Techniques for effective virtual sensor development and implementation with application to air data systems

Doctoral Dissertation
Doctoral Program in Aerospace Engineering (32.th cycle)

Techniques for effective virtual sensor development and implementation

with application to air data systems

Alberto Brandl
* * * * * *

Supervisors
Prof. M. Battipede, Supervisor
Prof. P. Gili, Supervisor

Politecnico di Torino

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Alberto Brandl

Turin, July 9, 2020

Summary

Between 2018 and 2019, two accidents with fatal results happened due to technical reasons on the Boeing 737 MAX 8. Although the catastrophes were related to multiple causes, the core was a fault on the sensor with the task of measure the angle of attack. Several alternatives have been studied during the last 50 years but, though a huge literature exists on the subject, most of the technological solutions proposed by researchers didn't go further Technology Readiness Level (TRL) 5. These alternatives can be found in literature as virtual sensors or synthetic sensors and they are studied to be applied in the analytical redundancy framework or as primary sensor in a more forward-thinking design.

At the time of writing this dissertation, another revolution is happening in the aeronautical industry related to inhabited and autonomous vehicles. This phenomena worth 19.3 billion $ in 2019 with a Compound Annual Growth Rate of 15.5 % between 2019 and 2025, resulting in 45.8 billion $ in 2025. The design of this kind of platform has some peculiarities, such as the high demand of innovative avionics involving autonomy and usage of Commercial Off-The-Shelf (COTS) components driven by the fast development of the market itself. These aspects result in difficulty in fulfilling the Size, Weight and Power (SWaP) requirements. An example of critical system that introduces an high number of components is the Air Data System (ADS). Unfortunately, a method able to meet the redundancy required by aeronautical regulations for the ADS is still not ready at the time of writing this dissertation. In fact, nevertheless of the number of studies on the topic of virtual sensors, the hardware redundancy is the unique solution for the above-mentioned safety-critical systems. However, the hardware redundancy might be not suitable for the Unmanned Aerial Vehicle (UAV) and Urban Air Mobility (UAM) fields.

The main reasons behind this slow transition between research and industrial application of synthetic estimation must be searched in the approach used to design the synthetic alternatives of the physical probes. The ideal estimator does not exist and the available alternatives cannot be compared between each other. To respond to the market needs, a basis for comparison must be provided. The approach proposed by this dissertation is inspired by the one followed in the case of traditional sensors: the definition of a design process based on a shared definition of the uncertainty of the sensor. Unfortunately, those metrics have never been proposed

in the past so this dissertation sets a first case of common definition. In fact, the results showed in literature focus on the analysis of the error timeseries and sometimes on some other global metrics as the mean error. This approach has some flaws that are described in this work.

At the beginning of this project, the main aim was the practical implementation of the virtual sensor called Smart-Air Data, Attitude and Heading Reference System (Smart-ADAHRS), which showed higher uncertainty when working with flight data than with simulated data. This dissertation hence proposes a set of techniques to design and optimize the performance of a synthetic sensor. In this particular case the core is a neural network, hence some of the proposed methods focus on the training of the network. Moreover, the findings in the neural network field given in this dissertation, as the Training Set (TS) analysis or the derivation of the Jacobian and Hessian matrices, are general and they are not limited to the case of Air Data estimation.

The dissertation starts with a detailed definition of the air data, followed by the literature review of the field of synthetic estimation. The mathematical aspects of the problem of estimation of the aerodynamic angles are described together with the description of the ill-posedness of the problem. A chapter is dedicated to the theoretical aspects that can help to move from the preliminary design of the virtual sensor to the tuning of the results. In the same chapter, the formulations for the Jacobian and Hessian matrices of a feed-forward Multilayer Perceptron (MLP) are derived. The experimental setup is described in a dedicated chapter followed by the proposed method of analysis and data augmentation. The final comparison shows the improved homogeneity of the uncertainty, which can be lowered to some degree for the angle of attack. The sideslip angle is also discussed, although the uncertainty due to the experimental setup reduces the confidentiality of the reference.

Acknowledgements

I would like to acknowledge the aircraft manufacturer Ing. Nando Groppo srl that provided the test aircraft for the data showed in this work, together with prof. Trainelli and prof. Rolando of Politecnico di Milano for leading the flight test campaign. Computational resources were provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://www.hpc.polito.it)

To my family.
To the passion of the lecturers.

Regrets, I've had a few
But then again, too few to mention
I did what I had to do
And saw it through without exemption
I planned each charted course
Each careful step along the byway
And more, much more than this
I did it my way

My Way, Frank Sinatra

Contents

List of Tables

List of Figures

Symbols

A_C Area of the cardinality

$a_{X,i}$ coordinate acceleration with respect to the reference system considered inertial, with components with respect to X, in $[\mathrm{m\,s^{-2}}]$

α Angle of Attack

β Angle of Sideslip

b Wingspan

i_D Down component

d_{nrd} Density-closeness function

i_E East component

i_{Body} ith Body axis

L_C Arithmetic mean of the first n values of cardinality

i_N North component

n_i proper acceleration as measured by an accelerometer on ith axis, in [g] or $[\mathrm{m\,s^{-2}}]$

x_{NN} Output of the neural network

p roll rate

ϕ roll angle

p_s static pressure

ψ yaw angle

q pitch rate

q_c impact pressure

r yaw rate

S Wing surface

θ pitch angle

x_T **or** x True or reference signal

$\mathbf{V}_\infty$ Aircraft (AC) velocity vector with respect to the surrounding control volume

x_{VS} Output of the virtual sensor

$\mathbf{W}$ Wind vector

$\tilde{x}$ Final estimation of the signal x

$\dot{x}$ Time derivative of the signal x

$\hat{x}$ Initial estimation of the signal x

Acronyms

AC Aircraft xix, , 2, 3, 6, 8, 14–18, 20, 21, 29, 54, 60, 83, 99, 101–103, 124, 131, 138, 151, 153–155

ADAHRS Air Data, Attitude and Heading Reference System xi, , 7, 55, 57

ADM Air Data Module , 18

ADS Air Data System iii, viii, xi, xiii, , 3–5, 7, 8, 10, 12–14, 17, 18, 26, 55, 151

ADU Air Data Unit , 55–57

AEKF Adaptive Extended Kalman Filter , 17, 18

AeroDAD Aerodynamics Derived Air Data , 4, 13

AHRS Attitude and Heading Reference System xi, xiii, , 7, 8, 17, 55, 58

ANN Artificial Neural Network , 12, 14, 15, 36, 56, 60

AOA Angle of Attack xi, xiii–xvi, , 2–8, 11–21, 25, 26, 28, 31, 54, 56, 58, 59, 83, 84, 94, 98–100, 103, 104, 116, 118, 120, 121, 124, 127, 130–135, 147, 149–151, 153, 154

AOS Angle of Sideslip x, xiii–xvi, , 2–5, 7, 8, 12–15, 17–21, 25, 26, 28, 31, 54, 56, 58–60, 65, 84, 98–100, 103, 104, 114, 116, 118, 120, 121, 124, 127, 132–135, 146–150, 153, 154

AUVSI Association of Unmanned Vehicle System International

AWS Automatic Weather Station , 21

CABS Computational Alpha-Beta System , 4, 14

CAD Computer-Aided Design , 101

CAS Calibrated Airspeed , 18

CDF Cumulative Distribution Function xiv, xvi, , 78, 80, 114, 130, 149, 154

CFD Computational Fluid Dynamics , 17, 130

CG Center of Gravity , 13, 57

COTS Commercial Off-The-Shelf iii, , 30

CPU Central Processing Unit , 128, 129

CRV Crew Return Vehicle , 13

CS Certification Specifications , 18

CV Cross Validation x, xiv–xvi, , 35, 40, 41, 65, 94, 95, 114, 120–122, 124, 127–131, 138–141, 145, 146, 148, 152–154

DA Data Augmentation xvi, , 130, 131, 135, 151

DAQ Data acquisition , 57

DoD Department of Defense (US government) , 9

DOF Degree Of Freedom , 14

DTFT Dropped Transonic Flight Test , 14, 15

DULV Deutsche Ultraleichtflugverband e. V. , 53

EAS Equivalent Airspeed

EASA European Aviation Safety Agency , 7, 9

EKF Extended Kalman Filter , 14–17, 21

EU European Union , 26

FADS Flush Air Data System , 12, 13, 23

FCS Flight Control System , 11, 13–15, 20, 22, 29

FDC Flight Dynamics and Control toolbox , 102

FDI Fault Detection and Isolation , 5, 15, 23

FEK FTE Electronic Kneepad , 56

FHA Functional Hazard Assessment

FOV Field Of View , 5

FP-NARX Functional Pooling Nonlinear AutoRegressive with eXogenous excitation , 14

FR Failure Rate xi, , 4, 8

FT Flight Test , 60

FTA Fault Tree Analysis

FTB-1 Flying Test Bed 1 , 14

FTE Flight Test Engineer , 56, 59, 83

FTI Flight Test Instrumentation ix, , 54, 55, 83, 85, 103

GA General Aviation

GAN Generative Adversarial Network ix, , 101, 105–107, 154

GES Global Exponential Stability , 17

GNSS Global Navigation Satellite System xi, , 15, 29, 31, 57

GPS Global Positioning System , 13, 17, 21, 26, 55

GPU Graphics Processing Unit , 128

GS Ground Speed , 16

HPC High Performance Computing , 128, 129

IAS Indicated Airspeed

IMU Inertial Measurement Unit , 14, 15, 17, 21, 57

INS Inertial Navigation System , 13, 18, 26

ISS International Space Station , 13

KF Kalman Filter , 16, 18, 20, 22, 31

LHS Left Hand Side

LIDAR Light Detection And Ranging , 23

LKF Linear Kalman Filter , 18

LM Levenberg-Marquardt , 27, 28

LMS Least Mean Squares , 18

LND Landing , 90, 91

LS Least Squares modeling , 16

MBCV Manoeuver-based Cross-Validation xv, xvi, , 120, 123, 131, 138, 147, 149

MBCVDA Manoeuver-based Cross-Validation with Data Augmentation xvi, , 127, 129, 131, 135

MEMS Micro-ElectroMechanical Systems , 57

MIDAS Modular and Integrated Digital Probe for SAT Aircraft Air Data System , 26, 29, 154

MISE Mean Integral Squared Error , 41

ML Machine Learning , 6, 11, 29, 38, 63, 69, 101, 105, 153, 154

MLP Multilayer Perceptron iv, viii, ix, xiii, , 15, 24, 26, 29, 36–38, 42–46, 48, 49, 63, 65, 71, 78, 97, 99, 101, 105, 108, 111, 113, 120, 124, 152, 153

MMU Mnemosine Main Unit , 55

MS Master of Science , 54, 59

MSE Mean Squared Error , 38, 71, 78, 130, 145

MTBF Mean Time Between Failures , 8

MTOW Maximum Takeoff Weight , 53

NASA National Aeronautics and Space Administration , 4, 9, 10, 12, 13

NED North East Down reference frame , 57

NN Neural Network xiv, , 13, 24–26, 28, 32, 35, 37, 38, 40–42, 63–68, 71, 93, 97–99, 101, 113, 115, 118, 120, 127, 130, 131, 136, 138, 140, 151, 152, 154

NSSE Normalized Sum-of-Squares Error xv, xvi, , 63, 94, 120, 123, 147

OAT Outside Air Temperature

ODE Ordinary Differential Equation , 102

PDF Probability Density Function ix, x, xiii, xiv, xvi, , 38, 39, 65, 68–72, 75, 81, 82, 105, 114, 124, 130, 149, 152, 154

PLA Power Lever Angle , 13

PLL Phase-Lock Loop , 31

RAMS Reliability, Availability, Maintainability, and Safety xi, , 8

RBF Radial Basis Function , 15

RPROP Resilient Propagation , 27, 28

RTSM Real-Time Stability Margins , 14

RVDT Rotary Variable Differential Transformer , 6

SADS Synthetic Air Data System , 4, 14–16

SFDIA Sensor Fault Detection, Isolation and Accommodation , 5

SFF Sensor Fusion Filter , 4

SHSS Steady Heading Sideslip xiii, , 60, 61, 120

Smart-ADAHRS Smart-Air Data, Attitude and Heading Reference System iv, viii, ix, xi, xiii, , 4, 24, 26–29, 31, 33, 35, 53–60, 84, 99, 110, 113, 151, 153

SSA Safety System Assessment , 3

SSE Sum-of-Squared Error , 63, 71

SWaP Size, Weight and Power iii, , 4, 5, 13, 14

TAS True Airspeed xiii, , 2, 14, 15, 19, 104

TAT Total Air Temperature

TO Takeoff , 90, 91

TRL Technology Readiness Level iii, , 24, 26, 53, 113, 151, 154

TS Training Set iv, ix, x, xiv–xvi, , 28, 32, 38, 41, 63–69, 71–75, 77–84, 94, 97, 102, 105, 114–116, 118–121, 124, 125, 127, 129–131, 138, 145, 146, 152–154

UAM Urban Air Mobility iii, , 151

UAT Universal Approximation Theorem viii, , 35, 38–41, 67, 68, 152

UAV Unmanned Aerial Vehicle iii, , 4–6, 15, 17, 21, 151

UKF Unscented Kalman Filter , 17, 20

ULM Ultra Light Machine , 53, 55

VAD Virtual Air Data , 4

VS Virtual Sensor xvi, , 6, 8, 23, 26, 29, 40, 68, 82, 94, 96–98, 114, 115, 120, 130, 131, 140, 145, 146, 150–155

WN White Noise , 19

Chapter 1

Introduction

Some of the analysis shown in this chapter have already been published in [1].

1.1 Definition and importance of the Air Data set

The term *Air Data* collects the entire set of flight parameters related to the reciprocal interaction of a flying body with the surrounding air. Its importance grounds on the well-known fact that this interaction is one of the sources of generation of the forces and moments acting on the aircraft.

The set includes:

- *static pressure* p_s
- *impact* and *dynamic pressure* q_c
- *aerodynamic angles* α (or AOA) and β (or AOS)
- *airspeed* V (with its various forms Indicated Airspeed (IAS), Calibrated Airspeed (CAS), Equivalent Airspeed (EAS), TAS)
- *Mach number* M
- *static temperature* T (or Outside Air Temperature (OAT))
- *total temperature* T° (or Total Air Temperature (TAT))

Based on these data, the application of basic flight mechanics and aerodynamics equations allows to obtain a second set of flight parameters. This second set could include the barometric altitude (with its several definitions), the speed of sound, the type of aerodynamic field (subsonic, transonic, supersonic, hypersonic), etc...

This work focuses on the determination of the aerodynamic angles. They are defined as the relative orientation between the *Body axes* reference frame and the *Air Trajectory* or *Wind axes* reference frame. Geometrically, β is the angle between the aircraft velocity vector $\mathbf{V}_\infty$ with respect to the surrounding control volume and the plane of symmetry of the AC. α is the angle between the projection of $\mathbf{V}_\infty$ on the plane of symmetry of the AC and the X Body axis. Figure 1.1 shows the definition of AOA and AOS.

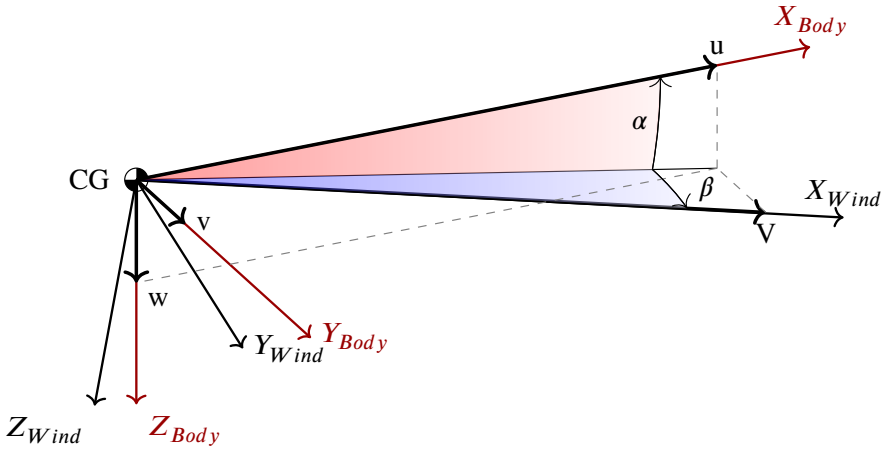


Figure 1.1: Aerodynamic angles definition

The choice of the X Body axis is not unique, however this does not affect the definition of α. In fact, taking as a reference a principal axis or the thrust axis the difference will result in a rotation that, under some slight assumptions, corresponds to a constant angle in the definition of α. The assumptions are as follows: the AC does not change significantly in mass or geometry and the thrust belongs to a constant thrust axis. AOA and AOS correspond to the Bryant angles in the rotation between the Wind axes and the Body axes $(\Psi, \Theta, \Phi) = (-\beta, \alpha, 0)$ (Rotation ZYX of the Wind reference frame to the Body reference frame).

The subset composed by TAS, AOA and AOS is also referred as the *air triplet*.

Generally, force F and moment M acting on an AC can be written as:

$$F = q_c S C_F \tag{1.1}$$

$$M = q_c c S C_M \tag{1.2}$$

where S is a reference surface (usually the wing area) and c is a reference length (usually the mean aerodynamic chord or the wing span b). Applying the Bryan's method to the nondimensional coefficients C_F and C_M, they result heavily dependent on α and β [2]. For this reason, some control systems uses these two angles as

feedback signals [3]. In fact, they represent a direct information on the dynamics of the aircraft itself. It must be noticed that C_F and C_M are nonlinear in α and β.

The application as feedback signals is not the only reason why it is important to measure the aerodynamic angles. The intrinsic nonlinearity of the functions $C_F = C_F(\alpha, \beta)$ and $C_M = C_M(\alpha, \beta)$ makes the flight unsafe and unpracticable when AOA approaches the stall region, which reduces the flight envelope in highly maneuverable vehicle [4]. Several phenomena arise from these non-linearities, for instance the aerodynamic stall or the spin. It is possible however to reduce the occurrence probability of these dangerous phenomena measuring AOA and AOS and providing a certain degree of *Situational Awareness* to the pilot, with traditional systems as an AOA indicator or a Stall Warning device.

Actually, several systems can benefit from the availability of these data:

- The fuel consumption can be optimized
- Autopilot systems can be improved not only in terms of fuel consumption, but also in passenger comfort, tracking error and precision landing capabilities thanks to gain scheduling based on AOA/AOS
- The pilot can be informed about the stability and control characteristics of the aircraft [5]
- Adaptive control systems can be employed
- Vehicle and structure health monitoring systems [6] can be implemented

The list could be extended. [7] reports possible applications in dropping objects, target tracking, geolocation, energy harvesting, trajectory optimization, and air traffic control. [8] adds gust suppression, aerial refueling, and localization of a ground target.

Now that the importance of an accurate measurement of the air data is clear, the vulnerability of the related physical sensors must be stressed. The system responsible to collect this set of data is the ADS. It is composed by several probes and vanes connected by pneumatic or electrical connections depending on the architecture. The implementation of a subset of the previously mentioned system brought to a series of accidents that sadly remember the concept of *safety critical* systems. The most recent example is the Boeing 737 MAX 8. In that case, a combination of events brought to the death of 346 people in two separated accidents. According to the report [9], there were 2 independent AOA sensors, one on each side of the fuselage. Boeing provides an alert called *AOA DISAGREE*, however it has not been considered a safety feature and not necessary to safely operate the AC. Being an optional feature, selected by approximately the 20 % of airlines, it was not selected by Lion Air. The Safety System Assessment (SSA) reports that all parameters based on corrected static pressure are impacted if the AOA vane

fails. The left AOA vane had a bias of 21° and the erroneous signal in input to the control system resulted in the fatal crashes. The FR of the AOA functional group as reported in [9] is shown in Table 1.1.

Table 1.1: AOA FR as reported in [9]

Item		*Loss of function*	*Misleading Data*
AOA	Left	$< 10^{-3}$	$< 10^{-5}$
	Right	$< 10^{-3}$	$< 10^{-5}$
	All	$< 10^{-7}$	$< 10^{-9}$

Apart from the exposure to the external agents, physical probes have also several design requirements that must be considered during the design of an ADS. Protruding probes must be positioned in a clean aerodynamic area, they are subjected to ice accumulation and hence they must be heated. Moreover, in some cases their weight is not negligible. The so-called SWaP requirements limit their applicability in innovative platforms like UAVs. This aspect will be thoroughly discussed in Sec. 1.2.

The ADS is also applied in the study of the atmosphere. In fact, it provides the measurements for the *wind estimation*, that is the estimation of the wind velocity vector. This field of study has recently gained more attention in terms of research impact than the Air Data estimation field. However, there exist a tight connection with the AOA and AOS estimation. This relation will be covered in the dedicated Sec. 1.5.

An alternative to the application of external protruding probes is the virtual or synthetic estimation of AOA/AOS.

The subject of Air Data estimation without external physical sensors can be found in academic and industrial literature under different reference names such as Virtual Air Data (VAD), Synthetic Air Data System (SADS) or Sensor Fusion Filter (SFF), if strictly based on sensor fusion. Other keywords can be Computational Alpha-Beta System (CABS) from Boeing, Aerodynamics Derived Air Data (AeroDAD) from National Aeronautics and Space Administration (NASA) and Smart-ADAHRS.

1.2 Comparison of traditional ADS systems with synthetic estimation

The past 25 years have shown a rapid grown in the interest for UAV. According to [10] and [11] nearly 80000 UAVs have been produced between 1994 and 2003, worth $3.9 billion. In 2019, the UAV market worth 19.3 billion $ and a significant growth of the UAV market up to 12% per year is expected in the next decade [12].

This phenomena pushed the debate about safe integration of UAV into national airspace [13, 14]. The problem arises from the necessity to guarantee the safety of the flight keeping the peculiarities of the UAV. Their atypical architecture requires advanced control systems and it implies strong SWaP limitations. As previously mentioned, the ADS traditionally adopted the hardware redundancy with voting system in order to increase the fault tolerance. However, a recent study by Freeman et al. [15] recalled how ADS can lead to catastrophic failure even in case of hardware redundancy combined with voting systems. At the same time, an increasing number of study developed an evolution in Fault Detection and Isolation (FDI) systems. A recent systematic literature review could be found in [16]. Current research projects show an evolution of the FDI with sensor fault accommodation named Sensor Fault Detection, Isolation and Accommodation (SFDIA) system [17, 18, 19]. Some applications specific for ADS can be found in [20, 21, 22, 23]. Innovative design frameworks can be found in literature, considering also the simulation of the fault injection, see for instance [24, 25, 26, 27]

At the time of writing this dissertation, various architectures of Air Data external physical sensors exist [28, 29, 30]. Current state-of-the-art air data sensors are made of several physical units, each of them requiring power supply, a deicing system when needed and a certain number of conditioning and computing modules. Furthermore, the external sensors should be positioned in a clear aerodynamic area, avoiding or reducing the mutual interference with other external sensors. For instance, propeller will produce a turbulent aerodynamic field that will induce oscillations on both the AOA vane and the multi-hole AOA probe. In addition, in small UAV applications where a camera constitutes the most common payload, the position of the external parts should not interfere or obstruct the Field Of View (FOV) of the camera itself. For these reasons, a number of authors have considered the analytical redundancy and synthetic estimation as useful solution to the problem [15, 31].

In Table 1.2 a list of commercial sensors for AOA is shown. However, as mentioned above, size and weight requirements may not be easy to meet during the design phase of unconventional architecture vehicle such as the majority of micro, small and medium size UAV. As the analysis of Table 1.2 suggest, the ADS architecture and size can vary substantially [32]. The lightest sensors are very simple but often they are not provided with an anti-ice or de-icing system. Current state-of-the-art ADS may be divided in two main groups:

1. *Conventional probes*: a different external unit for AOA, AOS, static and dynamic pressure (usually combined in the well-known Pitot-Static probe), temperature, recently analitically integrated by multi-sensor data fusion techniques [33];

2. *Multi-Function probes*: several multi-hole probes placed in particular position of the aircraft, usually the nose, integrated with a complex algorithm based

on curve calibration

In 2001, Tranchero and Latorre introduced their work stating that the approach for flight-critical systems was generally based on the definition of a proper level of hardware redundancy [34]. After 19 years, the strategy is quite similar, although a lot of effort has been put on the research and development of virtual sensors [35, 36, 37].

As anticipated, a VS is a software module able to provide a system with a given quantity without actually employing the physical sensors (or at least reducing the number of sensors) that would be needed to measure that quantity. This software implements an estimator that can be based on state observers, analytic formulations, Machine Learning (ML) techniques, or other types of algorithms. In research, the VS term usually stands for the core estimator, neglecting the interfaces or the other internal operations that must be implemented in a real application. It does not have the same significance of *analytical redundancy* because a VS is implemented in an analytical redundancy scheme but, in theory, it could be the unique source of the quantity.

In this framework, it is possible to understand one of the main advantages of using a virtual sensor. In fact, implementing a software solution brings to reduce the number of external physical sensors. This attenuate the inevitable and direct impact on the affordability of the system due to the high level of replication of equipment, as reported by Tranchero and Latorre.

Current research in UAV revealed additional limitations for designers. The restriction of available space and weight may be a question on system design, especially when talking about sensors.

In both cases, the classical hardware redundancy will multiply the number of units and connections by at least three, for a triplex physical redundancy, or even four. Moreover, in some cases the increased number of external units might not avoid reliability issues. As reported in [38], an investigation conducted by Airbus and Thales showed that an incorrect removal of machining oil during the manufacturing process of AOA resolver can bring to delayed or reduced AOA vane movement. This kind of fault could affect more than one sensor and hence could lead to delayed activation or non-activation of the AOA protection system. Eventually, the AC could exhibit a reduced controllability. In [39] the blockage of two AOA probes during climb led to the activation of a protection system on Airbus A321. In a worst case scenario, pilots could become not able to oppose to a nose down command if the Mach number increases. In the same Airworthiness Directive [39] the AOA sensors is claimed as necessary to maintain the required safety level of the aircraft.

Sensors based on moving parts might be preferred to multi-hole probe depending on the accuracy of the angle determination. Current state-of-the-art solutions involves potentiometers, Rotary Variable Differential Transformers (RVDTs) and

Table 1.2: Examples of Commercial Air Data Probes

Manufacturer	*Model*	*Weight [g]*	*Heater Power [W]*
UTC Aerospace	0012 AOA Transmitter	567	425
SpaceAge Control	4239-01	454	100
SpaceAge Control	101100 (micro air data boom)	142	unheated
SpaceAge Control	100900	5440	not available
AMETEK	Total Air Probe	900	not available
AMETEK	AOA transducer 25147A	816	270
AMETEK	AOA transducer 2568A	1814	270
Aerosonic	AOA	1360	150
Aerosonic	SWT	1360	190
Aerosonic	SWT	1360	450
Ack Emma LLC	CYA-100	56	unheated

synchro. Permanent magnet solution is presented in [40]. However, as reported in previously cited European Aviation Safety Agency (EASA) Airworthiness Directive ([38], [39]) moving parts might be subjected to delayed motion or even blockage. Alternative solutions have been discussed in the past literature as can be seen in [41], where moving vane and fixed fin equipped with strain gauge have been analysed. Another patent related to a multi-hole probe can be seen in [42]. The exposed part of the sensor must comply with safety regulations about de-icing, as could be the MIL-STD-810. Where possible, the external structure could be aerodynamically designed to passively avoid the ice build-up without heating (see [43]). As seen in Table 1.2 values between 150 W and 400 W per probe could be considered valid for the electrical heater consumption (see [44]-[45]). Hardware redundancy will multiply the power requirements. Hence, a reduction on the number of external probes might be considered a possible alternative to the current state-of-the-art solutions.

Table 1.3: Examples of Commercial ADAHRS

Manufacturer	*Model*	*Provides AOA/AOS*
Cobham	ADAHRS	no
Northropp-Grumman	LCR-300	yes
Honeywell	KSG 7200	no
Archangel	AHR300A	no

In some cases, the ADS is integrated with the AHRS to form the ADAHRS. Due to the high number of combinations between sensors and processing unit, comparing the ADAHRS architectures is not easy. Some examples of ADAHRS units are reported in Table 1.3. To the best of our knowledge, only one is able to

provide AOA or AOS signal among the ones listed. Often an additional equipment is required. RAMS performance of the entire ADS architecture must be taken into account as well. Table 1.4 shows related to FR and Mean Time Between Failures (MTBF) taken from [46], [47].

Table 1.4: ADS and AHRS RAMS Performance

Item	*FR* $[10^{-6}h^{-1}]$	*MTBF* $[h]$
AOA sensor	50	20000
Air Data Probe	20	50000
Electrical Connector	0.0163	$61.35{\cdot}10^{6}$
Pneumatic Tube	0.1104	$9.05{\cdot}10^{6}$
Air Data Computer	130	7692
Gyroscope or accelerometer	64	15625
GPS Antenna	20	50000
GPS Receiver	20	50000
Power Supply	31	32000

Another aspect that must be highlighted is the effort of system integration that the AC manufacturer undergoes when installing a new probe. Usually, the allocation of the probes comes after a series of decision on the AC design, further limiting the positioning aspect of the ADS. Frequently, the nose is the most desired place for the ADS probes but, even if it sounds strange, often there is no possibility to mount them. Moreover, a certain number of flight hours must be considered to characterize and calibrate the probe [48, 49]. Sometimes, unexpected aerodynamic phenomena appears such as quick condensations or shock interactions.

The analytical redundancy/VS solution is hence particularly useful when there is a control system which needs a reliable AOA/AOS signal and it is difficult to meet redundancy requirements or, in the worst case scenario, it is not even possible to place the traditional sensor in the right position.

1.3 State-of-the-Art of the estimation of Air Data

The concept of the measurement of the Air Data avoiding physical sensors is definitely not new. The possibility of a fault on a external physical sensor that is subjected to several environmental agents, objects and maintenance related issues, was already known in the 80s. With the advent of advanced flight control system and the application of Air Data as direct or indirect feedback, to find an alternative to the physical sensors became necessary, to avoid the failure of the mission, or even catastrophic events with human losses.

It must be noticed that there is no agreement in the documents of the major regulatory agencies on the definition of *fault* and *failure*. The most simple and

broad defintion is given by EASA. In the document CS-Definition *Definitions and abbreviations used in Certification Specifications for products, parts and appliances* [50] the two terms are considered with the same meaning.

> Fault (or) Failure means an occurrence which affects the operation of a component, part, or element such that it can no longer function as intended.

In this definition it is clearly indicated that both terms refer to a malfunctioning of some element of the aircraft. It is not specified if the effects of this occurrence are visible or detectable.

One of the most reported definition can be found in the MIL-STD-721C *Definitions of Terms for Reliability and Maintainability* [51]. This standard has been canceled without replacement the December 5, 1995 with the MIL-STD-721C Notice 2. However, several other MIL-STD and MIL-HDBK, which are active at the time of writing this thesis, keep the reference to it as vocabulary. The same definition that are reported here have been adopted by other agencies. It is written:

> Failure: The event, or inoperable state, in which any item or part of an item does not, or would not, perform as previously specified. Fault: Immediate cause of failure (e.g. maladjustment, misalignment, defect, etc.).

The main difference between these two definitions is the distinction and relationship applied by the Department of Defense (US government) (DoD) between *fault* and *failure*, reporting that the failure is caused by a fault. This implies the fact that a fault cannot be detected simply looking at the malfunctioning of an item, because it cannot be excluded that different kinds of fault could bring to the same effect (failure). As said before, this definition has been used in other vocabulary and the NASA-STD 8709.22 with Change 1 *Safety and Mission Assurance Acronyms, Abbreviations, and Definitions* [52] by NASA is a clear example. It extends the definitions reporting two alternative explanations. It is quoted:

> Failure: [1] Inability of a system, subsystem, component, or part to perform its required function within specified limits. [2] Non-performance or incorrect performance of an intended function of a product. A failure is often the manifestation of one or more faults (in [53] is added the important note "and is permanent.")
>
> Fault: [1] An undesired system state and/or the immediate cause of failure (e.g., maladjustment, misalignment, defect, or other). The definition of the term “fault” envelopes the word “failure” since faults include other undesired events such as software anomalies and operational anomalies. [2] An inherent defect in a product which may or may not ever manifest, such as a bug in software code.

The important step done by NASA is the specification that a fault may not manifest, so that the detection of the fault itself could be impossible. At the same time, it is stressed that a failure could be the result of simultaneous different faults. It is interesting to notice that in another NASA document, the *NASA Reliability and Maintainability (R&M) Standard for Spaceflight and Support Systems* [53] an important detail is added, stating that a failure is *permanent*. The ISO13372-2012 *Condition monitoring and diagnostics of machines - Vocabulary* [54] (paragraph 1.7, 1.8) defines as follows:

> Failure (1.7): termination of the ability of an item to perform a required function (1.9) Note 1 to entry: Failure is an event as distinguished from fault (1.8), which is a state. Note 2 to entry: Failure is the manifestation of a fault. Note 3 to entry: A complete failure of the main capability of a machine is a catastrophic failure (as defined by the end user)
>
> Fault (1.8): condition of a machine that occurs when one of its components or assemblies degrades or exhibits abnormal behaviour, which may lead to the failure (1.7) of the machine (1.10) Note 1 to entry: A fault can be the result of a failure, but can exist without a failure. Note 2 to entry: Planned actions or lack of external resources are not a fault.

Here, the definitions share several details with the previous ones, even if slightly more detailed. This document does not consider a simple malfunctioning as a failure but it relates the term to a complete termination of the function, differently to the *permanent* concept found in [53]. It is interesting to notice the Note 1 of the comma 1.8 explaining that a failure can cause a fault.

In literature, a definition can be found in [16]

> A fault is an unpermitted deviation of at least one characteristic property or parameter of the system from acceptable/usual/standard conditions. A fault may lead to a failure, which is a permanent interruption of the system ability to perform a required function under specified operating conditions.

Marzat et al again refer the failure to a *permanent* interruption.

This introduction is needed in order to anticipate one of the final statements of this thesis, the necessity of a common basis to permit the comparison of the several valuable works found in literature.

1.3.1 Taxonomy of synthetic ADS

In the last 45 years, researchers developed several solutions to the Air Data Estimation problem. In this dissertation, a total amount of 62 articles encompassing both air data estimation and wind estimation have been analyzed. The following classification can be proposed:

- solution based on *classical aerodynamic coefficients*: the flight parameters are evaluated by inversion of the classical dynamic model of the aircraft. Unfortunately, the dynamic identification of an aircraft hardly matches perfectly with the actual vehicle. When tested with sensor noise these methods showed clear degradation of performance [55]

- *model-based data fusion*: in this category fall every solutions for which the Air Data parameters are obtained with a Kalman filter, usually based on dynamic identification of the aircraft. Although the problem of the dynamic coefficient is not completely solved, these techniques can be properly designed to manage sensor noise [56, 57]. Moreover, kinematic models are often applied in place of the dynamic one.

- *model-learned estimation*: a method falls in this category if the dynamic model is not manually estimated but it is determined through an algorithm. In brief, every data-driven method based on ML that learns to estimate the desired flight parameters is collected in this class.

The last category is usually found in literature as *model-free* estimation. In this work, *model-learned* is preferred for two main reasons: the first one is functional and the latter is semantical. As a result of the theoretical relationship that grounds this problem, if an estimator evaluates the flight parameters from other measurements taken on board, depending on the architecture, it is likely based on a dynamic model of the aircraft. The second aspects is based on scientific nomenclature. In aeronautical engineering, the term *model* is usually applied to the dynamic model in a state-space formulation $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}, ...)$. On the contrary, a statistical model is a mathematical relationship between one or more random variables or, more formally, a set of probability distributions on the sample space [58]. Because ML comes from statistics, it might be useful to use some of the nomenclature of the field. The author's opinion is that the term *model-learned* might be more advisable to *model-free*, because it moves the attention to the important step of finding the model. As *black-box*, the term *model-free* can hide in part the design phase without stressing the interest on how the model (the statistical one) has been obtained and which aircraft model can describe the result.

1.3.2 Historical remarks

Historically, the first methods of AOA estimation were described in 1969, though referring that the idea of an AOA estimator was already 20 years old at that time [55]. One of the methods described in this report was later implemented and discussed in [59], earning the right to be referred as the Freeman's method. It applies a dynamical model to the acceleration measured onboard and, given the Flight Control System (FCS) state and the flight condition, it evaluates the

AOA. Figure 1.2 shows some of the most important steps in the evolution of the AOA/AOS estimators.

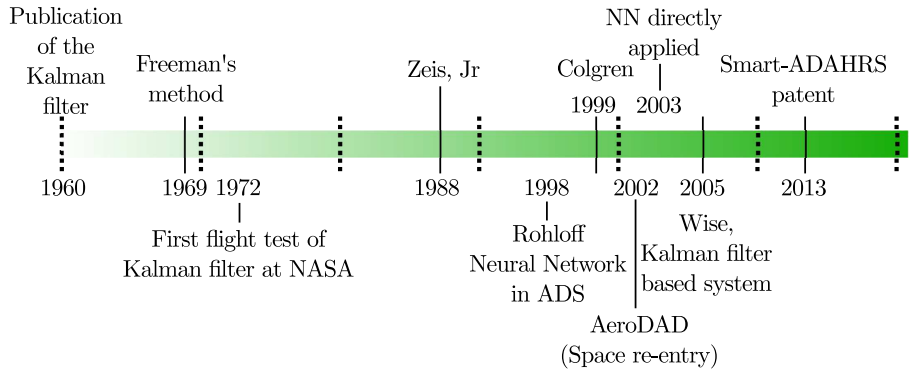


Figure 1.2: Historical development of the synthetic estimation of ADS

One of the first cited publication in 1988 regarding the estimation of the AOA is by Zeis, Jr. et al. [60] with further details on the Master's thesis [61]. The author proposes a regression based on an estimation of the C_L, the altitude and the Mach number. It is hence classifiable in the dynamical model method, even if a Kalman filter was not considered at that time. As an historical remark, Kalman published [62] in 1960 and the first flight test by NASA took place in 1972 [63]. The analysis begins with a first approximation of AOA considering $\beta = 0$ and neglecting the wind vector. Then, the load factor and the AOA are estimated using the aerodynamic coefficients of the aircraft. The author refers to the difficulty to judge the accuracy of the mechanical probes, reporting that it could reach the order of 1.5 to 2 times the actual value of AOA. The algorithm has been tested on a F15-A obtaining an uncertainty between 0.5° and 1.5° (depending on the test) through a range of 17° of AOA.

In 1998, Innocenti et al. proposes an analytic relationship, valid for missiles, between the turn rate and the AOA [4]. This research is conducted to design a proportional control system capable of pursuing highly maneuverable target, thanks to the possibility of flying at high AOA. The results showed an uncertainty of 4° when the turn rate is lower than $50\,^\circ\,\mathrm{s}^{-1}$ and 7° otherwise. The analysis was conducted on nonlinear simulation of two different maneuvers: an off-boresight manoeuver against a maneuverable target and a second one with the target conducting a Cobra maneuver. In this case, only the AOA is considered and it is evaluated starting from the turn rate, the maximum attainable AOA and the maximum attainable turn rate.

In the same year, Rohloff et al. publish one of the first model-learned method for ADS [64]. An Artificial Neural Network (ANN) is applied to estimate static pressure, dynamic pressure, AOA and AOS from a Flush Air Data System (FADS) with 11 pressure port. The method flew on the NASA F/A-18B of the NASA

Dryden Flight Research Center. The declared uncertainty is 1 % on static pressure, 2 % on dynamic pressure. No information have been given regarding the uncertainty on the aerodynamic angles, but the authors state that they should be comparable with the results previously obtained by the same group (0.4° for AOA and AOS).

Since 1999, Colgren et al. started publishing a relevant series of works where the set of onboard sensors of the aircraft is applied to estimate AOA and AOS. The motivation were the fulfillment of SWaP requirements, the correction of the angles measured using vanes and probes and the substitution of the physical probes in case of failure. The estimator was still an analytic formulation based on aerodynamic force evaluation and decomposition able to reach a declared uncertainty of 1° in [65] and a maximum error of 5° in [66]. Kalman filter is cited but reported as not mandatory. From the Inertial Navigation System (INS), Global Positioning System (GPS), altitude, mass, Center of Gravity (CG) position, angular velocity, q_c, Power Lever Angle (PLA), FCS and gear positions, both angles can be evaluated. The test was conducted on simulations and flight test data of an U2-S. The flight manoeuvers applied are: turn during wind turbulence, wing level sideslips and yaw doublets. Results were compared with a reference boom. Both method and system was patented in [67].

One important branch of the air data estimation, although represented by a lower number of works, is the application for space reentry operation. Due to the extreme conditions at which a vehicle is subjected during reentry, it is not possible to mount protruding sensors to measure air data. The first article is from NASA [68]. In this article, an estimator called AeroDAD has been applied as backup ADS for the International Space Station (ISS) Crew Return Vehicle (CRV) (at that time the X-38) which was equipped with FADS. For this kind of vehicle, there is the useful opportunity to take advantage of a known reference trajectory. In fact, this work proposes a deterministic algorithm using the ratio of the specific force measurements and an aerodynamic model to obtain aerodynamic angles, q_c and the Mach number. The ratio of the accelerations is directly related to an affine approximation of the AOA. The authors relate the perturbation from the reference trajectory to a change in AOA. q_c and AOS have been reconstructed by the aerodynamic derivatives disentanglement, neglecting flap and ailerons respectively for q_c and AOS. A consideration can be made on the fact that the size of ailerons and flaps for this particular kind of vehicle can be considered relatively small. Finally, the Mach number is minimized in an optimization loop considering the temperature value and the speed of sound. The velocity estimation has been found sensitive to error on air density. AeroDAD has been tested in simulated environment in two different test cases: nominal trajectory and off-nominal trajectory. Authors declare 2° of uncertainty on AOA, which spans a range from −5° to 40°, and 1° on AOS, ranging between −5° and 5°. 10 % uncertainty on the estimated airspeed in a range between 60 m s^{-1} and 1067 m s^{-1}.

In [69, 70], an NN has been implemented to identify the aerodynamic coefficient

of an AC.

Wise in 2005 sets the estimation process in a filtering framework. The patent for method and system [71] specifically refers to the application of Extended Kalman Filter (EKF) to solve the estimation problem. This solution, named CABS, was initially developed by Boeing to support advanced missile program in 1986, however the limited computational capability at that time made impossible the application. The recent aim in [71] was to furnish an hot backup to the nose boom of the X-45A, on which the algorithm was demonstrated, to fulfill SWaP requirements and to improve reliability and survivability performance. CABS is composed by two separated EKFs, one for the longitudinal and one for the lateral-directional plane, though a coupled solution was not excluded. The declared performance are 0.5° for AOA. An initial bias of 4°, further reduced to 2° was found for what concern the AOS. This bias has been directly correlated to the AC thrust. An important note is that they state that no degradation in stability margin or command tracking was observed in closed-loop analysis using the virtual sensor as primary ADS. The planned flight test activity considered Real-Time Stability Margins (RTSM) to measure pitch, roll and yaw stability margins on the AC during flight and doublet commands to demonstrate AOA/AOS tracking. The estimated signals have actually been compared with the one measured by the simplex mechanical (dual electric) nose boom with different AC configuration: gear up/down, weapon bay opened/closed. The input were the Inertial Measurement Unit (IMU) measurements, q_c from a Pitot tube and the FCS signals, considering also the thrust vectoring peculiarity of the X-45A.

One of the first direct application of neural techniques to the design of SADS can be found in [72] where a Functional Pooling Nonlinear AutoRegressive with eXogenous excitation (FP-NARX) network has been implemented and tested on a nonlinear 6 Degree Of Freedom (DOF) simulator, considering also wind and Dryden turbulence model. The input were vertical and longitudinal accelerations, q_c, TAS, elevator deflection, stabilizer position and pitch angle. AOS was not considered. The declared uncertainty on AOA is lower than 1.1°. This value has been obtained in 28 flights lasting 100 s in a range of 15.2° of AOA. AC configuration is not fixed, both landing, take-off and clean configurations have been tested. It is notable that three virtual sensors have been designed, one for each AC configuration.

In 2009, Nebula et al. published an article in which both EKF and ANN are applied on the estimation of AOA, AOS, Mach number and other variables. In this case, the application was a space autonomous vehicle called Flying Test Bed 1 (FTB-1) in atmospherical reentry. With respect to the work of Westhelle, this estimator has been tested also on real flight test data, obtained during the first Dropped Transonic Flight Test (DTFT)-1. Simulations have been conducted for the DTFT-2. Moreover, they have weather forecast available on board, uploaded before the beginning of the mission or at least before the deorbiting operation. The knowledge of the wind can greatly helps the final estimation performance. This

system showed a maximum error of 1° in a direct comparison with the timeseries lasting 20 s measured by physical probes. The input vector in this case is made by accelerometers, rate gyros and atmospheric forecast data. The ANN is used for the interpolation of the aerodynamic coefficient and not directly for the output elaboration, whereas the EKF is applied as state observer. DTFT-1 consists in a drop from 20 km altitude, controlled flight with constant AOA to reach Mach 1.05 before slowing down. DTFT-2 considered a drop from 24 km with constant AOA, followed by a AOA sweep at constant Mach in the transonic region before slowing down. As Nebula et al. [73] observes:

> The plain estimation of the air data quantities using only inertial measurements (zero-wind estimation) can be extremely inaccurate when inertial velocity and wind velocity are comparable.

