found at:

```
idrive_ros > script
```

A.4 MongoDB

In order to correctly install MongoDB the following steps must be followed:

- Download MongoDB community edition following instructions on:
<https://docs.mongodb.com/manual/tutorial/install-mongodb-enterprise-on-ubuntu/>;
- Create folder with MongoDB permission by using command

```
sudo chown -r mongodb:mongodb [folder path]
```

- Modify **/etc/mongodb.conf** by opening it with root privilege and change **dbpath** with path of created folder;
- Restart MongoDB with command

```
sudo service mongodb restart
```

A.5 ROS

To install ROS Kinetic version useful instructions can be found at:

```
http://wiki.ros.org/kinetic/Installation/Ubuntu
```

Appendix B

Programming documentation

Below we will describe in detail the code used for each algorithm used for correlation.

B.1 PCC

For PCC we have used Python module *scipy.stats.pearsonr(x,y)* which given two arrays x and y it calculates the correlation using Equation 5.5. Result are the correlation coefficient at position [0] which is -1 if perfect inverse correlation is found, +1 if linear correlation and p-value at position [1].

Due to the fact that the others algorithms have a greater value when no correlation is found we needed to change the output of *scipy.stats.pearsonr* to have a result that is comparable to the others algorithms. The considered output will be

```
1-scipy.stats.pearsonr(x,y)[0]
```

Applying this formula the correlation coefficient is ranged between 0 perfect correlation and 2 perfect inverse correlation.

B.2 ET

For ET we have applied Equation 5.10 merging in one for cycle first two sums.

```
def energy_test(x,y):  
    a = 0  
    b = 0  
    for i in x:  
        a += sum(abs(i-y))
```

```

        b += sum(abs(i-x))

    c = 0
    for l in y:
        c += sum(abs(l-x))
    abc = 2/(len(x)*len(y))*a-b/(len(x)**2)-
        c/(len(y)**2)
    return abc*(len(x)*len(y)/(len(x)+len(y)))

```

B.3 ARMA

For ARMA we have used Python module's *statsmodels.tsa.arima_model* method *arma* and module's *sklearn.metrics* method *mean_squared_error* and parameter from *params* to choose if down-sample the TS using a moving average (a value equal to length of population indicates no down-sample).

```
def arma(x, y, p, q):
```

```

    x_avg = np.mean(x.reshape(-1,
                             int(len(x)/params.splits_arma)),
                  axis=1)
    forecast = []

    for i in range(params.splits_arma):
        model = ARMA(x_avg,
                     order=(p, q))
        model_fit = model.fit(dispatch=0,
                              transparams=False,
                              trend='nc',
                              solver='nm')
        output = model_fit.forecast()
        forecast.append(output[0])
        x_avg=np.append(x_avg[1:],
                       output[0])

    out = interpolation.zoom(forecast,
                            zoom=len(y)/len(forecast))

    return mean_squared_error(out, y)

```

B.4 NNSBM

For NNSBM we have used Python module's *sklearn.metrics* method *pairwise_distances* that outputs a matrix with all pair distances between points of array *x* and *y*.

```
def nnsbm(x,y,k_tot):
    x=x.reshape(-1,1)
    y=y.reshape(-1,1)
    y = y-np.mean(y)+np.mean(x)
    same = 0
    total = 0
    rf = sklearn.metrics.pairwise_distances(x,y)
    ff = sklearn.metrics.pairwise_distances(x,x)
    for k in range(1,k_tot+1):
        for p in range(len(rf)):
            if(p-k>0 and p+k+1<len(x)):
                a = np.sort(rf[p][
                    p-k:p+k])
                b = np.sort(ff[p][
                    p-k:p+k+1])
            elif(p-2*k<0):
                a = np.sort(rf[p][
                    :p+k-1])
                b = np.sort(ff[p][
                    :p+k])
            else:
                a = np.sort(rf[p][
                    p-k:])
                b = np.sort(ff[p][
                    p-k-1:])
            res = np.sort(
                np.append(a,b))
            if(res[k] in b and
                not res[k] in a):
                same+=1
            total+=1
    return same/total
```

Appendix C

Manual

In this section we will describe in detail every script used to obtain final results with correlation probabilities. The pipeline, as showed in Figure 4.5, can be resumed in case of errors without loss of information and it allows to start after data preparation and from correlation analysis in order to change parameters of analysis. The pipeline is handled by bash script *data_extraction_for_correlation.sh*.

For each script we will describe:

- Functionality: implemented function;
- Required input: list of input to be passed;
- Parameters: list of parameters, both optional and mandatory;
- Output: shape and meaning of output.

C.1 Params.py

This file contains all parameters used for all the scripts, it must be imported in all scripts. We will see a list of all parameters:

- *resample frequency*: chosen frequency of down-sample defined in *date-time.timedelta* expression (for our work the frequency was 100Hz so it has been defined as 10ms);
- *percentage_drop*: minimum percentage of data in MongoDB collections in order not to drop the collection, if percentage is below the collection will be dropped as it does not contain enough information (for our work it is 0.10, if no drop is wanted it is enough to give 0 to this parameter);

- `collections_to_extract`: list of collections to extract from MongoDB, when a new data is inserted or one present has to be removed this parameter has to be updated;
- `velodyne_threshold`: minimum score of identified objects, below this value the object is not considered (for our work this value is 0.6 as we have seen in Chapter 5);
- `velodyne_max_distance`: maximum distance of identified objects, above this distance the object is not considered (for our work this value is 20m);
- `alfa`: α value for p-value to detect if Rear sub-sequence is greater or less than Front sub-sequence (for our work is 1.97);
- `percentile_step`: step of increasing or decreasing of considered percentile when extracting the ES (for our work is 1th);
- `n_min`: minimum number of iterations when calculating correlation thresholds (for our work is 100, higher than the standard 30 in order to obtain a thinner CI);
- `n_max`: maximum number of iterations when calculating correlation thresholds (for our work is 5000);
- `percentage_max_iterations`: maximum percentage of iterations when calculating correlation thresholds over the whole acquisition, when the first between `n_max` and `percentage_max_iterations` is reached the cycles stops (for our work is 0.05);
- `features`: list of features to be considered for correlation, this list is used for *extraction_correlation.py* and can be modified after the data preparation has been executed and DataFrame obtained, in this way different correlation analysis can be executed (for our work this list has been changing with different correlation analysis from all features to a single feature);
- `time_series`: as above but with respect to Physiological indexes (for our work this list has been changing following the same reasoning of features);
- `k_NNSBM`: value of k (`k_tot` in the code listed in Appendix B for NNSBM (for our work we have used `k=4` as we have seen in Chapter 5);

- `p_arma`: value of `p` for ARMA (for our work we have used `p=4` as we have seen in Chapter 5);
- `q_arma`: as above but for `q` value (for our work we have used `q=2`);
- `splits_arma`: moving average of `n` points in order to consider more of the sub-sequence as we have seen in Appendix B (for our work we have used 25);
- `durations`: list of used durations for correlation analysis, this value can change when changing correlation analysis (for our work we have first tried different durations and once obtained the best duration in term of ratio no-stress relevant / stress relevant we did our analysis with this duration that changes using different Physiological indexes);
- `shifts`: list of used shifts for correlation analysis, as above;
- `latencies`: list of used latencies for correlation analysis, as above;
- `percentiles`: list of used percentiles for correlation analysis from 0th to 100th (for our work we have used percentiles of [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45] and [55, 60, 65, 70, 75, 80, 85, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]).

C.2 Extract_mongo_collections.py

This is the first script that has to be executed, it extracts data from MongoDB collections. It loads collections defined in *params.py*, they are all saved in same way in MondoDB except collection *vel* that needed an implementation on its own. The collections will be then saved to a DataFrame.

Input

The script needs the acquisition to be present on MongoDB and MongoDB to be active. If requested a timestamp correction is performed using file *info.player.json*.

Parameters

- **-output**: Destination folder in which to save output files;
- **-timestamp**: Input folder of *info.player.json* used for timestamp correction, if no input folder is given or the file is not present or unreadable

the timestamp correction will not be performed, an error message will be displayed but the extraction will not stop;

- **-p**: option to plot results, if present the script will save plot results to output folder;
- **-csv**: option to save csv, if present the script will save CSV of results to output folder, this can be useful in case of errors and to increase modularity as the pipeline can be restored from error point using saved CSV.

Output

The script will output a DataFrame with data from MongoDB collections.

C.3 Extract_velodyne.py

This script will extract Velodyne information, it needs the file **.json* extracted using *velodyne_extraction.py* as input.

Input

The script needs the full path of file **.json* with identified objects from SECOND.