In [74] an Radial Basis Function (RBF) is used instead of MLP for the aerodynamic model inversion of a civil AC.

Another implementation of a cascaded EKF architecture can be found in [75]. In this work, the estimation is also conducted to enhance the FDI system capabilities using low-graded inertial and radio-navigation sensors. However, this solution called SADS could be classified with the model-based data fusion, because the first EKF is applied to obtain attitude and AC velocity from GNSS and IMU. The second EKF is actually the core of the air data estimation and it allows to observe for the first time the entire triplet TAS, AOA and AOS using an AC dynamic model and the FCS signals. Declared uncertainties are $2.5\,\mathrm{m\,s^{-1}}$ on TAS, 2° for AOA, 1° for AOS. These results have been obtained using both flight test data from the Ultrastick 120 testbed, a fixed-wing model, and from simulation of a Cessna 172. The simulated manoeuvers are level flight with 30° bank turns and steep 50° bank turns. The important step conducted in this research is the demonstration of the simultaneous non-observability of the airspeed and heading if the 2D horizontal problem and straight flight are considered. In fact, small heading changes (S-turn) are necessary in case of long straight-and-level flight. Unfortunately, it is not always possible to ask the pilot to maneuver in order to measure a flight parameter. One of the unknown variable was the throttle setting during flight tests, hence they have applied the value coming from the straight-and-level flight estimation. This was pointed as the major source of error during transients. [76] is a follow-up where the federated EKF architecture is again applied with a detailed observability analysis of the problem. Declared performance obtained in real environment with Ultrastick 120 are $2\,\mathrm{m\,s^{-1}}$ on TAS, 3° on AOA and 5° on AOS when bank angle is lower than 30°. The flight test points were doubles applied to the elevator, ailerons and rudder. Wind is considered and modeled as a First Order Gauss-Markov process. The algorithm resulted to be very sensitive to off-trim velocity but relatively insensitive to off-trim attitudes. Another aspects that Lie and Gebre-Egziabher reported is the possibility for a small UAV to experience a significant amount of wear during

a normal operation. This brings to the fact that it is not possible to obtain an accurate AC model. However, possible improvements proposed in the same article are the following:

- avoid the approximations $\dot{\phi} = p$ and $\dot{\theta} = q$, use the nonlinear kinematic equation
- consider the current gravity projection instead the one at the trim point
- use First Order Gauss-Markov model for the process noise
- apply nonlinear dynamic model or model scheduling

Because the dynamic model is linear, the approximation of constant airspeed in the definition of the coefficient of the state matrix brings the error to increase. The increased sensitivity at low airspeed in the estimation of the aerodynamic angles is not new and it also affects the classical sensors. Figure 1.3 is taken from [77]. The effect of the derivative $\frac{dV}{d\alpha}$ relects on the SADS because the a priori estimates is corrected using the Ground Speed (GS) innovation. However, if the GS innovation changes only a little because $\frac{dV}{d\alpha}$ is low, there is a little correction on α and β. It must be noticed that the SADS is very important at low speed.

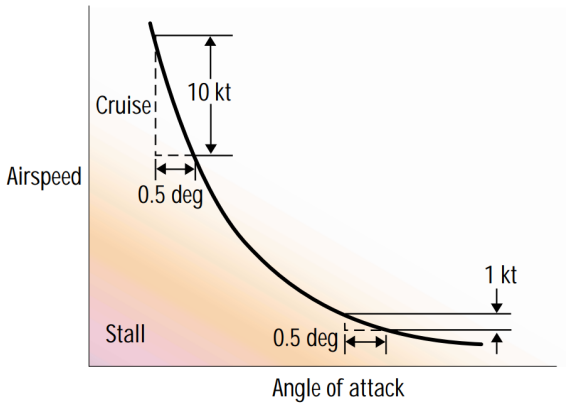


Figure 1.3: Airspeed - AOA relationship (from [77])

In this kind of state observer, it is important how the wind is considered. The wind model is actually the main difference in a lot of papers.

A series of publication can be found on the Pitot-free analytical redundancy. An example can be [78] where the velocity and wind vectors are estimated using Least Squares modeling (LS) and EKF with a final error about $1\,\mathrm{m\,s^{-1}}$ and $2\,\mathrm{m\,s^{-1}}$. In [79] an analytical redundancy approach for Pitot tube failure has been proposed using nonlinear Kalman Filter (KF). The wind state vector has been modeled as

a random walk process. A comparison between the performance of EKF and Unscented Kalman Filter (UKF) is also conducted. The algorithms have been tested on flight data coming from 2 different UAVs. In this case, the wind vane is installed and the final performance on the airspeed estimation is about $1\,\mathrm{m\,s^{-1}}$ and $2\,\mathrm{m\,s^{-1}}$.

In [80] the estimation is conducted with the aim of estimation, online calibration and fault diagnosis for ADS. They have implemented a cascaded estimator with an EKF for the AHRS data fusion, a wind velocity observer considering slowly-time varying wind and the AOA/AOS computation as final step. The authors refer that adverse conditions can influence the aerodynamic coefficients of the UAV and that the aerodynamic model can become inaccurate during agile manoeuvers with high AOA/AOS. Moreover, the application of the kinematic model in place of the dynamic model makes this solution aircraft-independent. In this work, there wasn't any direct measurement of the parameters and hence the comparison with classical sensors was not possible. However, in this case there is an analysis of the impact of the aircraft mass and architecture. In fact, the algorithm has been tested with flight test data coming from 3 different UAVs with a mass of 4 kg, 20 kg and 200 kg respectively, as demonstration of the independence from the aircraft. It is worth noticing that the article showed that maneuvers with variations in attitude can bring to the Global Exponential Stability (GES) of the estimator error dynamics. In fact, the rank of the observability Grammain matrix rises to 4 in case of flight with variation in pitch and yaw, whereas the flight at constant attitude makes the rank equal to 2. This analysis has been conducted with both fixed Pitot probe and self-aligning probe. The flight tests are made by loops at different altitudes, also depending on the UAV. The manoeuvers are: 8 loops at 500 ft and $30\,\mathrm{m\,s^{-1}}$ with 20 m of altitude oscillation or airspeed variation, and 10 loops between 1000 ft and 600 ft.

Several version of Adaptive Extended Kalman Filter (AEKF) are also demonstrated in [81], in case of fault injection. This paper compares the performance of the proposed algorithms in both simulated and flight scenarios. Two simulated fault scenarios are considered, respectively with multiple and simultaneous faults, with 3-2-1-1 aileron command. In case of operative scenario, the flight test data coming from Cessna Citation II aircraft have been enriched with simulated fault injection. Despite for the complete analysis, the duration of the timeseries is quite low, 50 s for simulation and 60 s for flight test.

In 2016, an AEKF has been presented for high performance AC to estimate the air triplet and pressures in high load factor manoeuvers. The algorithm is based on GPS, IMU and a self-aligning probe and it implements the assumption of slow-varying wind $\dot{\mathbf{W}} = \mathbf{0}$. It has been tested on data coming from a light-military training simulator. To improve the significance of the simulation, the ADS has been modeled in 2 parts: the first one simulates the ADS measurements. This part is composed by Computational Fluid Dynamics (CFD)-derived functions to consider the point of installation, the self-aligning dynamics based on a II order system and

the noise model. The second part models the Air Data Module (ADM), which evaluates the Air Data parameter using the simulated probe measurements and using Least Mean Squares (LMS) polynomials. In this way, both the measurement error and the algorithmic error are simulated. This is important, because the more the error are accurately modeled, the more it it would reduce the gap between a flight test and the simulation. The flown manoeuvers are the following:

- fast nose-up manoeuver (elevator pulse deflection 20° lasting 2 s) with constant vertical wind ($10\,\mathrm{m\,s^{-1}}$)
- fast yaw manoeuvre (pulse rudder deflection of 10° lasting 2 s) with constant lateral wind ($10\,\mathrm{m\,s^{-1}}$),
- robustness test: fast vertical gust disturbance (as Certification Specifications (CS)-25 recommends, see paper for formulas), no command variation

Please consider that the robustness test violates the basic assumptions of the model, as the same authors wrote, in which the time-variation of the wind components has been considered negligible. The uncertainty is declared lower than 1°, increasing up to 5° in case of vertical gust. This last result is also interesting because the presence of a vertical gust violates the model. In this paper, the wind is evaluated during steady-state flight and then kept constant during high load factor segments by the KF implemented.

Recently, [83] applied an AEKF for the estimation of AOA and CAS. This work is more focused on fault detection because, as Alcalay et al. observes, *Even if very improbable, simultaneous and consistent faults of two or three sources are difficult to detect*. The method has been tested on 50 min of flight data.

One of the most recent work is [84]. This article has been published in 2019 and it proposed the estimation of AOA and AOS for reducing the calibration costs of the ADS of an advanced AC. In fact, as they underline in this article, the AOA calibration can be conducted with relevant long flight test and, moreover, the AOS is more difficult to calibrate because it is more sensitive to the lateral winds. The state observer is a Linear Kalman Filter (LKF) for the wind, using CAS and INS whereas AOA/AOS are analytically evaluated once the velocity vector components are available. The following assumptions have been implemented:

- In this case, W_D has been considered negligible
- Slow wind time-variation ($\frac{\partial W_N}{\partial t} = \frac{\partial W_E}{\partial t} = 0$)

These assumptions are mostly related to neglect the ground effects and they can be considered valid at sufficient high altitude. However, being the wind vector estimated in a KF, the second assumption does not mean that the time derivatives of the horizontal wind components will be zero, because they are propagated according

to the White Noise (WN) implemented as plant noise. This solution has been tested on P1HH and M346 data and it has been compared with an high-accuracy nose-boom vanes in 1600 s of levelled flight, climb and descent. These flight phases have been selected because they constitute the common procedure for flow angle calibration. The declared difference between the reference and the estimation is always lower than 2° for the aerodynamic angles and 2 m s^{-1} for the wind estimation.

1.4 Aerodynamic Angles estimation

The estimation of AOA/AOS has been heavily studied in literature. A strong similarity has been found in the proposed methods and the solution to the problem seems to remain an open question in this field.

Define the *kinematic problem of AOA/AOS estimation* as the inverse problem of estimation of the aerodynamic angles based only on kinematic quantities and reasoning. This problem is generally ill-posed and the following brief analysis can identify the source of ill-posedness.

Figure 1.4 shows a vector representation of the problem. If $\| \mathbf{V}_\infty \|$, $\mathbf{a}$, ω_B could be measured, α and β can not be determined instantaneously. In fact, a_{tras} is obtained from Eq. 1.3

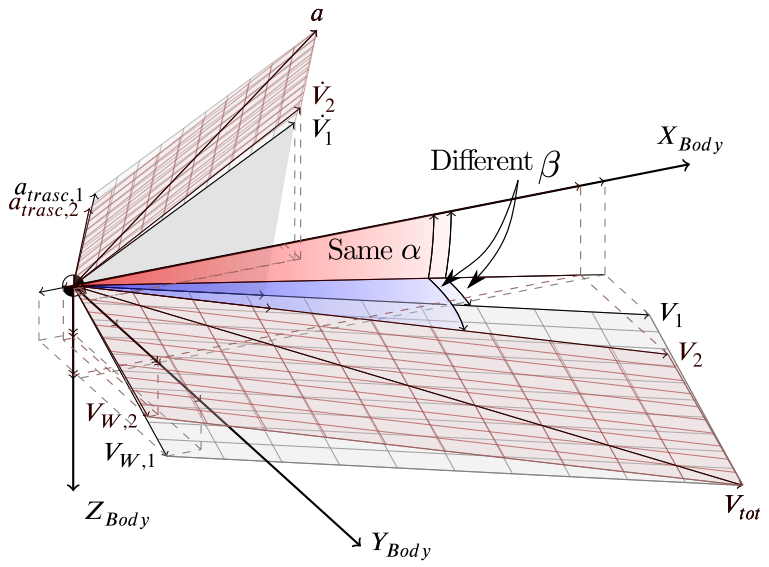


Figure 1.4: Vector representation of the problem of estimation of AOA/AOS. The same measurements of TAS, n, ω_B and V_{tot} brings to infinite $\mathbf{V}$ vector and hence the ill-posedness of the problem.

$$\mathbf{a_{tras}} = \omega_\mathbf{B} \times \mathbf{V} \tag{1.3}$$

Cross product is a Lie algebra and hence it does not have an inverse operator. In brief, there are infinite velocity vector $\mathbf{V}$ with the given measured module $\| \mathbf{V}_\infty \|$ that composed with the angular velocity vector close the problem with the right acceleration $\mathbf{a}$. A possibility is to consider also the time derivative of $\| \mathbf{V}_\infty \|$ which gives some constraints on $\dot{\mathbf{V}}$. In that case, some solution could be found and the ill-posedness comes from the fact that it is not possible to discriminate between $\dot{\mathbf{w}}$ and $\dot{\mathbf{V}}$ using an accelerometer.

The ill-posedness of the kinematic problem results in the fact that the system is globally unobservable. The observability analysis can be seen in [76].

However, a possibility to determine the aerodynamic angles could be solve the dynamic problem instead of the kinematic one. This avoid the fact that the kinematic problem is ill-posed. In fact, the $\mathbf{a}$ vector is generated by the aerodynamics and the propulsion system. An estimation of the acceleration can relate α and β to $\mathbf{a}$ so that the ambiguity is somehow reduced. Unfortunately, the knowledge of the dynamic model is needed to relate the aerodynamic angles to the accelerations. This means that the sensor will be strongly AC dependent. Moreover, once the model is known with a certain degree of accuracy, also the input variables to the model itself are needed, which means the knowledge of the state variables related to the FCS and to the engine. Sometimes the input variables of the AC are not available or the designer prefers to avoid relying on a high number of signals that could be suggested to a failure.

1.5 Wind estimation

As before mentioned, the research on the estimation of the wind vector $\mathbf{W}$ is strictly related to the estimation of AOA/AOS. This connection is so tight that the major part of the recent works belongs to the both field of study. In fact, the knowledge of the air triplet, which can be seen as the knowledge of the velocity vector $\mathbf{V}$, together with a measurement of the speed with respect to the ground allows to close the problem using the well known Eq. 1.4.

$$\mathbf{V}_{ground} = \mathbf{V} + \mathbf{W} \tag{1.4}$$

As stated by Rhudy et al. in [7], the inevitable presence of the wind is one of the major problem in the estimation of the AC states and parameters. Moreover, the same author recently refers that the validation of these techniques is very challenging because it is difficult to directly measure the wind experienced by an aircraft.

Most of these works applies KF using the assumptions similar to the one described in Sec. 1.3. [7] integrates a set of kinematic equations using a UKF assuming $V_{pitot} = u$. Results have been compared with the measurement coming from a

ground weather station (with 6° of resolution) corrected with a power law to consider ground friction effects. Constant wind direction estimation error: 16.7°. [85] applies a variant of the EKF called Three-step EKF.

Although the estimation of AOA/AOS finds more interests in fixed-wing AC, the wind estimation can also be conducted on rotorcraft and multicopter. In this case, the difficulty arise from the flux related to the rotor. In [86] a variational technique is applied. In [87] the estimation takes advantage of a link between propeller power measurement and wind velocity.

One of the most interesting works is [8]. Authors classify the methods in two classes: the first class is based on the wind triangle relationship (as Eq. 1.4), whereas the latter comprises approaches using the aircraft dynamics. They show 4 methods using Eq 1.4:

- estimation of the horizontal planar wind using Pitot-Static and planar GPS velocity estimates
- as the previous method but including the Down component of the wind
- estimation using IMU measurements
- extension of the previous formulations using also the wind vanes

An influential work is [88], when the wind estimation itself has been identified as possible important application of the UAV. In this work, an EKF is designed modeling the wind shear as first-order Markov process with 32 km of correlation distance. This method has been validated through simulation and flight test using data from Automatic Weather Station (AWS) of the Korea Meteorological Administration (at ground level, 15 km far) with usual power law correction.

One of the problems of the estimation $\mathbf{W}$ in small UAV applications, is that the frozen wind field approximation is not applicable. In other words, the wind velocity variations are similar to the vehicle speed [89]. Nevertheless, the knowledge of the wind field is crucial for the autonomous energy harvesting research by means of gust soaring.

Recently, vision-based systems have also been implemented [90].

An important theoretical result can be found in [91]. In this work, the condition number of the observability Gramian is used to measure the observability of the problem. The result is that the problem is globally unobservable. However, it is conditionally observable. They apply Lie derivatives and F-norm of the observability matrix. Authors write and prove a theorem regarding the observability of the matrix related to the slow-varying wind assumption.

A 2 stage estimation is implemented in [92]. The preliminary result obtained with Levenberg-Marquardt is then corrected with a Three-step EKF. This method showed an error between 0.1° to 0.3° on simulation.

A frequency separation method is implemented in [93].

1.6 Absence of a shared metrology

As can be clearly understood from Section 1.3, the solutions found in literature are not comparable between each other. Each relevant work hides some details or some others are simply not considered so that it is quite impossible to define which solution is the best one. Actually, once the similarities among the various algorithms have been highlighted, declared uncertainty should be more homogeneous.

The first analysis that can be conducted on this topic is a comparison between the architectures studied in literature. Hence, the selection of which parameters must be examined is very important. In this thesis, the following list has been considered:

- The study led to the estimation of the entire air data triplet $\{V, \alpha, \beta\}$ or to a subset of it. This choice can bring to analysis that are restricted to a limited part of the flight envelope. For instance, neglecting the estimation of β limit the study to the longitudinal plane and the demonstration can be incomplete
- The analysis is based or not on inertial measurements. The error injected by the sensor noise might not be neglected in some cases, especially if data fusion isn't conducted
- Filtering process. The estimation is conducted using a state observer (e.g. Kalman filter) or complementary filters are applied. The design of the KF can be simplified assuming restrictions on the problem as linearity or clean atmosphere environment
- The algorithm is based on machine learning techniques.
- The origin of the data is numerical simulation or flight test data. In the first case, the characteristics of the simulator can deeply affect the final significance of the data. In the second case, several limitations can bring to an uncomplete coverage of the flight envelope. This aspect will be discussed in Chapter 3.
- The aircraft configuration. The air data estimator is of particular interest during take-off and landing procedures, when usually flaps and/or slats are extracted. The demonstration of the results sometimes is carried out in every flight configuration, such as landing gear up/down or weapon bay opened/closed, and some other works prefer to concentrate on a single situation at a time.
- FCS values are used in the estimation or not. Due to the theoretical reasons that have been discussed Sec. 1.4, the control surface position can affect the final uncertainty obtained by the estimator. However, the availability of those signal is often put into discussion and hence some researchers preferred to avoid the implementation of the FCS in the algorithm.

The results of the analysis of the state-of-the-art are collected from Table 1.5 to Table 1.10. It is worth noticing that the absence of metrological procedures significantly reduces the reliability of the values reported in Table 1.10.

Table 1.5: Classification of the methods for estimation of Air Data, in terms of final aim

Scope	*Frequency*	*Percentage*
Estimation	19	34.55 %
Estimation for Situational Awareness	2	3.64 %
Estimation for SWaP	2	3.64 %
Estimation for AP, Model Tuning or performance improvement	5	9.09 %
FDI	6	10.91 %

Table 1.6: Classification of the methods for estimation of Air Data, in terms of architecture

Architecture	*Frequency*	*Percentage*
Classical	11	20 %
Model-based	15	27.27 %
EKF	9	16.36 %
AEKF	3	5.45 %
*Other**	3	5.45 %
Model-Learned	3	5.45 %
Other	3	5.45 %

* *Other* collects any architecture that applies different physical phenomena than the ones studied in this dissertation (Light Detection And Ranging (LIDAR), Optical flow) or FADS

This long introduction is hence needed to define the framework in which this work is integrated. Since Air Data VSs have been already proposed, the second step is to optimize the design and the performance of the sensor itself. The main aim of this dissertation is to provide a set of methods that can help to design VS. The term *design* in engineering often traduces in comparison of the obtained solutions. Actually, the design starts from the specifications, which in turn are defined based on consolidated reference metrics. For this reason, except for a training procedure, most of the proposals are based on metrics that can be used to compare Air Data VS. At the end of this dissertation, it will be shown that the reliable and repeatable uncertainty for this kind of VS can be defined.

Table 1.7: Classification of the methods for estimation of Air Data, in terms of estimated flight parameters

Parameters	*Frequency*	*Percentage*
AOA	3	5.45 %
AOA+AOS	3	5.45 %
AOA, AOS, pressures	3	5.45 %
AOA, AOS, other	4	7.27 %
AOS	1	1.82 %
Air Triplet	2	3.64 %
AOA + TAS + Wind (no AOS)	1	1.82 %
Air data + Wind	1	1.82 %
AOA + Other	1	1.82 %
Entire set of Air Data	1	1.82 %

Table 1.8: Classification of the methods for estimation of Air Data, in terms of applied sensors

Sensors	*Frequency*	*Percentage*
INS	21	38.18 %
GNSS	7	12.73 %
ADS	11	20 %
FADS	2	3.64 %
FCS	9	16.36 %
Model (any kind)	5	9.09 %
Other	12	21.82 %

1.7 Focus and scope of this study

Previous sections described the problem and how it is currently faced in literature. A gap has been identified from a metrological point of view that impedes to actually compare the solutions available. As it will be better detailed in Sec. 1.8, this study starts from a patented method that was at the beginning of the first flight test campaign when this PhD project has begun. Historically, the Smart-ADAHRS patent has been published in 2013 and reached TRL 4 after the demonstration in simulated environment. Although some research on the effect of the noise has been conducted in [95], the performance dropped once Smart-ADAHRS was installed on-board. The main question at the time of starting this PhD project was related to possible alternative ways to train the MLP in order to improve the performance of the estimator. Moreover, the possibility to certify this kind of system based on NN was an open question in this field and it still remains an open question at the time of writing this dissertation. In this framework, a comparison with the other

Table 1.9: Classification of the methods for estimation of Air Data, in terms of data origin

Data origin	*Frequency*	*Percentage*
Simulation	8	14.55 %
Flight test	12	21.82 %
Simulation and Flight Test	9	16.36 %
Wind Tunnel	2	3.64 %

Table 1.10: Classification of the methods for estimation of Air Data, in terms of uncertainty

Target	*Uncertainty Class* [°]	*Frequency*
AOA	0.2 - 0.4	2
	0.4 - 0.6	8
	1 - 1.2	7
	>1.2	7
AOS	0.2 - 0.4	2
	0.4 - 0.6	4
	1 - 1.2	6
	1.4 - 1.6	1
	>1.8	10

methods that can be found in literature is surely interesting. However, it is not the main scope of this dissertation. In fact, the identification of the gap at the basis of this problem was considered a priority, such that the attempt of establish an uncertainty associated to the various estimators could lead to a systematic comparison. This dissertation proposes several ideas, some of them are design methods, some others are more related to the design itself. For sake of clarity, a similarity with the polynomial fitting can be observed. Indeed, the final aim of polynomial fitting is to define the coefficients of a polynomial. Once the degree of the polynomial is given, the final polynomials are usually not indicated as different methods, as it is commonly done with applied NNs. In fact, moving the attention from terms like *model-free* to *model-learned* and considering two estimators different methods only when there is a structural difference (i.e. in the architecture) could focus the study on the actual procedure applied to find the final estimator. This lead to the problem of finding the best design flow to be followed to design a NN-based estimator. To follow this path, it is necessary 1) to associate an uncertainty to the estimator, such that it is possible to compare the method with the other available in literature and 2) to try several training and data analysis methods and to compare them.

This dissertation provides a set of methods and their mathematical background, with the chance of starting a path directed to the definition of a certified design

flow. In Chap. 6 a comparison of the results obtained following the proposed methods is conducted. The comparison starts from the definition of a *Baseline* NN. The Baseline NN is an example of how the design of a NN-based estimator was conducted and it works as a reference model. Finally, Chap. 7 suggests a procedure that could be the base for a certified design flow.

1.8 The case study: the Smart-ADAHRS algorithm

This dissertation shows some methods that can be applied to the design of synthetic sensor of aerodynamic angles. To demonstrate their advantages and disadvantages, they have been applied to the Smart-ADAHRS algorithm. The Smart-ADAHRS is a patented technology that has been developed by Politecnico di Torino for the estimation of the aerodynamic angles using NN [94]. The patent is now property of Aerosmart srl and it is implemented as VS in the European Union (EU) project Modular and Integrated Digital Probe for SAT Aircraft Air Data System (MIDAS) [96]. To the best of the author knowledge, this is the first case of synthetic estimation of aerodynamic angles on a certifiable ADS, developed following the design process in accordance with the aeronautical regulations. Usually, the TRL of the other algorithms is low, generally they have not been tested in operative environment. On the contrary, this algorithm has reached a stable design and the knowledge of its peculiarities allows the author to be confident in the application of the proposed methods. In this algorithm, the AOA and AOS are obtained by sum of an initial estimation $\hat{\alpha}$ (or $\hat{\beta}$) to a correction term $\Delta\alpha$ (or $\Delta\beta$) evaluated by a NN, as in Eq. 1.5. These estimations are based on data coming from conventional pressure probes and GPS/INS. In this way, only one external sensor is required. An high-level schematic is shown in Figure 1.5.

$$\begin{aligned} \alpha \approx \tilde{\alpha} &= \hat{\alpha} + \Delta\alpha_{NN} \\ \beta \approx \tilde{\beta} &= \hat{\beta} + \Delta\beta_{NN} \end{aligned} \tag{1.5}$$

The patent does not give strict indications on the architecture of the NN or how the NN must be trained. However, in recent works a main structure has been maintained. The most tested NN is a feed-forward fully connected MLP. The number of neurons and layers is limited, usually 1-2 layers with 10-20 neurons each. Although static, its simplicity should be a great advantage. In fact, even if deep learning is very common [97] at the time of writing this dissertation and interesting theoretical results have been discovered [98], the aeronautical industry still have some hesitation in the application of such untamable systems. In base to the classification given in Sec. 1.3, Smart-ADAHRS belongs to the *model-learned*

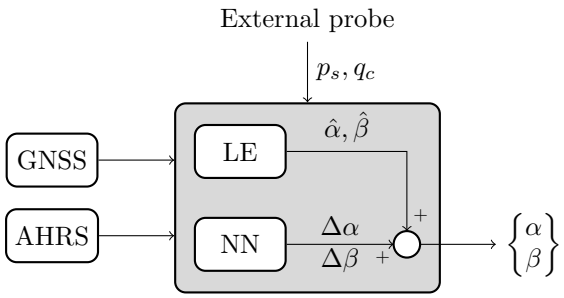


Figure 1.5: General schematic of the Smart-ADAHRS

category. One of the advantages of this category is that the application of a learning procedure able to autonomously define a model avoids the amount of computation needed to manually fulfill the same scope.

At the beginning, the Levenberg-Marquardt (LM) method was applied as heuristic rule during training (see [99], [100]), which adapt a parameter λ to pass from a standard gradient descent approach for large value of λ to a Gauss-Newton formula for small value of λ. It represents an example of trust region approach applied to Gauss-Newton method. The main mathematical description for this method is reported in Eq. 1.6, where $\boldsymbol{Z}$ is the Jacobian matrix of the error function with respect to the weights whereas $\boldsymbol{w_{old}}$ and $\boldsymbol{w_{new}}$ represents respectively the old and new weight vectors expressed in the weight space. $\epsilon\left(\boldsymbol{w_{old}}\right)$ is the residual error applying $\boldsymbol{w_{old}}$.

$$\boldsymbol{w_{new}} = \boldsymbol{w_{old}} - \left(\boldsymbol{Z}^T\boldsymbol{Z} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{Z}^T\epsilon\left(\boldsymbol{w_{old}}\right) \tag{1.6}$$

Although LM method avoids the calculation of the Hessian matrix, it is quite heavy in terms of memory and computational cost due to the evaluation of the Jacobian matrix. Moreover, there are some implications using the partial derivative of the error function for the direct modification of the weight matrix. The unforeseeable behaviour of the derivative itself could indeed bring to very slow learning or disturbances in the training procedure. To address this problem, the optimization algorithm has been changed with the Resilient Propagation (RPROP) in which the weight update step is function only of the sign of the derivative. For a complete description of the method please see [101]. General weight update step is reported in Eq. 1.7.

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \tag{1.7}$$

where

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & \text{, if } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)} & \text{, if } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0 & \text{, else} \end{cases} \tag{1.8}$$

and

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{+}\Delta_{ij}^{(t-1)} & \text{, if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^{-}\Delta_{ij}^{(t-1)} & \text{, if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & \text{, else} \end{cases} \tag{1.9}$$
$$\text{where } 0 < \eta^{-} < 1 < \eta^{+}$$

Findings show that the LM algorithm is usually the best one in terms of speed convergence and training error but, at the same time, RPROP allows to manage large amount of data. See [102] for further optimization algorithm comparison.

The selection of the input signal has been described in previous research [103, 104, 95, 1, 105, 106, 107, 108, 109, 110]. The current design is the one reported in Eq. 1.10.

$$\Delta\alpha = f_{\alpha}\left(\hat{\alpha}, q_c, \dot{q}_c, a_x, a_y, a_z, \theta, \phi, p, q, r\right) \tag{1.10}$$

$$\Delta\beta = f_{\beta}\left(\hat{\beta}, q_c, \dot{q}_c, a_x, a_y, a_z, \theta, \phi, \psi, p, q, r\right) \tag{1.11}$$

where $\hat{\alpha}$ (or $\hat{\beta}$) is the initial estimation of the AOA (or AOS), q_c and $\dot{q}_c$ are the dynamic pressure and its time derivative, a_i is the coordinate acceleration measured along i-th Body Axis, p, q, r are the angular speed respectively around the X_{Body}, Y_{Body} and Z_{Body} axes, ϕ is the roll angle, θ is the pitch angle, ψ is the yaw angle.

Ideally, an infinite TS would be needed to train the NN and ensure that there are no peaks in the estimation error. However, an infinite TS is obviously not available in a realistic application. The selection of a suitable set of input-target pair is one of the most difficult part of the NN design, together with the definition of which signals apply for the input pattern. Smart-ADAHRS learns from data logging coming from real flight tests, when the number of possible maneuvers is limited by fuel consumption, pilot experience, available time and flight test purpose. In fact, the analysis of the TS is one of the most described aspect of this work.

Table 1.11 collocates the Smart-ADAHRS algorithm in the classification given in Sec. 1.6.

Table 1.11: Smart-ADAHRS classification

Description	*Parameter*
Estimated quantities	α, β
Inertial measurements in input	Yes
Filtering process	No
ML	Yes (MLP)
Data origin	Simulation and flight data
AC configuration	Clean
FCS in input	Generally no. Added in the MIDAS project

1.8.1 Fault injection and real system noise simulation

The Smart-ADAHRS algorithm has already been tested under several conditions. An accurate flight simulation is an important tool in aeronautical engineering for several applications [111]. In this case, the simulations have been exploited in order to define the performance of the VS also in case of degraded conditions. This has been analyzed in [107]. At the time of writing the article, the simulator of the G70 that will be shown in Chap. 5 was not completed. Moreover, the analysis has been conducted in order to preliminary define the behaviour of the output signal in case of fault injection. To obtain realistic results, each sensor in input to the Smart-ADAHRS has been modeled in order to reproduce its dynamic, noise and fault behaviour according to the standards [112]. The resulting blocks have been collected in a library that can be used with any Simulink AC model. In Figure 1.6 and Figure 1.7 the general schematics of the inertial and pressure sensors and GNSS receiver are shown. In Table 1.12 the list of parameters that can be configured for inertial and pressure sensors is reported. In Table 1.13 it is reported the list of faults that can be injected in the simulation.

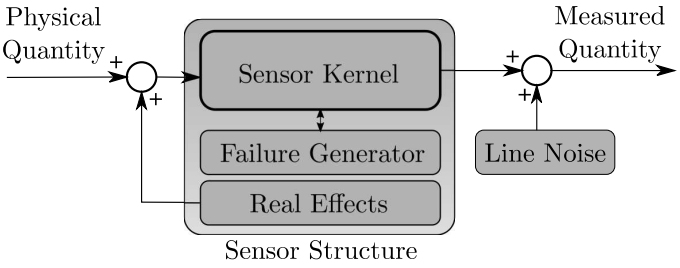


Figure 1.6: General schematic of the sensor subsystem.

Results showed in [107] provide some hints on which are the most influencing faults that can happen on the preceding sensors. The GNSS faults seems to be well-faced if applied for a limited time. However, the estimation error become unacceptable in case of null output of the inertial sensors. Moreover, the effect of the

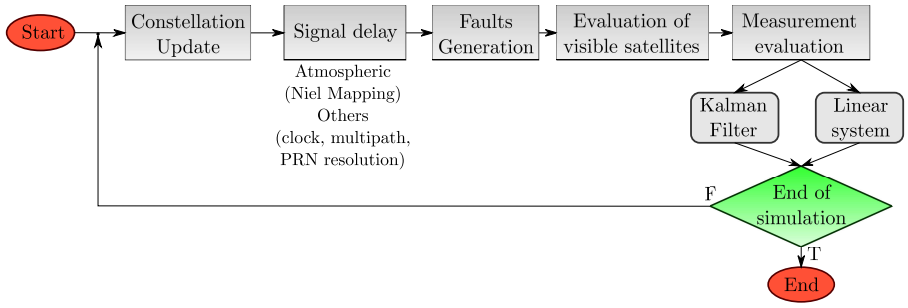


Figure 1.7: General schematic of the GNSS receiver subsystem.

Table 1.12: List of the model parameters (inertial and pressure sensor). UM stands for the unit of measurement of the quantity of interest

Full scale [UM]	Dynamic model equivalent mass [-]
Threshold [UM]	Dynamic model equivalent damping coefficient [-]
Fixed bias [UM]	Dynamic model equivalent third degree stiffness [-]
Bias stability from turn-on to turn-on [UM]	Dynamic model equivalent fifth degree stiffness [-]
Bias temperature stability [UM/K]	Natural frequency (phase = -90 deg) [rad/s]
White noise input process power [UM^2]	Damping factor [-]
AR model parameters	Misalignment matrix [-]
Scale factor temperature sensibility [%/K]	Floating point precision of ADC [UM]
Operating temperature range [K]	Bias compensation (added to the input)
Bias reference temperature [K]	Misalignment and sensitivity compensation matrix (left multiplied by (input + bias))
Reference temperature [K]	

temperature on the inertial sensors can also be detrimental only if the temperature sensitivity is higher than the values typically reported on datasheets of COTS sensors. An example of results is shown in Figure 1.8.

Table 1.13: List of faults for the GNSS sensor

GNSS Receiver faults	*Description*
KF fault	KF stops to update
Phase-Lock Loop (PLL) fault	Zero satellites in view
Satellite fault	Satellites in view randomly loss
Mask angle change	Change the minimum elevation angle (it can be defined depending also on the azimuth angle)
ADAHRS faults	*Description*
Null output	Output signal set to 0
Temperature fault	Temperature increasing with given slope

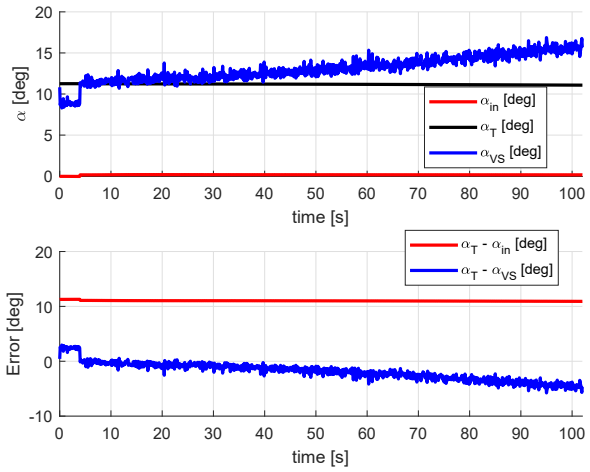


Figure 1.8: Smart-ADAHRS response in case of a temperature increase on the accelerometers of 2 K/s with an high temperature sensitivity of 0.02 K^{-1} (subscripts VS, T and in stand respectively for *Virtual Sensor*, *True angle* and *initial estimation*)

1.8.2 Study of the most influential variables using genetic programming

In order to understand which are the most influencing variables in the estimation of AOA/AOS, the genetic programming method has been implemented. Actually, a complete study of the math behind the problem defined in Sec. 1.4 would be necessary but it is still unavailable in literature.

The genetic programming consists in a heuristic search of the nonlinear function that fits the data given in input. Once the function is obtained, it is possible to understand which are the unused variable and the most effective one. The applied

dataset has been normalized with the same procedure used for training the NN. Two tests have been executed, the first one using the entire dataset and the second one using only the TS that allows to obtain the best performance of the NN. The description of this TS can be seen at the end of this dissertation in Chap. 6. The results of the genetic programming can be seen in Table 1.14 and Figure 1.9.

Table 1.14: Results of the genetic programming algorithm

	Using the entire dataset	*Using only the TS*
Function	$\overline{\Delta\alpha} = 0.23757\overline{\theta} + 0.027312\overline{\phi} + 0.08094\overline{\dot{q}_c}\overline{a}_z + 0.13559 * 0.27425^{\overline{q}_c} - 0.11023 + -0.22356\overline{a}_z - 1.3746\overline{\hat{\alpha}} + -0.18109\overline{\phi}^2$	$\overline{\Delta\alpha} = 0.12544 * 0.25117^{\overline{q}_c} + 0.10618\overline{\theta} * 0.25117^{\overline{q}_c} - 0.07588 + -\sin(\sin(\sin(0.14322\overline{\hat{\alpha}}0.25117^{\overline{q}_c}))) + -0.317\overline{a}_z - 1.088\overline{\hat{\alpha}} - 0.22746\overline{\phi}^2$
R^2 coefficient	0.99655505	0.99636089
Unused variables	$\overline{a}_y, \overline{q}, \overline{r}$	$\dot{q}_c, \overline{a}_y, \overline{p}, \overline{r}$

The nonlinear functions obtained with this method are very similar and they show a strong dependence on the $\overline{\phi}$ angle and on the impact pressure $\overline{q}_c$. Moreover, $\overline{a}_y$ and $\overline{r}$ are unused in both cases and $\dot{\overline{q}}_c$ is used only in the first case multiplied by $\overline{a}_z$ and with a low correlation coefficient. In both cases, the R^2 coefficient is higher than 0.996, as confirmed by Figure 1.9. The computational time required by the two tests has been 5.5 hours with a total amount of 1.1×10^{12} function evaluations when the entire database was applied, and 3 hours and 11 minutes for the second test, with 5.7×10^{11} function evaluations. For this reason, even if compelling results are obtained in terms of R^2 coefficient, this method is considered computationally more expensive then the NN training.

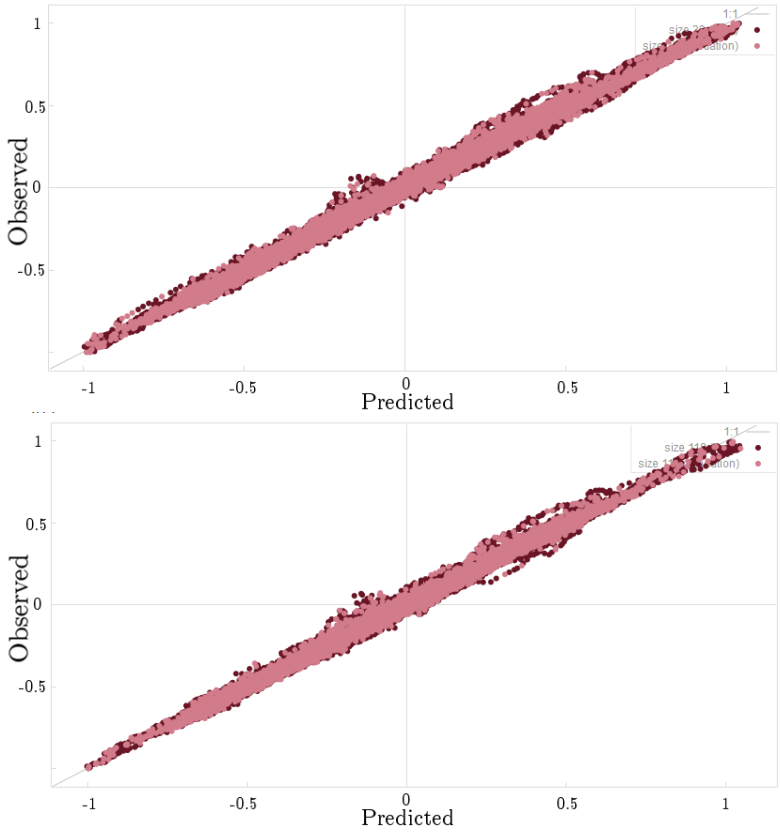


Figure 1.9: Observed vs Predicted plot obtained with Genetic Programming

Chapter 2

Theoretical aspects of neural networks

In line with the philosophy of providing a follow-up to the previous research on Smart-ADAHRS, some mathematical and, in particular, statistical theory about NN is given in this chapter. Taking advantage of the aspects shown in this chapter, it is possible to improve the performance of the estimator, or to understand the origin of localized high uncertainty that might be difficult to reduce. Sec. 2.1 defines the structure of the NN considered in this dissertation. Sec. 2.2 shows the point of view of multivariate nonlinear regression and UAT is introduced as justification for the approach in Sec. 2.3. Because of the fact that the error results in the sensor uncertainty, Sec. 2.4 reports some of the known limits of the statistical approximation using NN and it introduces the CV method. Part of this chapter has been already submitted in [113].

2.1 Neural Network structure definition

In this section, a brief definition of the quantities that will be appied in this thesis is conducted. The convention is similar to the one applied in [114, 115] but some symbols have been re-defined to adapt to the formulation.

Henceforth, it is assumed the following convention. Given $f : \mathbb{R} \to \mathbb{R}$ and a matrix $\boldsymbol{A} \in \mathbb{R}^{nm}$ it is possible to write:

$$f(\boldsymbol{A}) = \boldsymbol{B} \tag{2.1}$$

such that

$$\boldsymbol{B} \in \mathbb{R}^{nm} \quad \text{with} \quad \boldsymbol{B} = \begin{pmatrix} f(a_{11}) & f(a_{12}) & \cdots & f(a_{1m}) \\ f(a_{21}) & f(a_{22}) & \cdots & f(a_{2m}) \\ \vdots & \vdots & \ddots & \vdots \\ f(a_{n1}) & f(a_{n2}) & \cdots & f(a_{nm}) \end{pmatrix} \tag{2.2}$$

ANN is a broad term referring to a lot of different architectures with a common topology. A neural network is a sorted triple (N, V, w) with two sets N, V and a function w, where N is the set of neurons (also called nodes or hidden units) and V a set $\{(i,j) \mid i,j \in \mathbb{N}\}$ whose elements are called connections between neuron i and neuron j. The function $w : V \to \mathbb{R}$ defines the weights, where $w((i,j))$, the weight of the connection between neuron i and neuron j, is shortened to $w_{i,j}$. As convention, it is assumed that $j = 0$ ($w_{i,0}$) can be used to point the bias of the i-th neuron. Depending on the point of view, w_{ij} can be 0 or even undefined for the connections that do not exist in the network [116]. Briefly, an ANN consists of a set of neurons with weighted interconnections among them. This thesis is focused on MLP, a kind of ANN described by a directed graph from a set of inputs to a set of outputs, where neurons are organized in ordered *layers*. The first layer is the input layer, the last one is the output layer and the other ones are referred to the hidden layers. Although this definition can slightly vary in literature, in this thesis it is assumed that the connection can exist only between consecutive layers in a *fully-connected* and *feed-forward* architecture.

Each neuron computes a value called *activation* from the biased linear combination of the output values of the preceding neurons. The weights constitutes the coefficients of the linear combination and the bias itself. For convenience, the result of the biased linear combination of the input of the i-th neuron of the k-th layer is called $s_i^{(k)}$. Figure 2.1 shows a general schematic and notation of the quantities involved in a layer.