Parameters

- **-input**: Full path of **.json* file;
- **-output**: Destination folder for output files;
- **-timestamp**: Input folder of *info.player.json*, see previous subsection for details;
- **-p**: if present the script will save plots in output folder;
- **-csv**: if present the script will save csv in output folder, see previous subsection for details.

Output

The script will output two DataFrames, one with number of identified objects for each frame and one with minimum distance of identified objects for each frame.

C.4 `Extraction_features.py`

This is the main script for data preparations, the script will merge information from `extraction_mongo_collection.py`, `extraction_velodyne.py` and Matlab® scripts. The script will perform down-sampling, filtering, drop of collections with too few information and extract features from mongo collections such as Timeheadway, Jerk and Speed.

Input

The script needs all DataFrames from previous scripts, Matlab® scripts will automatically save them to DataFrames, while Python scripts will return DataFrame as output and saving them only if parameter `-csv` has been given. This script automatically saves the CSV as result because it can be used for different correlation analysis without executing the data preparation from 0.

Parameters

- `-input`: Input folder in which Matlab® scripts have saved CSV;
- `-output`: Output folder in which to save CSV and eventual plots;
- `-timestamp`: Input folder of `info.player.json` file;
- `-p`: if present the script will save plots in output folder.

Output

The script output is a DataFrame that contains all information extracted from previous scripts and of feature extracted. The output is automatically saved in a CSV because it might be used multiple times for correlation analysis.

C.5 `Correlation_extraction.py`

This is the main script for correlation analysis, it contains the algorithms explained in Chapter 5 which code has been presented in Appendix B, the algorithm to extract ES and all iterations to extract correlation thresholds, actual number of increasing and decreasing events and output the correlation probabilities.

Input

The script needs the DataFrame extracted from *Extraction_features.py*.

Parameters

- **-input:** Input folder containing DataFrame extracted from *extraction_features.py*;
- **-output:** Output folder to save CSV and eventual plots;
- **-p:** if present the script will save plots in output folder;
- **-alg:** chosen algorithm for correlation analysis between PCC,ET,NNSBM and ARMA;
- **-f:** if present the script will try to find CSV files of correlation thresholds and increasing, decreasing percentage thresholds, if not present the script will always recalculate them. This is useful in case of errors since calculating correlation thresholds and increasing, decreasing percentages can be computationally demanding and take up to hours depending on the length of acquisition.

Output

The script will output a DataFrame with all probabilities for every time characteristic, percentile, features and TS.

Appendix D

Questionnaire

These are the pre and post driving questionnaires proposed to participant of the experiment as defined in the protocol[63] with goal to understand if there has been any change in driver's mood.

Pre-Driving Questionnaire

1. What is your gender?
 - Male
 - Female
 - Other
 - Prefer not to say
2. How old are you?
3. Do you own a driving license?
 - Yes
 - No
4. How long have you had your driving license?
 - Less than 5 months
 - From 5 months to 1 year
 - From 1 year to 2 years
 - From 2 years to 5 years
 - For more than 5 years
5. Do you or your family own a car?

- Yes
 - No
6. How often do you drive a car?
- Never in last 3 months
 - Twice a month
 - Once a week
 - Twice or three times a week
 - Everyday
7. The following questions relate to your experience as a **driver** in an everyday driving scenario. Please circle the appropriate number - there are no right or wrong answers.
- I feel unsafe at high speeds:
Not at all / Somewhat / Moderately / Very much
 - I get uncomfortable with heavier congestion:
Not at all / Somewhat / Moderately / Very much
 - Getting close to other vehicles make me feel uncomfortable:
Not at all / Somewhat / Moderately / Very much
8. The following questions relate to your experience as a **passenger** in an everyday driving scenario. Please circle the appropriate number - there are no right or wrong answers.
- I feel unsafe at high speeds:
Not at all / Somewhat / Moderately / Very much
 - I get uncomfortable with heavier congestion:
Not at all / Somewhat / Moderately / Very much
 - Getting close to other vehicles make me feel uncomfortable:
Not at all / Somewhat / Moderately / Very much
9. How do you consider your driving style?
- Very Defensive
 - Defensive
 - Aggressive
 - Very Aggressive
10. Which is your favourite driving style as passenger?