The activation value $a \in \mathbb{R}$ is the image of the activation function $\phi : \mathbb{R} \to \mathbb{R}$, which can be linear or nonlinear. This composition brings to the recursive formula in Eq. 2.3.

$$a_i^{(k)} = \phi\left(\sum_j \left(w_{i,j}^{(k)} a_j^{(k-1)} + v_i^{(k)}\right)\right) = \phi\left(s_i^{(k)}\right) = \phi_i^{(k)} \tag{2.3}$$

where the superscript (k) is the index of the considered layer, $w_{i,j}^{(k)}$ is the weight of the connection from neuron j to neuron i at the k-th layer. $k \in 0, 1, \cdots, l, l+1$ with 0 representing the input layer, 1 representing the first hidden layer, l the number of hidden layer, $l+1$ the output layer. n_k is the number of neurons contained in the k-th layer. Because the previous convention for the bias value of a neuron can bring to some indexing problem, it has been preferred to collect the biases of the k-th layer in a new vector $\boldsymbol{v}^{(k)}$. Subscript f is also applied to point out the output layer for sake of clarity. The last member of the equation is simply a short form to

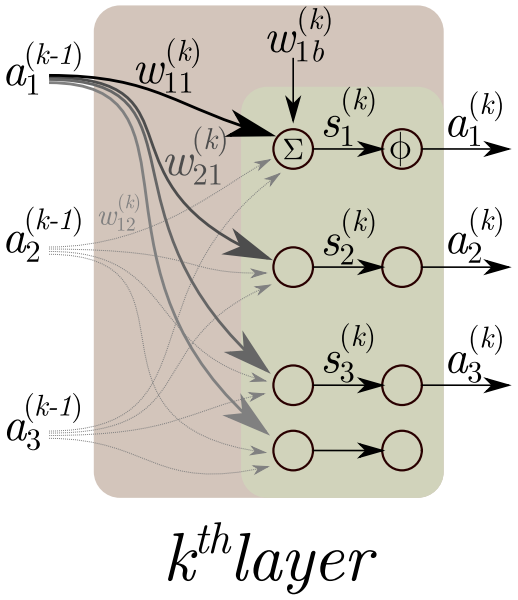


Figure 2.1: General schematic and notation of a layer of a MLP. The red area collects the weights of the layer whereas the green one collects the operations.

relieve the formulation. Moreover, in this thesis is assumed that the same function ϕ is applied to every neurons belonging to the hidden layers and it is considered *sigmoidal*. Two implementations are considered, the *logistic function* as in Eq. 2.4 and the *hyperbolic tangent* as in Eq. 2.5.

$$\phi(s) = \frac{1}{1+e^{-s}} \tag{2.4}$$

$$\phi(s) = \tanh s \tag{2.5}$$

There is the possibility for the input and output neurons to have the *identity* function as activation function, for instance $y = s^{(f)}$, in addition to the sigmoidal function.

This description allows to write the equation of a single hidden layer MLP with linear input and output node. The matrix form can be seen in Eq. 2.6, where the size of the matrices have been highlighted.

$$\begin{aligned} \tilde{y} &= \underset{1\times n_1}{\boldsymbol{w}^{(f),T}} \phi\left(\underset{n_1\times 1}{\boldsymbol{s}^{(1)}}\right) + v^{(f)} = \boldsymbol{w}^{(f),T}\boldsymbol{\phi}^{(1)} + v^{(f)} = \\ &= \boldsymbol{w}^{(f),T}\phi\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{v}^{(1)}\right) + v^{(f)} \\ &\text{where} \quad \boldsymbol{W}^{(1)} \in \mathbb{R}^{n_1\times n_0}, \boldsymbol{v}^{(1)} \in \mathbb{R}^{n_1}, v^{(f)} \in \mathbb{R} \end{aligned} \tag{2.6}$$

Henceforth, it is assumed that the studied NNs have only one output node and that the input and output nodes are linear.

2.2 Multivariate nonlinear regression framework

Similarly to [114], it is possible to define *regression* as follows:

Definition *Regression is the problem of infer the value of one or more continous target variables t_k, given the value of a D-dimensional vector $\mathbf{x}$ of input variables. In a probabilistic framework, regression consists in modelling the distribution $p(t_k|\mathbf{x})$ minimizing the value of a chosen loss function.*

Henceforth, the subset of the set $\mathcal{T}$ of tuples $\mathcal{T}_i = (\mathbf{x}_i, \mathbf{t}_i)$ applied to model the $p(t_k|\mathbf{x})$ distributions is called TS whereas the subset applied for testing is called Test Set, with no abbreviation to avoid misunderstandings. The set $\mathbf{X}$ includes the sample observed from the *input space*, $\mathbf{T}$ the sample from the *target space*. The TS and the Test Set are drawn from $\mathbf{X}$ and $\mathbf{T}$. Both the input space and target space can assume the structure of vector spaces and it is common to refer to dimensions of $\mathbf{X}$ and $\mathbf{T}$ as they would be vector spaces, even if they are not. The direct sum of the input space and the target space is generally called the *hypercube*. Please note that $\mathbf{X}$ and $\mathbf{T}$ are actually *multisets* because it is assumed here that the multiplicity of identical elements affects the training. However, the multiset notation is dropped here, also in view of the typical ML notation.

As a recall, the MLP asymptotic solution for training with minimization of the Mean Squared Error (MSE) is the following:

$$y_k(\mathbf{x}|\mathbf{w}^*) = \int t_k p(t_k|\mathbf{x})\, dt_k \tag{2.7}$$

where y_k is the k-th output of the MLP, $k \in 1 \dots K$ and K is the dimension of the target space. $\mathbf{w}$ stands for the weight vector of the MLP and $p(.)$ represents the PDF [114].

Eq. 2.7 can be interpreted as follows: given the set of weights $\mathbf{w}^*$ that minimize the MSE with respect to the TS and given the input vector $\mathbf{x}$, the output of an MLP is the expected value of the distribution of the given target. This means that MLP actually gives the regression of t_k conditioned on $\mathbf{x}$ and that MLP is a valid application for the proposed method. Figure 2.2 shows an example based on linear fitting.

2.3 The UAT

This section shows the results from [117] by Cybenko. This theorem is reported with the name of UAT. Several authors worked on the universality of approximation of the NNs, for instance see also [118, 119, 120].

First of all, some definitions are needed. Let I_n be the n-dimensional unit cube, the so-called hypercube. $C(I_n)$ is the space of continuous functions on I_n while

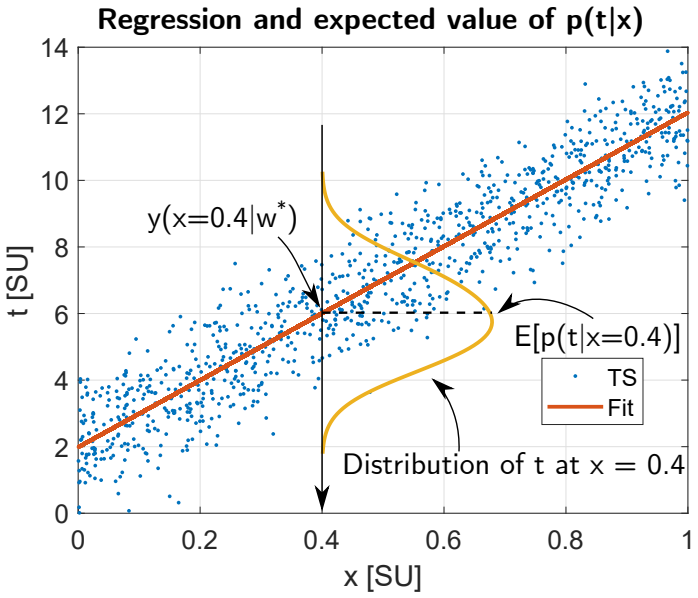


Figure 2.2: Example showing the relationship between regression and conditional PDF of the target on the input value.

$M(I_n)$ is the space of finite, signed regular Borel measures on I_n. the definition of *discriminatory function* is needed.

Definition *A function* σ *is* discriminatory *if for a measure* $\mu \in M(I_n)$

$$\int_{I_n} \sigma(\boldsymbol{W}\boldsymbol{x} + v)\, d\mu(\mathbf{x}) = 0 \tag{2.8}$$

for all $\boldsymbol{W} \in \mathbb{R}^{1\times n_0}$ *and* $v \in \mathbb{R}$ *implies that* $\mu = 0$.

The definition of *sigmoidal* in [117] is slightly more general than the one given in Eq. 2.4 and 2.5. However, a lemma is proven [117] stating that any continuous sigmoidal function is discriminatory.

Theorem. *Let* σ *be any continuous discriminatory function. Then finite sums of the form*

$$\tilde{y}(\mathbf{x}) = \sum_{j=1}^{n_f} w_j^{(f)} \sigma\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{v}^{(1)}\right) \tag{2.9}$$

are dense in $C(I_n)$. *In other words, given any* $y(\mathbf{x}) \in C(I_n)$ *and* $\epsilon > 0$, *there is a sum,* $\tilde{y}(\mathbf{x})$, *of the above form, for which*

$$|\tilde{y}(\mathbf{x}) - y(\mathbf{x})| < \epsilon \quad \textit{for all} \quad \mathbf{x} \in I_n. \tag{2.10}$$

This theorem proves that a NN can actually represent any function on the n-dimensional hypercube. Refer to [117] for additional details on how UAT can be applied to a single hidden layer NN. However, as stated by the author itself, they focused on existence. The number of neurons is unknown and the statistical risk must be anayzed separately as in Sec. 2.4.

2.4 Approximation and Estimation error bounds

This section gives the statistical definitions of *approximation error* and *estimation error* as in [121]. They will be used as introduction to the CV method and to provide an introduction to some of the most important results from Barron. Given a set of random variables (the observations) $\xi_i \in \Xi$ with common distribution P, the aim of regression is to estimate the regression function y from Eq. 2.7, where subscript k has been dropped for convenience.

Please note the difference between the target values t, the regression function y and the approximation of the regression function $\tilde{y}$. The target t consists of the measured values, its uncertainty depends on the measurement equipment and on the followed measurement process. y is actually the ideal solution that can be obtained once a criterion is adopted, for instance the least-squares analysis. $\tilde{y} \in \mathbb{Y}$ is the proposed approximation. It is possible to define the *excess loss* as in Eq. 2.11.

$$l\left(y, \tilde{y}\right) = \mathcal{L}_P\left(\tilde{y}\right) - \mathcal{L}_P\left(y\right) \geq 0 \tag{2.11}$$

where $\mathcal{L}_P : \mathbb{Y} \to \mathbb{R}$ is called the *loss function* and it is defined as follows:

$$\mathcal{L}_P\left(\tilde{y}\right) = \mathbb{E}_{\xi \sim P}\left[\gamma\left(\tilde{y}; \xi\right)\right] \tag{2.12}$$

The loss function measures the quality of the approximation of y using $\tilde{y}$. It depends on the choice of the *contrast function* $\gamma : \mathbb{Y} \times \Xi \to [0, \infty)$. A common choice in regression is the *least-squares contrast* $\gamma\left(\tilde{y}; \xi\right) = \left(\tilde{y}\left(\mathbf{x}\right) - t\right)^2$.

It is interesting to introduce the *model selection problem*. Let $\left(Y_m\right)_{m \in \mathcal{M}_n}$ be a family of candidate models from which the final one will be chosen. From an operative point of view, this family can be composed by the various NN obtained during the trial-and-error design of the VS. The minimum contrast estimator can be written as $\hat{y}_m\left(D_n\right)$. However, the following considerations can be made. When Y_m is small, there is a lower bound on the excess loss that can be achieved.

$$l\left(y, Y_m\right) = \inf_{\tilde{y} \in Y_m}\left\{l\left(y, \tilde{y}\right)\right\} \leq l\left(y, \hat{y}_m\left(D_n\right)\right) \tag{2.13}$$

$l\left(y, Y_m\right)$ is called *approximation error* and it is the lower bound for $l\left(y, \hat{y}_m\left(D_n\right)\right)$. The approximation error depends on the size of Y_m, so that the number of candidate models. This situation results in underfitting the dataset. On the other hand, if

Y_m is large, $\hat{y}_m(D_n)$ will likely overfit the data. In fact, it can be proven that the *risk of the estimator* follows Eq. 2.14.

$$\mathbb{E}\left[l\left(y, \hat{y}_m\left(D_n\right)\right)\right] \approx l\left(y, Y_m\right) + \alpha_n D_m = \text{approximation error} + \text{estimation error} \tag{2.14}$$

where $\alpha_n > 0$ does not depend on m and $\alpha_n = \frac{\sigma^2}{n}$ when the least-squares contrast function is used and $\text{var}\left(t|\mathbf{x}\right) = \sigma^2$. On the contrary, $D_m = \dim Y_m$. Hence, too many possibilities introduce an higher expectation on the excess loss. $\alpha_n D_m$ is called *estimation error* and the trade-off resulting from this analysis is the famous *bias-variance dilemma*.

A class of methods that can be used to reduce the risk is based on CV. CV procedures start from splitting the data and then averaging or voting the estimator of the risk corresponding to different data splits. This definition is very general and it is necessary to specify it for the aim of this dissertation. In this work, the first step consists in the partition of the dataset. The second step is selecting how many TS are desired from the obtained partitions and then the generation of the various TSs is conducted. Once the various TSs are obtained, the training phase of the NN is conducted iteratively on the various TSs, measuring the error metric on the part of the dataset excluded by the TS. At the end of the process, a set of NN is obtained, each one associated with a test error and the NN with the minimum test error is chosen.

According to [123] the Mean Integral Squared Error (MISE) between a neural network and the target function y is bounded by Eq. 2.15

$$\mathcal{O}\left(\frac{C_y^2}{n}\right) + \mathcal{O}\left(\frac{nd}{N}\log N\right) \tag{2.15}$$

where n is the number of neurons, d is the input dimension of the function, N is the number of training observations. C_y is the first absolute moment of the Fourier magnitude distribution of y defined in Eq. 2.16.

$$C_y = \int |\,\omega\,|\,|\,\mathcal{F}\left(y\right)|\,d\omega \tag{2.16}$$

Unfortunately, the term C_y is not known a priori in this case study. In fact, at the time of writing this dissertation, the expression of the functions α and β given with respect to the aircraft state vector is still unknown, except for rare cases. However, this section rigorously define the situation faced in this dissertation. The measured values cannot be interpolated due to measurement error, but an unknown function exist behind them. This function can be represented by a NN and this can be proven by the UAT. It is hence possible to find some candidate models with a double search, the first on the network architecture and the second one from data. The final excess loss has a lower bound. The last question is if the function

actually exist, however this is still an open question. Analysis of NN that differs from this framework can exist. However, at the time of writing this dissertation, those analyses should be firstly supported by a mathematical foundation that could encourage the study.

2.5 Derivatives of a neural approximation

In this section, the derivatives of a neural approximation are obtained. They are useful because numerical differentiation would be subjected to the unknown regularity of the function in the neighborhood of calculation. On the other hand, the formulations derived here allow to evaluate the exact gradient and the Hessian matrix of an MLP in a given point, but the result can be strongly local.

2.5.1 Gradient of an MLP with 1 hidden layer

In this subsection, the partial derivative of the function represented by a single hidden layer MLP with respect to one of its variable is obtained.

Deriving Eq. 2.6 with respect to the input variables

$$\frac{\partial \tilde{y}}{\partial x_i} = \sum_{k=1}^{k=n_1} w_k^{(f)} \frac{\mathrm{d}\phi_k^{(1)}}{\mathrm{d}s_k^{(1)}} \frac{\partial s_k^{(1)}}{\partial x_i} \tag{2.17}$$

where

$$\frac{\partial s_k^{(1)}}{\partial x_i} = w_{k,1}^{(1)} \tag{2.18}$$

and considering sigmoidal activation function it is possible to write

$$\begin{aligned} \frac{\mathrm{d}\phi_k^{(1)}}{\mathrm{d}s_k^{(1)}} &= \phi\left(s_k^{(1)}\right)\left(1-\phi\left(s_k^{(1)}\right)\right) \quad \text{if } \phi \text{ follows Eq. 2.4} \\ \frac{\mathrm{d}\phi_k^{(1)}}{\mathrm{d}s_k^{(1)}} &= 1-\phi^2\left(s_k^{(1)}\right) \quad \text{if } \phi \text{ follows Eq. 2.5} \end{aligned} \tag{2.19}$$

Hence, Eq. 2.17 can be rewritten considering, for instance, the logistic function:

$$\begin{aligned} \frac{\partial \tilde{y}}{\partial x_i} &= w_1^{(f)}\phi\left(s_1^{(1)}\right)\left(1-\phi\left(s_1^{(1)}\right)\right)w_{1,i}^{(1)} + \\ &+ w_2^{(f)}\phi\left(s_2^{(1)}\right)\left(1-\phi\left(s_2^{(1)}\right)\right)w_{2,i}^{(1)} + \cdots = \\ &= \boldsymbol{w}^{(f),T} \odot \phi\left(\boldsymbol{s}^{(1),T}\right) \odot \left(1-\phi\left(\boldsymbol{s}^{(1),T}\right)\right)\boldsymbol{W}_{*,i}^{(1)} \end{aligned} \tag{2.20}$$

where $\odot$ stands for the Hadamard or entrywise product. Although the second and the third factors of Eq. 2.20 implicitly depends on the variable of differentiation, they do not change if the derivative is taken with respect to a different variable and hence they can be collected in a row vector named $\boldsymbol{b}^{(1),T}$. This vector represents the derivatives of the activation functions at a given layer. It is hence possible to write the gradient of the single hidden layer MLP as in Eq. 2.21.

$$\nabla^T \tilde{y} = \left(\boldsymbol{w}^{(f)} \odot \boldsymbol{b}^{(1)}\right)^T \boldsymbol{W}^{(1)} \text{ or } \nabla y = \boldsymbol{W}^{(1),T} \left(\boldsymbol{w}^{(f)} \odot \boldsymbol{b}^{(1)}\right) \tag{2.21}$$

where

$$\begin{aligned} \boldsymbol{b}^{(1)} &= \phi\left(\boldsymbol{s}^{(1)}\right) \odot \left(1 - \phi\left(\boldsymbol{s}^{(1)}\right)\right) \quad \text{if } \phi \text{ follows Eq. 2.4} \\ \boldsymbol{b}^{(1)} &= 1 - \phi^2\left(\boldsymbol{s}^{(1)}\right) \quad \text{if } \phi \text{ follows Eq. 2.5} \end{aligned} \tag{2.22}$$

2.5.2 Gradient of an MLP with 2 hidden layers

In case of 2 hidden layers, Eq. 2.6 modifies in Eq. 2.23.

$$\begin{aligned} \tilde{y} = \underset{1\times n_l}{\boldsymbol{w}^{(f),T}} \phi\left(\underset{n_l\times 1}{\boldsymbol{s}^{(2)}}\right) + v^{(f)} = \boldsymbol{w}^{(f),T} \phi\left(\boldsymbol{W}^{(2)} \phi\left(\boldsymbol{s}^{(1)}\right) + \boldsymbol{v}^{(1)}\right) + v^{(f)} \\ \text{where } \boldsymbol{W}^{(2)} \in \mathbb{R}^{n_2 \times n_1}, \boldsymbol{v}^{(2)} \in \mathbb{R}^{n_2} \end{aligned} \tag{2.23}$$

whereas Eq. 2.17 can be generalized as follows:

$$\frac{\partial \tilde{y}}{\partial x_i} = \sum_{k=1}^{k=n_l} w_k^{(f)} \frac{\mathrm{d}\phi_k^{(l)}}{\mathrm{d}s_k^{(l)}} \frac{\partial s_k^{(l)}}{\partial x_i} \tag{2.24}$$

It is possible to write

$$\frac{\partial s_1^{(2)}}{\partial x_1} = w_{11}^{(2)} \frac{\mathrm{d}\phi_1^{(1)}}{\mathrm{d}s_1^{(1)}} \frac{\partial s_1^{(1)}}{\partial x_1} + w_{12}^{(2)} \frac{\mathrm{d}\phi_2^{(1)}}{\mathrm{d}s_2^{(1)}} \frac{\partial s_2^{(1)}}{\partial x_1} + \cdots + w_{1,n_2}^{(2)} \frac{\mathrm{d}\phi_{n_2}^{(1)}}{\mathrm{d}s_{n_2}^{(1)}} \frac{\partial s_{n_2}^{(1)}}{\partial x_1} \tag{2.25}$$

It is possible to generalize Eq. 2.19. For instance, for the logistic sigmoidal function Eq. 2.19 becomes Eq. 2.26

$$\frac{\mathrm{d}\phi_k^{(j)}}{\mathrm{d}s_k^{(j)}} = \phi\left(s_k^{(j)}\right)\left(1 - \phi\left(s_k^{(j)}\right)\right) \tag{2.26}$$

and remembering Eq. 2.18 it is possible to write

$$\begin{cases} \dfrac{\partial s_1^{(2)}}{\partial x_1} = \boldsymbol{W}_{1,*}^{(2)} \odot \phi\left(\boldsymbol{s}^{(1),T}\right) \odot \left(1 - \phi\left(\boldsymbol{s}^{(1),T}\right)\right) \boldsymbol{W}_{*,1}^{(1)} \\ \dfrac{\partial s_2^{(2)}}{\partial x_1} = \boldsymbol{W}_{2,*}^{(2)} \odot \phi\left(\boldsymbol{s}^{(1),T}\right) \odot \left(1 - \phi\left(\boldsymbol{s}^{(1),T}\right)\right) \boldsymbol{W}_{*,1}^{(1)} \\ \quad \dots \end{cases} \tag{2.27}$$

Eq. 2.27 can be arranged in matrix form as

$$\frac{\partial \boldsymbol{s}^{(2)}}{\partial x_i} = \boldsymbol{W}^{(2)} \text{diag}\left(\boldsymbol{b}^{(1)}\right) \boldsymbol{W}_{*,i}^{(1)} \tag{2.28}$$

It is hence possible to write the Jacobian matrix of the input of the activation function with respect to the input of the MLP as in Eq. 2.29

$$\mathbf{J}\boldsymbol{s}^{(2)} = \boldsymbol{W}^{(2)} \text{diag}\left(\boldsymbol{b}^{(1)}\right) \boldsymbol{W}^{(1)} \tag{2.29}$$

Analogously to what has been done for a single hidden layer MLP, It is finally possible to rearrange Eq. 2.24

$$\begin{aligned} \frac{\partial \tilde{y}}{\partial x_i} &= w_1^{(f)} \phi\left(s_1^{(2)}\right)\left(1 - \phi\left(s_1^{(2)}\right)\right) \frac{\partial s_1^{(1)}}{\partial x_i} + \\ &+ w_2^{(f)} \phi\left(s_2^{(2)}\right)\left(1 - \phi\left(s_2^{(2)}\right)\right) \frac{\partial s_2^{(1)}}{\partial x_i} + \dots = \\ &= \boldsymbol{w}^{(f),T} \odot \phi\left(\boldsymbol{s}^{(2),T}\right) \odot \left(1 - \phi\left(\boldsymbol{s}^{(2),T}\right)\right) \frac{\partial \boldsymbol{s}^{(1)}}{\partial x_i} \end{aligned} \tag{2.30}$$

Collecting the derivatives of the activation function of any layer, as previously done for the first layer in 2.22, as in Eq. 2.31,

$$\begin{aligned} \boldsymbol{b}^{(i)} = \phi\left(\boldsymbol{s}^{(i)}\right) \odot \left(1 - \phi\left(\boldsymbol{s}^{(i)}\right)\right) \quad &\text{if } \phi \text{ follows Eq. 2.4} \\ \boldsymbol{b}^{(i)} = 1 - \phi^2\left(\boldsymbol{s}^{(i)}\right) \quad &\text{if } \phi \text{ follows Eq. 2.5} \end{aligned} \tag{2.31}$$

the gradient of the 2 hidden layers MLP can be hence written as in Eq. 2.32

$$\nabla^T \tilde{y} = \left(\boldsymbol{w}^{(f)} \odot \boldsymbol{b}^{(2)}\right)^T \mathbf{J}\boldsymbol{s}^{(2)} \text{ or } \nabla \tilde{y} = \mathbf{J}\boldsymbol{s}^{(2),T}\left(\boldsymbol{w}^{(f)} \odot \boldsymbol{b}^{(2)}\right) \tag{2.32}$$

2.5.3 Gradient of an MLP with n hidden layers

Previous analysis can be generalized to any number of hidden layers simply recognizing the recursive pattern in the equations. Here, the results are reported.

$$\nabla\tilde{y} = \mathbf{J}\boldsymbol{s}^{(l),T}\left(\boldsymbol{w}^{(f)} \odot \boldsymbol{b}^{(l)}\right) \tag{2.33}$$

with

$$\mathbf{J}\boldsymbol{s}^{(i)} = \boldsymbol{W}^{(i)}\text{diag}\left(\boldsymbol{b}^{(i-1)}\right)\mathbf{J}\boldsymbol{s}^{(i-1)} \text{ and } \mathbf{J}\boldsymbol{s}^{(1)} = \boldsymbol{W}^{(1)} \tag{2.34}$$

2.5.4 Hessian matrix of an MLP with n hidden layers

The Hessian matrix collects the second partial derivatives of a scalar field. Eq. 2.24 can be derived again to obtain the second partial derivative with respect to x_i and x_j.

$$\frac{\partial^2\tilde{y}}{\partial x_j \partial x_i} = \sum_{k=1}^{k=n_l} w_k^{(f)} \frac{\partial}{\partial x_j}\left(\frac{\mathrm{d}\phi_k^{(l)}}{\mathrm{d}s_k^{(l)}}\frac{\partial s_k^{(l)}}{\partial x_i}\right) \tag{2.35}$$

The chain rule can be applied noticing that the output of a neuron is the composition of the activation function with s.

$$\frac{\partial}{\partial x_j}\left(\frac{\mathrm{d}\phi_k^{(l)}}{\mathrm{d}s_k^{(l)}}\right) = \frac{\mathrm{d}^2\phi_k^{(l)}}{\mathrm{d}s_k^{(l),2}}\frac{\partial s_k^{(l)}}{\partial x_j} \tag{2.36}$$

$$\frac{\partial^2\tilde{y}}{\partial x_j \partial x_i} = \sum_{k=1}^{k=n_l} w_k^{(f)}\left(\frac{\mathrm{d}^2\phi_k^{(l)}}{\mathrm{d}s_k^{(l),2}}\frac{\partial s_k^{(l)}}{\partial x_j}\frac{\partial s_k^{(l)}}{\partial x_i} + \frac{\mathrm{d}\phi_k^{(l)}}{\mathrm{d}s_k^{(l)}}\frac{\partial^2 s_k^{(l)}}{\partial x_j \partial x_i}\right) \tag{2.37}$$

The second derivative of the activation function with respect to its variable is shown in Eq. 2.38.

$$\begin{aligned}
&\text{if } \phi \text{ follows Eq. 2.4:}\\
&\frac{\mathrm{d}^2\phi_k^{(j)}}{\mathrm{d}s_k^{(j),2}} = \phi\left(s_k^{(j)}\right)\left(1-\phi\left(s_k^{(j)}\right)\right)\left(1-2\phi\left(s_k^{(j)}\right)\right) = \frac{\mathrm{d}\phi_k^{(j)}}{\mathrm{d}s_k^{(j)}}\left(1-2\phi\left(s_k^{(j)}\right)\right),\\
&\text{if } \phi \text{ follows Eq. 2.5:}\\
&\frac{\mathrm{d}^2\phi_k^{(j)}}{\mathrm{d}s_k^{(j),2}} = -2\phi\left(s_k^{(j)}\right)\left(1-\phi^2\left(s_k^{(j)}\right)\right) = -2\phi\left(s_k^{(j)}\right)\frac{\mathrm{d}\phi_k^{(j)}}{\mathrm{d}s_k^{(j)}},
\end{aligned} \tag{2.38}$$

As it has been done for the Jacobian matrix, it is possible to recognize that the output of this kind of MLP is simply a biased linear combination of the preceding activation values. Because of the fact that this holds true also for the input value of any neuron, it is possible to directly write down the recursive Eq. 2.39.

$$\begin{cases} \dfrac{\partial^2 s_n^{(m)}}{\partial x_j \partial x_i} = \displaystyle\sum_{k=1}^{k=n_{m-1}} w_{nk}^{(m)} \left(\frac{\mathrm{d}^2 \phi_k^{(m-1)}}{\mathrm{d} s_k^{(m-1),2}} \frac{\partial s_k^{(m-1)}}{\partial x_j} \frac{\partial s_k^{(m-1)}}{\partial x_i} + \frac{\mathrm{d}\phi_k^{(m-1)}}{\mathrm{d} s_k^{(m-1)}} \frac{\partial^2 s_k^{(m-1)}}{\partial x_j \partial x_i} \right) \\ \dfrac{\partial^2 s_n^{(1)}}{\partial x_j \partial x_i} = 0 \quad \forall i, j \end{cases} \tag{2.39}$$

The last step is rearranging Eq. 2.39 with the last form of Eq. 2.38

$$\begin{cases} \dfrac{\partial^2 s_n^{(m)}}{\partial x_j \partial x_i} = \displaystyle\sum_{k=1}^{k=n_{m-1}} w_{nk}^{(m)} \frac{\mathrm{d}\phi_k^{(m-1)}}{\mathrm{d} s_k^{(m-1)}} \left(\left(1 - 2\phi_k^{(m-1)}\right) \frac{\partial s_k^{(m-1)}}{\partial x_j} \frac{\partial s_k^{(m-1)}}{\partial x_i} + \right. \\ \qquad \left. + \dfrac{\partial^2 s_k^{(m-1)}}{\partial x_j \partial x_i} \right) \\ \dfrac{\partial^2 s_n^{(1)}}{\partial x_j \partial x_i} = 0 \quad \forall i, j \end{cases} \tag{2.40}$$

Although a generalization of the Hessian matrix for vector-valued function exists, the matrix form is defined for scalar-valued functions. Because Eq. 2.40 shows the second-order partial derivative with respect to the MLP input variables of the $s_n^{(m)}$ function, which is defined for each neuron as the biased linear combination applied to the activation function, it is immediate to understand that an Hessian matrix will be defined for each neuron. This could bring to some problems if the computational power is limited. However, for a 13 inputs MLP with 200 neurons there are a total number of $13 \times 13 \times 200 = 33800$ values to be stored. Hence, this limitation does not apply for the typical application in this dissertation.

A compact form of Eq. 2.40 might be obtained using the Einstein notation. However, it is preferred here a matrix notation based on the expansion to block matrices of the various elements. In addition, it is possible to define a new vector $\boldsymbol{c}^{(i)}$ as done for $\boldsymbol{b}^{(i)}$ for simplify the notation, as shown in Eq. 2.41.

$$\begin{aligned} \boldsymbol{c}^{(i)} &= \left(1 - 2\phi\left(\boldsymbol{s}^{(i)}\right)\right) \quad \text{if } \phi \text{ follows Eq. 2.4} \\ \boldsymbol{c}^{(i)} &= -2\phi\left(\boldsymbol{s}^{(i)}\right) \quad \text{if } \phi \text{ follows Eq. 2.5} \end{aligned} \tag{2.41}$$

In particular, vectors $\boldsymbol{b}^{(i)}$, $\boldsymbol{c}^{(i)}$ and each n-th row of the weight matrix $\boldsymbol{W}_{n,*}^{(m)}$ can be expanded in a block diagonal matrix, where each block is again a diagonal matrix with repeated elements from the original vector. This can be seen from Eq. 2.42 to Eq. 2.44 in both expanded form and using the Kronecker product $\otimes$.

$$\boldsymbol{B_2}^{(m)} = \begin{pmatrix} b_1^{(m)}\boldsymbol{I}_{n_0} & & & \\ & b_2^{(m)}\boldsymbol{I}_{n_0} & & \\ & & \ddots & \\ & & & b_{n_m}^{(m)}\boldsymbol{I}_{n_0} \end{pmatrix} = \text{diag}\left(\boldsymbol{b}^{(m)}\right) \otimes \boldsymbol{I}_{n_0} \tag{2.42}$$

$$\boldsymbol{C_2}^{(m)} = \begin{pmatrix} c_1^{(m)}\boldsymbol{I}_{n_0} & & & \\ & c_2^{(m)}\boldsymbol{I}_{n_0} & & \\ & & \ddots & \\ & & & c_{n_m}^{(m)}\boldsymbol{I}_{n_0} \end{pmatrix} = \text{diag}\left(\boldsymbol{c}^{(m)}\right) \otimes \boldsymbol{I}_{n_0} \tag{2.43}$$

$$\boldsymbol{W_2}_n^{(m)} = \begin{pmatrix} w_{n1}^{(m)}\boldsymbol{I}_{n_0} & & & \\ & w_{n2}^{(m)}\boldsymbol{I}_{n_0} & & \\ & & \ddots & \\ & & & w_{n,n_{m-1}}^{(m)}\boldsymbol{I}_{n_0} \end{pmatrix} = \text{diag}\left(\boldsymbol{W}_{n,*}^{(m)}\right) \otimes \boldsymbol{I}_{n_0} \tag{2.44}$$

A similar operation must be conducted on the Jacobian matrix $\mathbf{J}\boldsymbol{s}_{n,*}^{(m-1)}$. In this case, to ease the computational form, it is useful to expand the Jacobian matrix in a block diagonal matrix where each block is a row of the original Jacobian, transposed to become a column. In this way, the product between the first order partial derivative in Eq. 2.40 becomes the following compact matrix product $\boldsymbol{J_2}^{(m)}\boldsymbol{J_2}^{(m),T} = \frac{\partial s_k^{(m-1)}}{\partial x_j}\frac{\partial s_k^{(m-1)}}{\partial x_i}$.

$$\boldsymbol{J_2}_n^{(m)} = \begin{pmatrix} \mathbf{J}\boldsymbol{s}_{1,*}^{(m),T} & & & \\ & \mathbf{J}\boldsymbol{s}_{2,*}^{(m),T} & & \\ & & \ddots & \\ & & & \mathbf{J}\boldsymbol{s}_{n_m,*}^{(m),T} \end{pmatrix} \tag{2.45}$$

As previously mentioned, the Hessian matrix of a vector-valued function is usually written as a tensor. Here, on the contrary, it is assumed that the Hessian matrices of the activation functions at a given layer can be collected in a block diagonal matrix, where each block refers to a neuron as follows:

$$\mathbf{H}\boldsymbol{s}^{(m)} = \begin{pmatrix} \mathbf{H}s_1^{(m)} & & \\ & \ddots & \\ & & \mathbf{H}s_{n_m}^{(m)} \end{pmatrix} \tag{2.46}$$

where the subscript n has been dropped and the $\boldsymbol{s}$ has been written in bold text, to distinguish with the classical Hessian of a single activation function of a neuron s_n.

It is hence possible to re-arrange the addends of the summation expressed in Eq. 2.40 as the blocks of a block diagonal matrix $\tilde{\boldsymbol{H}} s_n^{(m)}$. Eq. 2.48 and Eq. 2.49 are obtained where blocks have been generally expressed as $\boldsymbol{H}_i$ to avoid subscript and superscript repetition.

$$\tilde{\boldsymbol{H}} s_n^{(m)} = \boldsymbol{W_2}_n^{(m)} \boldsymbol{B_2}^{(m-1)} \left(\boldsymbol{C_2}^{(m-1)} \boldsymbol{J_2}^{(m-1)} \boldsymbol{J_2}^{(m-1),T} + \mathbf{H}\boldsymbol{s}^{(m-1)} \right) = \tag{2.47}$$

$$= \begin{pmatrix} \boldsymbol{H}_1 & & & \\ & \boldsymbol{H}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{H}_{n_{m-1}} \end{pmatrix} \tag{2.48}$$

The sum of the blocks $\boldsymbol{H}_i$ that constitutes $\tilde{\boldsymbol{H}} s_n^{(m)}$ finally gets the Hessian matrix of the input to the activation function of a neuron n at a given layer (m).

$$\mathbf{H} s_n^{(m)} = \sum_{k=1}^{n_{m-1}} \boldsymbol{H}_k \tag{2.49}$$

As a side note, the second partial derivatives of the function represented by the MLP are continuous and hence the order of differentiation does not matter as a consequence of the Schwarz's Theorem.

Finally, the Hessian matrix of the overall MLP $\mathbf{H}y$ can be written as follows:

$$\mathbf{H}\tilde{y} = \boldsymbol{W_2}^{(f)} \boldsymbol{B_2}^{(l)} \left(\boldsymbol{C_2}^{(l)} \boldsymbol{J_2}^{(l)} \boldsymbol{J_2}^{(l),T} + \mathbf{H}\boldsymbol{s}^{(l)} \right). \tag{2.50}$$

2.5.5 Generalization in case of data mapping layers

In some cases, it is useful to add some mapping layers to an MLP. In details, instead of conducting a pre-processing operation to map data into another set, it is possible to insert the map directly inside the MLP. In this thesis, data is scaled and shifted to fit the $[-1;1]^d$ hypercube with the common operation also called *data normalization*.

This kind of function can be written as in Eq. 2.51.

$$\begin{aligned} \overline{x} &= m(x) = \frac{\overline{x}_{max} - \overline{x}_{min}}{x_{max} - x_{min}} (x - x_{min}) + \overline{x}_{min} = a_x (x - x_{min}) + \overline{x}_{min} \\ \overline{t} &= n(t) = \frac{\overline{t}_{max} - \overline{t}_{min}}{t_{max} - t_{min}} (t - t_{min}) + \overline{t}_{min} = a_t (t - t_{min}) + \overline{t}_{min} \end{aligned} \tag{2.51}$$

If the MLP is modified as in Figure 2.3, so that the input and the output of the network are indicated with overlined symbols, previous relations are modified as follow.

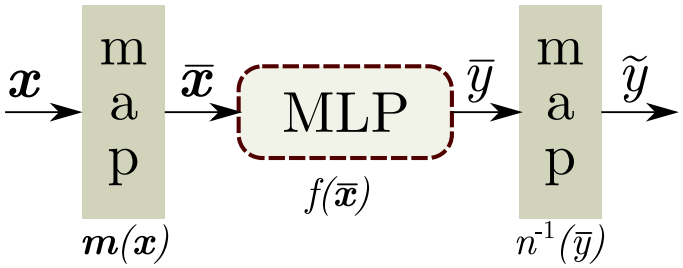


Figure 2.3: MLP with mapping functions for the input and output variables

$$\frac{\partial \tilde{y}}{\partial x_i} = \frac{\partial \tilde{y}}{\partial \overline{y}} \frac{\partial \overline{y}}{\partial \overline{x}_i} \frac{\partial \overline{x}_i}{\partial x_i} = \frac{a_{x,i}}{a_t} \frac{\partial \overline{y}}{\partial \overline{x}_i} \tag{2.52}$$

$$\begin{aligned} \frac{\partial^2 \tilde{y}}{\partial x_j \partial x_i} &= \frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{y}}{\partial x_i}\right) = \frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{y}}{\partial \overline{y}} \frac{\partial \overline{y}}{\partial \overline{x}_i} \frac{\partial \overline{x}_i}{\partial x_i}\right) = \\ &= \frac{\partial \tilde{y}}{\partial \overline{y}} \frac{\partial \overline{x}_i}{\partial x_i} \frac{\partial^2 \overline{y}}{\partial x_j \partial \overline{x}_i} = \frac{\partial \tilde{y}}{\partial \overline{y}} \frac{\partial \overline{x}_i}{\partial x_i} \frac{\partial^2 \overline{y}}{\partial \overline{x}_j \partial \overline{x}_i} \frac{\partial \overline{x}_j}{\partial x_j} = \\ &= \frac{\partial \tilde{y}}{\partial \overline{y}} \frac{\partial \overline{x}_i}{\partial x_i} \frac{\partial \overline{x}_j}{\partial x_j} \frac{\partial^2 \overline{y}}{\partial \overline{x}_j \partial \overline{x}_i} = \frac{a_{x,i} a_{x,j}}{a_t} \frac{\partial^2 \overline{y}}{\partial \overline{x}_j \partial \overline{x}_i} \end{aligned} \tag{2.53}$$

Hence, in case of mapping function as in Eq. 2.51, from Eq. 2.52 and Eq. 2.53 it is clear that the previous formulations for $\nabla \tilde{y}$ and $\mathbf{H}\tilde{y}$ can be simply generalized with some scalar coefficients coming from the mapping functions themselves.

2.5.6 Example with analytical function

In this section, the previous formulations are applied to a known function in order to check the validity of the derivation. An MLP has been trained to map a paraboloid function as in Eq. 2.54.

$$z = f(x_1, x_2) = x_1^2 + x_2^2; \ \nabla f = (2x_1, 2x_2); \ \mathbf{H}f = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \tag{2.54}$$

The result of the training of a 10-hidden layers MLP on a training set constituted by 50×10^3 points distributed as $\mathcal{N}(0, 8)$ can be seen in Figure 2.4.

Figure 2.5 and Figure 2.6 shows the results.

The deviation from the numerical differentiation and from the theoretical value are very small. The source of the difference between the dotted lines and the black lines is on the accuracy of the neural approximation, whereas the deviation from the red line and the black line comes from the accuracy of the numerical differentiation.

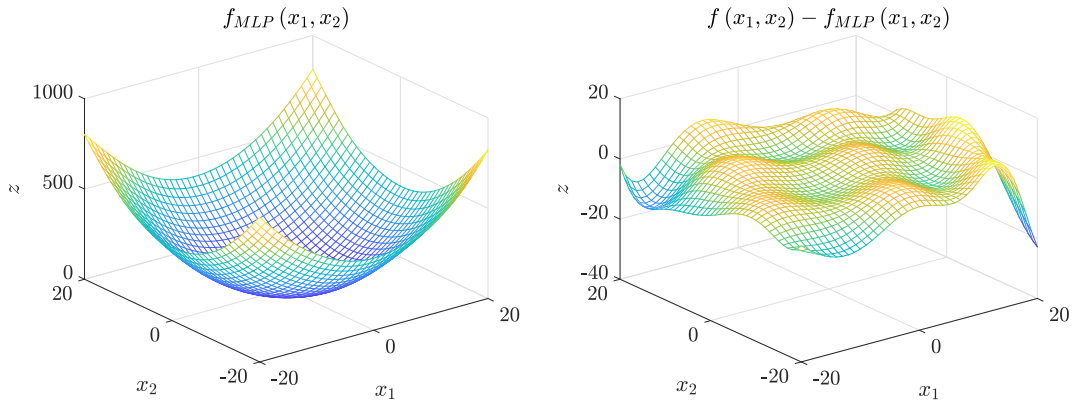


Figure 2.4: Plot of the neural approximation $f_{MLP}(\boldsymbol{x})$ and of the deviation from $f(\boldsymbol{x})$

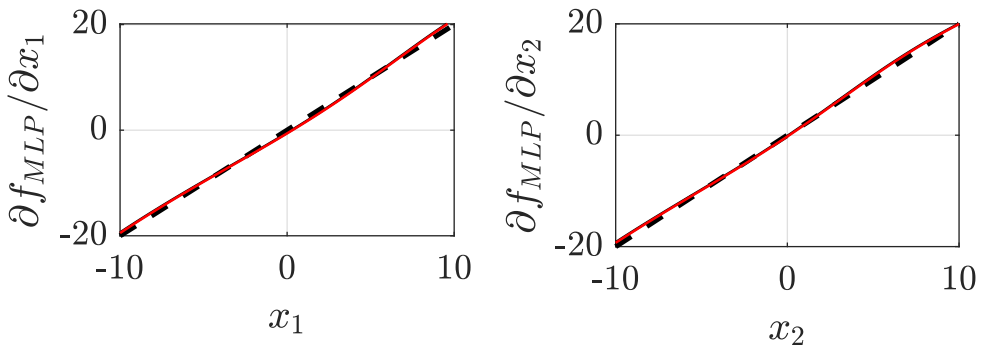


Figure 2.5: ∇f_{MLP} plot. (The dotted line is the theoretical trend as in Eq. 2.54, the black line is the derivative evaluated using numerical differentiation, the red line is the result of the formulation)

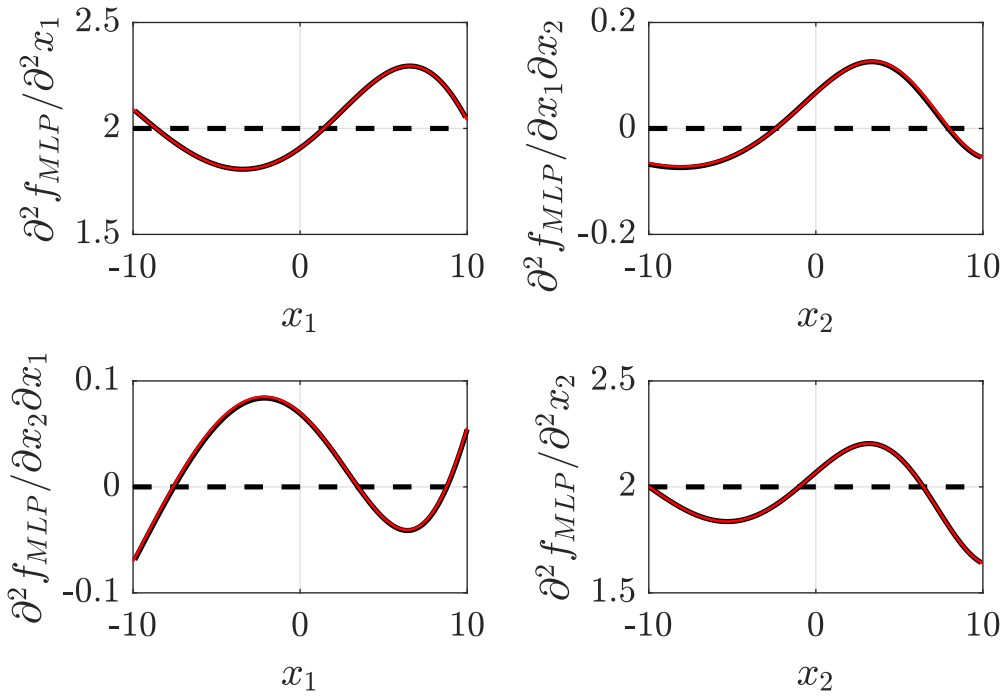


Figure 2.6: $\mathbf{H}f_{MLP}$ plot. (The dotted line is the theoretical value as in Eq. 2.54, the black line is the derivative evaluated using numerical differentiation, the red line is the result of the formulation)

Chapter 3

Experimental setup and flight data description

This chapter has been partially published in [106].

3.1 Experimental setup

This dissertation focuses on the definition and demonstration of a set of methodologies that can be applied during the design of a virtual sensor. An important step of the development of a technology is the demonstration in operative environment, that is moving from TRL 4 (Technology validated in lab) to 5 (Technology validated in relevant environment) and higher [124]. As before mentioned, this step is not always conducted in literature and this aspect has been taken as one of the classification rules to define a taxonomy of the methods found in literature. Smart-ADAHRS has been extensively tested with flight data and this chapter briefly describes the experimental setup, the flight tests and the resulting dataset. Four different entities worked together to take mutual advantage of their experience and capabilities: AeroSmart srl, a startup owning the patent of Smart-ADAHRS; two academic research groups, one at Politecnico di Torino and the other at Politecnico di Milano; and Ing. Nando Groppo srl, a small aircraft manufacturing company.

3.2 The G70 aircraft

The test aircraft is an Ultra Light Machine (ULM) named G70, manufactured by the Italian company Ing. Nando Groppo srl. G70 is shown in Figure 3.1 and it is an high-wing, single engine aircraft with fixed tricycle landing gear. The main structure is made of steel, with panels in aluminium. It received the type certificate from Deutsche Ultraleichtflugverband e. V. (DULV) for a Maximum Takeoff Weight (MTOW) of 600 kg. The engine is a Rotax 912ULS with a maximum power of

74.5 kW whereas the propeller is an Helix H50F 1.75 R-S-19-2 with three blades, fixed pitch. Two fuel tanks are located inside the wings, with a maximum capacity of 50 L each. Table 3.1 collects the main characteristics of the G70.

Table 3.1: Test aircraft main characteristics

Parameter	*Symbol*	*Value*
Wingspan	b	8.9 m
Length	-	6.22 m
Wing surface	S	10.95 m^2
MTOW	m	600 kg

Figure 3.1: Picture of the test vehicle G70 with the AOA/AOS boom mounted under the right half-wing

The G70 has been used during a Master of Science course at Politecnico di Milano entitled "Flight testing" [125] and it was analysed in several Master of Science (MS) thesis from the same university [126, 127]. The test AC has been equipped by a FTI suite named *Mnemosine* and a technological demonstrator of the Smart-ADAHRS. The final experimental setup can be seen in Figure 3.2. The subsystems are listed in Table 3.2.

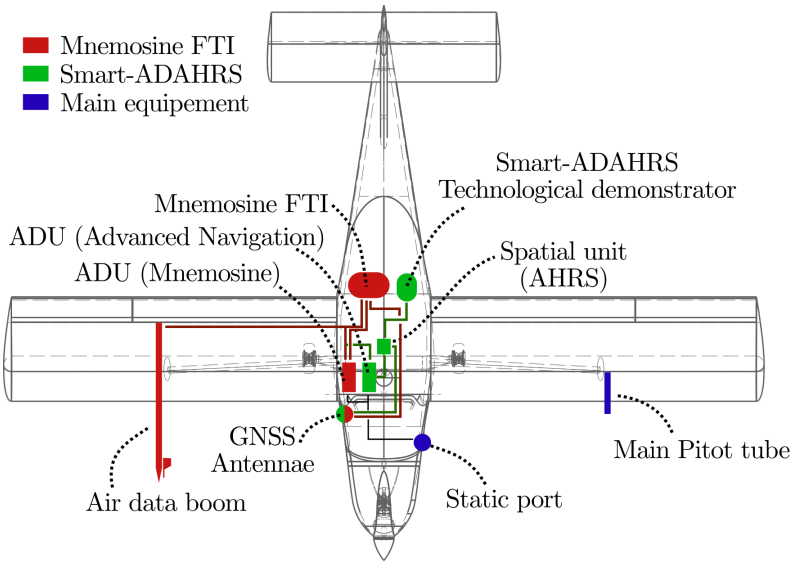


Figure 3.2: High-level schematic of the experimental setup on the top view of the G70 aircraft.

Table 3.2: Subsystems of the test equipment

System	*Model (Producer)*	*Role*	*Usage*
ADAHRS	Spatial (Advanced Navigation)	Main	Input to Smart-ADAHRS
ADS	Spatial (Advanced Navigation)	Main	Input to Smart-ADAHRS
Air Data Boom	- (Aerosonic)	Main	Target for Smart-ADAHRS
AHRS	MTi (Xsens)	Redundancy	Input to Mnemosine
GPS	LEA-6R (ublox)	Redundancy	Input to Mnemosine

3.3 The FTI Mnemosine

The Mnemosine FTI suite development started in 2005 at Politecnico di Milano. Its design has been tailored for ULM and it consists of a low cost, low intrusive and flexible solution for flight testing [128]. This system has been demonstrated capable of supporting certification procedures [129] It is composed by a Mnemosine Main Unit (MMU) that groups the main nodes of the system, including the GPS unit, a card manager to store data and the galvanically-isolated electrical power supply for the entire FTI. On-ground operations can be performed using an auxiliary external battery. Only the Air Data Unit (ADU) has been maintained independent, to keep

the installation as close as possible to two air data booms mounted under each wing, at mid-span. The right wing boom has been equipped with two vanes for AOA and AOS measurements and the left one with a Pitot-static system. The ADU is composed by a micro-controller Olimex STM32-5107 board, transducers and a signal conditioning module. Moreover, thanks to two omnidirectional antennae and a WiFi Telemetry Unit complying with 802.11n wireless protocol, Mnemosine provides on the fly data monitoring on a ground station. To monitor relevant acquired quantities in real time and mark specific events during the tests, the Flight Test Engineer (FTE) is equipped with the FTE Electronic Kneepad (FEK) equipped with a Top Switch lever and an Event Marker button. See Table 3.3 for the list of the acquired parameters list. Additional information is found in [128].

Table 3.3: Mnemosine acquired parameters list (from [128])

Parameter	*Group*	*Rate [Hz]*
Earth Centred Earth Fixed position	GPS	4
Earth Centred Earth Fixed velocity	GPS	4
Time reference	GPS	4
Body-axis acceleration	Inertial	50
Body-axis angular rate	Inertial	50
Body-axis magnetic field	Magnetic	50
Static Pressure	Air Data	10
Dynamic Pressure	Air Data	10
Angle Of Attack	Air Data	10
Side Slip Angle	Air Data	10
Outside Air Temperature	Air Data	10
Flight Controls	Flight Controls	10
Position Stick	Flight Controls	10
Force Engine	Power Plant	10
RPM Top State	Utility	10
Top Counter	Utility	10
Event Counter	Utility	10

3.4 Smart-ADAHRS technological demonstrator

A technological demonstrator of the Smart-ADAHRS has been developed with the initial aim of recording the set of signals that would be applied as input to the ANN. The demonstrator is not completely independent from Mnemosine. In fact, Mnemosine is the only one system that record the data coming from the AOA/AOS boom. This system is based on two programmable electronic platforms, namely Arduino and Raspberry. The final version of the system will embed the

inertial and air data sensors in the same box. However, at this design phase, a commercial ADAHRS (Spatial from Advanced Navigation) has been mounted externally the demonstrator. It consists of two different units, an ADU and an integrated Micro-ElectroMechanical Systems (MEMS)-based GNSS/IMU platform [130, 131]. Pneumatic signals coming from the pressure probes are divided and simultaneously measured by Mnemosine and Advanced Navigation ADU. Three serial ports for the connection of the Spatial, Air Data and Data acquisition (DAQ) units are available, together with the power connector, a RJ45 and a safety ground link. The specifications are described in Table 3.4. The list of acquired parameters is reported in Table 3.5.

Table 3.4: Demonstrator description

Size (l x w x h) [mm]	232 x 128 x 55
Weight [kg]	0.89
Power Supply [VDC]	7 – 30
Power consumption [W]	<5 (typical)

Table 3.5: List of parameters acquired by the technological demonstrator. Acquisition rate 50 Hz

Parameter
Time
Latitude
Longitude
Height
North East Down reference frame (NED) velocity vector
Body acceleration vector
GForce
Roll angle
Pitch angle
Yaw angle
Angular velocity vector
Absolute Pressure
Differential Pressure
Airspeed
Barometric altitude

The inertial unit has been carefully placed in direct contact with the structure in a position near the average CG of the aircraft, according to the instruction given by the manufacturer. GNSS antennae, both for Mnemosine and Smart-ADAHRS, have been placed on the airframe structure near the windshield to avoid limitations

on the antenna field of view. The raw output of the magnetometers has been analysed to check the presence of dynamic magnetic interference, measuring only very slight effects in engine-on conditions. For this reason, magnetometers in the Spatial unit have been kept active, considering their important role for the final accuracy and precision. The output fusion algorithm of the Spatial is not known and the manufacturer reports only that is a Kalman-like method.

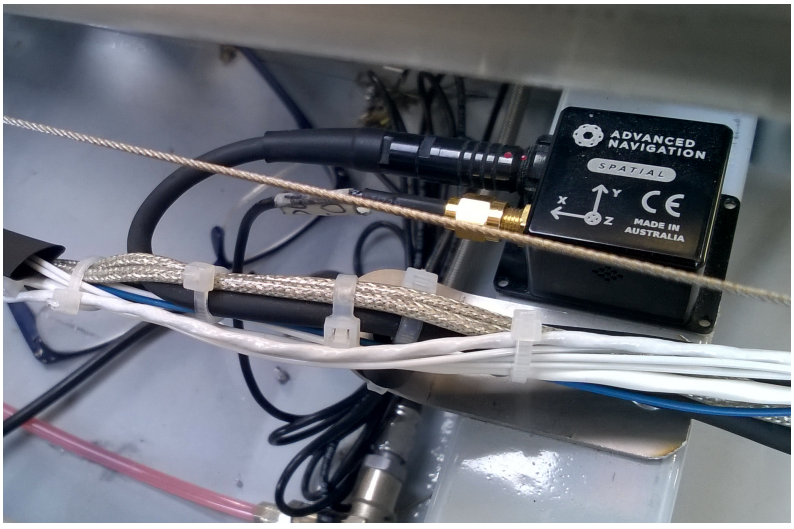


Figure 3.3: Installation of the AHRS (Spatial by Advanced Navigation)

Figure 3.1 shows the G70 from the right side, displaying the air data boom with AOA and AOS vanes. In Figure 3.4 Smart-ADAHRS and Mnemosine control unit installation are shown. See Figure 3.3 for the positioning of the Spatial unit.

From the analysis of previous flight tests, since the campaign of June 2017, a modification on the air data boom mounting the AOA and AOS vanes has been carried out. Indeed, the dynamic characteristics of the previous equipment were not suitable for the training procedure. A new boom, less subjected to oscillation, has been mounted. Its dynamical features have been estimated and reported in Table 3.6.

Table 3.6: Air Data-boom dynamic properties

Natural frequency [rad/s]	62.83
Damped frequency [rad/s]	62.83
Damping ratio [-]	0.0053

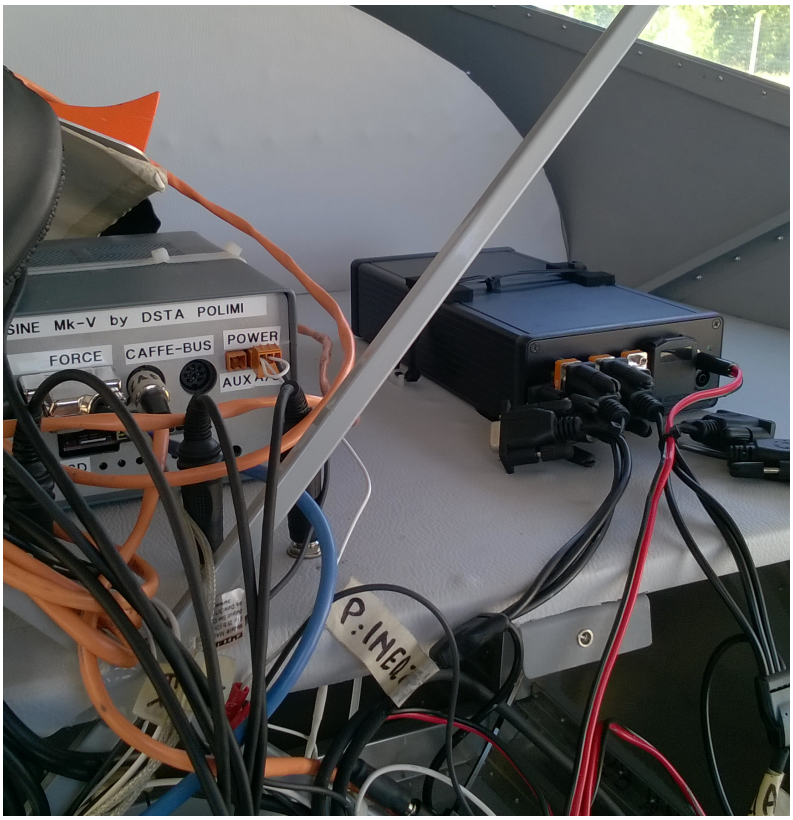


Figure 3.4: Installation of Mnemosine (on the left) and Smart-ADAHRS (on the right)

3.5 Flight test procedure and selection

Several flight test campaigns have been conducted by Politecnico di Milano since 2016. Most of them were related to the certification of the G70. The role of Politecnico di Torino was limited to provide suggestions and requests of test points to define a rich dataset for the test of the Smart-ADAHRS algorithm. As before mentioned, in several case the FTE role has been covered by MS student of the Flight Testing course. Unfortunately, most of them have to be excluded during the first data selection. In fact, the partial dependence of the Smart-ADAHRS from the Mnemosine made impossible to retrieve the data recorded by the AOA/AOS boom in several test points. Moreover, the analysis of the results brought the definition of a set of necessary test points but none of them have never been flown. For these reasons, the most complete flight campaign and the only considered valid to the aim of this dissertation is the one flown in June 2017. Figure 3.5 to 3.8 show some segments of the flight campaign conducted on June 10th, 2017, where it is possible to observe the typical correlations between the flight parameters and the position of the flight control surfaces [2].

After the Flight Test (FT) campaign has been concluded, the selection of proper time windows and manoeuvers has been carried out. As first decision, the sections with extracted flap have been discarded. This is because, in deflected flap conditions, the underlying aircraft model parameters change, therefore considering take-off and landing conditions with respect to the same model would have implied possible gross approximations. However, this should not be seen as a limitation on the application of Smart-ADAHRS, because the validity of the method is general. In fact, with a proper training, the ANN should learn a generic underlying model for deflected and un-deflected flap conditions. The trajectories on the phase space for those two configurations should be in fact separable and it is only a matter of amount of data and proper selection of the training set to learn the generic model. Further investigations on the training operation to evaluate this aspect will be conducted.

Table 3.7: Composition of the selected dataset

Flight ID	*Manoeuvers*	*Duration*
1	Sawtooth glides, Dutch-Roll	2320 s
2	Sawtooth glides, Dutch-Roll	1970 s
3	Phugoid	420 s
4	SHSS	480 s
5	Sawtooth glides	580 s
6	Sawtooth glides, Phugoid	1900 s
7	Sawtooth glides	900 s

Table 3.7 shows the test points contained in the selected dataset. During a *sawtooth glide* the AC follows a series of glide segments, increasing and decreasing the altitude between predefined bounds at constant vertical speed. Another test is the *phugoid (stick-fixed)* excitation. In this case, several techniques exist to excite the phugoid mode. Generally, the pilot commands an elevator doublet and then keeps the stick fixed. During an *SHSS*, the pilots increases the AOS in both positive and negative direction using rudder and ailerons. To excite the *Dutch-Roll*, the pilot applies an alternating rudder command and then tries to sustain the Dutch-Roll mode so that a clear oscillatory motion is generated. In this test, the pilot should find the right input frequency before stopping the rudder command. It must be noticed that the Dutch-Roll mode of the G70 appeared so damped that it has never been observed during flight tests.

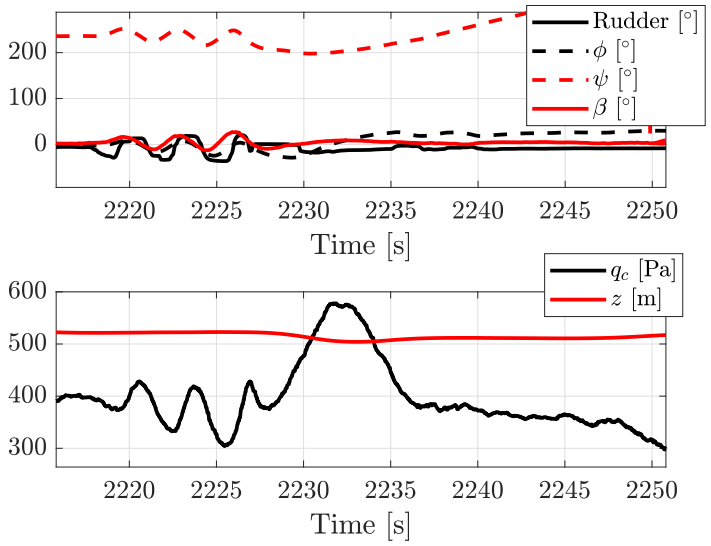


Figure 3.5: Dutch-roll test executed on June 10th, 2017

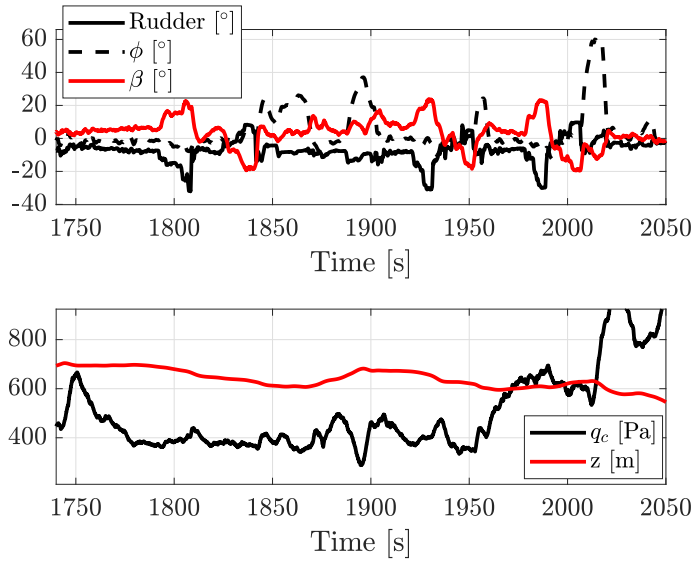


Figure 3.6: SHSS test executed on June 10th, 2017

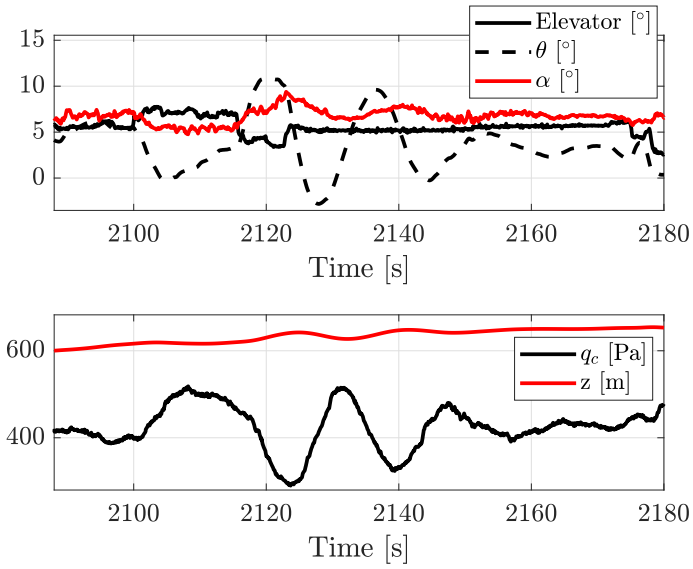


Figure 3.7: Phugoid test (stick-fixed) executed on June 10th, 2017

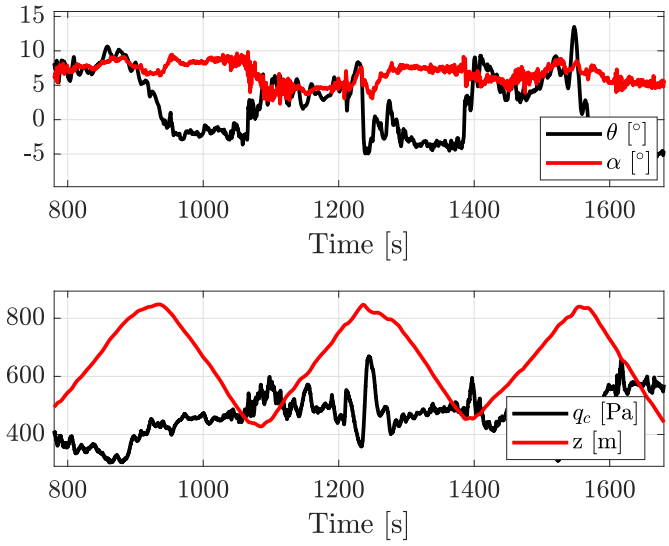


Figure 3.8: Sawtooth glides executed on June 10th, 2017

Chapter 4

Statistical methods and data mining possibilities

The analysis that are shown in this section, together with some of the results, have been partially submitted for publication in [113].

ML is applied is several engineering problems [132, 133, 134, 135]. The problem of using an MLP to estimate a flight parameter reduces to two main aspects: data collection and data exploitation. It is not a surprise, because it comes from the application of a ML technique to learn a model from data. This chapter describes statistical methods to exploit the maximum amount of information from those data. Some of them require the concept of data mining, that is the extraction of relevant information from large datasets. Chapter 5 will focus on the collection of more data.

4.1 Basic methods and proposal

Before undertaking the investigation of novel methods, the conventional methods are analyzed. Mainly, only two results have been shown in literature. The first one is, in case of NN based estimator, the training error. If plotted against the training epoch number, this value gives a global indication on the training phase. It can indicate if the training phase converged or not. Partitioning the TS it is also possible to monitor the potential overfitting condition. In fact, if the training error decreases together with a clear increase on the test error, there is high probability that the NN is overfitting the TS.

The training error can have different representations. In case of TS partitioning, their common definitions for Sum-of-Squared Error (SSE) and NSSE are reported in Eq. 4.1 and 4.2.

$$SSE_i = \sum_{n=1}^{N} \Omega_i(n)\{y\left(\boldsymbol{x^n}; \boldsymbol{w}\right) - \boldsymbol{t^n}\}^2 \tag{4.1}$$

$$NSSE_i = \frac{1}{N_i} \sum_{n=1}^{N} \Omega_i(n)\{y\left(\boldsymbol{x^n}; \boldsymbol{w}\right) - \boldsymbol{t^n}\}^2 \tag{4.2}$$

where

$$\Omega_i(n) = \begin{cases} 1 & \text{if } n \in \boldsymbol{v}_i \\ 0 & \text{if } n \notin \boldsymbol{v}_i \end{cases} \tag{4.3}$$

and $\boldsymbol{v}_i$ represents a vector containing the indexes of samples selected for the i-th set and i can represent training, test or validation.

Actually, for many reasons, this value is just an indication on the training phase, that is only one of the steps on the design of a synthetic sensor. First of all, usually this value refers to the input, output and target directly applied to the NN, which can be mapped to a different space. Hence, the metric driving the training does not assume physical values. Moreover, the map can be nonlinear, making impossible to evaluate the metric in the physical space. Secondly, this metric is a global indication on a subset of the TS. It does not give any indication on local behaviour of the function. Lastly, it is not related to the aeronautical field.

A second method applied in literature is the comparison of the target and output timeseries. With this method, it is possible to measure the local error at any instant. To properly represent the results, a significant amount of time must be plotted together with the error timeseries itself. Moreover, discarding unimportant flight phase is preferred, to avoid a bias on the analysis. This analysis can be used for a visual and numerical interpretation of the results. It is possible to find particular situation where the error increases and to understand (almost qualitatively) if the function follows the aircraft dynamics. Moreover, spikes and outliers can be easily discovered both on the target signal and on the output of the estimator. Unfortunately, this kind of analysis can become very difficult due to the dimension of the problem (dimension of the input and output spaces). The analysis of only one variable with respect to the time does not give any information on the correlation with the other input variables. At the same time, to ensure the functioning of the synthetic sensor, the estimator must be tested on a long series of timestamps, which are difficult to be analysed.

In this dissertation, the estimator sums two values: an initial angle $\hat{\alpha}$ or $\hat{\beta}$ and the neural correction $\Delta\alpha$ or $\Delta\beta$. The industrial response to this solution is usually skeptical on the neural correction segment. In fact, the impossibility to manage the

mathematical model represented by the NN always brings to ask if the estimation error is limited. This aspect brought to analyse both the final deviation between the virtual sensor and the target and also the single neural correction with respect to the initial value.

For these reasons, some statistical aspects must be considered.

To summarize, the following methods are proposed and classified.

Analysis of the results

- Comparison of the absolute deviation between the initial and final estimation
- Analysis of the deviation PDF
- Sensitivity analysis and uncertainty estimation

Analysis of the TS

- Hypercube coverage to verify that TS includes the Test Set
- TS analysis based on similarity functions

Training of the MLP

- Re-training and selection
- Manoeuver-based CV

Other support methods for the analysis of both the TS and the results

- Use of flight data mining
- Feature map for data visualization

Figures 4.1, 4.2, 4.3, 4.4 collect the functional diagrams for the techniques applied in this dissertation.

4.2 Comparison of the deviation between the initial and final estimation

Introducing the deviation between the initial estimation error and the final estimation error as in Eq. 4.4

$$\epsilon = \mid \tilde{\alpha} - \alpha_t \mid - \mid \hat{\alpha} - \alpha_t \mid \tag{4.4}$$

it is possible to measure how much the NN improved the initial estimation. α is substituted with β when analysing AOS. In this thesis, the comparison is made on the following statistics: mean value of ϵ, standard deviation of ϵ and maximum value of ϵ [136]. If the metric based on ϵ is negative, it means that the NN actually improved the estimation from $\hat{\alpha}$. Figure 4.5 shows an example of possible chart. The value on the abscissa corresponds to a segment of the dataset.

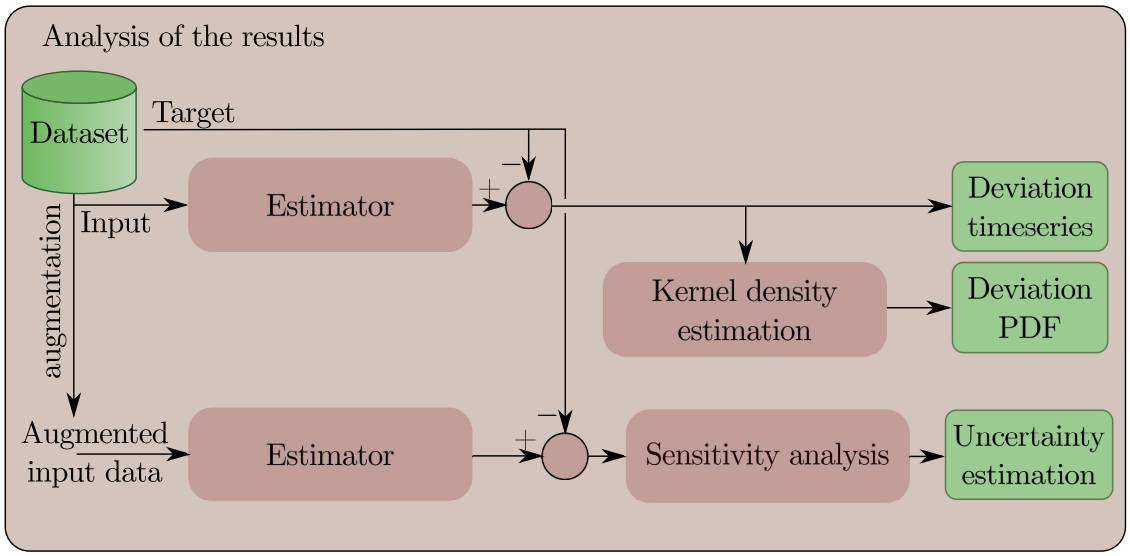


Figure 4.1: Functional diagram of the methods for the analysis of the results

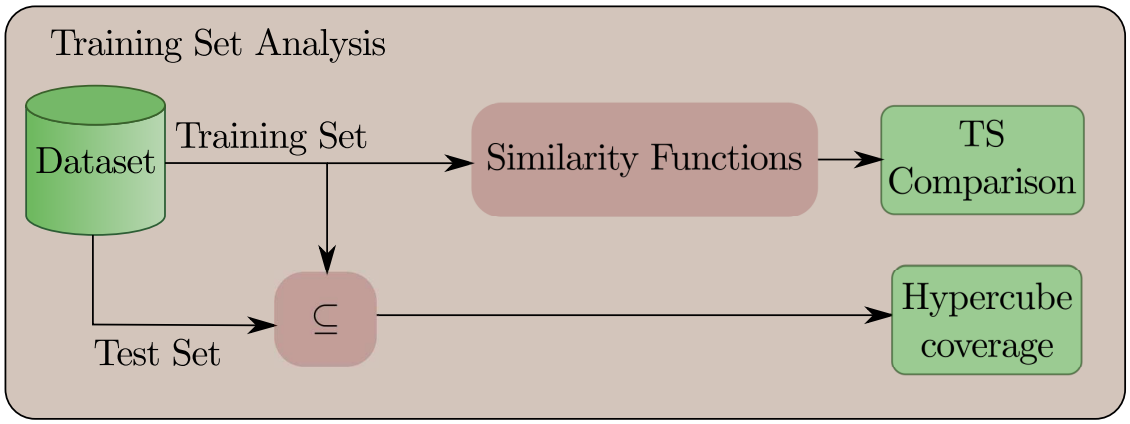


Figure 4.2: Functional diagram of the methods for TS analysis

4.3 Hypercube coverage to verify that TS includes the Test Set

Because of the weights of the NN are obtained by means of nonlinear regression, the distribution of the TS is fundamental. In following sections the TS will be deeply analysed. For now, it is sufficient to introduce the chart of the single marginal distribution of the input and target variables. This chart is here represented as a classical box-and-whisker plot [137]. An example can be seen in Figure 4.6.

This chart can help in the initial phase of design to select the TS so that at least the desired flight envelope is included in it. More important is the comparison of the same chart obtained for the test set. In fact, this method allows to check

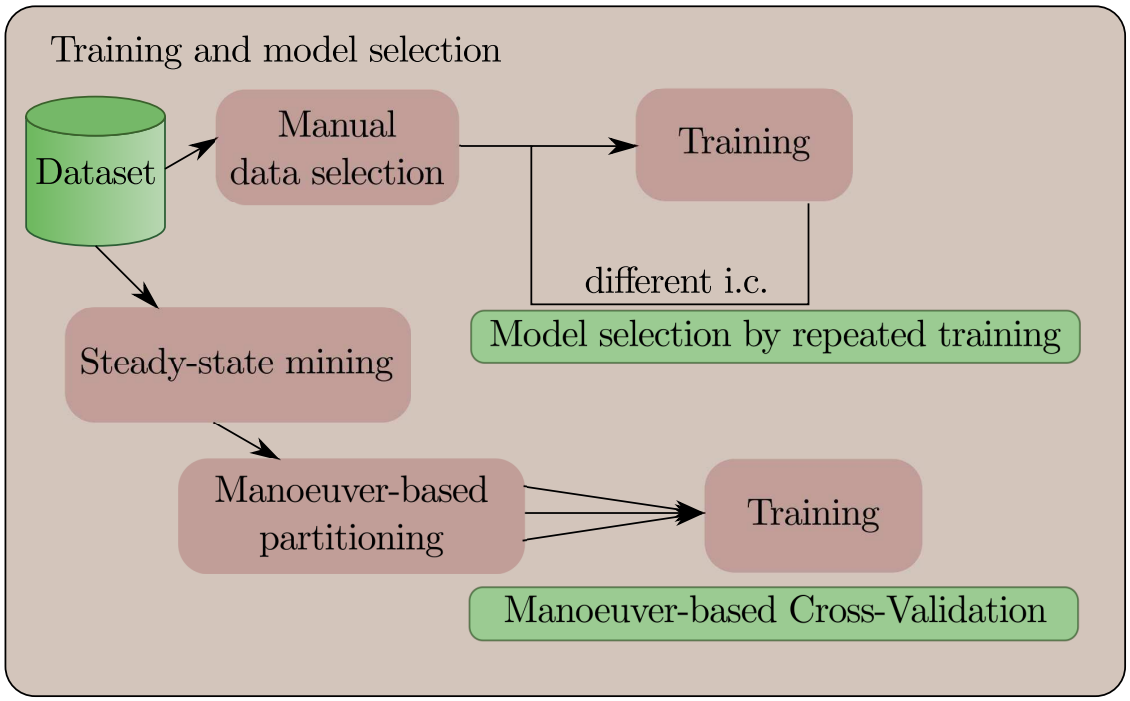


Figure 4.3: Functional diagram of the methods for training and model selection

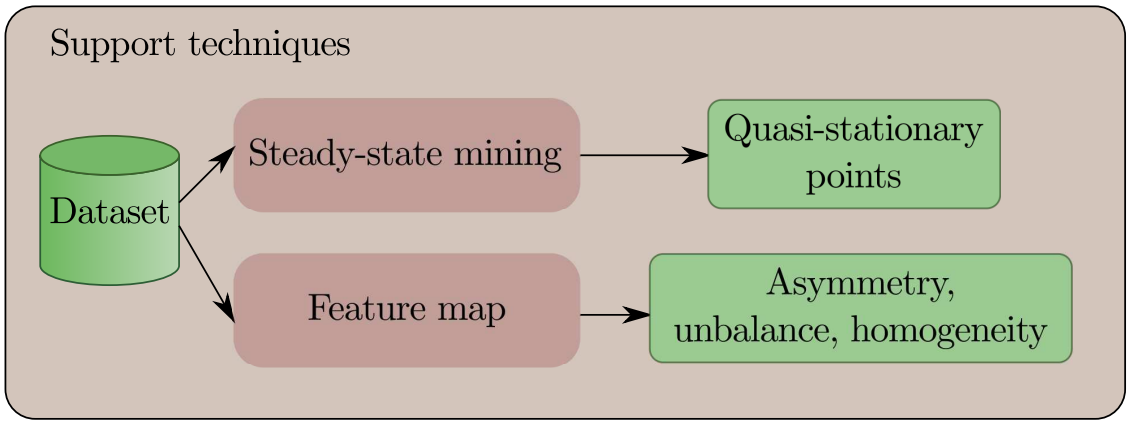


Figure 4.4: Functional diagram of the support methods based on data mining and visualization

if the TS is a superset of the test set or, in other words, the inclusion of the test set in the TS applied to define the virtual sensor. If some points are not included in the TS there are two possibilities: the output of the NN is erroneous or the NN can still generalize the target from extrapolation. However, it must be noted that this condition should be avoided. In fact, if TS is a superset for any test set means that, in case of asintotical training, the UAT can be applied. On the contrary, to

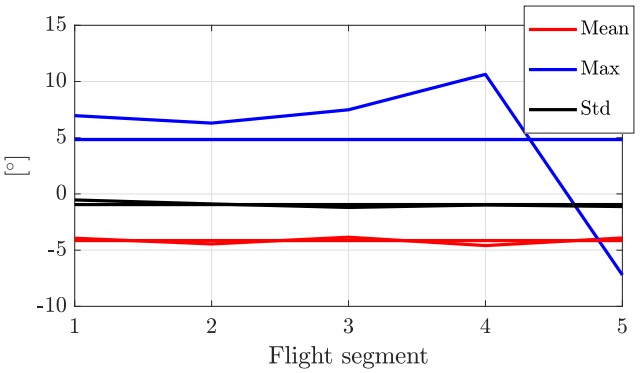
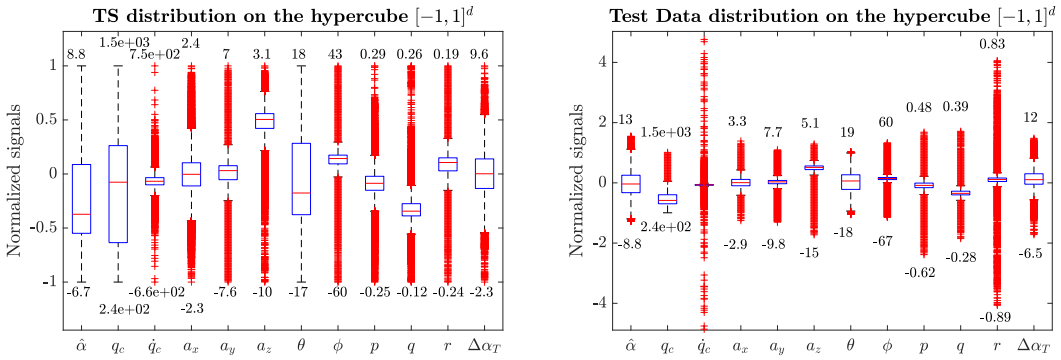


Figure 4.5: Example of statistics of ϵ



(a) TS distribution

(b) Test set distribution

Figure 4.6: Distribution of the test set on the hypercube defined by the TS (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s]).

the best of the author's knowledge, outside of the hypercube defined by the UAT nothing can be said on the nonlinear map represented by the NN. Moreover, an inclusive TS makes every test an additional check of the nonlinear map on internal points, adding confidence on the final sensor. The possibility of an entire set of points not included means that the behaviour of the map on that region has not been controlled by the training phase.

4.4 PDF of the deviation

Another important analysis that can be conducted is the estimation of the PDF of the deviation between the target and the output of the VS. This method allows to better understand how the final error of the sensor is distributed. An example

can be seen in Figure 4.7 where a kernel density method has been used to estimate the deviation PDF. For instance, from Figure 4.7 can be seen that the mean error is close to 0° and that the distribution of the error is symmetric and unimodal.

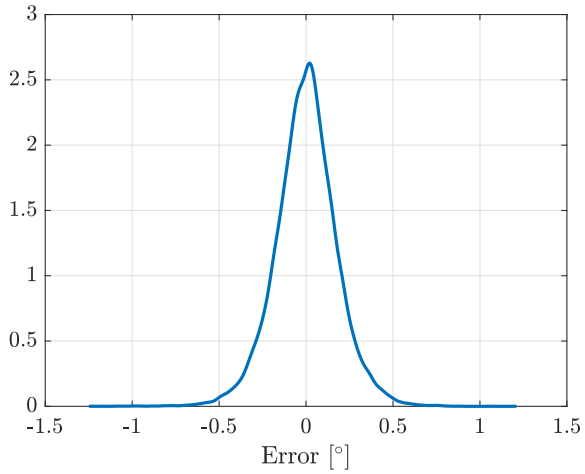


Figure 4.7: Example of error PDF

4.5 TS analysis based on similarity functions

In engineering, the source of data is usually an experimental measurement. This means that data is affected by noise and errors and the effort to reduce the uncertainty on the measurement usually results in increased costs. At the same time, during flight tests the measurement uncertainty is higher than lab experiment conducted in controlled environment. Sometimes, it would be necessary to plan, cancel or repeat additional tests depending on availability of the aircraft, weather conditions, overall budget. Final dataset is hence the result of a list of trade-off.

When a ML algorithm is hence implemented, the selection of data is of fundamental importance. A lack of methods to select data is observed, when dealing with supervised regression [138, 139]. Regularization methods are usually applied to handle noise and outliers, and outliers detection methods can be separately used to recognize unvalid data.

This section suggests a method based on similarity measures for the analysis of the TS effectiveness and, in turn, for the creation of the TS itself. This method will give answer to some of the typical questions in applied ML: is this training set valid? Is the generalization error related to the learning algorithm or to the measurement error? Are there some points of the training set that should be removed? Some aspects of this section can fit with the generic formulation showed in [140].

4.5.1 Mathematical background

Henceforth, the analysis is conducted for a 1-dimensional target space. However, this is not a limitation because the analysis can be independently conducted on every target component.

Definition *Defining* χ_j *as*

$$\chi_j = \{\textit{multiset of indentical elements } \mathbf{x}_i \in \mathbf{X}\}$$

with cardinality $l_j = m_{\mathbf{X}}(\mathbf{x}_i)$*, where* $m_{\mathbf{X}}(\mathbf{x}_i)$ *represents the multiplicity of* $\mathbf{x}_i$ *in* $\mathbf{X}$*, it is possible to write*

$$t|_{\chi_j} = \left\{ t_i \in \mathbf{T} \,\middle|\, (\mathbf{x}_i, t_i) \in \mathcal{T}, \mathbf{x}_i \in \chi_j \right\} \tag{4.5}$$

It is hence possible to manually evaluate $p\left(t|_{\chi_j}\right)$ to verify if χ_j meets the statistical requirements, which occurs if there are enough values to obtain a reliable PDF or if the distribution is normally distributed or it is multimodal. Eventually, exploring the reasons of the multimodality would contribute to find the searched outliers. To obtain $p\left(t|_{\chi_j}\right)$ a kernel density estimation method can be applied.

4.5.2 The algorithm

The main drawback in the proposed approach is that finding several identical input vectors corresponding to different target values is quite tough in real applications. For instance, in engineering and practical applications, the noise will affect every measurements impeding to build the situation reported in Eq. 4.5.

Eq. 4.5 basically suggests to analyze how the target distributes in case of identical input vectors. This is intractable in engineering, but also in other fields, due to the measurement error. In fact, also in controlled experiments, input vectors will never be actually the same. The solution proposed takes advantage of clustering methods. Think about snowflakes. A common sentence is that there is no possibility of finding two identical snowflakes. However, if some details can be removed, it is more likely to find two *similar* snowflakes. Thinking to the snowflakes as the input vectors, it means that confusing a set of input vectors with only one vector, it is possible to determine the distribution of the target values corresponding to this particular set. In details, it is possible to write the relation 4.6

$$\mathbf{S}_i = \{\mathbf{x} \in \mathbf{X} \mid \mathbf{x} \sim \mathbf{x}_{c,i}\}, \mathbf{x}_{c,i} \leftarrow \forall \mathbf{x} \in \mathbf{S}_i, \implies p(t \mid \mathbf{x} \in \mathbf{S}_i) = p\left(t\Big|_{\mathbf{S}_i}\right) \tag{4.6}$$

where $\mathbf{x}_{c,i}$ is the centroid of the i-th cluster $\mathbf{S}_i$ (subset of $\mathbf{X}$). This assumption helps to re-write the definition reported in Eq. 4.5.

The proposed algorithm can be disentangled as follows:

- CLUSTERING of a subset of the input space (hence, excluding the target)
- EVALUATION of the proposed similarity functions
- COMPARISON of the TS

It is important to divide the sole input space in subsets, without considering the value of the corresponding target observations. In fact, if the target is included into the partitioning, it will affect the comparison analysis. Observing a spike in the target value, it can be noticed that its enormous entity would likely affect the definition of the subsets, by assigning it to a cluster which is not the closest in the input space. In this case, the analysis of the target values would not show any anomaly whereas, if the spike is assigned to the closest cluster in the input space, the proposed approach allows to observe an outlier.

Please note that a multimodal PDF does not strictly mean that the MLP will hardly estimate the expected value. If the points on the subdivision are far enough, the target values may not be associated to the same data generating distribution. Briefly, the cluster is too large and the modes belong to different zones of the hypercube.

Obviously, this method is affected by the choice of the number of clusters, that is their size. However, an interesting result on the connection between multimodality and cluster size will be showed later on.

The squared error evaluated over the entire test set and the SSE are used to verify if the proposed similarity functions are actually working. A non-dimensional error can be introduced as follows:

Definition *Given a model M and its corresponding training error in terms of MSE, a target value t and the approximation of t called $\hat{t}$ obtained with M, it is possible to write:*

$$E_n = \frac{\left(\hat{t} - t\right)^2}{MSE_{train}}$$

Difficulties arise, however, when the learning paradigm is not able to represent the underlined solution. In other words, if a single hidden neuron feedforward NN is applied to learn a map different from a sigmoidal function, the application of this approach cannot lead to substantial improvements on the final generalization error. In fact, the final regression will always be a sigmoid and the reduced complexity of the regression model could imply several comparable local minima of the error function. In other words, several different sigmoidal functions fitting the given TS may exist, but any of them can represent the actual correlation of the processed data.