- Very Defensive
- Defensive
- Aggressive
- Very Aggressive

11. Have you ever ridden an autonomous vehicle?

- Yes
- No

12. On a scale from 1 to 10 (being 1 **Not Trusting** and 10 being **Fully Trusting**) please rate your overall trust in autonomous vehicles and their safety:

1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10

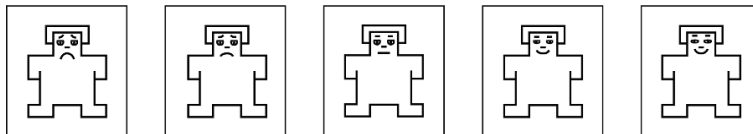
Emotional State Analysis

SAM (Self-Assessment Manikin) Scale

Choose the figure that better describes your emotional state

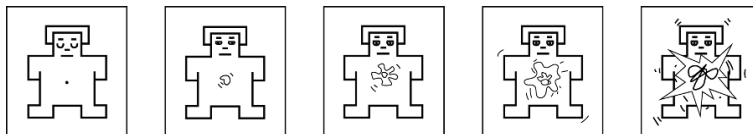
1. Emotive state

From very distressed to very euphoric



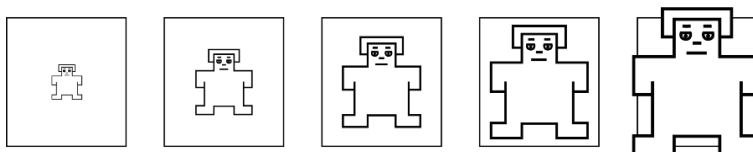
2. Degree of agitation

From very calm to very agitated



3. Degree of self-control

From no control to full control of situation



Post-Driving Questionnaire

1. How do you rate the experience of autonomous driving?

From 0 very negative to 5 very positive

0 / 1 / 2 / 3 / 4 / 5

2. Did you feel comfortable wearing physiological sensors?

From 0 very uncomfortable to 5 very comfortable

0 / 1 / 2 / 3 / 4 / 5

3. Would you like to repeat the autonomous driving?

From 0 absolutely no to 5 absolutely yes

0 / 1 / 2 / 3 / 4 / 5

Please read each statement and circle the appropriate number to indicate how you felt during the drive about various situations and driving characteristics. There are no right or wrong answers. Do not spend too much time on any one statement.

1. Vehicle speed?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

2. Vehicle acceleration?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

3. Vehicle jerk?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

4. Vehicle speed on turns?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

5. Vehicle braking?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

6. Distances from other vehicles?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

7. Distances from pedestrians?

Very Uncomfortable / Uncomfortable / Comfortable / Very Comfortable

Please read each statement and circle the appropriate number to indicate how you felt during the drive. There are no right or wrong answers. Do not spend too much time on any one statement.

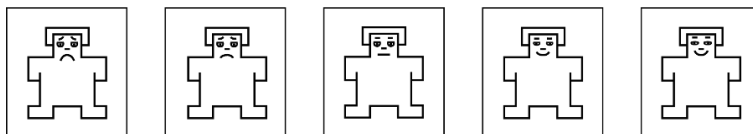
1. I felt nervous
Not at all / Somewhat / Moderately / Very Much
2. I felt jittery
Not at all / Somewhat / Moderately / Very Much
3. My body felt tense
Not at all / Somewhat / Moderately / Very Much
4. I felt tense in my stomach
Not at all / Somewhat / Moderately / Very Much
5. My body felt relaxed
Not at all / Somewhat / Moderately / Very Much
6. My heart rate was racing
Not at all / Somewhat / Moderately / Very Much
7. I felt my stomach sinking
Not at all / Somewhat / Moderately / Very Much
8. My hands were clammy
Not at all / Somewhat / Moderately / Very Much
9. My body felt tight
Not at all / Somewhat / Moderately / Very Much

Emotional State Analysis

SAM (Self-Assessment Manikin) Scale

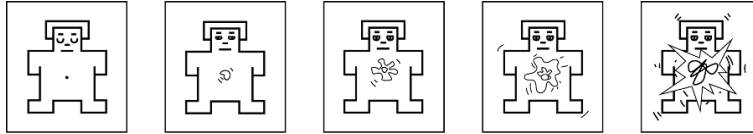
Choose the figure that better describes your emotional state

1. Emotive state
From very distressed to very euphoric



2. Degree of agitation

From very calm to very agitated



3. Degree of self-control

From no control to full control of situation