4.5.3 Similarity functions

Previous sections depicted the problem and provided a method for the TS analysis. This section introduces two similarity functions for the evaluation and comparison of TS.

The Density-Closeness function

Analyzing the TS data distribution with respect to the Test Set, two observations can be made. Firstly, suppose that every cluster has enough points to determine a PDF on the target variable and consider null the measurement error. In this situation, a test point close to a centroid would give a low generalization error. However, the assumption of densely populated clusters is not always applicable. In fact, it may happen that the TS has only one isolated point close to the test point. In this case the associated cluster is not considered reliable. The reason of such a situation may origin from a bad design of the experiment or even in the decision to avoid to observe some conditions, both to reduce the overall experiment cost and impossibility to provide measurements in such conditions. Another situation is encountered when the distance inside the hypercube between the test point and the TS increases. This involves, together with the previous observation, testing the regression with a test point surrounded by a lot of elements of the TS. Again, the generalization error should be low. However, the assumption that each test point is surrounded by a lot of points might be too strong. Summarizing, it would be advisable to have $\mathbf{X}$ such that the coverage of the input space is dense and homogeneous. This corresponds to an high number of small clusters, defined by a lot of points. For sake of clarity, see Figure 4.8.

Figure 4.9 shows the position of the TS entries superimposed to the logarithm of E_n. It can be observed, as expected, that the approximation error is reduced in correspondence to the blue dots, representing the TS. Moreover, some connections can be visible between the low error regions. These connections actually represents the regions of good generalization. Please consider that the figure does not show necessarily an overfit condition.

For these reasons, a function is proposed to associate the approximation error to the input vector. This value involves both closeness of a centroid with the density of points that define the cluster itself. This *Density-Closeness function* will give an evidence for the proposed approach.

Definition *Given a partition $\mathcal{P}$ of the input space $\mathbf{X}$ and an element $\mathbf{x}_i$ and defined n_j, r_j respectively as the cardinality and the radius of the subset j such that $\mathbf{x}_i \in \mathbf{S}_j$, it is possible to define the* Density-Closeness function $d_{nrd} : \mathbf{X} \to \mathbb{R}$ *as follows:*

$$d_{nrd}(\mathbf{x}_i) = \frac{n_j}{r_j \left\| \mathbf{x}_i - \mathbf{x}_{c,j} \right\|_2} \tag{4.7}$$

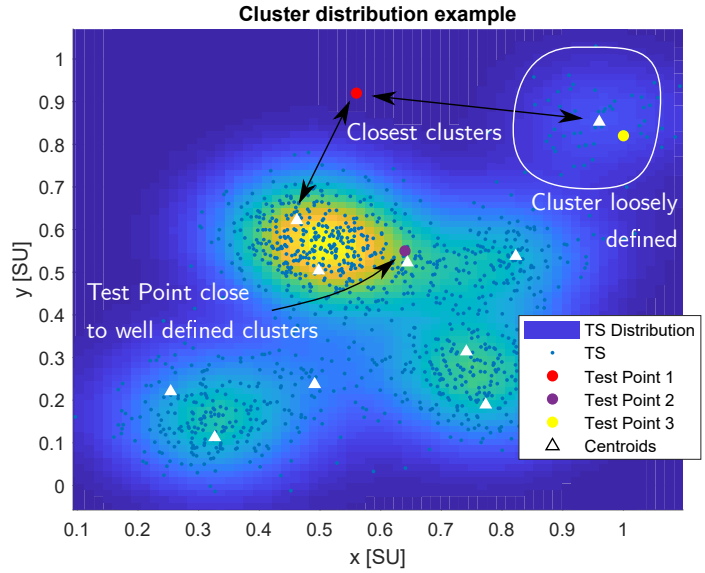


Figure 4.8: Effect of the distribution of the TS entries on the accuracy at several position on the input space.

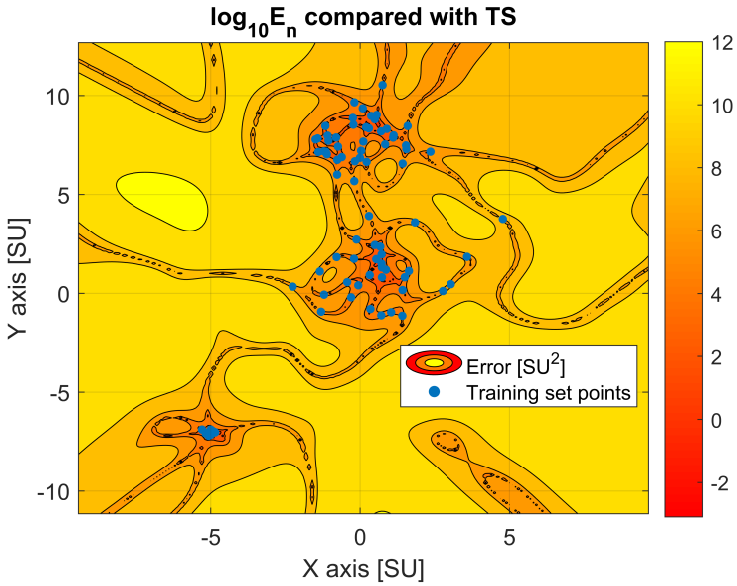


Figure 4.9: Comparison of the logarithmic normalized approximation error $log_{10}(E_n)$ with availability of training points (analytic test case). SU stands for System Unit

Intuitively, if the model distribution is tested in a region where the approximation is statistically valid, the generalization error would be low. To be considered

statistically valid, a certain amount of points must be used to fit the distribution and the proposed analysis must be sufficiently refined to detect the local effects. The intuitive results are reported in Eq. 4.8 and 4.9, where d stands for distance.

$$\begin{cases} \text{n/r high} \\ \text{d low} \end{cases} \implies \frac{n}{rd} \quad \text{high} \implies E \quad \text{low} \tag{4.8}$$

$$\begin{cases} \text{n/r low} \\ \text{d high} \end{cases} \implies \frac{n}{rd} \quad \text{low} \implies E \quad \text{high} \tag{4.9}$$

Please note that it is also possible to generalize Eq. 4.7. In fact, in place of the cluster radius r_j, it is possible to consider any other measure of the cluster concentration, which could be geometrical or statistical (for example standard deviation of the D-dimensional distribution inside each cluster).

In Sec. 4.5.4 some initial results will be shown. In those cases, a general trend can be observed such that a minimum value for d_{nrd} can be associated to a maximum error. In this way, a TS could be enhanced in some region, increasing the corresponding d_{nrd}. At least for low dimensional problems, a summarizing value can be determined from this map as the ratio of points in the hypercube characterized by d_{nrd} at least equal to a given value.

Similarity function calculated from cardinality of the set of the stationary points

If the TS is collected from practical experiments, it could happen that some entries result from erroneous measurements (for example: thermal distortion of a ruler, aerodynamic deflection of the sensor support in a wind tunnel setup, detachment of the sensor itself during the test). In some cases, this kind of error might be avoided with a proper design of the experiment. However, the cost of the experiment is always an important figure on the trade-off between the repetition of the test and the decision to exploit the obtained measurements. On the other hand, a new algorithm can be applied on data that was previously collected for other purposes. In this case, the experiment setup has been designed for another test without taking care of some details that might be important for the new application.

This section deals with two different kinds of stochastic errors, as reported in Eq. 4.10.

$$t_{meas} = \tilde{t}_{real} + w + u \tag{4.10}$$

where $\tilde{t}_{real}$ represents the real target considering the nominal distortion effects of the sensor, $w \sim p_{noise}(w)$ models the stochastic noise and $u \sim p_{additive}(u)$ models a less likely but much more disruptive error. Hence u represents the unexpected totally erroneous measurement, such as a damage on the sensor. Everything else

acting on the measurements is contained in w. It is hence possible to consider t as the union of a set of valid observations and a set of outliers e_i as follows:

$$\mathbf{t} = \{t_1, t_2, t_3, \dots t_n, e_1, e_2, \dots e_j\} \tag{4.11}$$

where t_i stands for a single target measurement, e_i for an erroneous target measurement and $\mathbf{t}$ for the entire set of targets.

The analysis of the target PDF inside each cluster can give some important insights on this problem. In fact, if $\mathbf{t}$ has a non-negligible error in a particular position of the hypercube, and other measurements have been collected in similar conditions, the density distribution of the target will tend to become multimodal. In this case, the situation reported in Figure 2.2 modifies in Figure 4.10.

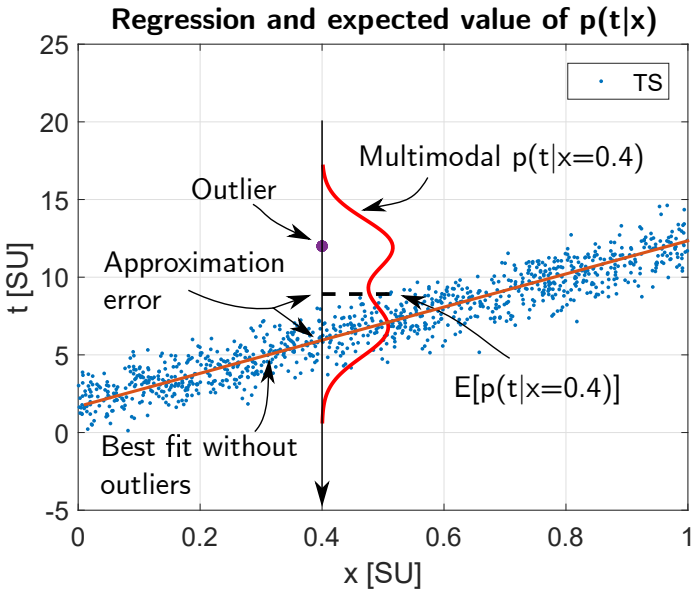


Figure 4.10: Effect of the presence of outliers and noise on the conditional PDF of the target given the input

Hence, it is possible to estimate the target PDF with kernel density methods, as proposed in Sec. 4.5.1 and 4.5.2, and to evaluate the cardinality of the set of the stationary points. In this paper, $p\left(t\big|_{\mathbf{S}_i}\right)$ has been obtained as discrete function.

Recall the stationary point definition as

Definition *A stationary point of a function f is every point x_0 where f is differentiable and $\frac{df}{dx}(x_0) = 0$*

It is possible to apply the definition to the discrete functions comparing each value with the previous one. It is easy to imagine that the cardinality of the set

of the stationary points is affected by the partition of the input space. However, if the erroneous entries are close to other training points, they will be assigned to the same cluster unless the subdivision is further refined.

Hence, features like maxima, extreme points and constant segments of $p\left(t\big|_{\mathbf{S}_i}\right)$ are dependent on the number of subdivisions. This dependency gives information about the presence of outliers in $\mathbf{t}$.

Notation *Let $\mathcal{P}$ be a partition of the input space $\mathbf{X}$ and $\mathbf{T}$ the target space, a discretization of the target PDF inside each subset $\mathbf{S}_i \in \mathcal{P}$ is written as $p\left(\left[t_n\big|_{\mathbf{S}_i}\right]\right)$.*

In this work, minima are not considered in the overall count. Hence only maxima, constant values and points on the domain boundaries have been counted. For these reason, the following similarity function can be defined:

Definition *Let $\mathcal{P}$ be a partition of the input space $\mathbf{X}$ with cardinality τ and $\mathbf{T}$ the target space. Given the discretized target PDF inside each subset $p\left(\left[t_n\big|_{\mathbf{S}_i}\right]\right)$, it is possible to evaluate C as the maximum value of the cardinality of the sets of constant and the maximum values of $p\left(\left[t_n\big|_{\mathbf{S}_i}\right]\right)$ inside each subset.*

$$C(\tau) = \max_{i=1\ldots\tau} \left| \left\{ t \in \mathbf{T} \middle| \max p\left(\left[t_n\big|_{\mathbf{S}_i}\right]\right) \vee p\left(\left[t_n\big|_{\mathbf{S}_i}\right]\right) = p\left(\left[t_{n-1}\big|_{\mathbf{S}_i}\right]\right) \right\} \right| \tag{4.12}$$

Moreover, given the k-th set in which Eq. 4.12 is maximum, it is possible to define C_2 as follows:

$$C_2(\tau) = \frac{C(\tau)}{\mid \mathbf{S}_k \mid} \tag{4.13}$$

Definition *Let $\mathbf{X}$ and $\mathbf{T}$ be respectively the input and target spaces. Given a family of partitions $\{\mathcal{P}\}$ with increasing number of subdivision from 1 to l where l is the total number of TS entries, the area of the cardinality A_C can be defined as:*

$$A_C = \frac{l-1}{2N} \sum_{n=1}^{N} \left(C(\tau_n) + C(\tau_{n+1}) \right)$$

evaluated with trapezoidal numerical integration with N usually equal to l. Moreover, L_C can be defined as the arithmetic mean of the first J values of $C(\tau)$

$$L_C = \frac{1}{J} \sum_{j=1}^{J} C(j) \quad where \quad J < l$$

By extension, it is possible to define A_{C_2} as the area of $C_2(\tau)$.

In this paper, J has been set to the value at which if the clusters are equally distributed, they will contain 5 entries. It means around 9 - 10.

Three effects can be indirectly observed from the trend of this function. If the sample is drawn from an ideal distribution without the w and u noise, the value of $C(\tau)$ should quickly drop, thanks to the smoothness of the function to be learned. Otherwise, the term u adds observations that affect the target PDF until they are isolated, when τ becomes high. On the other hand, the w term smoothes the PDF, reducing the area A_C.

In Sec. 4.5.4 some visual and numerical demonstration of the proposed similarity functions will be given.

4.5.4 Test Case: analytic function

This section shows an example of the results obtained with the method based on similarity functions. The test case is the regression of an analytic function. This test case is effective in showing a TS that does not cover the manifold of definition of the function. Some observations will be drawn in case of noise and outliers injection. Those observations will give evidence that this technique is also effective in the determination of bad tuples $(\mathbf{x}, \mathbf{t})$.

Consider the following real-valued function:

$$\begin{aligned} f :&\mathbb{R}^2 \to \mathbb{R}, \\ (x, y) &\mapsto \cos\left(0.1(x^2+y^2)\right) e^{-0.05(x^2+y^2)} + 0.5e^{-(x-5)^2-(y-5)^2} + 0.5e^{-(x+5)^2-(y)^2} \end{aligned} \tag{4.14}$$

The function in Eq. 4.14 is a valid example of a composition of exponentially decaying functions. With a partial knowledge of the map, fitting this kind of function is very hard. In fact, although the three addends affect the entire function domain, it might not be sufficient to learn the position of the maximum and the width of the single bells.

Figure 4.11 shows the function f together with the TS and Test Set.

Ten different TS have been sampled with different sizes and distributions from the Equation 4.14. The first two TS have 92 entries, other two have 500 entries and the last six have 2800 entries. Among the TS with the same size of 92 and 500 entries, the spatial distribution of the entries changes. This would demonstrate that not only the size of the TS affects the quality of the approximation but also the location of the TS on the hypercube. This gives the proof that the proposed method shows sensitivity on this subject.

The largest TS has been used to test the sensitivity of the method to noise and outliers injection. Six different TS of 2800 entries have been drawn with every

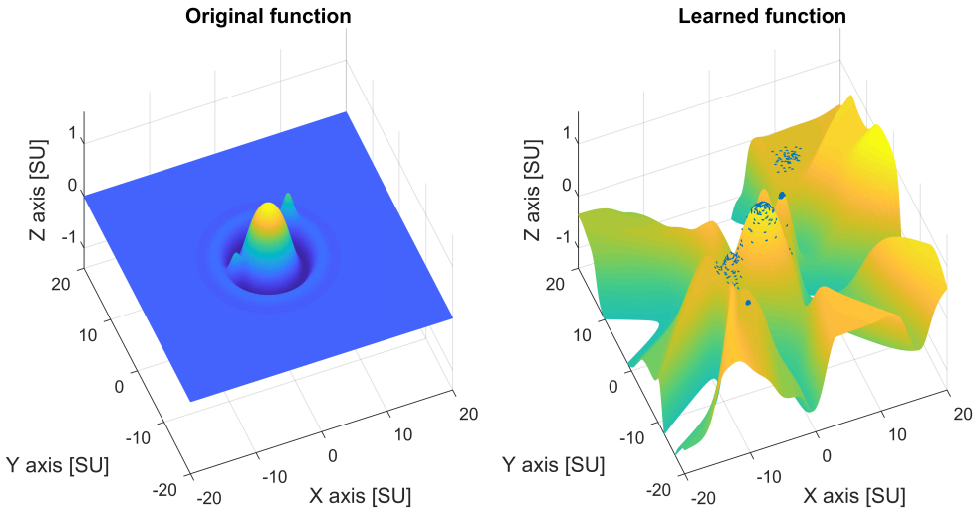


Figure 4.11: Original function to be learned (left) compared with a neural fitting (right). The blue dots on the right figure represents the TS.

combination between injected outliers, no outliers, no noise and two level of noise. A gaussian noise has been applied to the target values to detect the effect of the uncertainty on the data. Eventually, some erroneous entries have been manually added to the TS. Table 4.1, 4.2 and 4.3 collect the information on the various TS.

Table 4.1: Mean and variance of the family of generating distributions of the TS

	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	$\mathcal{N}_8$	$\mathcal{N}_9$
$[\mu_x, \mu_y]$	[0, 0]	[0, 0]	[0, 1]	[3, 1]	[-5, -7]	[0, 8]	[10, 15]	[5, 5]	[-5, 0]
$\left[\sigma_x^2, \sigma_y^2\right]^T$	[1, 1]	[64, 64]	[1, 1]	[4, 4]	[0.01, 0.01]	[1, 1]	[1, 1]	[0.01, 0.01]	[1, 1]

The applied MLP for this test case has 2 hidden fully connected layers with 20 neurons each, organized in a Feed Forward architecture.

4.5.5 Observations

The density-closeness function values is showed in Figure 4.12. In this scatter plot, the d_{nrd} value has been correlated to the normalized approximation error E_n, showing the expected relationship between higher d_{nrd} and lower E_n. It must be recalled that E_n is a squared error normalized by the $\text{MSE}_{\text{train}}$. Hence, if to a particular input vector corresponds $E_n = 1$, it means that the approximation error equals the averaged squared error obtained after training, on the applied TS. Figure 4.12 shows the CDF of d_{nrd} related to the first test case and also the

Table 4.2: TS point distributions

TS	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	$\mathcal{N}_8$	$\mathcal{N}_9$	$\sigma^2_{p_w}$
1	0	0	30	2	20	40	0	0	0	0
2	0	30	0	2	0	0	0	20	40	0
3	100	0	0	10	140	0	50	100	100	0
4	50	0	0	300	0	0	50	50	50	0
5	0	0	1000	500	300	1000	0	0	0	0
6	0	0	1000	500	300	1000	0	0	0	0.04
7	0	0	1000	500	300	1000	0	0	0	0.16
8	0	0	1000	500	300	1000	0	0	0	0
9	0	0	1000	500	300	1000	0	0	0	0.04
10	0	0	1000	500	300	1000	0	0	0	0.16

Table 4.3: TS size

TS	*# Entries*	*# Extra entries*
1	92	0
2	92	0
3	500	0
4	500	0
5	2800	0
6	2800	0
7	2800	0
8	2800	5
9	2800	5
10	2800	5

respective mean values have been highlighted. Figure 4.12 shows the complements of the TSs and no further selection or processing have been applied to them. The decrease appears to be power law in this case. Moreover, It is clear that the big TS (corresponding to TS 5) produces a better fit than the TS 1, as expected. In Figure 4.12 the corresponding points of the TS 5 shifted downwards and rightwards. In fact, the TS 5 has a lot more entries in regions where the TS 1 leads to a poor fit. At the same time, a cloud of points distributed around $(d_{nrd}, E_n) = (10^4, 1)$ can be observed, which is missing in the scatter plot related to the TS 1.

It is interesting to see the effect of feature extraction on the density-closeness function. Removing two input variables brought an increase on the d_{nrd} mean value, mainly due to the dimensionality reduction acting on the l_2-norm at denominator. This means that d_{nrd} cannot be applied to compare TS with dimensions of the input space that differ too much.

The effect of the presence of the outliers is clear from Figure 4.13. In fact,

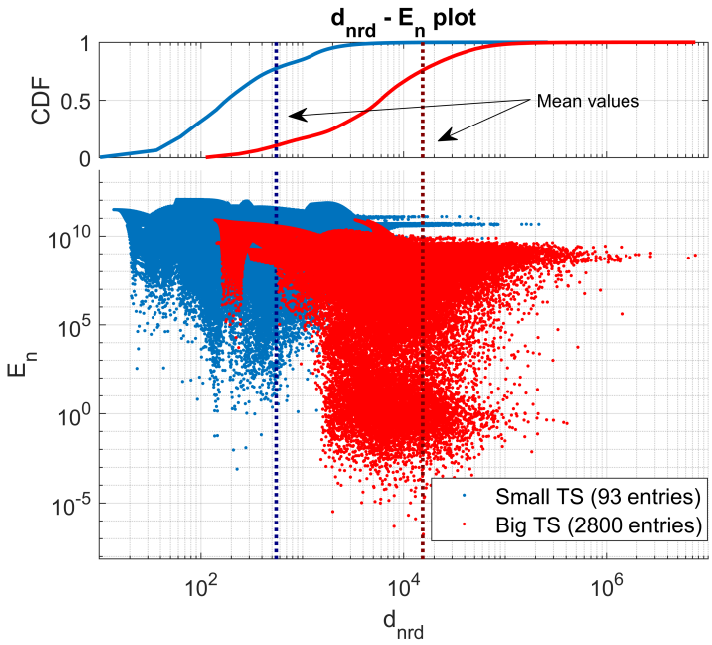


Figure 4.12: CDFs of the Density-Closeness function (upper plot) and corresponding E_n (lower plot) for two different TS. Please note that the d_{nrd} and E_n scales are logarithmic. The dashed lines represents the mean values of the d_{nrd} distributions.

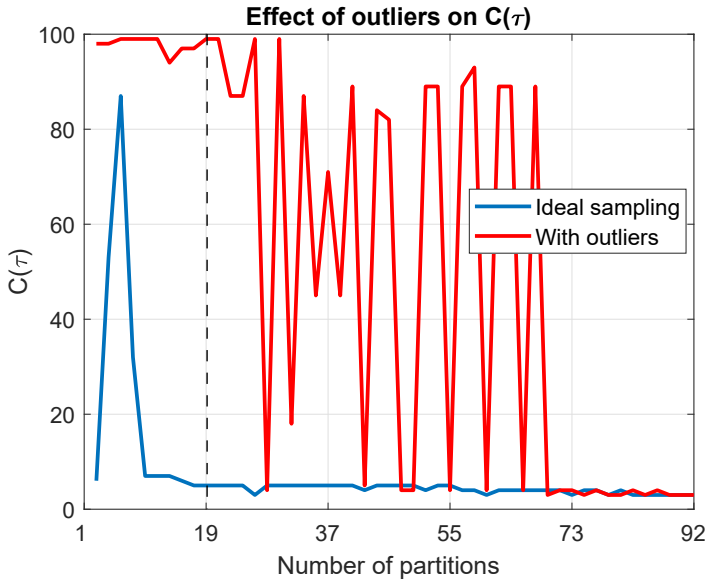


Figure 4.13: Analysis of the effect of outliers on the $C(\tau)$ similarity function. J is represented by the dashed line

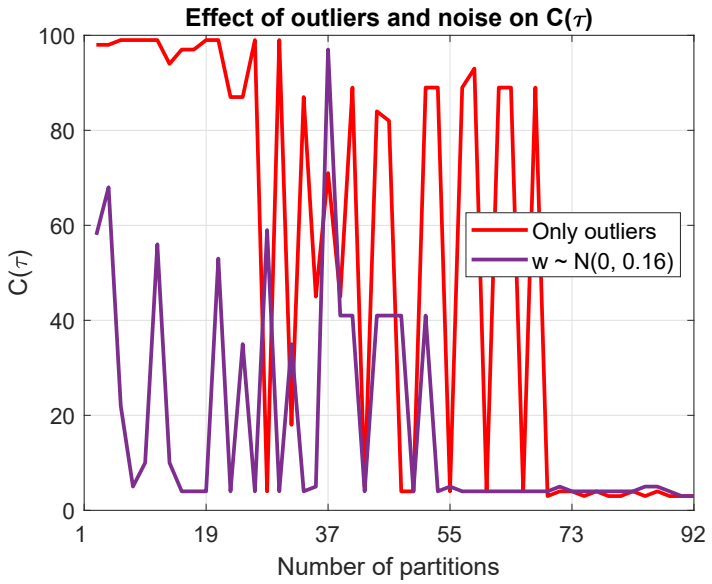


Figure 4.14: Analysis of the effect of noise on the $C(\tau)$ similarity function in case of outliers.

because the injected errors differs from the value assumed by the other entries, close in the input space, the target PDF in the corresponding cluster contains a lot of stationary points. The trend of $C(\tau)$ drops with a low number of subdivisions, when only ideal observations are contained into the sample. Otherwise, in case of

outliers, the cardinality drops when the number of subdivision is higher. At the same time, the noise injection also affects this figure, as can be seen in Figure 4.14. The smoothness induced by the noise broadens the target PDFs and the kernel density estimation fits smoother distributions, with a lower number of stationary points. The corresponding function, chosen to summarize this behaviour, allows a differentiation in the various TS.

The entire list of results from 100 tests have been summarized in Table 4.4, 4.5 and Figure 4.15. Several trends can be observed from Figure 4.15. L_C is significantly higher in TS 2 than in TS 1 and also A_C increases slightly. In fact, although the introduction of points with mean in $[0, 0]$ with $\mathcal{N}_2$, and the points around $[5, 5]$ and $[-5, 0]$ introduced by 20 points of $\mathcal{N}_8$ and 40 of $\mathcal{N}_9$, there is no improvement on the description of the function. In fact, $\mathcal{N}_2$ has been made purposely broad so that the 30 points likely fall far from the peak. Similarly, $\mathcal{N}_8$ is a narrow distribution and the removal of the points from the other distributions likely damaged the discretization of the objective function. From TS 3 to 4, the number of points associated to the most interesting distributions $\mathcal{N}_{1,8,9}$ has been halved, maintaining the size of the TS at the same value of 500 entries. This affected negatively on the A_C values, which increased. The steep increase of A_C and L_C passing from TS 4 to 5 is associated to the significant increase, with a factor of 5.6, on the number of points of the TS 5, which is associated to a general higher value of $C(\tau)$ with respect to TS 4. The effect of noise injection can be seen from TS 5 to 7, showing a slight reduction of both similarity functions. The injection of outliers in TS from 8 to 10 increases significantly the values of both A_C and L_C with respect to previous TSs. At the same time, the reduction of $C(\tau)$ shown in Figure 4.14 is confirmed in the decreasing trend from TS 8 to 10.

A possible limitation of this approach is that the union of TS and Test Set is only a partial sampling of the entire hypercube. The density-closeness function is only a partial representation and hence, looking for the value such that the error is the same as the training error could be wrong. However, this means that it is valid at least for the given Test Set and hence it is possible to define a better TS from this point of view.

4.6 Flight data mining

The following analysis has been already published in [108, 141]. As previously mentioned, a quasi-uniform distribution of flight data inside the hypercube is beneficial to avoid the minimization of the mean error mainly in local area rather than on the entire hypercube. However, due to the size of the dataset both in terms of cardinality of the elements and in terms of the number of variables involved, some tools are needed to verify the performance of the VS during particular flight conditions.

Table 4.4: Mean and standard deviation (between parentheses) values over 100 tests of the similarity functions based on the cardinality of the stationary points for the analytic function dataset

TS	*1*	*2*	*3*	*4*	*5*
L_C	7.26 (5.13)	25.59 (15.24)	18.32 (9.60)	15.63 (9.72)	26.50 (11.53)
A_C	385.48 (294.86)	525.05 (239.68)	748.50 (333.75)	927.23 (417.52)	2307.82 (442.62)
A_{C_2}	115.01 (100.05)	121.02 (74.30)	206.02 (113.83)	264.94 (141.84)	744.35 (150.52)

Table 4.5: Mean and standard deviation (between parentheses) values over 100 tests of the similarity functions based on the cardinality of the stationary points for the analytic function dataset with noise and error injection

TS	*6*	*7*	*8*	*9*	*10*
L_C	19.55 (9.81)	18.40 (8.81)	94.11 (2.10)	77.06 (9.00)	64.43 (11.17)
A_C	2234.98 (411.53)	2224.86 (455.35)	3515.15 (334.87)	2856.19 (398.00)	2692.92 (423.38)
A_{C_2}	728.18 (139.93)	723.87 (154.64)	880.22 (115.26)	830.73 (137.58)	797.64 (143.01)

It can be demonstrated that the lift coefficient C_L of an AC is linearly dependent on the AOA when AOA is small [2]. The slope $C_{L,\alpha}$ and the vertical intercept C_{L0} come from the aerodynamic properties of the airfoils of the wing and of the tail, composed with the aerodynamic of the other bodies such us the fuselage and the landing gear. The most common flight phase can be approximated to stationary flight in linear conditions and, for this reason, it is required to meet good performance of the algorithm in this conditions. At the same time, some so-called trim shots have been planned and conducted during the flight test campaign of 2017 also to determine the $C_L - \alpha$ curve. They consist in several seconds of trimmed flight (constant speed, altitude and null angular rates) corresponding to the longitudinal equilibrium condition, repeated for different AOA. If the mass of the AC is known, it is possible to estimate the lift coefficient and define the linear section of the $C_L - \alpha$ curve. However, ideal longitudinal equilibrium conditions are rarely obtained in flight test, due to residual variation of the flight parameters and the noise acting on them. The analysis is based on dedicated test points and the assumption of validity of the measurement grounds on experience of both pilot and FTE, FTI performance, atmospheric and meteorological reasons.

To ease the data analysis, the method based on flight data mining proposed in

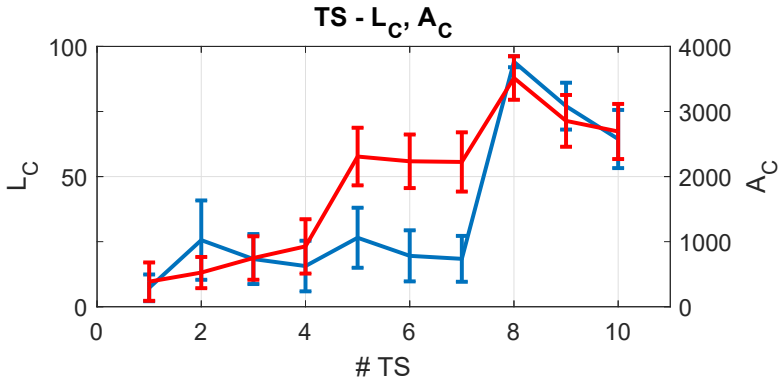


Figure 4.15: Mean and standard deviation values of L_C (blue line) and A_C (red line) among different TS for the analytic function dataset

this section automatically extracts a subset of the dataset corresponding to those particular manoeuvres, in particular the stationary and quasi-stationary flight conditions. Once the set of stationary condition has been extracted, it is possible to test the performance of the AOA/AOS estimator. Moreover, this procedure allows to obtain an higher number of available points with respect to the traditional procedure of planning and analyzing several trim conditions. Eventually, this automatic trim detection algorithm allows to estimate the balance between the entries of the dataset corresponding to dynamic conditions and the one related to stationary conditions. In this dissertation, the ratio has been estimated in 1 stationary condition entry over 100000 dynamic condition entries and this value has been considered high enough to negatively influence the training phase. In [141], a data-reduction method has been applied to reduce the unbalance between dynamic and stationary flight conditions in the TS.

The flight data mining algorithm is based on a down sampling technique followed by a linear multivariable score assignation. Relaxing the constraints on the flight parameters, that is with the assumption that a slight deviation from the ideal value can be accepted, it is possible to define the *quasi*-stationary and *quasi*-symmetric flight conditions. In case of the analysis of Smart-ADAHRS, it is not specifically required a perfect longitudinal equilibrium. Hence, this method provides a set of data corresponding to flight conditions close to the ideal longitudinal equilibrium, without any manual analysis of the dataset and without planning dedicated manoeuvers.

The main procedure is based on the assignation of a value called *score* to a given instant. The maximum score is given to symmetric and stationary equilibrium flight condition so that it is possible to select the desired instant observing this value. Actually, the score is not assigned directly to the n-dimensional vector associated to the flight condition but firstly assigned independently to the various

flight parameters and eventually the final score is obtained from the signal scores (e.g. the average). See Algorithm 1 for details.

At the beginning, the signals are divided in non-overlapping time windows. The length of the time windows is constant and it has been taken as 5 s. The length of the time-window is obtained as a trade-off between finding an actual stationary point and the efficiency of the algorithm. In fact, longer time window will bring to a more reliable evaluation of the flight condition whereas shorter time window will bring to extra points in the final result. Afterwards, some statistics of the signal during each time window are evaluated. Because of this statistical analysis, the original sampling frequency of the signal can have an important influence on the final result. In fact, the number of elements in each subdivision must be sufficiently high such that the sample estimators are statistically valid. In this work, original sampling period is about 0.05 s, corresponding to 100 elements per interval, which is sufficiently high. The original sampling period comes from the sampling frequency of the FTI.

Algorithm 1: Quasi-Stationary and Quasi-Symmetric flight conditions detection algorithm

Data: Flight data
Result: Set of quasi-symmetric and quasi-stationary flight conditions

```
Partitioning in 5 s-long time windows;
foreach time-window do
    foreach signal do
        Evaluation of the statistics of the signal during the time-window;
        Assign a score to each signal;
    end
    FinalScore = average(signal scores at the current time-window);
    if FinalScore > threshold then
        Store as valid time-window
    end
end
```

For each interval, the sample mean, sample standard deviation and the deviation between the minimum and maximum values covered by the signal are calculated. A score is hence assigned to each interval depending on the descriptive values calculated beforehand. The decision on which statistics consider for the score assignment grounds on the type of signal evaluated. In some cases, the score depends on how much the sample mean is close to a given value. For instance, the angular rates must be zero in stationary conditions. In other cases, when the interest falls in observing a constant signal, the sample standard deviation or the maximum deviation drives the score assignment. For instance, the closer the maximum deviation

of the impact pressure is to zero, the higher is the assigned score.

A piecewise linear function has been implemented to assign the score but several other possibilities exist. The standard deviation of the signal during the time-window has been added. In this way, the dispersion of the data around the desired conditions affects the score assigned by the algorithm. This helps to increase the score of the stable sample with respect to sample affected by high variability.

When every signal has been sampled and the signal score has been assigned, the final sample score is evaluated as the mean of the scores among the signals.

Once the scores have been obtained, the selection of the stationary and quasi-stationary points can be carried out. This procedure involves the definition of a minimum score threshold necessary to pick or not a sample, given the scores assigned to each sample. The previous steps are applied equally to each flight test. For this reason, the scores are comparable among different flights. However, it has been observed that a normalization procedure greatly simplifies the decision process. In fact, subtle differences exist between quasi-stationary and quasi-symmetric conditions. To better explain this, it must be recalled that this method can obtain a set of points on the flight envelope slightly relaxing the trim condition constraints. In some cases, very low time derivatives and deviations can be observed. However, the attitude could be too much asymmetric to be neglected. At the same time, a slightly more symmetric flight condition but corresponding to higher deviation on the measures than the previous case can be considered valid. The problem of defining when a flight condition is quasi-symmetric and quasi-stationary is obviously ill-posed. This method converts the problem into the definition of a a threshold on the score, that unfortunately fails to be a metric. It is not yet clear whether the normalization of the score helps or not the solution.

4.6.1 The score assignment process

This subsection shows more details on how a sample of a signal has been related to a scalar value. This step is crucial for the effective functioning of the algorithm. Two different solutions have been applied.

Mathematically speaking, given a signal $x_i = x_i(t)$ it is possible to extract a sample $x_i[n] = E[x_i(t)]$ with $(n-1)\,t_s \leq t < nt_s$. Various statistics θ of the sample $x_i[n]$ can be measured. Eventually, the signal score s is assigned to the n-th sample based on the corresponding $\theta(n)$ as in the following relation:

$$s = s(\theta(n)) = s(n) \tag{4.15}$$

The first approach on the functional form for s is piecewise linear. The *triangle function* can be generally defined as in Eq. 4.16. An example can be seen in Figure 4.16.

$$\Lambda(\theta) = \begin{cases} 1 - \frac{|\theta|}{\theta_{th}} & \text{for } |\theta| < \theta_{th} \\ 0 & \text{for } |\theta| \geq \theta_{th} \end{cases} \tag{4.16}$$

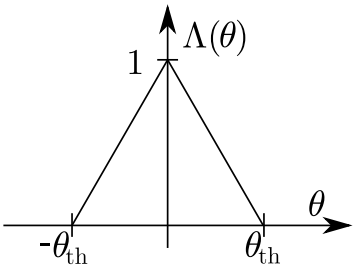


Figure 4.16: Example of triangle function $\Lambda = \Lambda(\theta)$

Setting $s = \Lambda(\theta(x[n]))$ a first evaluation of the sample score can be obtained. For instance, if the q_c signal is considered, the maximum acceptable Δq_c during 5 s can be set to 100 Pa. In this case, if the maximum deviation is higher than Δq_c then a null score will be assigned. At the same time, if the maximum deviation is between 0 Pa and 100 Pa, then a proportional score will be assigned to that given sample.

However, it has been found that extending the previous analysis to more than one single statistic brings to higher coefficient of determination with respect to using only one statistic. To this aim, Eq. 4.16 can be modified considering similarities to a triangulation problem.

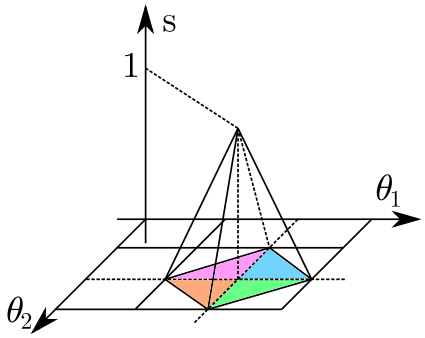


Figure 4.17: Generalization of the score function

Given $\theta_{1,1}, \theta_{1,2}, \theta_{2,1}, \theta_{2,2} \in \mathbb{R}$ with $\theta_{1,1} < \theta_{1,2}$ and $\theta_{2,1} < \theta_{2,2}$ it is possible to identify four regions as reported in Figure 4.18. For convenience, each region is identified by the same index of one of the two border vertices.

Let $P \in \mathbb{R}^2$ a point belonging to the $O\theta_1\theta_2$ space, it is possible to write

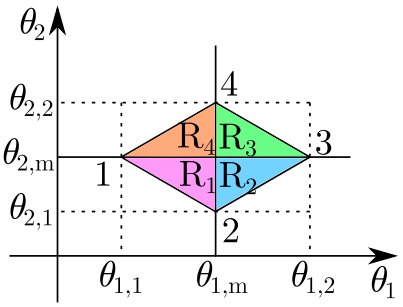


Figure 4.18: Definition of the regions in a generic $O\theta_1\theta_2$ plane

$$P \in \mathcal{R}_1 \Leftrightarrow \begin{cases} \theta_2 > \theta_{2,s_1}(\theta_1) \\ \theta_1 < \theta_{1,m} \\ \theta_2 < \theta_{2,m} \end{cases}, \quad P \in \mathcal{R}_2 \Leftrightarrow \begin{cases} \theta_1 \geq \theta_{1,m} \\ \theta_2 < \theta_{2,m} \\ \theta_2 > \theta_{2,s_2}(\theta_1) \end{cases}, \tag{4.17}$$

$$P \in \mathcal{R}_3 \Leftrightarrow \begin{cases} \theta_1 \geq \theta_{1,m} \\ \theta_2 \geq \theta_{2,m} \\ \theta_2 < \theta_{2,s_3}(\theta_1) \end{cases}, \quad P \in \mathcal{R}_4 \Leftrightarrow \begin{cases} \theta_1 < \theta_{1,m} \\ \theta_2 \geq \theta_{2,m} \\ \theta_2 < \theta_{2,s_4}(\theta_1) \end{cases} \tag{4.18}$$

where $s_{1,2,3,4}$ stand for the 4 straight lines defining the border.

Once the r-th region in which P belongs to has been found, it is possible to write Eq. 4.19 which describes each planar face of the pyramid as follows:

$$\nabla s_r \cdot (\boldsymbol{\theta} - \boldsymbol{\theta}_r) = s_r(\boldsymbol{\theta}) - s_r(\boldsymbol{\theta}_r) \tag{4.19}$$

where $\boldsymbol{\theta}$ represents the vector of coordinates of the point P. Moreover,

$$\nabla s_r \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_r) = s_r(\boldsymbol{\theta}_m) - s_r(\boldsymbol{\theta}_r) = 1 \tag{4.20}$$

$$\nabla s_r \cdot (\boldsymbol{\theta}_{r+1} - \boldsymbol{\theta}_r) = s_r(\boldsymbol{\theta}_{r+1}) - s_r(\boldsymbol{\theta}_r) = 0 \tag{4.21}$$

For simplicity, Eq. 4.21 contains a little abuse of notation. In fact, $r+1$ becomes 1 for $r = 4$.

The following linear system can be written:

$$\begin{bmatrix} \theta_{1,m} - \theta_{1,r} & \theta_{2,m} - \theta_{2,r} \\ \theta_{1,r+1} - \theta_{1,r} & \theta_{2,r+1} - \theta_{2,r} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial s_r}{\partial \theta_1} \\ \dfrac{\partial s_r}{\partial \theta_2} \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{4.22}$$

which brings to Eq. 4.19,

$$\frac{\partial s_r}{\partial \theta_1} = \frac{\theta_{2,r+1} - \theta_{2,r}}{\det \mathbf{G}} \tag{4.23}$$

$$\frac{\partial s_r}{\partial \theta_2} = -\frac{\theta_{1,r+1} - \theta_{1,r}}{\det \mathbf{G}} \tag{4.24}$$

representing the two partial derivatives of the score function s with respect to θ_1 and θ_2, where $\mathbf{G}$ is the matrix at the LHS of Eq. 4.22.

Eventually, the score assigned to any sample $x_i[n]$ is evaluated as follows:

$$s(x_i[n]) = \begin{cases} \nabla s_r(\boldsymbol{\theta}(x_i[n]) - \boldsymbol{\theta}_r) & \text{if } P \in \mathcal{R}_r \text{ with } r \in \{1,2,3,4\} \\ 0 & \text{if } P \notin \mathcal{R}_r \text{ with } r \in \{1,2,3,4\} \end{cases} \tag{4.25}$$

With this formulation, two statistics can be considered. In this paper, the standard deviation of the sample is always applied as second statistic. This help to increase the score for stable samples. Table 4.6 shows the values used in this dissertation.

Table 4.6: List of signals and corresponding statistics

Signal	*First statistic*	*Value*	*Second statistic (Standard deviation) value*
Angular rate	Sample mean	$0.01\,\mathrm{rad\,s^{-1}}$	$0.005\,\mathrm{rad\,s^{-1}}$
Altitude	Max deviation	$1\,\mathrm{m}$	$0.5\,\mathrm{m}$
Vertical speed	Sample mean	$1\,\mathrm{m\,s^{-1}}$	$0.5\,\mathrm{m\,s^{-1}}$
Acceleration	Sample mean	$0.05\,\mathrm{g}$	$0.025\,\mathrm{g}$
Impact pressure	Max deviation	$100\,\mathrm{Pa}$	$50\,\mathrm{Pa}$
Pitch angle	Max deviation	$2°$	$1°$
Roll angle	Sample mean	$1°$	$0.5°$
Yaw angle	Max deviation	$1°$	$0.5°$

4.6.2 Results for G70 data

This section shows the results obtained on the flight dataset by the proposed score-assignment method.

To evaluate the capability of the method, the $C_L - \alpha$ plot is obtained. Because no engine measurement is available on-board, a fuel consumption linearly descreasing with time has been assumed. In this way, it is possible to evaluate the lift coefficient as $C_L = \frac{mg}{q_c S}$, where m is the aircraft mass, g is the gravitational acceleration, q_c

is the impact pressure and S is the wing surface area. The samples resulted to be organized on a straight line, with slope $C_{L\alpha}$ and zero α_0 close to the independent analysis previously conducted by [127]. To avoid nonlinear effects, only the samples with $\alpha < 10°$ have been taken in consideration. A maximum error of $-0.06\,\%$ on $C_{L\alpha}$ and of $-2.27\,\%$ on α_0 has been obtained. The quality of the linear regression has been measured with the coefficient of determination.

Actually, the effects of $C_{L\delta_e}$ and asymmetric flight condition should be accounted in a post-processing correction. Shortly, according to [142], the slope obtained by the regression should be properly called $C^*_{L\alpha}$. However, the difference between $C^*_{L\alpha}$ and $C_{L\alpha}$ has been here dropped for sake of clarity.

It is interesting to note that the obtained samples organize on three straight lines. In fact, some of them corresponds to flaps down in Take-Off (about 14°) and Landing (about 36°) conditions. Because the flap angle deflection, for leading edge plain flap, is equivalent to offset the null lift direction with an angle proportional to the δ_f, it is possible to identify two analytic values of $\Delta\alpha$ to compare our results. In fact

$$\Delta\alpha = \tau\delta_f \tag{4.26}$$

In this work, to assess the accuracy of the method, the effect given by the flap on α_0 has been evaluated. Considering a 2-dimensional $\tau_{2D} = 0.5$ [143] and a surface ratio $S_f/S = 0.66$, the 3-dimensional value for $\frac{\partial\alpha}{\partial\delta_f}$ becomes $\tau_{3D} = 0.33$. Unfortunately, this value is valid up to 20° of flap deflection. The κ' parameter [144] has been used to extend the evaluation beyond this limit, leading to the following more general formulation:

$$\tau_{3D} = \tau_{2D} * \kappa' * \frac{S_f}{S} \tag{4.27}$$

Therefore, the following values are used in this paper $\tau_{3D,14°} = 0.33$, $\tau_{3D,36°} = 0.27$.

Looking at the Figure 4.19 it is possible to identify all the sample points obtained with $s(i) \geq 0.667$ and $\delta_f = 0°$.

Figure 4.20 shows the samples corresponding to flaps in Takeoff (TO) condition, whereas Figure 4.21 collects the sample obtained in Landing (LND) condition.

It is important to notice that no flight control surface data has been directly applied into the score assignment. In fact, the flap deflection angle δ_f has been used only in post-processing to distinguish Figure 4.19–Figure 4.21, in order to clearly identify the three linear regression functions. Figure 4.22 gives a global view on the $C_L - \alpha$ curves found by the algorithm.

The comparison of the regression with the independent analysis conducted by Battaini [127] confirms the validity of the method. Table 4.7 collects the results obtained. A difference about $-0.06\,\%$ has been obtained on $C_{L\alpha}$ with respect to

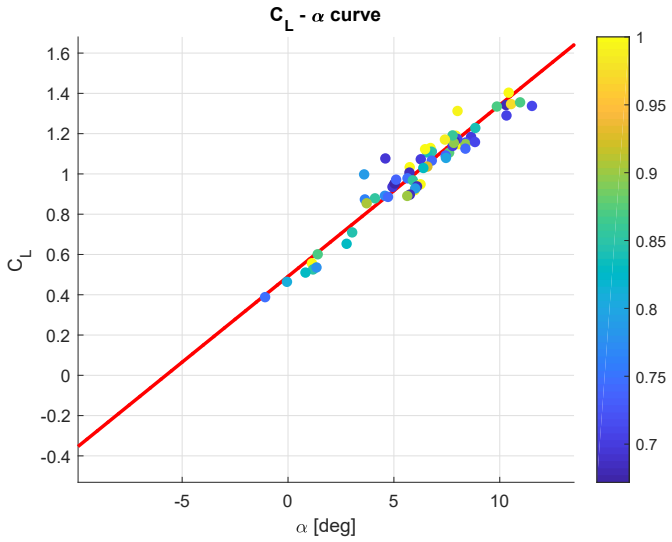


Figure 4.19: $C_L - \alpha$ plot, clean condition, $\alpha_0 = -5.7673°$, $C_{L,\alpha} = 0.085\,136\,°^{-1} = 4.8779\,\text{rad}^{-1}$, $R^2 = 0.95812$. Scatter plot color corresponds to the sample score.

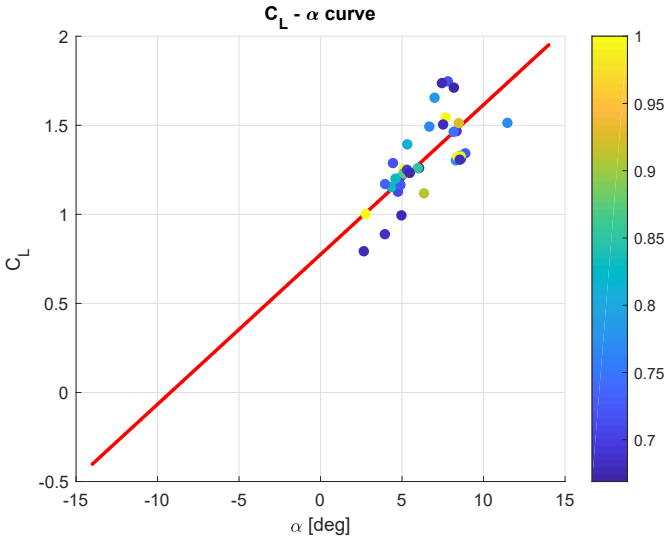


Figure 4.20: $C_L - \alpha$ plot, TO condition, $\alpha_0 = -9.2128°$, $C_{L,\alpha} = 0.084\,072\,°^{-1} = 4.817\,\text{rad}^{-1}$, $R^2 = 0.70726$. Scatter plot color corresponds to the sample score.

the manual derivation, neglecting the effect of the elevator. For what concern the α_0, the difference is about $-2.27\,\%$.

To evaluate the accuracy of the regression in TO and LND conditions, a manual estimation has been carried out. Hence, although the coefficient of determination of the linear regression is lower than 0.9 for the TO and LND conditions, the obtained results can be considered a good demonstration of the trim identification method,

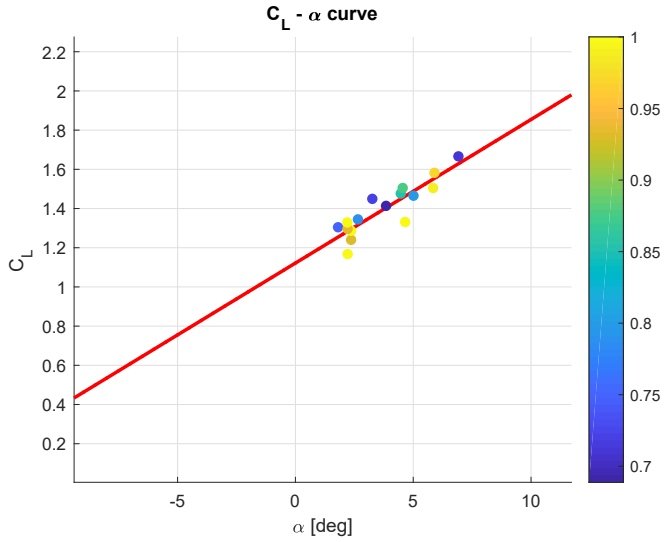


Figure 4.21: $C_L - \alpha$ plot, LND condition, $\alpha_0 = -15.3087°$, $C_{L,\alpha} = 0.073\,267\,°^{-1} = 4.1979\,\mathrm{rad}^{-1}$, $R^2 = 0.88861$. Scatter plot color corresponds to the sample score.

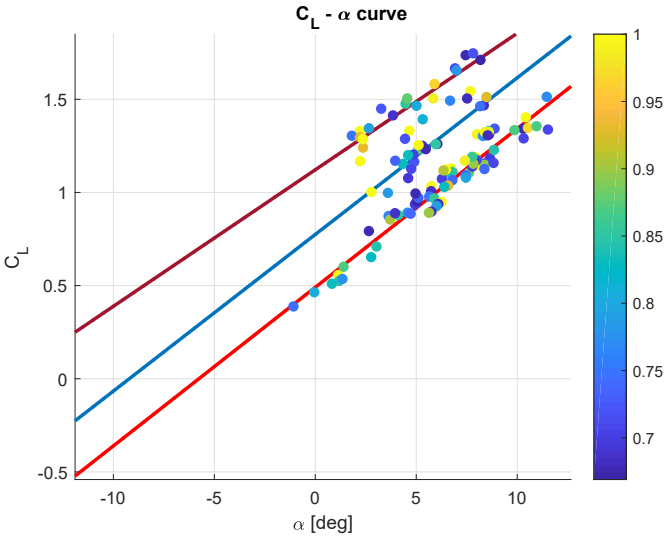


Figure 4.22: $C_L - \alpha$ plot, global view of the three regression lines

due to the accordance with the analytical estimation.

4.7 Re-training and selection

This section describes a procedure able to increase the independence from the initial condition of the training operation. In fact, the training operation is actually

Table 4.7: Result on the estimation of $C_{L,\alpha}$ and α_0

Source	$C_{L,\alpha}$	α_0	R^2
Battaini [127]	4.881	$-5.9015\,\text{deg}$	-
Regression, clean	4.8779	$-5.7673\,\text{deg}$	0.95812
Regression, TO	4.817	$-9.2128\,\text{deg}$	0.70726
Regression, LND	4.1979	$-15.3087\,\text{deg}$	0.88861

an iterative procedure that tries to solve the non-convex problem of identifying the weights able to minimize the value of the error. If the initial condition changes, the path followed by the solution can change depending on the regularity of the error surface. Repeating the training operation using the same dataset but re-initializing the weights of the NN results in different sets of weights. Given this procedure and defining a criterion, a NN can be chosen among the various obtained. In this dissertation, the minimum global error obtained both in training and test is applied.

A second observation can be made. The final training error can change a lot among the various solutions or not. It is worth noticing that training with simulated data results in the first case, whereas using flight data brings to the second one. A more homogeneity in the final training error gives an indication of convergence of the training operation.

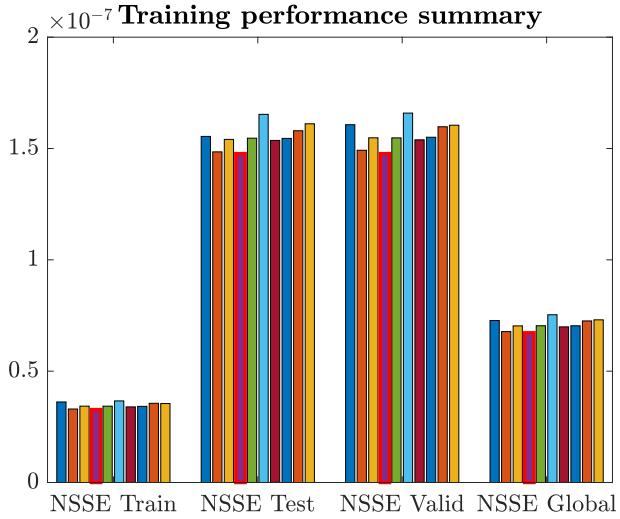


Figure 4.23: Repeated training and selection of the best NN (The red border identifies the selected NN)

4.8 Manoeuver-based Cross-Validation

The definition of which manoeuver must be conducted in order to minimize the final uncertainty of the virtual sensor is still an open question of this field. During the flight test campaign, several flight segments or flight manoeuvers have been flown. Moreover, some of them have been repeated at a different flight condition (e.g. airspeed) or even at the same one. Thanks to the availability of these data and its internal structure, it could be interesting to carefully analyze which manoeuver acts positively on the uncertainty of the synthetic estimation. In this way, it could be suggested to the designer of a synthetic sensor a guideline of flight campaign.

However, the amount of data available is too large to try every possible combination of manoeuver. From a statistical point of view, it is possible to see this situation as the *model selection problem* [121]. As mentioned in Sec. 2.4, it could be applied a CV method.

Taking advantage of the CV methods, it is possible to define which subset of data brings to the best estimation of AOA. Classic CV well applies for homogeneous stationary data, each point is considered independently without the consideration that the observation comes from the response of a dynamic system.

This section proposes a version of the k-Fold CV method. If the various folds are obtained from consideration on the flight condition instead of using a statistical partitioning method, the subset that results in the minimum error would be composed by a set of flight conditions. Hence, dropping some assumptions of independence of the partitions, the result of the model selection algorithm can be analyzed from a aeronautical point of view. It must be verified that the number of points is at least statistically valid. For this reason, to the analysis it is necessary to add the ratio of the size of the Test Set with respect to the TS.

The result of the Quasi-Stationary detection algorithm to the dataset can be used to automatically split a huge dataset coming from flight test. The consideration that each manoeuvre is conducted after the determination of a trim shot can be used to partition the flight test database in smaller subsets, reducing the manual search for trim points. This method is called Manoeuver-based CV.

Once the flight test database has been partitioned, the classical k-Fold CV procedure can be carried out. In this dissertation, the value reported is calculated as the ratio between the deviation from the average NSSE and the average itself. This allows to immediately compare the results.

Figure 4.24 shows an example of the error comparison among various tests. In that case, the selection of the 5th subset as TS induces a reduction of the 81.4 % on the NSSE with respect to the average. This important reduction is an example of how much the selection of the TS can affect the final uncertainty of the VS.

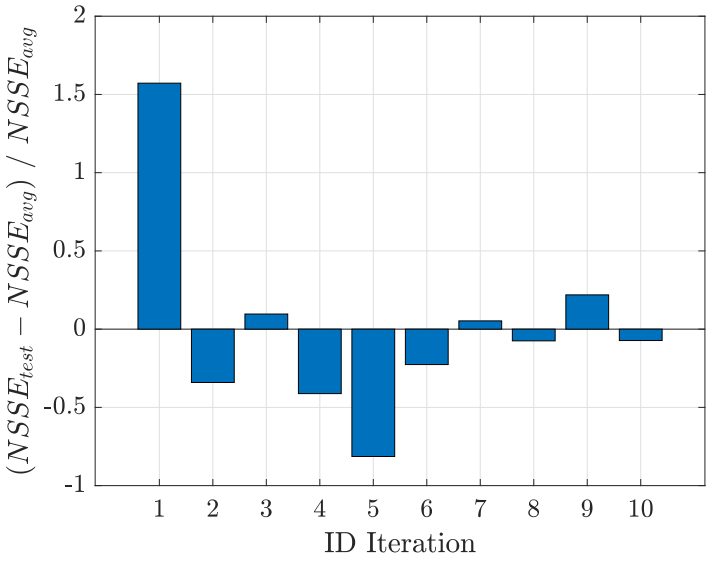


Figure 4.24: Example of 10-fold CV result.

4.9 Feature map for flight data reduction

One of the open question in the synthetic sensor field is the guarantee that the estimator does not show peak error in unknown conditions. Although, this obviously greatly depends on the architecture of the sensor, a common procedure can help finding issues in the estimation.

Unfortunately, the aircraft is a coupled multivariable system which size does not allow to analyze each variable singularly, even less so check every flight condition. The data visualization community has researched this kind of problem for the last 20 years and several methods exist. Among those method, the feature mapping is a possibility.

This method is a possible data reduction based on feature mapping applicable for the aeronautical engineering field. Feature mapping means to reduce the size of the problem to a lower number of variables so that it is possible to represent data on a chart. In this way, some information are surely lost but the data reduction policy can be chosen in way of obtaining still rich information from data.

Defining X and Y as follows:

$$\begin{aligned} X &= \overline{q} + \overline{u} + \frac{\sqrt{2}}{2}\overline{p} + \frac{\sqrt{2}}{2}\overline{r} + \frac{\sqrt{2}}{2}\overline{v} \\ Y &= \overline{\beta} + \overline{\phi} + \frac{\sqrt{2}}{2}\overline{p} + \frac{\sqrt{2}}{2}\overline{r} + \frac{\sqrt{2}}{2}\overline{v} \end{aligned} \tag{4.28}$$

where the line above the symbols represents the non-dimensional quantity. The original aircraft state vector is reduced to size 2. On this feature map it is hence

possible to plot the estimation error obtaining a 2D chart as in Figure 4.25. This chart shows how the error is distributed along the various flight conditions. The $X = 0$ coincides with the *Quasi-Stationary Flight Conditions*. The origin of the chart represents the symmetric and quasi-stationary flight conditions as defined in [108], whereas in case of asymmetric flight, maintaining the quasi-stationarity, the point moves along the $X = 0$ axis. The abscissa of a point is related to the presence of a pitch rate or an acceleration along the X-Body axis. Hence, longitudinal non-stationary flight condition are encountered moving along the X-axis whereas the asymmetric fight is shown along the Y-axis. Mixed dynamics brings the points to move away from both the axes.

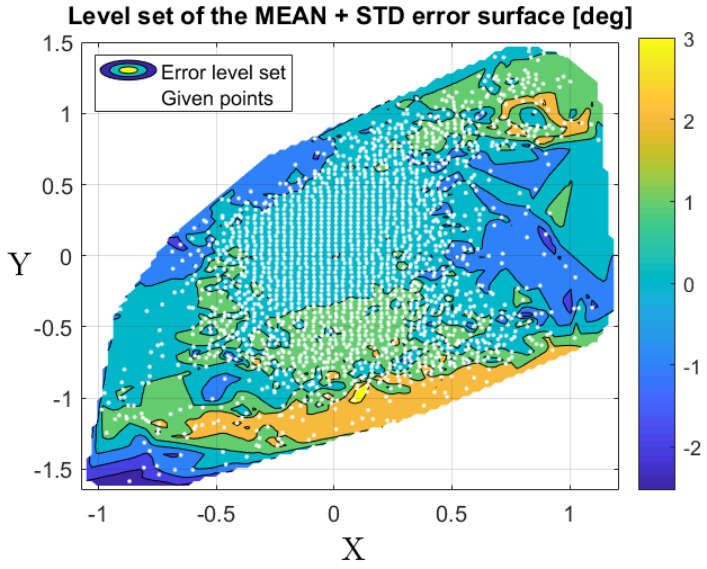


Figure 4.25: Feature map example

4.10 Sensitivity Analysis and Uncertainty estimation

This section describes a method that can be applied to VS in order to estimate its uncertainty. As anticipated in Chap. 1, in this field the value reported is more related to an error bound than an actual statistical analysis. In that case, the given value is more subjective and less repeatable. However, there are common procedure already studied in metrology that can be applied. Following these methods the estimation will be repeatable and objective.

In this section, two main techniques are described for uncertainty propagation: the first one is based on a truncated Taylor series using the theoretical results of

Chap. 2, while the second is the study of the nonlinear uncertainty propagation. It will be clear that the first technique is not able to provide an acceptable value due to the highly nonlinearity of the MLP.

It is important to conduct these analysis on the entire dataset and not only on the test set. In fact, once the network has been trained, the regions corresponding to the TS are operative regions for the VS. A common error in this field is to report data only for the Test Set, forgetting that this partitioning is only needed in order to verify the generalization of the NN, but the VS can work also with values that have been selected for the TS.

4.10.1 Linearization of the MLP

Given the covariance matrix $\mathbf{\Sigma}^x$ of the variables in input to the MLP, it is possible to write the Eq. 4.29.

$$\begin{aligned} y &= y\left(\mathbf{x}_0\right) + \mathbf{J}\Delta\mathbf{x} + \mathcal{O}\left(\Delta\mathbf{x}^2\right) \\ \mathbf{\Sigma}^y &= \mathbf{J}\mathbf{\Sigma}^x\mathbf{J}^T \end{aligned} \tag{4.29}$$

However, the truncated Taylor series is not a good representation of the MLP due to the high nonlinearity of the function. Hence, the results given by this analysis can be erroneous.

4.10.2 Nonlinear analysis

To estimate the uncertainty of the sensor fully considering its nonlinearity, the function is tested with a Monte Carlo simulation using a Gaussian distribution on the input variables and analysing the distribution of the output around the nominal values. Hence, for any point of the dataset it can be written as follows:

$$\mathbf{x}_{i,MC} = \mathbf{x}_i + \Delta\mathbf{x} \text{ where } \Delta\mathbf{x} \sim N\left(\mathbf{0}, \mathbf{\Sigma}^x\right) \tag{4.30}$$

The number of points generated around the nominal point $\mathbf{x}_i$ is of great importance because it must be sufficiently high to allow a statistical analysis but the computational cost can increase. In fact, this analysis must be conducted on the entire dataset and the final number of points will be the size of the dataset times the number of generated points around each one. In this dissertation, 100 points have been generated for each entry of the dataset.

4.10.3 Presentation of the results

As typically requested by the aeronautical industry, the uncertainties must be reported on a chart $O\alpha\beta$. In this way, the uncertainty is specified for the actual

values of the aerodynamic angles. As mentioned before, the evaluation of the uncertainty must be executed for the entire available dataset and hence the heatmaps will cover the entire envelope of AOA/AOS. However, any pair (α, β) can represent several flight conditions. For sake of clarity, it is not specified, for instance, if the flight is stationary or there is a linear or angular acceleration. For this reason, the mean and standard deviation value of each resultant distribution are calculated and then grouped by bins on the α, β chart. The resultant distribution of expected values and sample standard deviations will be analyzed statistically, if there are enough points (here corresponding to different flight instants) to define a distribution. In this way, it is possible to study the expected value of the standard deviation in each bin and therefore associate an uncertainty of the estimation to the particular bin. An example is shown in Figure 4.26.

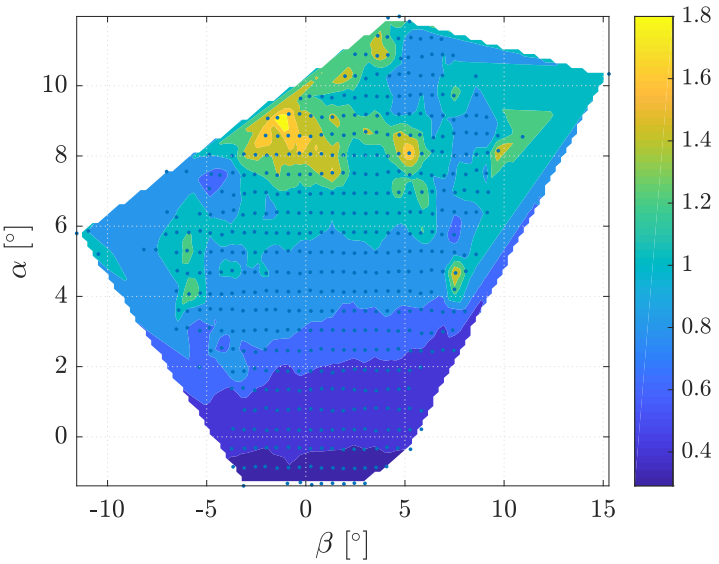


Figure 4.26: Example of uncertainty map. Values of the colorbar are in [°]

The dispersion of the sample standard deviation is also of great importance. In fact, if several fight conditions are grouped inside the same (α, β) bin, it is interesting to study if the flight condition has an effect on the final uncertainty. Statistically speaking, It must be checked if the sample standard deviation at a given pair (α, β) is biased from the flight mechanics point of view. Finally, the obtained charts should be compared with the estimation error, to try to understand if the classical approach gives at least an estimation of the uncertainty.

Due to the structure of the estimator, this analysis is repeated twice. In fact, although the final uncertainty is metrologically important, the effect of the NN on the initial estimation can greatly help the design of the VS.

Chapter 5

Data augmentation

Some of the analysis shown here, mostly regarding the modeling of the sensors, have already been published in [107].

5.1 The reasons behind data augmentation

The case study of this dissertation is the Smart-ADAHRS algorithm, which kernel is an MLP. During the design of an MLP, the selection of the data applied for training is crucial. Although several works focuses on the architecture of the NN, the availability of data both for training and testing the NN allows to improve performance or, at least, to gain confidence on the results.

Based on the author's experience, due to practical reasons related to the flight tests, data-sets coming from actual flights tend to share some similarities. Figure 5.1 shows a typical coverage on the AOA/AOS envelope. A common shape is a cross, due to the difficulty of flying with high AOA and at the same time high AOS. Also, the envelope might be covered asymmetrically, due to intrinsic asymmetry of the AC, i.e. effects due to the propeller. Another common problem concerns negative AOA.

In addition, the same concept of incomplete data-set can be applied to static and dynamic flight conditions. Using the algorithm published in [108], a set of quasi-stationary flight conditions is extracted from the dataset. The extracted subset can be seen in Figure 5.2.

The analysis of the corresponding output of Smart-ADAHRS in these conditions showed an higher uncertainty in stationary conditions than in unsteady conditions. This lack of accuracy has been associated to an unbalanced training operation, between stationary and dynamic flight conditions. As common practice to assess the AC performance, several trim shots have been scheduled, from which two or three points are applied to fit linear models. Several aspects affect the quality of these points from the air turbulence to the discretization of the flight control surface

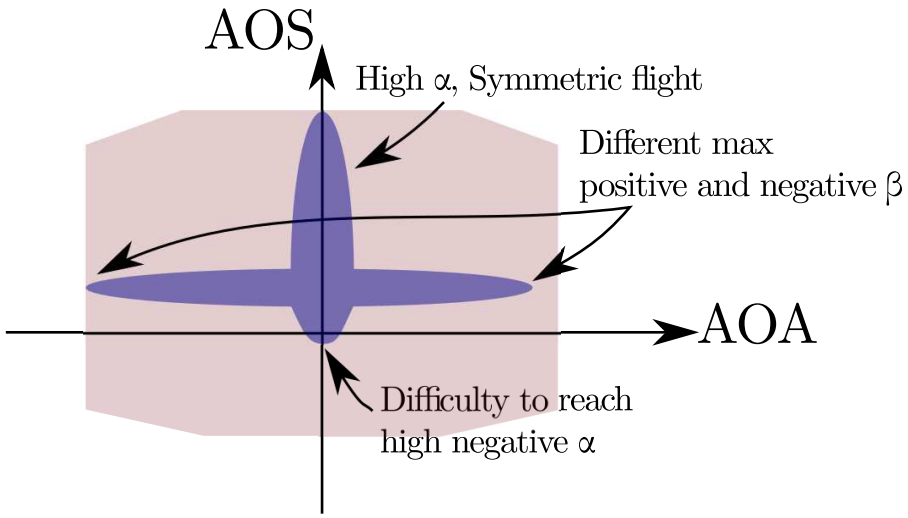


Figure 5.1: Typical cross-shaped distribution of available flight data (blue) on the required AOA/AOS envelope (red)

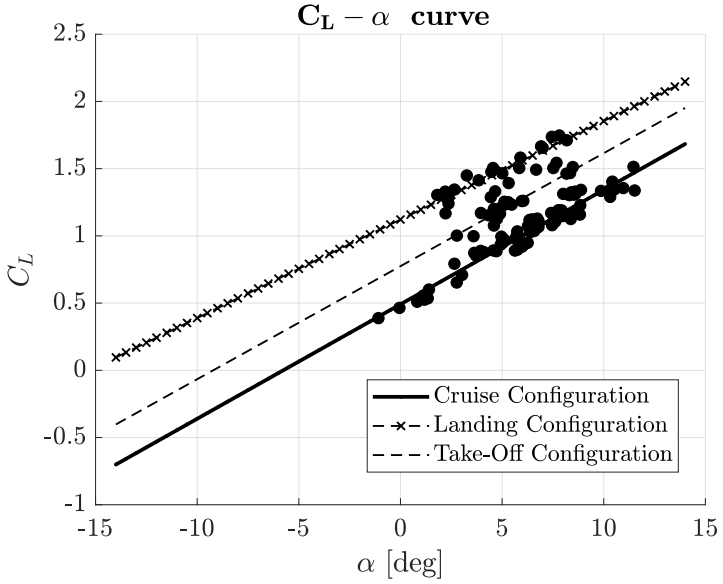


Figure 5.2: Distribution of measured quasi-stationary and quasi-symmetrical flight conditions

position. However, these points result very close in the phase space, meaning that their Euclidean distance is low with respect to the maximum distance among all the recorded entries. In short, the trim shots collected during flight tests are very similar, low dispersion of airspeed, attitude and flight control surface positions.

Although it is difficult to plan additional flight tests to fill the red area in Figure 5.1 or to add points in particular dynamic or static conditions, it might be

easier to simulate or artificially generate those points and add them to the data-set. In this thesis, the cross-shaped distribution has been solved with additional requirements on the manoeuvers to be flown in the flight test campaign of 2017. However, the same re-planning was not enough to reduce the strong unbalance between stationary and dynamic entries. To obtain a better balance between dynamic and stationary entries, the data augmentation techniques is applied.

Due to the particular application of this dissertation, there are two main ways to conduct the data augmentation method. The first one takes advantage from the fact that the data-set is made by the input, output and state variables of an AC. Hence, the first method is to conduct a flight simulation. The second one is using a very recent ML technique named GAN. In this case, a particular ML paradigm is used to attempt to generate data from a multivariate distribution. These two methods are described in this chapter together with their typical drawbacks and results.

As a remark, it is questionable to artificially add data to the training set of a NN with the aim to improve the training, using data coming from a flight identification based on the same set of data. In fact, as it will be clearer at the end of this chapter, the second solution is still not valid to improve the training operation. However, it must be recalled that a flight simulator is designed based on an additional set of data, that is the geometrical and technical data of the AC itself. This add a definitely new set of information about the AC that is only partially self-contained inside the data-set. The idea is to take advantage of the Computer-Aided Design (CAD) drawings, engine and propeller datasheets, certification reports and studies of the AC to complete the information coming from the raw flight test recordings.

5.2 Flight simulation

From a general point of view, training an MLP means to find a nonlinear map between the input and the target entries that can reduce, or even minimize at the limit, some global metric based on the error of estimation. In this framework, the only possibility is to apply a huge set of data to train, validate and test the MLP. However, it must be pointed out that in this particular case, data belongs to a particular class of generation model, that is an AC. To better focus on this, please consider a classical ML application as it could be the image classification. In this case, the classifier is trying to find a map that is, up to now, hidden from a mathematical point of view. The presence of a dog or a cat in a picture is something that is difficult to model in a deterministic way, although the recent findings allow to think that an actual function *could* exist. The application of this dissertation is totally different, because the variables are strictly related among each other by the aircraft kinematics and dynamics. These relationships have been already studied in engineering and they allow, although not perfectly, to match with measurements

conducted in flight. Briefly, it is possible to write an implicit equation as in Eq. 5.1 that must be verified by the AC data.

$$f\left(t, \boldsymbol{x}, \dot{\boldsymbol{x}}\right) = 0. \tag{5.1}$$

The idea is to generate a set of data that verifies Eq. 5.1 to improve the coverage of the training hypercube in those regions in which the TS is poorly defined. This can be done with a flight simulator.

5.2.1 Structure of the Flight Simulator

The AC has been modeled in MATLAB®/Simulink implementing a nonlinear Ordinary Differential Equation (ODE). The simulator is based on the Flight Dynamics and Control toolbox (FDC) toolbox [145]. The aerodynamic database has been determined in two phases. At the beginning, preliminary results have been obtained from the DIGITAL DATCOM [146]. Secondly, the DATCOM methods from [147] have been applied to manually tune some of the aerodinamic derivatives. Since the control derivatives related to the rudder are not available on the DIGITAL DATCOM, also [144] has been used to evaluated them. The result is the entire aerodynamic database of the aircraft which values are given in terms of α. Figure 5.3 shows the comparison on the $C_L - \alpha$ chart between the simulated and real quasi-stationary conditions.

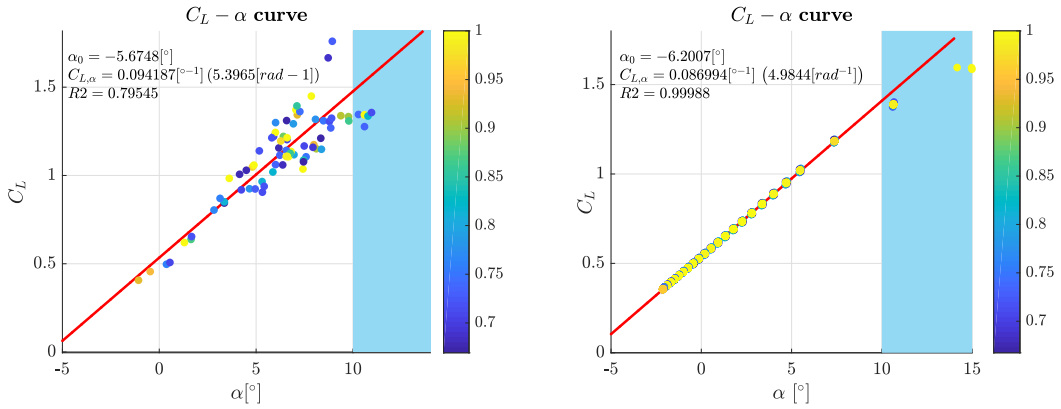


Figure 5.3: Comparison between simulated and flight test quasi-stationary conditions on the $C_L - \alpha$ plot

The engine Rotax® 912ULS has been modeled according to the performance declared on the datasheet published by the producer [148]. Unfortunately, some uncertainties exist. The geometry of the propeller, an Helix H50F 1.75 R-S-19-2, is unknown and classified by the producer. Actually, the Rotax® 912ULS can be coupled with a series of propeller, two-blade or three-blade, and this add an uncertainty on the matching between flight simulation and flight test data. However,

some results can be found from a series of Master Thesis from Politecnico di Milano [127, 126].

The validation of the flight simulator is a difficult task. The main reason derives from the fact that the throttle has not been logged during flight tests. This makes impossible a perfect matching in a direct comparison feeding the simulator with the same input recorded in flight.

However, it is possible to compare the parameters typically considered also by the aeronautical regulations, that is the stick-fixed dynamical modes of the aircraft. It must be mentioned that on this kind of vehicle, equipped with the available FTI, some of the test cannot be properly conducted. For instance, the proper exitation of the elevator to test the Short-Period mode and the one to the rudder to test the Dutch-Roll mode have not been applied. Moreover, the test were mainly conducted to verify compliance to the LFT-UL regulation and it simpy prescribes that any *short period* mode must be heavily dumped. A comparison of the natural frequencies, periods and time constants obtained from flight test in [126] and from linearization of the AC model can be seen in Table 5.1. The damping values that should be added to the oscillatory modes have not been compared because they were not available in [126].

Table 5.1: Comparison of the stick-fixed aircraft modes

Mode	*Simulation*	*Flight Test*
Phugoid	$0.36\,\mathrm{rad\,s^{-1}}$, period 18.02 s	$0.35\,\mathrm{rad\,s^{-1}}$, period 18 s
Short-Period	$3.11\,\mathrm{rad\,s^{-1}}$, period 2.9 s	$4.18\,\mathrm{rad\,s^{-1}}$, period 1.5 s
Dutch-Roll	$1.40\,\mathrm{rad\,s^{-1}}$, period 4.49 s	Not observed
Spiral	36 s	Not conducted
Roll mode	0.17 s	Not conducted

To generate a comprehensive dataset it could be possible to design a set of autopilots and to simulate a set of manoeuvers. Moreover, the sensor noise can be simulated as it has been done during the analysis shown in Chap. 1. However, a first test has been conducted simply perturbing a set of trim shots and propagating the AC dynamics. First of all, the minimization of a scalar cost function J based on the time-derivative of the state vector brought to a series of 25 equilibrium points (the so-called trim shots). For each one, 19 perturbations have been applied and 100 s of trajectory have been simulated. The perturbations consist in altering the airspeed, α and β singularly, and other mixed perturbations of both airspeed, α and β. At the end of the data generation phase, a dataset with more than 475k entries has been generated. Table 5.2 collects the main setup of the simulation.

Figure 5.4 shows the distribution of the simulated data on the AOA/AOS envelope. It is clear from Figure 5.4 that the stationary condition is covered by the simulated data.

Table 5.2: Simulation setup

Parameter	*Value*
# of stationary points	25
# of simulated trajectories	475
TAS range	from $19\,\mathrm{m\,s^{-1}}$ to $55\,\mathrm{m\,s^{-1}}$
AOA range	from $-7°$ to $15.6°$
AOS range	from $-5.2°$ to $5.6°$
ΔV perturbation	up to $5\,\mathrm{m\,s^{-1}}$
$\Delta\alpha$ perturbation	up to $5°$
$\Delta\beta$ perturbation	up to $5°$
Simulation time	$100\,\mathrm{s}$

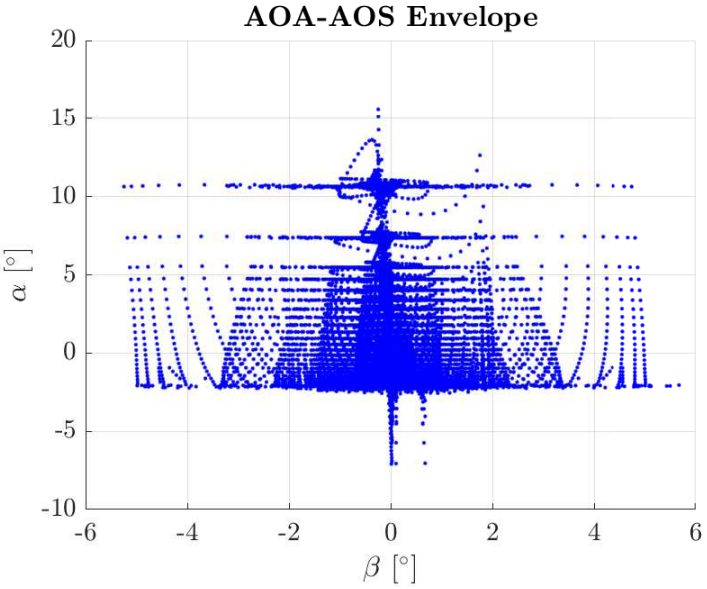


Figure 5.4: Coverage on the AOA/AOS envelope of the simulated data

5.3 Build a generative model to conduct data augmentation

The flight simulation is an important tool, able to strongly reduce the cost of the experiment in flight. However, to build a model is not an easy task. The knowledge on the geometry of the aircraft is not always perfect. The formulations of the aerodynamic database are based on empirical analysis and the validation of the model itself requires a flight test campaign. In this section, another possibility is studied, that should be able to augment the dataset in the stationary region.

In 2014, Goodfellow et al. presented a popular paper proposing the GAN. Recently, GANs are gaining a lot of attention in the field of ML and they have been applied in a lot of fields. This paradigm showed compelling results in distribution fitting in high dimensional problems, such as image creation. At the same time, GANs are not subjected to the overfitting problem, because the output of the generative model is not directly compared to the target [149].

While a discriminative model is trained to fit the posterior PDF directly, a generative model is implemented to solve the more general problem of finding the joint PDF. Generally, it is preferred to use discriminative models over generative ones. However, in [150] Ng and Jordan showed that two different regimes exist, depending on the training set size. The size of this TS is small (about 100 entries, corresponding to the number of the extracted stationary conditions) so a generative model should reach its asymptotic error faster than a discriminative one.

Training a GAN requires particular attention due to its peculiar structure. For example, non-convergence can lead the GAN to underfit [151]. Several techniques have been proposed to improve GAN training [152, 153, 154, 155].

5.3.1 Introduction to Generative Adversarial Networks

A GAN is composed by two algorithms or players respectively called the Generator and the Discriminator. The training operation, or better the update operation on the weights is not directly conducted on the estimation error of the Generator with respect to some target. It is indeed the competition between the two players to lead the training procedure. The Discriminator applies logistic regression and it classifies if the input vector comes from the real data distribution or it is a fake vector. The Generator aims to generate realistic vectors such that the Discriminator is not able to classify them as fake. Hence, this learning process is not an optimization problem, but it is instead a minimax game and the solution is a Nash equilibrium. It is convenient to introduce a value function $V(G, D)$ [149].

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (5.2)$$

GAN can be formally described as structured probabilistic models with latent variables $\mathbf{z}$ and observed variables $\mathbf{x}$ [153].

Actually, the Generator and the Discriminator are simply two functions G and D depending on the some parameters $\theta^{(G)}$ and $\theta^{(D)}$. Usually, they are implemented as MLPs. The domain of G is the so-called *latent space*, a sample space associated to a uniform or gaussian probability distribution. Two differentiable cost functions are associated respectively to the Generator $J^{(G)}\left(\theta^{(G)}, \theta^{(D)}\right)$ and the Discriminator $J^{(D)}\left(\theta^{(G)}, \theta^{(D)}\right)$.

A tuple $\left(\theta^{(G)}, \theta^{(D)}\right)$ which is a local minimum of $J^{(D)}$ with respect to $\theta^{(D)}$ and local minimum of $J^{(G)}$ with respect to $\theta^{(G)}$ constitutes the Nash equilibrium before mentioned [149].

The cost functions, based on cross-entropy minimization, can be expressed as follows:

$$J^{(D)} = -\frac{1}{2}\left(\mathbb{E}_{x\sim p_{\text{data}}} \log D\left(x\right) + \mathbb{E}_{z\sim p_z} \log\left(1 - D\left(G\left(z\right)\right)\right)\right) \tag{5.3}$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{z\sim p_z} \log D\left(G\left(z\right)\right) \tag{5.4}$$

Several versions of GANs exist in literature, usually focusing on image creation. The main differences between them are in the definition of the cost functions, which intrinsically define the minimax game. In this dissertation, the classical framework has been adopted. Training a GAN is still an open issue because it is subjected to various phenomena typical of the game problem. The saturation of the game due to the perfect discriminator and the mode collapse are only two of the difficulties that can arise training a GAN. In the perfect discriminator case, the game saturates because D easily classifies from the beginning the examples coming from the real data distribution from the ones coming from G. Mode collapse happens when G learns to produce only a single example that D is not able to classify as fake. However, during training, D is easily trained to classify that example, so that G again collapses on another mode. Several techniques have been proposed in literature to improve this aspect, one of them is the minibatch discrimination. Adding a minibatch discrimination layer allows to maintain a certain level of entropy in the generated distributions. See [152] for additional details.

5.3.2 TrimGAN

Because at the beginning D quickly becomes perfect, a supervised pretraining procedure is conducted on G, represented in Figure 5.5 by the dashed arrows. The number of pretraining epoch has been obtained by a trial and error procedure with the observation that very few epochs are enough to avoid the perfect discriminator phenomena. In this work, only 10 pretraining epochs are conducted. The adversarial training starts after the pretraining of G. Every main step, D is trained once and G twice. This unbalance can again be justified by the perfect discriminator phenomena. D is optimized with respect to Eq. 5.3 using an RMSProp update scheme. The input is a mini-batch made by two stacked arrays, one coming from the real dataset (in this case stored in a CSV file), the other coming from G. The target is made by two stacked binary arrays, the first section corresponds to the real data and is set to 1, whereas the second section is set to 0. On the contrary, the Generator is trained indirectly using an Adadelta optimizer with respect to

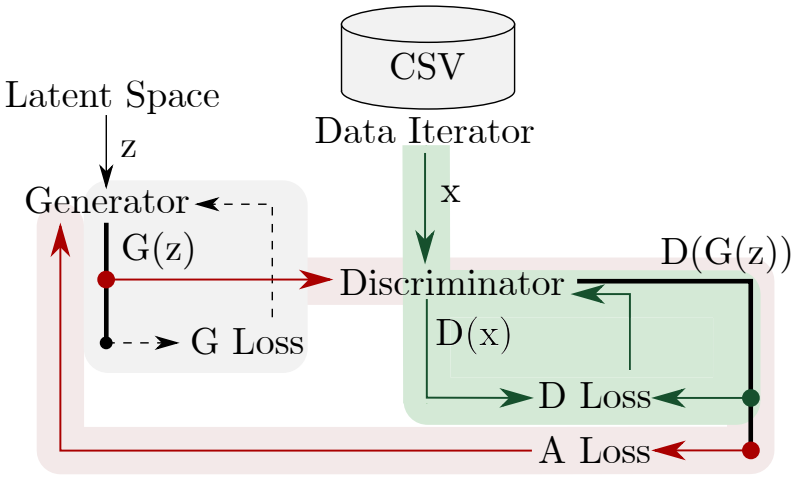


Figure 5.5: TrimGAN training outline (dashed gray lines represents the pretraining operation, green lines the Discriminator training procedure, red lines represents the Generator training procedure, A loss stands for the generator loss function in adversarial training)

Eq. 5.4. The loss, reported on Figure 5.5 as *A Loss* where A stands for *adversarial* to discriminate it from the *G Loss*, is calculated with respect to the output of D with a sample generated by G. In this way, the weights of G are optimized in way of generating samples classified as real by D.

5.3.3 GAN results

In this section some preliminary results regarding the generation of stationary flight state vectors using GAN are shown. The effects of the training parameters and of the hyperparameters of the two networks are not here considered. Further studies will be conducted on this aspect. At the beginning, TrimGAN has been implemented on low dimensional test case. In this case, the aircraft state vector has been condensed in two variables, $\mathbf{x} = [\alpha, C_L]$. Hence, the lift coefficient C_L is directly modeled as random variable instead of being evaluated from the mass and impact pressure. Two tests have been conducted. In the first subcase, α comes from the automatic trim detector algorithm whereas C_L has been obtained by the linear fitting reported in [108]. Hence, the first subcase is a linear fitting problem. The second subcase considers the actual α and C_L found by the autotrim algorithm, hence a nongaussian error is intrinsically present in the data. Table 5.3 collect the training configuration for both the simplified test cases.

GAN has shown promising results in the low dimensional test cases. The marginal distributions of each variable is shown in Figure 5.6 and 5.7. In both cases, the random variables assume realistic values and the distribution of the generated data is close to the original one.

Table 5.3: Training configuration for dim (**x**) = 2

Parameter	*Pre-training*	*Discriminator*	*Generator*
Learning rate	0.001	0.0005	0.5
# of epoch (partial/total)	2	1/10k	1/10k
Algorithm	RMSProp	RMSProp	AdaDelta
MLP architecture	N/A	[1010]	[10]
Batch size	N/A	60	60

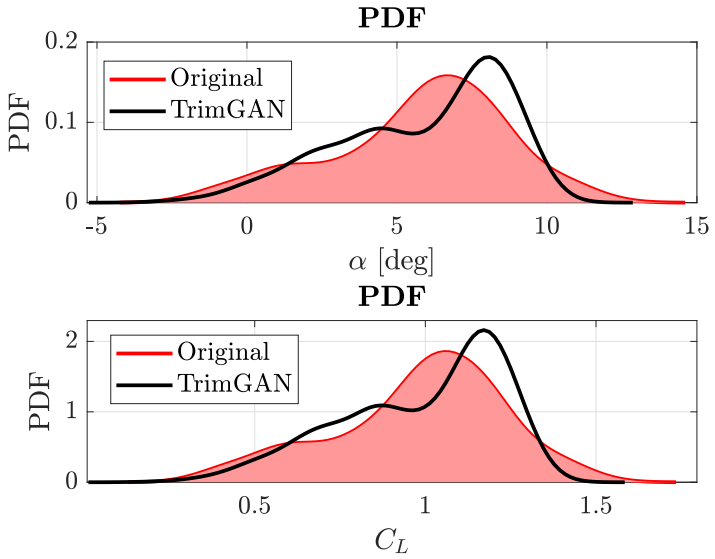


Figure 5.6: Marginal distributions of α and C_L of the original dataset and of the generated samples for Test Case 1 (Linear regression)

Figure 5.8 and 5.9 shows the loss function and the accuracy of both the Discriminator and the Generator networks. Because the first two Test Cases are simplified versions of the main problem, it is not surprising that the loss function quickly reaches the convergence. This quick convergence is also visible on the Discriminator accuracy that reaches 0.5, at least as an average value.

Eventually, the $C_L - \alpha$ curves can be compared. They have been represented in Figure 5.10 and 5.11. Test Case 1 can be considered a Toy Problem, however in Test Case 2 the positive correlation is weaker than in Test Case 1 and the joint distribution of the two variables shows promising results.

The same analysis has been conducted on a complete set of variables with dim $(x) = 17$. In this case, the lift coefficient C_L is evaluated as additional variable at the end of the training, directly on the generated data. This means that the correlation between the variables is hidden by an additional formula that is not explicit in the code or in the dataset. The training configuration is reported in

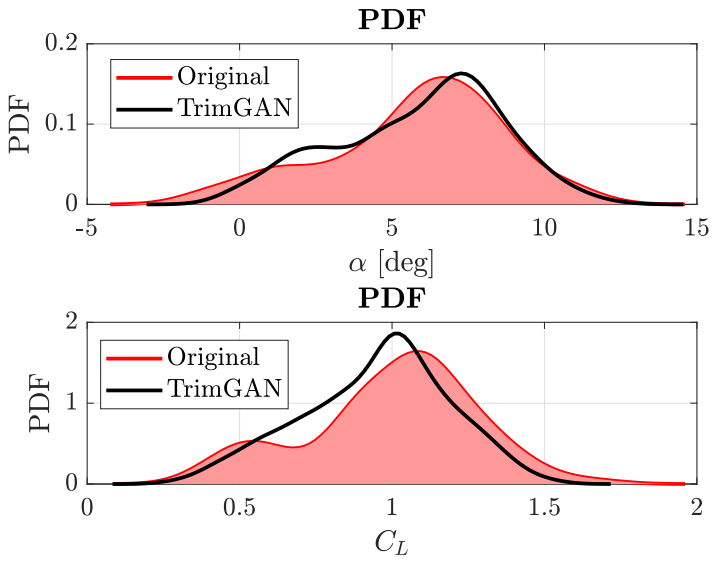


Figure 5.7: Marginal distributions of α and C_L of the original dataset and of the generated samples for Test Case 2 (Real data, $\dim(x) = 2$)

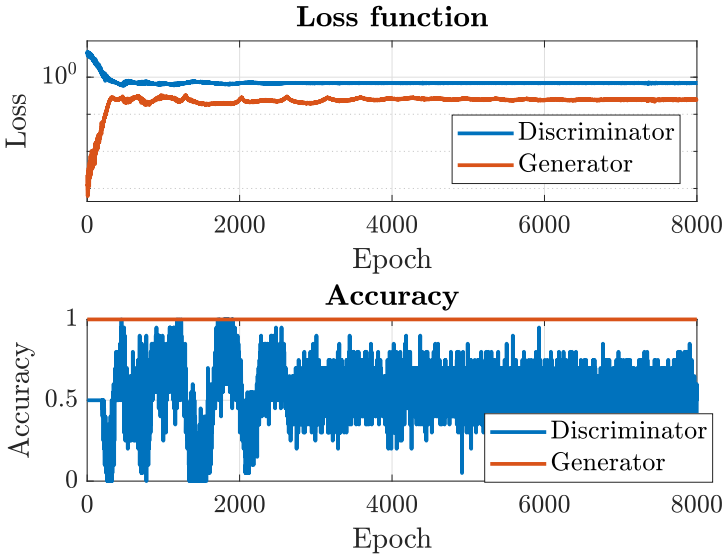


Figure 5.8: Loss and accuracy values for Test Case 1 (Linear regression)

Table 5.4.

In this case, the three linear regression lines that were underlined in the dataset have been lost. The positive correlation can still be observed.

Even if state variables of the samples drawn from the p_{model} assumed realistic values and even if the joint probability density function $p_{C_L,\alpha}(x, y)$ seems similar,

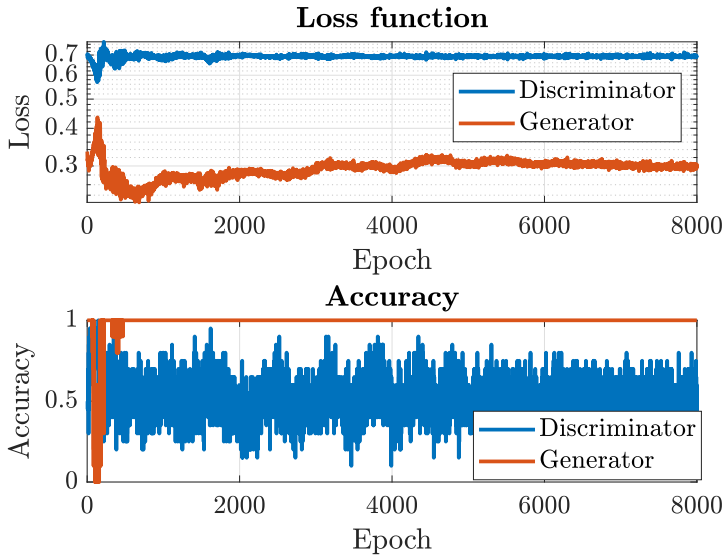


Figure 5.9: Loss and accuracy values for Test Case 2 (Real data, dim $(x) = 2$)

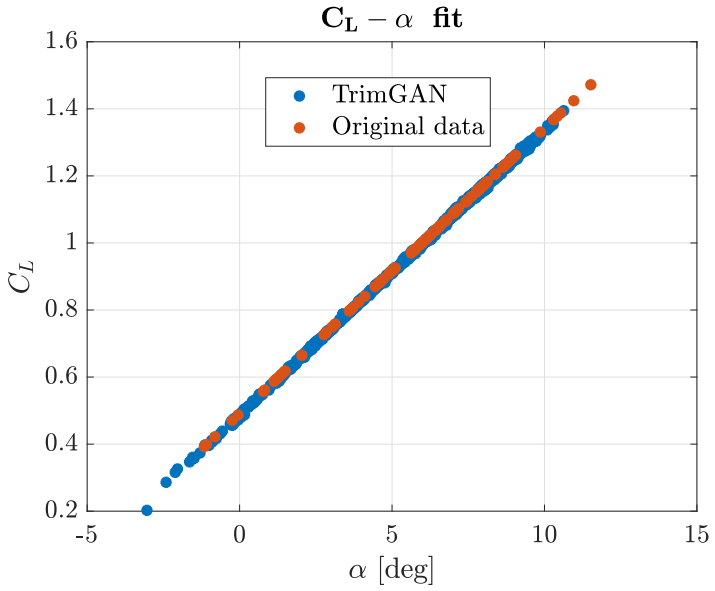


Figure 5.10: Comparison of the $C_L - \alpha$ curves for Test Case 1 (Linear regression)

at the moment the state vectors obtained by the TrimGAN are still not eligible for the training of Smart-ADAHRS. In fact, the uncertainty introduced by this new dataset is higher than the one already obtained by Smart-ADAHRS. This aspect should not be seen as a limitation of the methodology because these results must be considered preliminary.

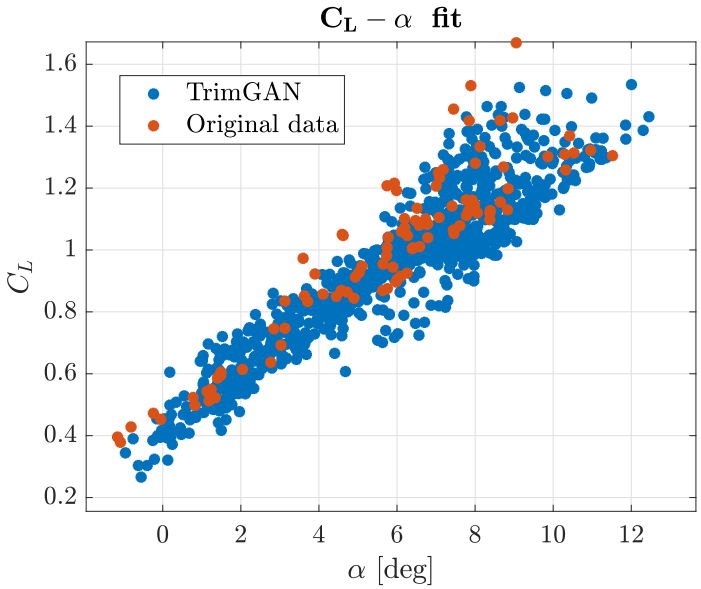


Figure 5.11: Comparison of the $C_L - \alpha$ curves for Test Case 2 (Real data, dim $(x) = 2$)

Table 5.4: Training configuration for dim $(\mathbf{x}) = 17$

Parameter	*Pre-training*	*Discriminator*	*Generator*
Learning rate	0.001	0.000 005	0.000 000 05
# of epoch (partial/total)	2	20/10k	1/10k
Algorithm	RMSProp	RMSProp	AdaDelta
MLP architecture	N/A	[2020]	[20]
Batch size	N/A	60	60

Chapter 6

Final comparison

Previous chapters described a set of methods applicable to the design of a synthetic sensor based on NN for aerodynamic angle estimation. The description focused on the theoretical foundation of the proposed methods and on the interpretation of the individual results.

One of the aim of this dissertation is to show the performance that the same NN architecture can reach, if trained with different methods. In 2013, the patent of Smart-ADAHRS has been published and the method was at TRL 4. The NN was tested in simulated environment and the demonstration in relevant environment was beginning in 2016. The performance, moving from simulation to flight test, dropped and it was necessary to tune the training of the NN to cope with sensor noise and uncertainty propagation. From this point of view, the comparison with other techniques such as Kalman filter was out of the main scope of this dissertation. On the other hand, the definition of a design path for a NN-based estimator passes from the comparison of the various methods that can be used to obtain the final set of weights.

This point of view resembles the regression point of view. For instance, in polynomial fitting the final product is a polynomial function, regardless of how it has been obtained. Several methods can be used to fit a dataset using the same degree of the polynomial. Thinking to MLP as a function obtained by means of multivariate regression, it is interesting to compare several training or data analysis methods in order to improve the performance of the MLP, without necessarily changing the architecture.

This chapter provides the comparison of the results obtained with a synthetic sensor designed with the methods proposed in the previous chapters, with respect to a first attempt of design obtained with the classical method. This first MLP, henceforth called *Baseline network*, is obtained through the traditional method that composed the state-of-the-art of the training at the beginning of this PhD project. To fully understand the advantages and disadvantages of the methods, four classes of estimators are designed. This chapter begins with the description

of the four classes of estimators in Sec. 6.1. Sec. 6.2 reports the analysis of the PDFs and CDFs of the error. In Sec. 6.3 the uncertainty of the various solutions is estimated and compared. In Sec. 6.6 the comparison of the error in the time domain is conducted to show the limitations of the analysis of the timehistories. In Sec. 6.7 the TS is analysed in terms of similarity functions. The feature maps are shown in Sec. 6.5. At the end of this chapter, the proposed methods have been applied to design a VS for the estimation of the AOS.

6.1 Definition of the compared estimators

As stated in Chapter 1, the absence of a shared metrology introduces a certain difficulty in the design of a synthetic sensor. In fact, to conduct the comparison between the solutions obtained following different design flows, a set of rules based on some metrics must be defined. During the design of a synthetic sensor, a long series of comparisons is conducted among the various solutions to fully understand the behaviour of the estimator. In this dissertation, the methods proposed in Chapter 4 and 5 are applied for two main reasons: to ease the design of a synthetic sensor and to ensure the estimated uncertainty of the sensor itself. To analyze their advantages and disadvantages, several estimators have been designed following different methods and the results have been compared. A Baseline estimator is designed following the only some fundamental steps, as common in literature. It is used as scientific control for the other solutions. Due to the huge time needed by some architectures, not every solutions has been analyzed. However, this should not be seen as a limitation because the conclusions should be already clear.

Table 6.1: Classes of estimators

Class name	*Architecture*
Baseline	[13], [20 20], [20 20 20]
Multiple flight tests	[13], [20 20], [20 20 20]
Manoeuver-based CV	[13], [20 20], [20 20 20]
Data augmentation	[13]
Data augmentation Manoeuver-based CV	[13]

Table 6.2: Additional hyperparameters common to the various tests

Activation function	sigmoidal
Codomain of the input and target map	$\{\forall x \in \mathbb{R} \mid -1 \leq x \leq 1\}$
Maximum number of iteration	1000
Sampling time	0.5 s

For sake of clarity, Table 6.1 collects the classes of estimators used to verify the proposed methods. Table 6.2 gives details that are in common to the various estimators.

6.1.1 Definition of the baseline estimator

The TS of the baseline networks is composed by a single flight segment containing sawtooth glides and excitation of the phugoid mode (Flight ID 6 in Table 3.7). These manoeuvers are mainly contained into the longitudinal plane. Table 6.3 collects the main details of the Baseline estimator.

Figure 6.1 shows the coverage of the (α, β) envelope given by the selected TS. It is clear from Figure 6.1 that some regions are not considered by the TS, for instance $\{\forall\,(\alpha,\beta) \mid 4° < \alpha < 6° \wedge -5° < \beta < 0°\}$. A high (or even low) uncertainty in those regions cannot be taken as granted. Testing the VS with values contained in those regions will only verify the generalization capabilities of the NN. The marginal distributions that define the hypercube for the Baseline NNs is shown in Figure 6.2. The hybercube of the Baseline shows several outliers, corresponding to the red crosses. This is an index of too narrow distributions and this can negatively affects the final performance. Figure 6.3 shows the standard deviation of the residuals.

Table 6.3: Baseline definition

Number of flight test for training	1
Size of TS	38 000
Time duration of the TS	1900 s
Architectures	1 hidden layer 13 neurons
	2 hidden layer [20 20] neurons
	3 hidden layer [20 20 20] neurons
Input pattern :	$\alpha_{in}, q_c, \dot{q}_c, a_x, a_y, a_z, \theta, \phi, p, q, r$
Network output:	$\Delta\alpha$
Heuristic rule	Levenberg-Marquardt
TS partitioning	70 % training, 15 % validation, 15 % test

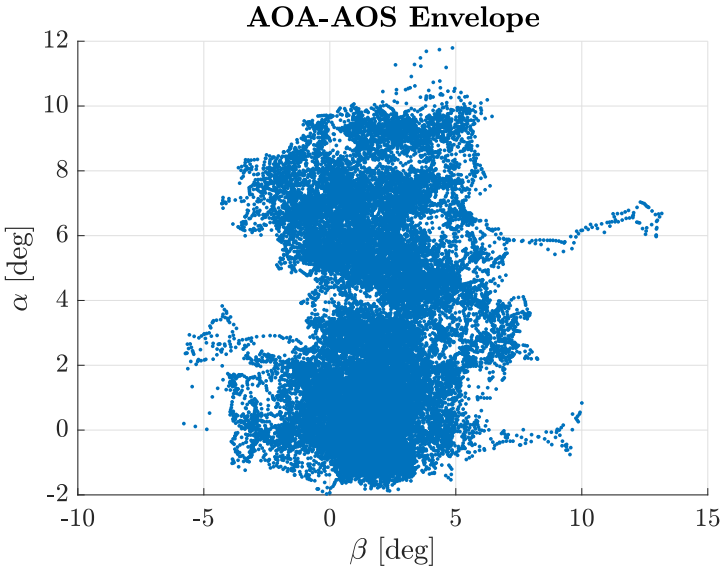


Figure 6.1: Distribution of the TS on the AOA/AOS envelope, baseline

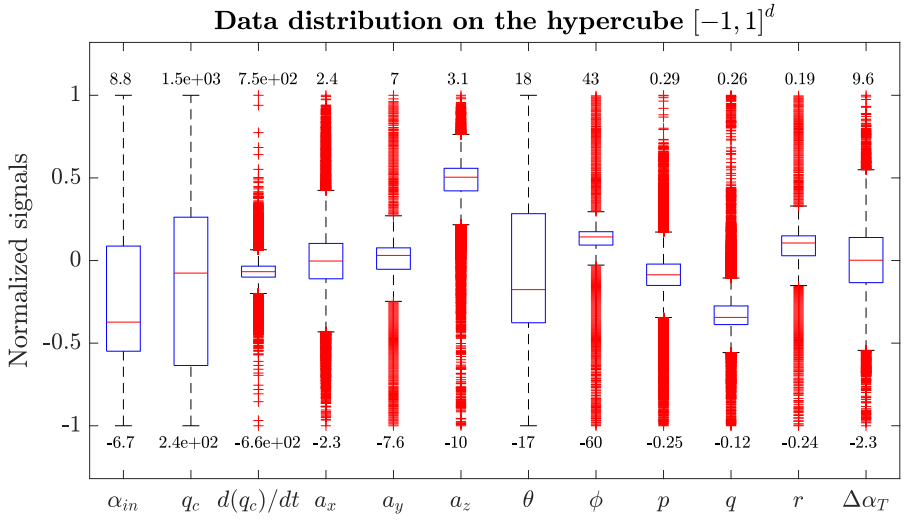


Figure 6.2: Marginal distributions of the TS, baseline (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

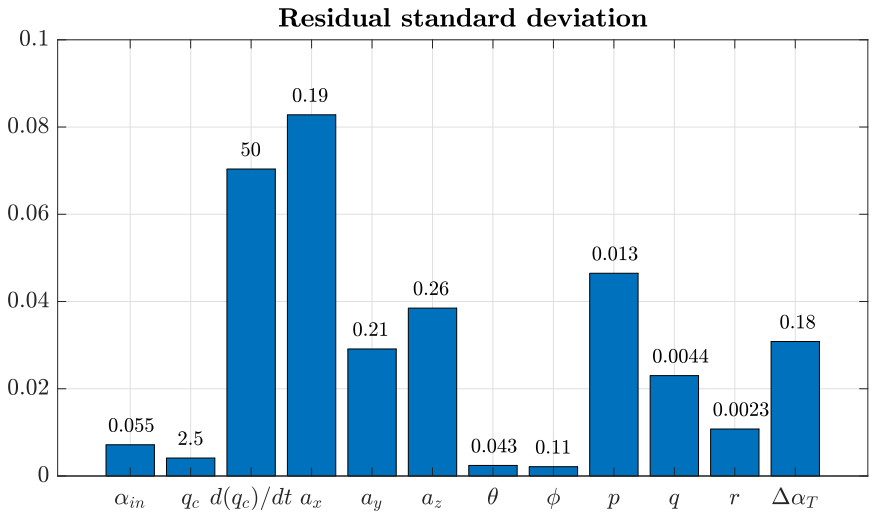


Figure 6.3: Standard deviation of the residuals, baseline (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

6.1.2 Multiple flight test training

The same architecture showed in Table 6.3 is applied on a dataset bigger than the one of the baseline, composed of several flight tests.

Manoeuvers out of the longitudinal plane have been added to the TS. A more generic training is in accordance with the practical habit of applying supervised learning on a wide portion of the input-target space. In this way, the test set becomes more likely a subset of the TS obtaining better generalization error performance [114]. For this class of estimator, also the selection of one of the NNs among several re-training is conducted. The selected network is one over 10 training.

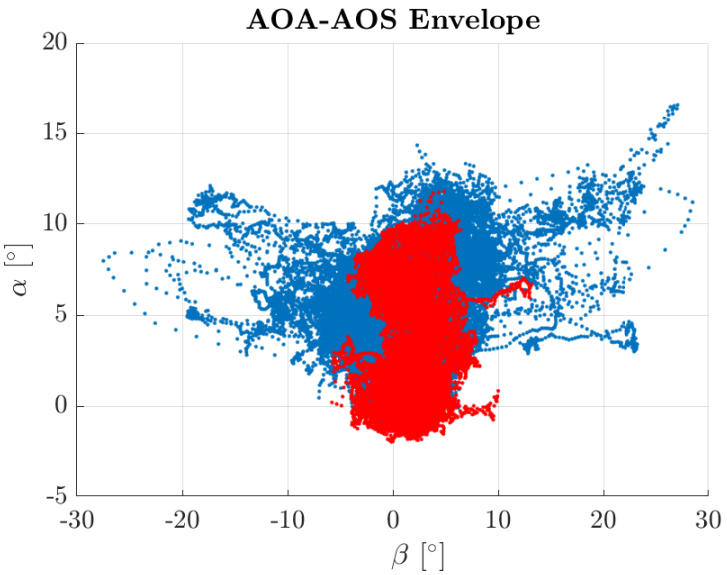


Figure 6.4: Distribution of the baseline TS on the AOA/AOS envelope (red) superimposed to the one of the multi-flight test

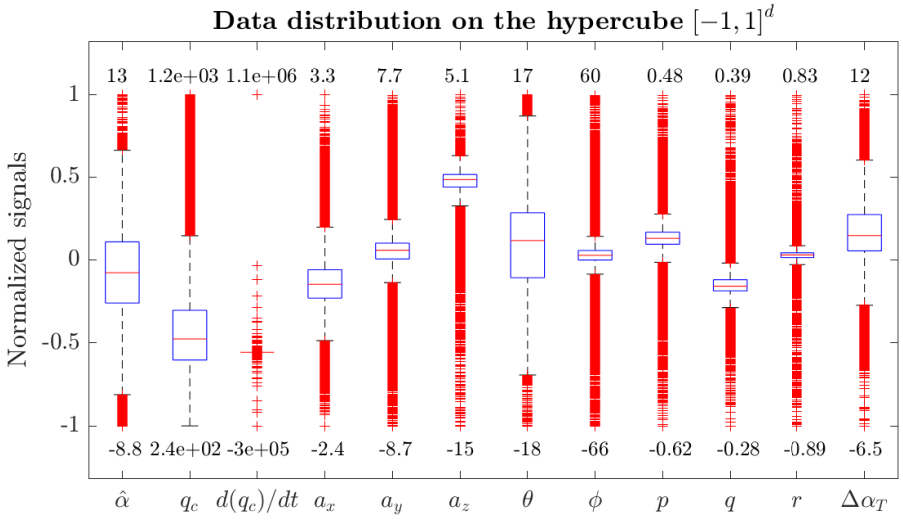


Figure 6.5: Marginal distributions of the TS, multi-flight test (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

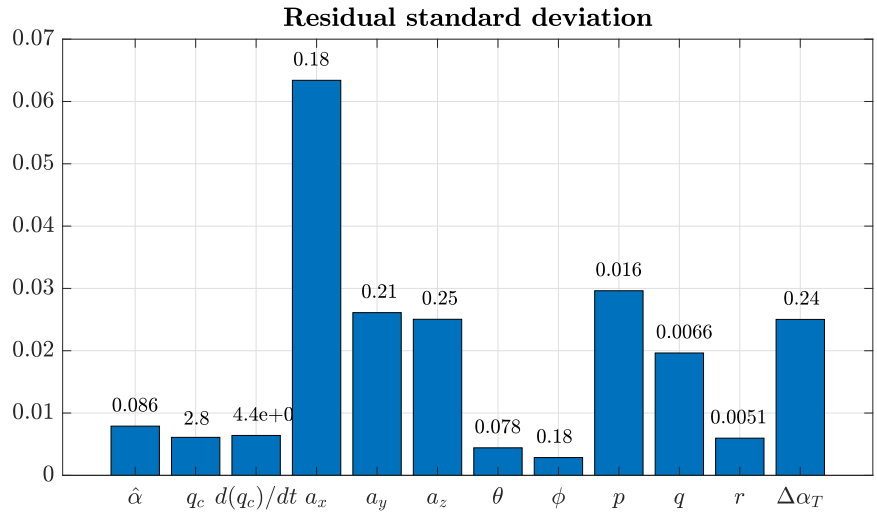


Figure 6.6: Standard deviation of the residuals, multi-flight test (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

6.1.3 Manoeuver-based CV network

This section describes the results obtained with the application of the manoeuver-based CV to the training of the MLP. As a recall, this method applies the entire dataset and it tries different combinations of the subsets of the dataset to find the best TS. The number of partitions have been set to 10 (10-fold CV).

Figure 6.7 shows the different coverage on the AOA/AOS envelope of the TS allowed by the CV technique.

Figure 6.8 shows the definition of the hypercube using the selected subset of the TS. The standard deviation of the residuals is reported in Figure 6.9.

Also in this case, the analysis has been repeated varying the architecture of the NN. It could be expected that different NNs trained from scratch with different capacity might select a different TS. It is quite notable that the selected subset of data for training is always the same, that is the fifth one as can be seen from Figure 6.10. This could be the indication that the selected subset is actually the most adapt for the VS design, given the available dataset.

As negative outcome, the selected TS is almost the entire dataset. The best combination excludes from the TS only some sawtooth glides from different flights. In details, the entire segment of sawtooth glides of the flight ID 6 is excluded, together with a section of the same manoeuver conducted in flight ID 1 and 2. However, this is not completely a negative aspect. In fact, it means that the other manoeuvers are essential for the training operation. The exclusion of some sawtooth glides can also be caused by an unbalancement of the available manoeuvers inside the dataset. In fact, there are 5 flight tests containing sawtooth glides (with several repetitions for each one), whereas only one flight segment contain SHSS, 2 flight segments contain Phugoid excitation and other 2 contain Dutch-Roll excitation.

It must be noticed that the ratio of test points on training points is 26.68 % and it is the higher among the other combinations, as can be seen from Figure 6.10.

The results of the MBCV method can be visualized both in terms of statistics of the training and validation error. Figure 6.11 shows the comparison of the NSSE obtained using the 10 different subsets with respect to the average value. Figure 6.12 shows the mean, maximum and standard deviation of the deviation ϵ expressed by Eq. 4.4 among the various manoeuvers. As a remark, a negative value means an improvement given by the NN with respect to the estimator $\hat{\alpha}$.

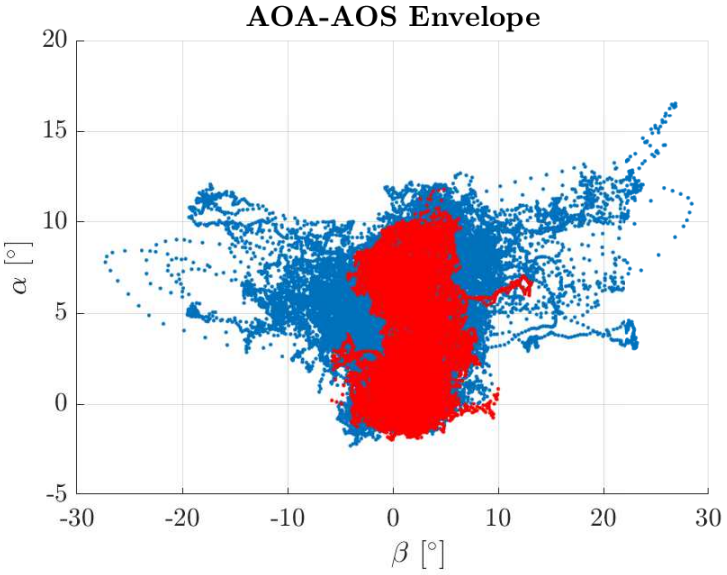


Figure 6.7: Distribution of the baseline TS on the AOA/AOS envelope (red) superimposed to the one selected by the CV method.

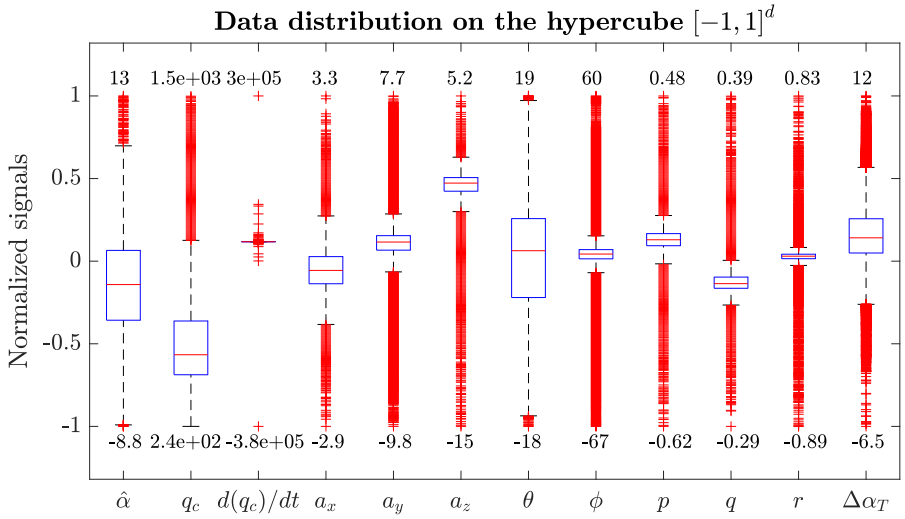


Figure 6.8: Marginal distributions of the TS, CV test (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

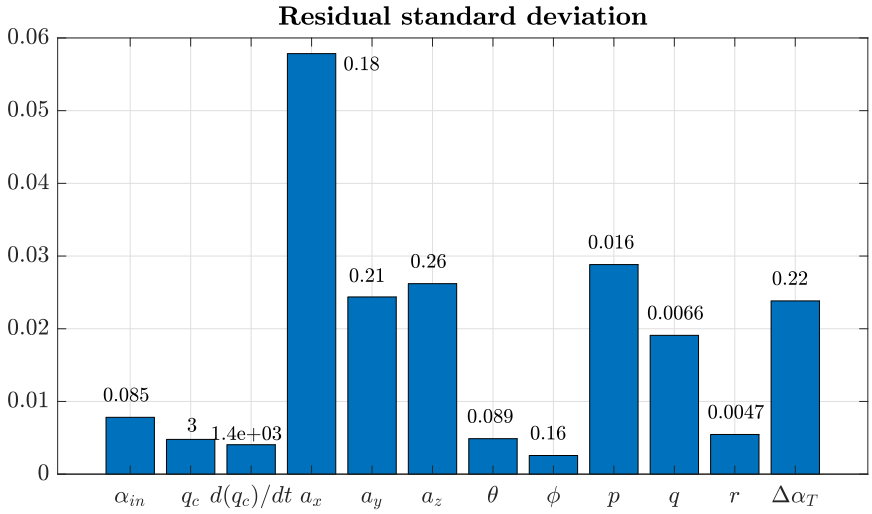


Figure 6.9: Standard deviation of the residuals, CV test (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

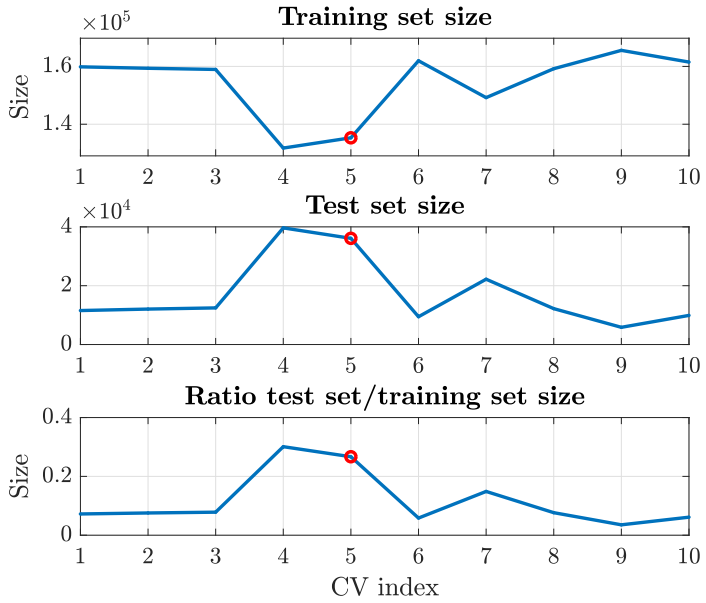


Figure 6.10: Size of the various combinations analysed during the manoeuver-based CV. The red circle points at the best partition in terms of training performance.

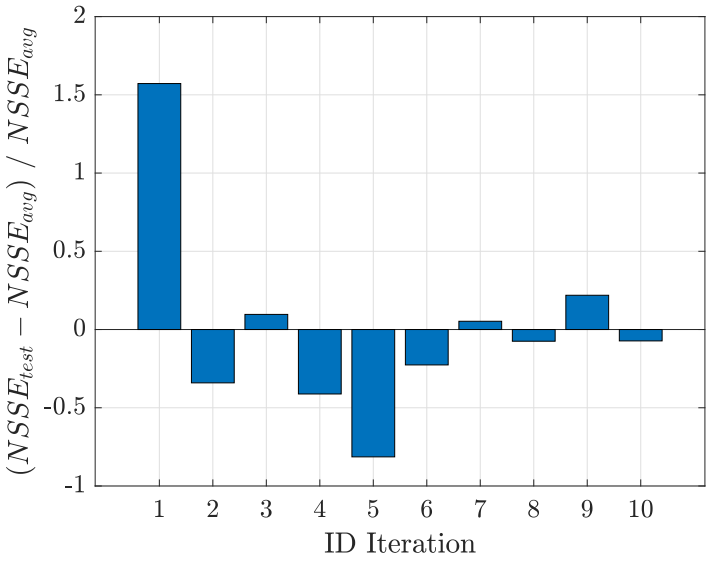


Figure 6.11: Validation NSSE values obtained with the application of the MBCV method

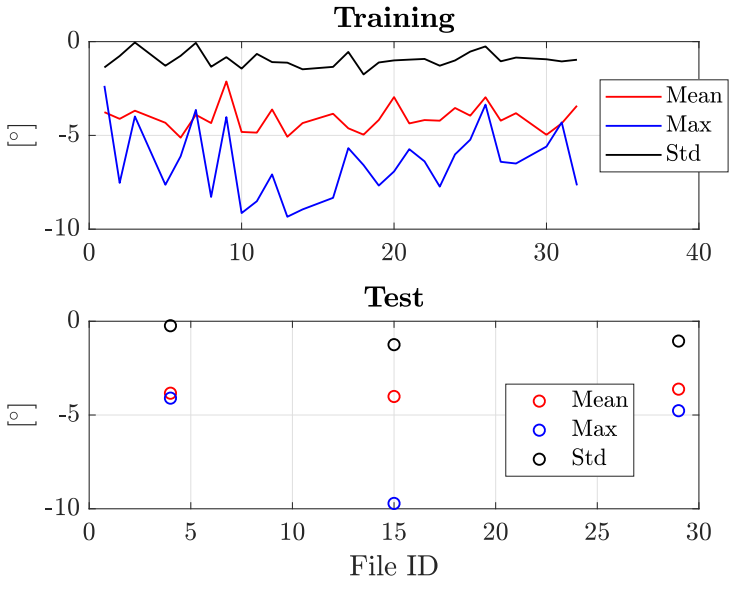


Figure 6.12: Statistics of ϵ among the various manoeuvers

6.1.4 Training with data augmentation

This section describes the results obtained with the data augmentation techniques to the training of the MLP.

Figure 6.13 shows the different coverage on the AOA/AOS envelope of the TS allowed by the CV technique.

Figure 6.14 shows some differences with respect to the corresponding charts of the previous methods. The introduction of a set of entries in an highly concentrated region results in more narrowed PDFs of the various variables. This can negatively affect the training phase.

As can be seen in Figure 6.15, the data coming from the flight simulator has reduced the standard deviation of the residuals, except for the q variable.

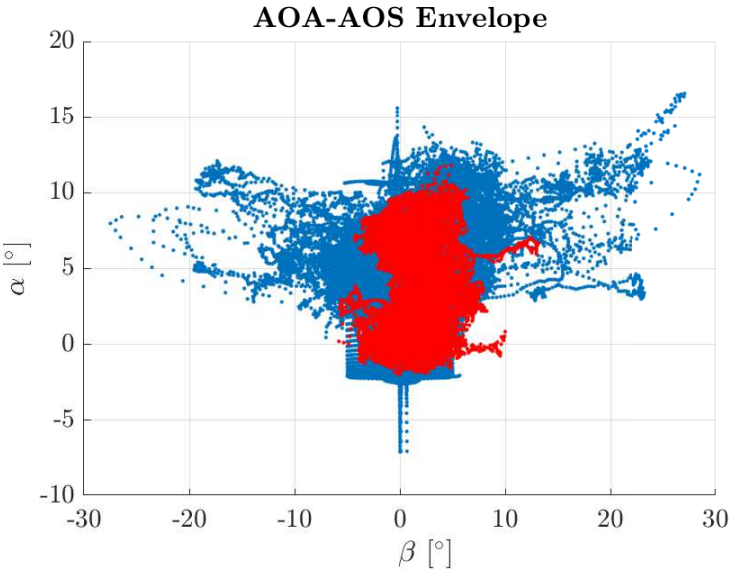


Figure 6.13: Distribution of the baseline TS on the AOA/AOS envelope (red) superimposed to the one of the data augmentation test

On the augmented dataset, it is possible to run the automatic trim detection algorithm of Chapter 4 to verify if the region of the quasi-stationary flight conditions has been actually improved in terms of coverage. Moreover, it is possible to partially verify the consistency of the simulations with the real AC. In Figure 6.16 the $C_L - \alpha$ plot is shown. For this class of estimators, only the single hidden layer [13] is designed.

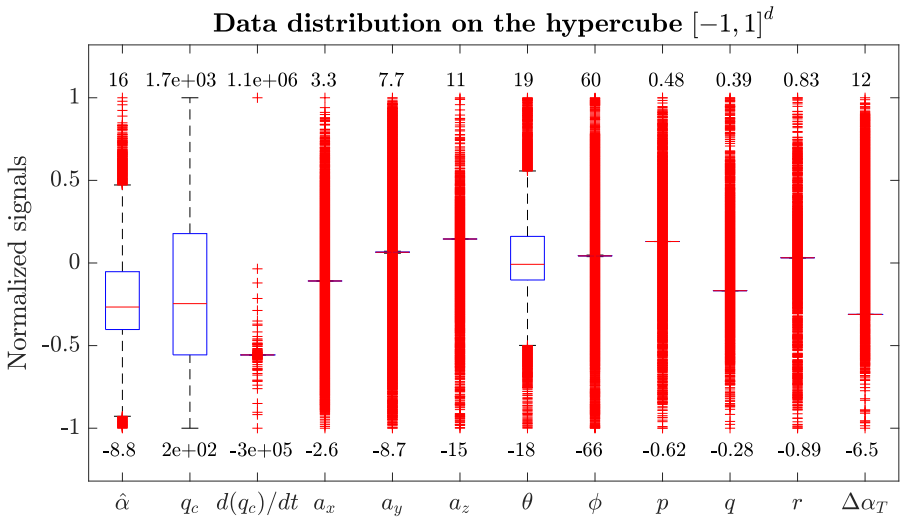


Figure 6.14: Marginal distributions of the TS, data augmentation test (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

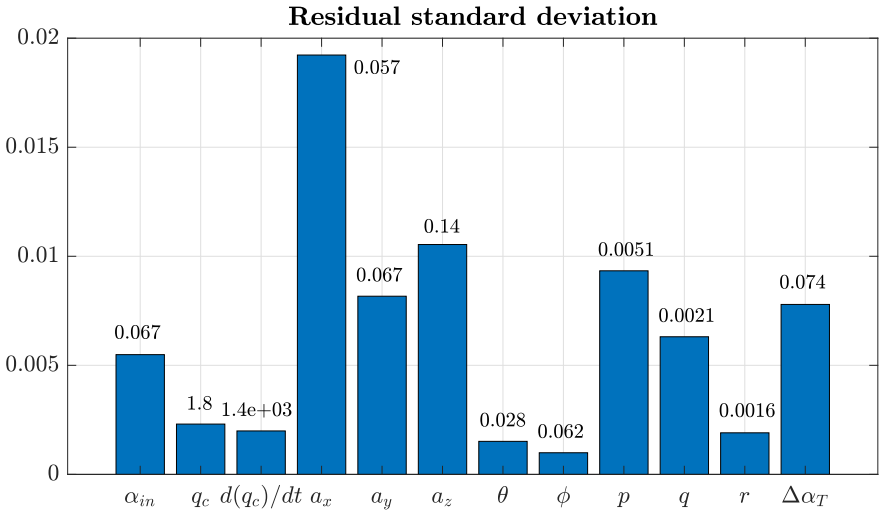


Figure 6.15: Standard deviation of the residuals, data augmentation test (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

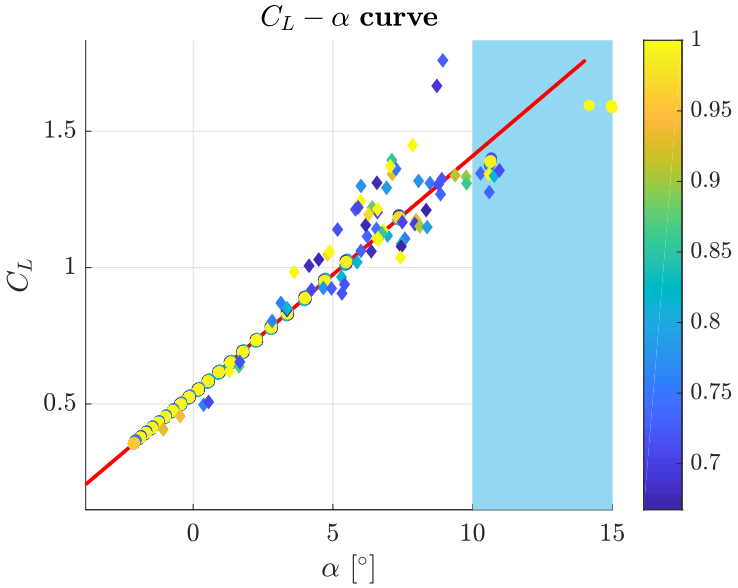


Figure 6.16: Comparison between simulated and flight test quasi-stationary conditions on the $C_L - \alpha$ plot (the circles represent the simulated points, the diamonds the flight test points) $\alpha_0 = 6.2007°$, $C_{L,\alpha} = 0.086\,994\,°^{-1}$ ($4.9844\,\mathrm{rad}^{-1}$), $R2 = 0.99988$

6.1.5 Manoeuver-based CV training with data augmentation

Due to the huge size of the dataset, the training conducted in Sec. 6.1.4 might become intractable. To understand if the combination of the methods can be beneficial in terms of final uncertainty, the manoeuver-based CV is conducted on the augmented dataset. For sake of clarity the method is henceforth reported as MBCVDA. Moreover, for aforementioned reasons, It becomes important if the CV procedure will discard the simulated trajectories. In fact, this behaviour indicates a poor matching of the simulated aircraft with the real one or that no improvement is added with the data augmentation. Also for this class, only the single hidden layer network is designed. As can be seen from Figure 6.20, the ratio of the cardinality of the test set on the cardinality of the TS is only 9.7 %, lower than the acceptable.

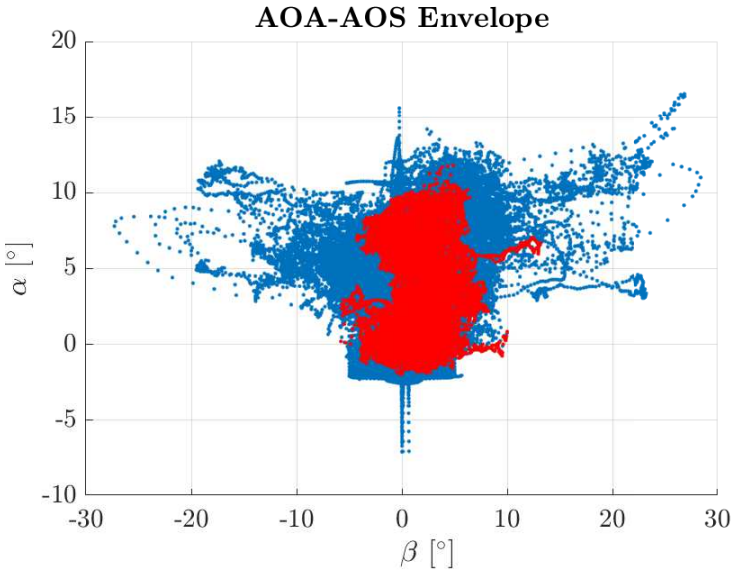


Figure 6.17: Distribution of the baseline TS on the AOA/AOS envelope (red) superimposed to the one selected by the CV method on the data augmented dataset

6.1.6 Analysis conducted and computational cost

Due to the time needed for each analysis and because of the fact that in some case the analysis can be considered unneeded on a particular solution, only the most important tests have been conducted. This section reports the test plan that brought the results that have been reported in this Chapter. To conclude this section, the time needed to train the NNs is reported in Table 6.4. The Baseline and the multiflight NNs have been trained on a personal laptop in parallel mode

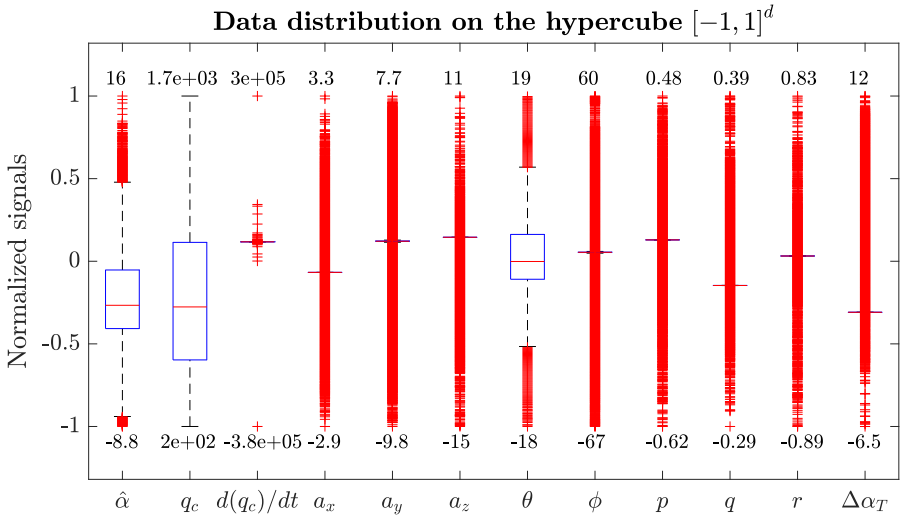


Figure 6.18: Marginal distributions of the data augmentation, CV test on augmented dataset (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

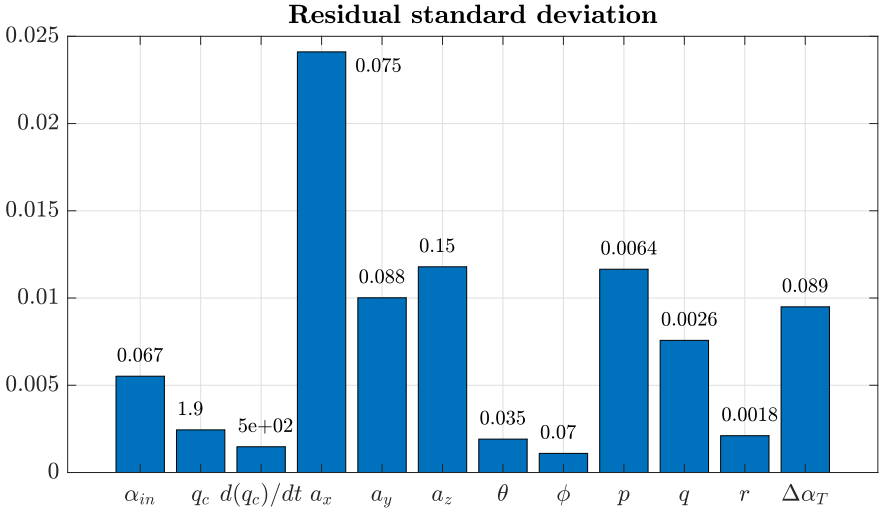


Figure 6.19: Standard deviation of the residuals, CV test on augmented dataset (Unit of measurements as follows: angle in [°], angular rate in [rad s^{-1}], pressure in [Pa], time in [s])

with computation shared on the Central Processing Unit (CPU) Intel i7-7700HQ and on the Graphics Processing Unit (GPU) nVidia GeForce GTX1050. The CV and data augmentation methods have been carried out on the High Performance Computing (HPC) facility of the Politecnico di Torino called Legion. Legion is

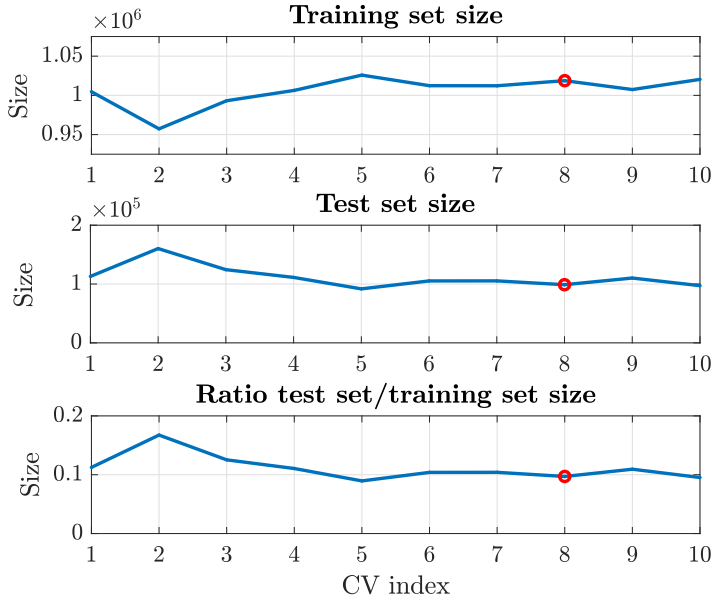


Figure 6.20: Size of the various combinations analysed during the manoeuver-based CV with data augmentation. The red circle points at the best partition in terms of training performance.

equipped with Intel Xeon Scalable Processors Gold 6130 2.10 GHz 16 cores and nVidia Tesla V100 SXM2 - 32 GB - 5120 cuda cores. For additional details on the HPC available at Politecnico di Torino visit `http://hpc.polito.it`.

Table 6.4: Time needed for training and hardware list

Method	*[13]*	*[20 20]*	*[20 20 20]*
Baseline	15.4 s	237.5 s	557.4 s
Multiple flights	329.9 s	420.1 s	581.8 s
Manoeuver-based CV	620 s	988.2 s	1447.8 s
MBCVDA	20 812.7 s	N/A	N/A

The sensitivity analysis lasted 1300 s and 4550 s on the HPC, depending on the architecture. The computation of the TS analysis based on similarity measurements has been conducted only on the Manoeuver-based CV solution because it was considered the most promising. The analysis has been splitted in 10 jobs that required more than 9.5 hours (85 CPU hours) and 20 GB of RAM each, corresponding to a total amount of 850 CPU hours.

6.2 Deviation PDF comparison

Before showing the results of the uncertainty estimation, the PDF of the deviation between the target α_T, the output of the initial estimation $\hat{\alpha}$ and the output of the VS $\tilde{\alpha}$ are given in this section. Although this analysis can be seen as part of the uncertainty estimation, it still gives more understandable results than the analysis of the timeseries. As for the uncertainty analysis, this evaluation is conducted on the entire dataset, not only on the Test Set.

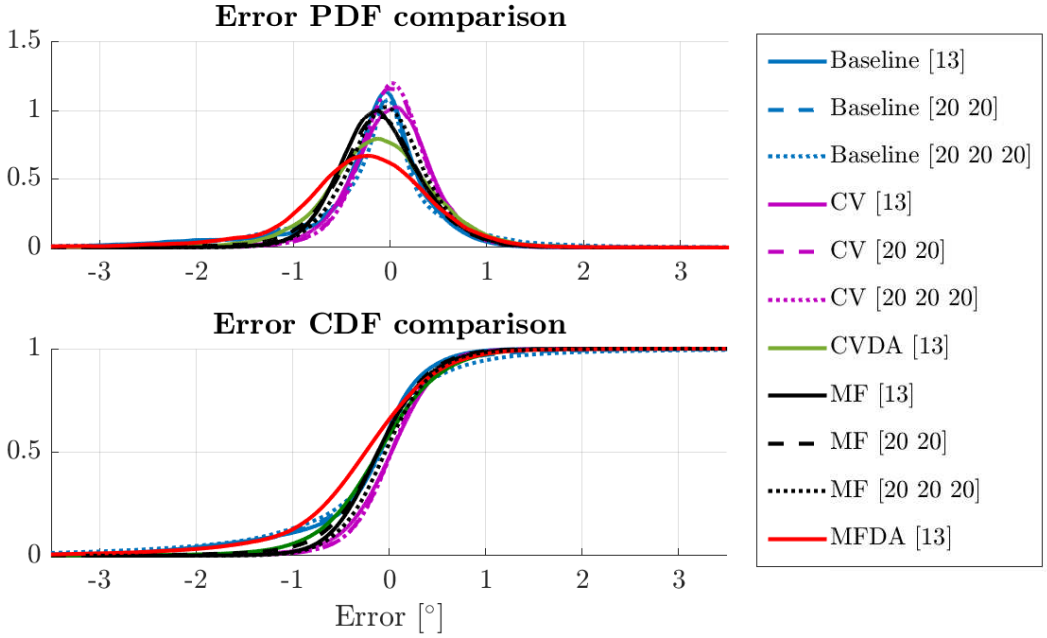


Figure 6.21: Comparison of the PDFs and CDFs obtained with various architectures and methods on the entire dataset for the AOA estimator.

Figure 6.21 shows the PDFs and CFDs obtained with the various methods on the entire dataset. It can be seen that the CV methods greatly improves the Baseline results, whereas the DA results in broader distributions. As anticipated, the architecture has little effect on the results and there is not a strong correlation between the number of connections and the final error distribution. Only in the case of the multiple flight TS there is an important improvement with a 3-hidden layer NN with respect to a lower number of hidden layers. Moreover, being the multiple flight NN selected among 10 re-trained networks, and because of the fact that the training metric MSE is homogeneous among the various NNs, this improvement might not be accounted to overfitting.

6.3 Uncertainty comparison

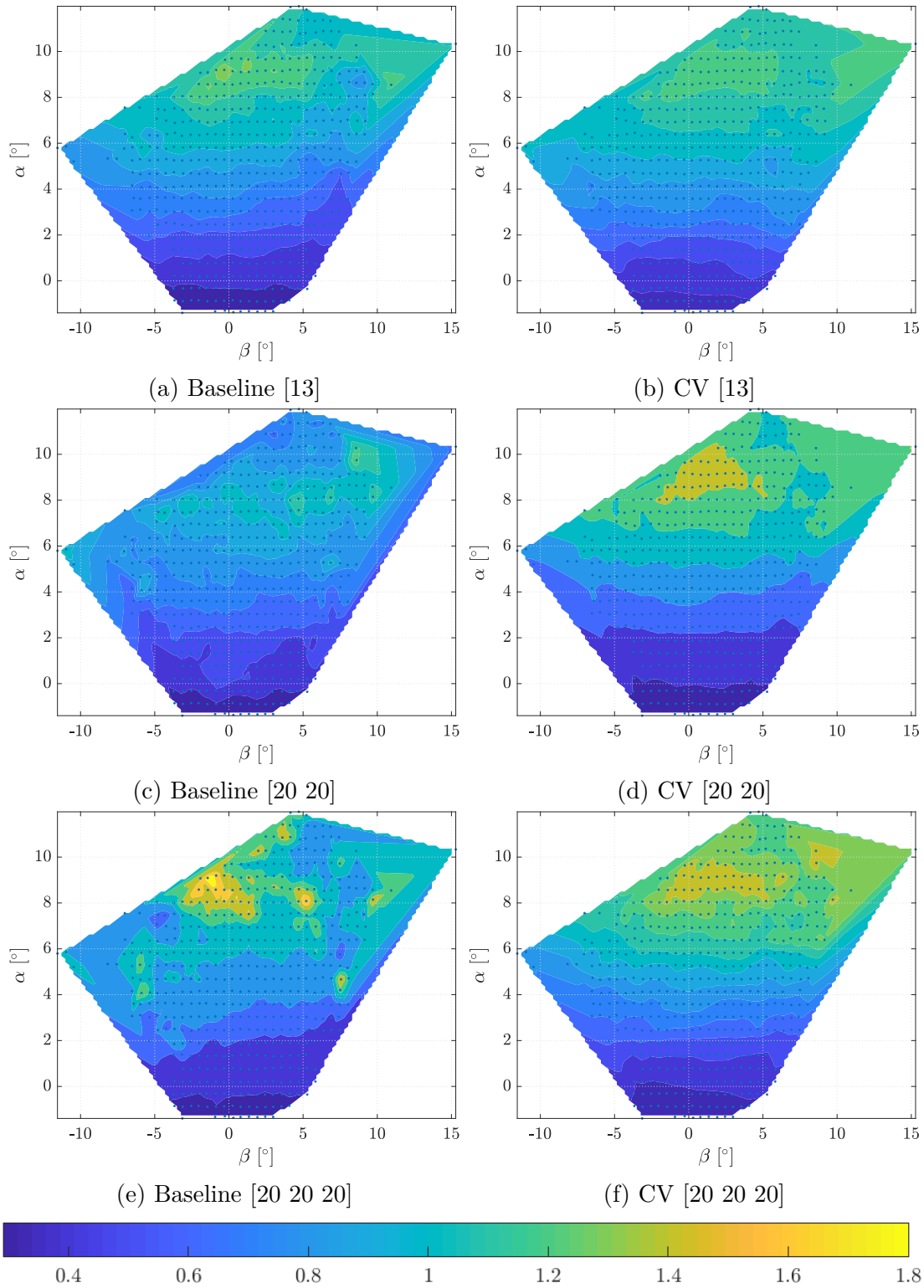
This kind of synthetic sensor must be characterized with a final uncertainty value with respect to the reference quantity. However, the behaviour of the isolated NN can also be interesting. For this reason, also the distribution of the deviation of the NN response with respect to nominal output has been analysed. As mentioned in Chapter 4, the results reported in this section are based on the entire reference dataset, containing both the TS and the Test Set. The following heatmaps contain the entire dataset considered in this dissertation.

Figure 6.22 shows that the increased capacity of the network is not generally beneficial in terms of final uncertainty, independently from the applied training technique. Moreover, it seems that the major improvement given by the Manoevuer-based CV is an increased homogeneity of the uncertainty.

Figure 6.23 shows the maximum error due to the uncertainty propagation. This figure clarifies the positive effects of the CV, that avoids peaks in the estimation error.

Figure 6.24 shows that the Manoeuver-based CV reduces the dispersion between flight conditions at the same pair (α, β). On the contrary, the sensitivity to the flight condition increases with the network capacity. Please note that the color scale in Figure 6.24e has been cut to improve the visibility of the other maps. In fact, the Baseline [20 20 20] network reaches a peak of 1.2° of standard deviation of the uncertainty value.

As mentioned before, the DA techniques can be applied when the model used to generate the augmented dataset matches perfectly to the desired AC. Any difference between the two would bring to error in the training phase of the NN. Excluding the extreme case of non-convergence of the training, the results can still be unacceptable or, as happen in this case, they can make useless the application of the DA itself. This has been shown in Figure 6.25, where 3 NNs with the same architecture have been trained respectively using multiple flight tests, multiple flight test and DA and MBCVDA. Although MBCVDA reduces the uncertainty at low AOA with respect to simply using multiple flights to build the TS, the results show no improvement on the uncertainty of the VS with respect to the MBCV method, which remains the best in terms of every estimated metrics.



(a) Baseline [13]

(b) CV [13]

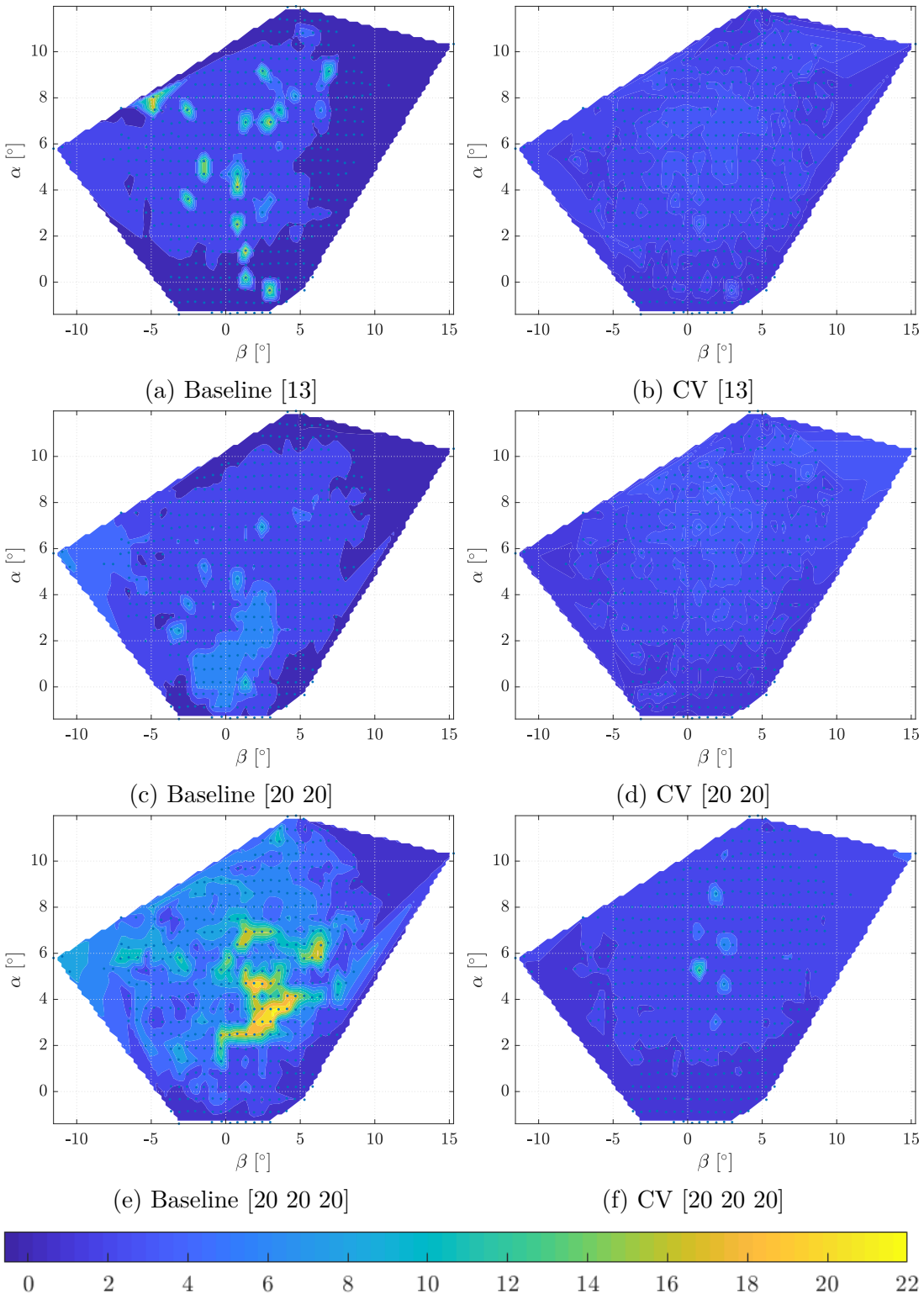
(c) Baseline [20 20]

(d) CV [20 20]

(e) Baseline [20 20 20]

(f) CV [20 20 20]

Figure 6.22: Comparison of the 1σ distributions on the AOA/AOS envelope obtained using different architectures and techniques



(a) Baseline [13]

(b) CV [13]

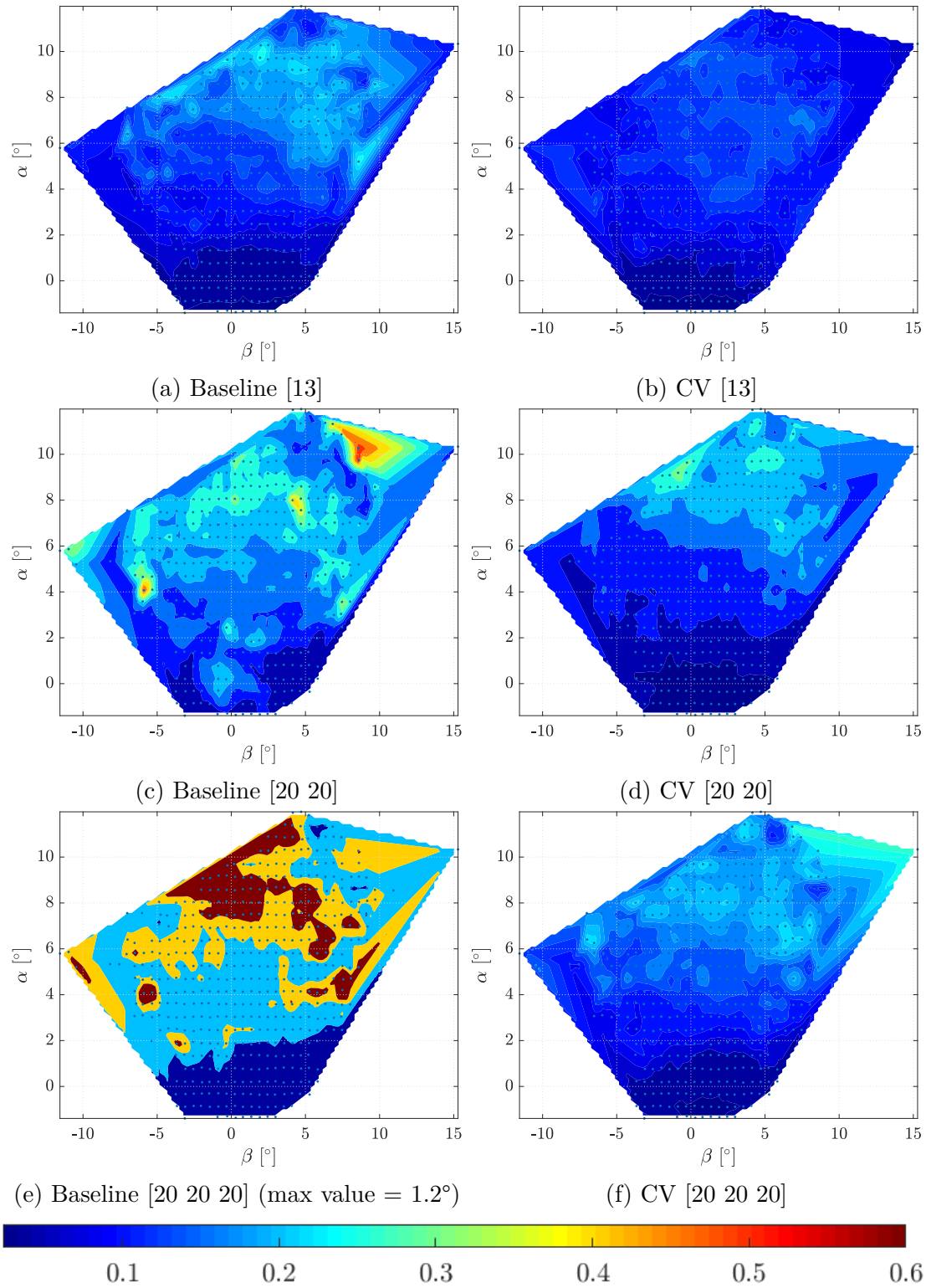
(c) Baseline [20 20]

(d) CV [20 20]

(e) Baseline [20 20 20]

(f) CV [20 20 20]

Figure 6.23: Comparison of the maximum error distributions on the AOA/AOS envelope obtained using different architectures and techniques



(a) Baseline [13]

(b) CV [13]

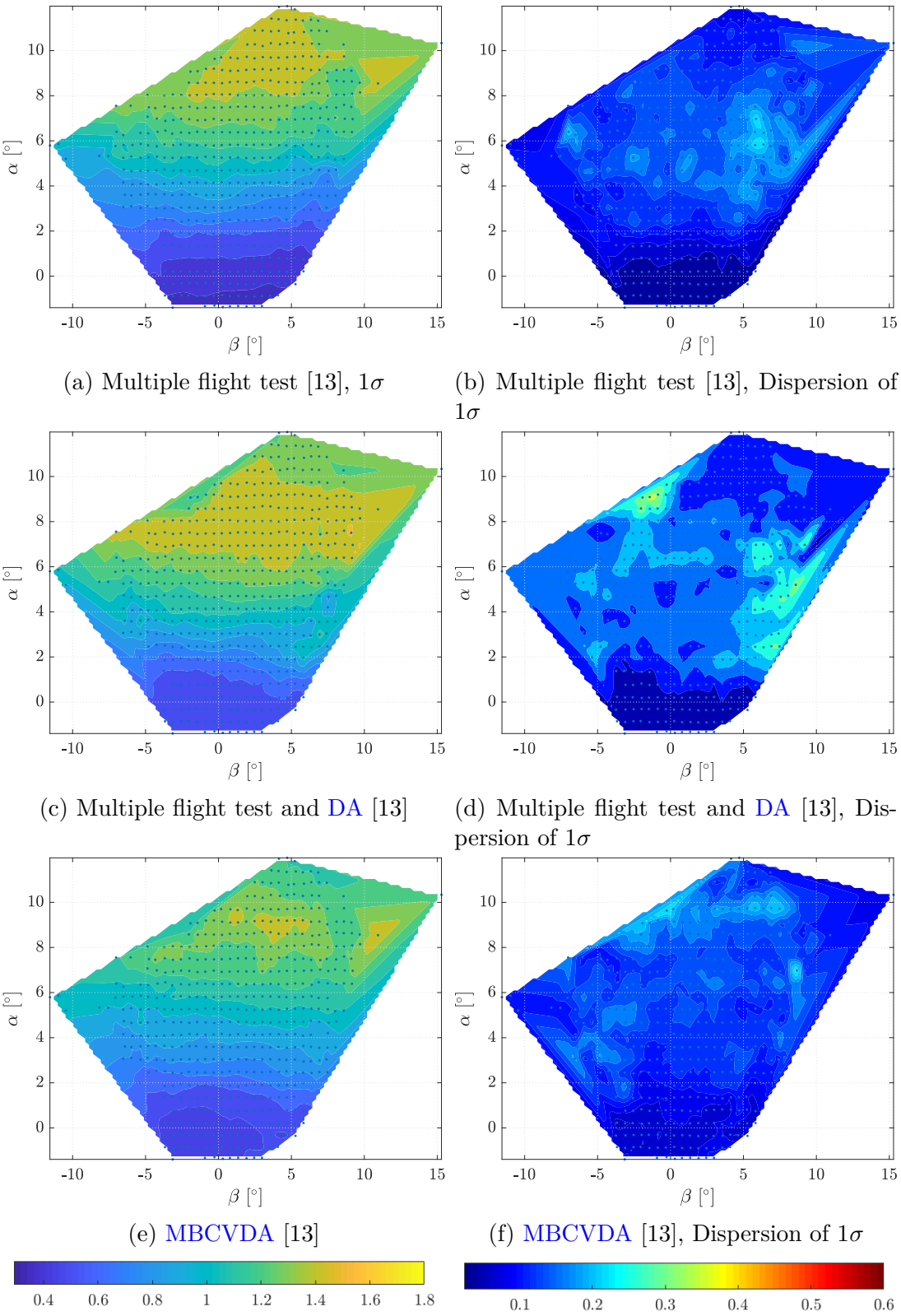
(c) Baseline [20 20]

(d) CV [20 20]

(e) Baseline [20 20 20] (max value = 1.2°)

(f) CV [20 20 20]

Figure 6.24: Comparison of the standard deviation of the uncertainty distributions on the AOA/AOS envelope (Effect of the flight condition at equal pair (α, β)



(a) Multiple flight test [13], 1σ

(b) Multiple flight test [13], Dispersion of 1σ

(c) Multiple flight test and DA [13]

(d) Multiple flight test and DA [13], Dispersion of 1σ

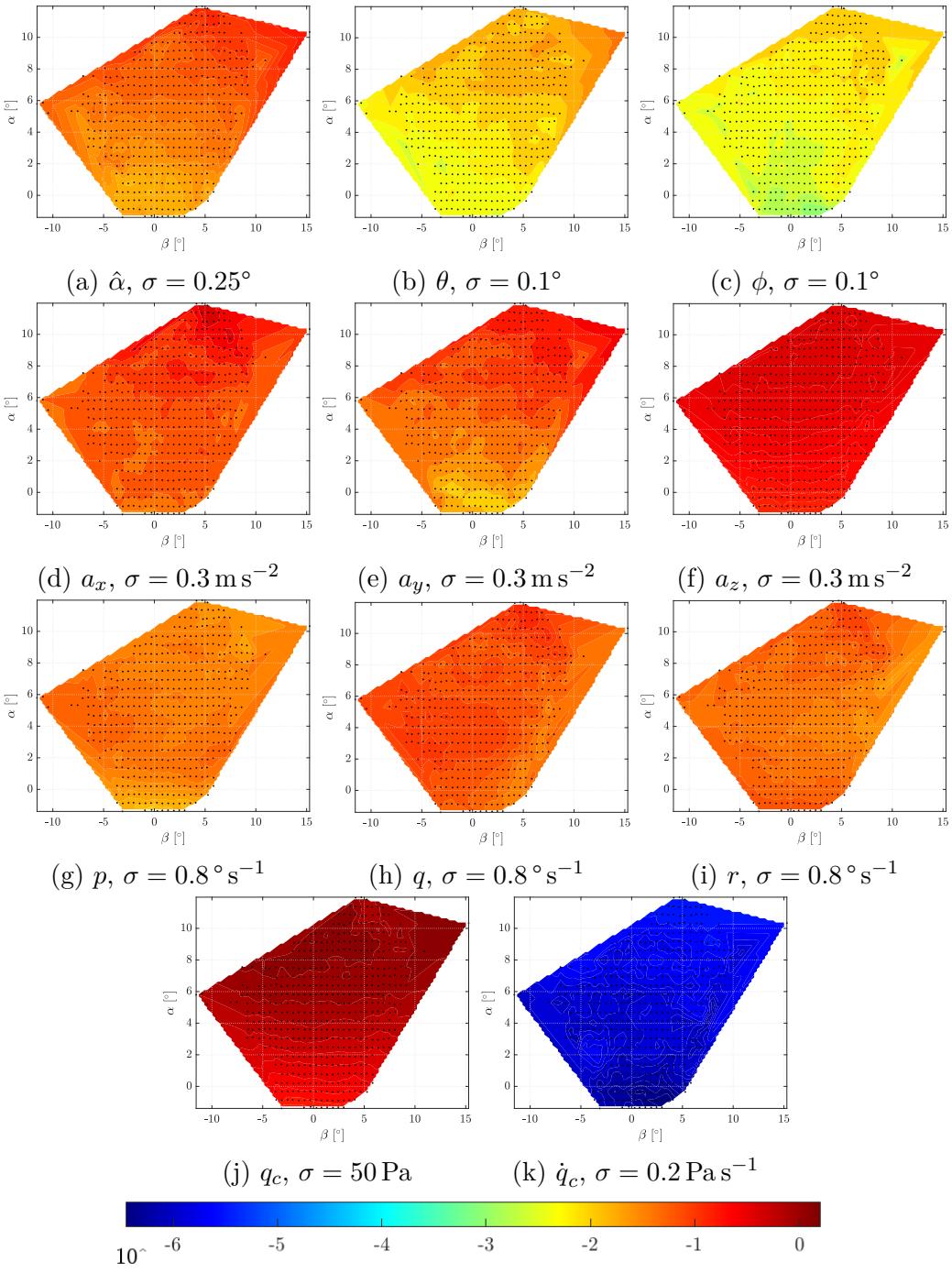
(e) MBCVDA [13]

(f) MBCVDA [13], Dispersion of 1σ

Figure 6.25: Comparison on the AOA/AOS envelope of using multiple flights, DA and MBCVDA

6.4 Analysis of the most influential input variable

The analysis shown in Sec. 6.3 can be repeated analyzing the effect of each input variable one at a time. Using the same input covariance matrix applied in Sec. 6.3, it is possible to understand which is the most influential input variable with important consequences on the necessary equipment for an actual implementation. A white noise has been injected with the standard deviation reported in the figure. This value has been defined by experience and previous works. Results obtained for the CV [20 20] are shown in Figure 6.26. The reported heatmaps show that, except for the $\dot{q}_c$ signal which seems to have a very little effect, the contribution are almost homogeneous. Both the attitude angles (θ and ϕ) acts with less then 0.01° in the final uncertainty. For the CV [20 20] network, the major source of uncertainty is the impact pressure. These results can be compared with the one obtained at the end of Chap. 1 using genetic programming. Also in that case $\dot{q}_c$ has a low impact on the final performance and the same holds for the ϕ angle. For what concern the acceleration a_y, the genetic programming discarded that variable, whereas the NN training revealed a certain sensitivity to a_y. Both analysis, genetic programming and NN training, confirm the importance of a reliable measurement of the impact pressure q_c.



(a) $\hat{\alpha}$, $\sigma = 0.25°$ (b) θ, $\sigma = 0.1°$ (c) ϕ, $\sigma = 0.1°$

(d) a_x, $\sigma = 0.3\,\mathrm{m\,s^{-2}}$ (e) a_y, $\sigma = 0.3\,\mathrm{m\,s^{-2}}$ (f) a_z, $\sigma = 0.3\,\mathrm{m\,s^{-2}}$

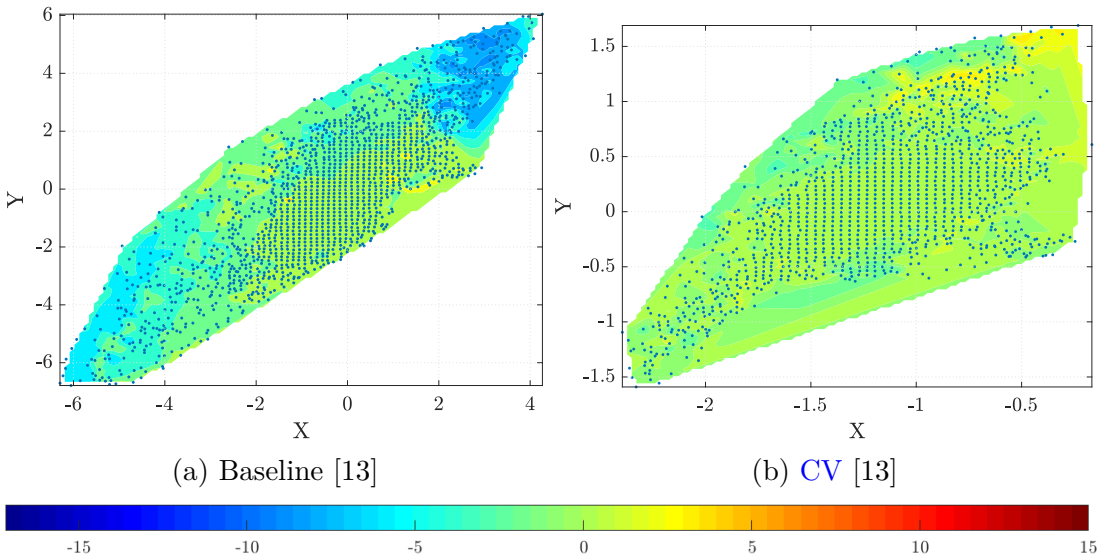
(g) p, $\sigma = 0.8\,°\,\mathrm{s^{-1}}$ (h) q, $\sigma = 0.8\,°\,\mathrm{s^{-1}}$ (i) r, $\sigma = 0.8\,°\,\mathrm{s^{-1}}$

(j) q_c, $\sigma = 50\,\mathrm{Pa}$ (k) $\dot{q}_c$, $\sigma = 0.2\,\mathrm{Pa\,s^{-1}}$

Figure 6.26

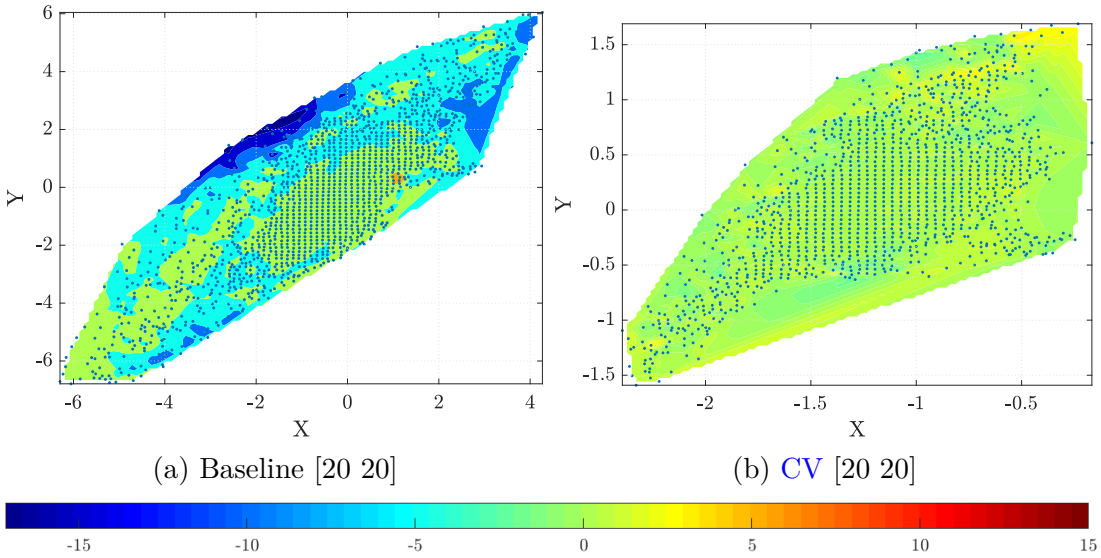
6.5 Feature maps comparison

This section shows the comparison of the feature maps. They differ from the uncertainty charts in the variables used in the representation. In fact, the feature map is based on the calculation of nondimensional variables X and Y. The comparison is shown from Figure 6.27 to Figure 6.29. Also in this case, the MBCV method shows homogeneity in the estimation error also with respect to different AC state vectors. The Baseline network, on the contrary, increases the estimation error when the flight condition is highly asymmetrical or dynamic. The MBCV method shows also that the selection of the TS can actually reduce the ambiguity given by the NN architecture.



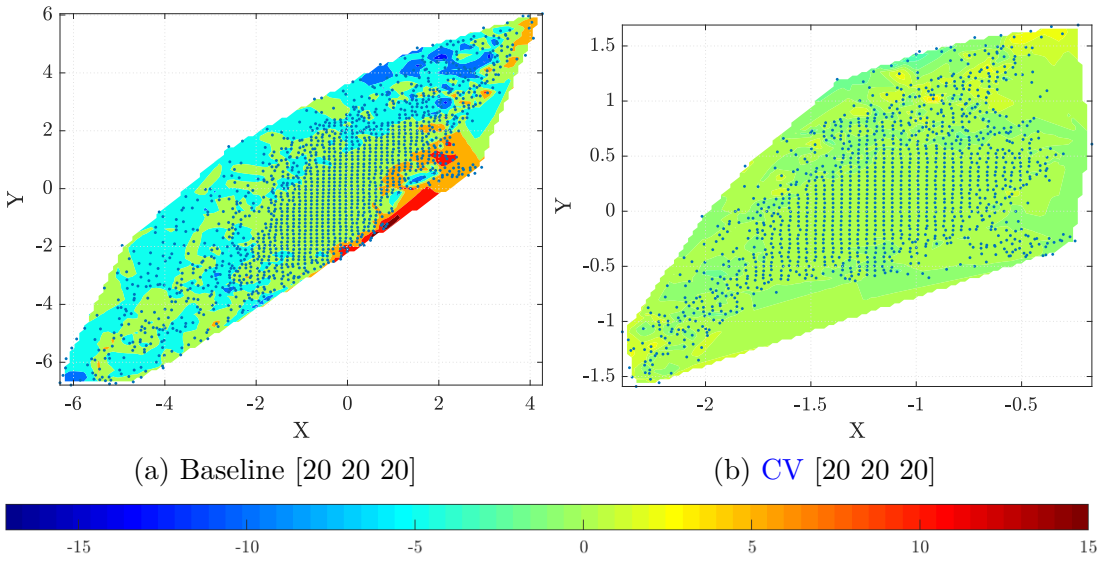
(a) Baseline [13] (b) CV [13]

Figure 6.27: Comparison of the feature maps (part 1/3)



(a) Baseline [20 20] (b) CV [20 20]

Figure 6.28: Comparison of the feature maps (part 2/3)



(a) Baseline [20 20 20] (b) CV [20 20 20]

Figure 6.29: Comparison of the feature maps (part 3/3)

6.6 Timeseries comparison

As mentioned before, the comparison of the timeseries is not a satisfactory instrument. However, this section gives some details from this point of view.

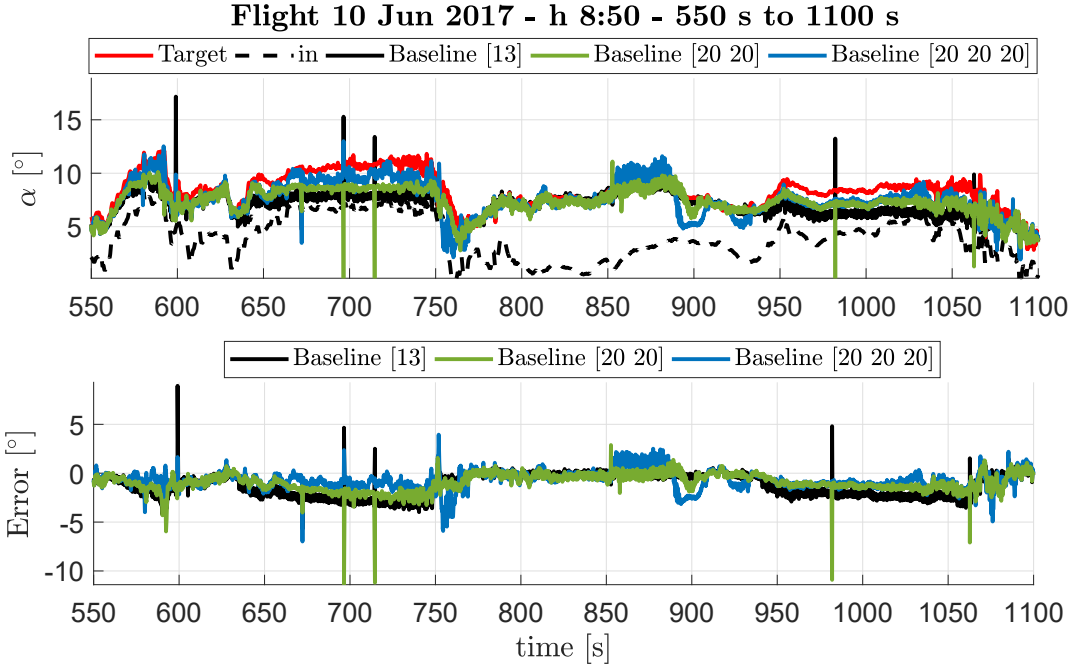


Figure 6.30: Details of the results on a particular section of a flight.

Figure 6.30 shows the effect of the network capacity during a particular flight condition. Some outliers have been observed in every architecture, at the same instant. These outliers has not been injected by $\hat{\alpha}$ and hence they come directly from the evaluation of $\Delta\alpha_{NN}$ using the NN. An apparent beneficial effect of the increased NN capacity can be understood from this plot. However, this is not confirmed by Figure 6.31, where the Baseline [20 20 20] VS clearly deviates from the other VSs during flight conditions that seems coped well even with 1 hidden layer with 13 neurons. This could be a representation of the overfitting encountered using an increased capacity on a set of data with unsufficient cardinality. The observed deviation persists for an unneglectable time length (about 45 s) and a drop of 5° has been observed at 890 s.

Figure 6.32 shows the results of the VS designed following the manoeuver-based CV method. No outliers have been observed and the error is less correlated to the flight condition than in Figure 6.30.

The same observations can be gathered from Figure 6.33 and 6.34. In Figure 6.34 can also be found another case of high capacity network (CV [20 20 20]) with an

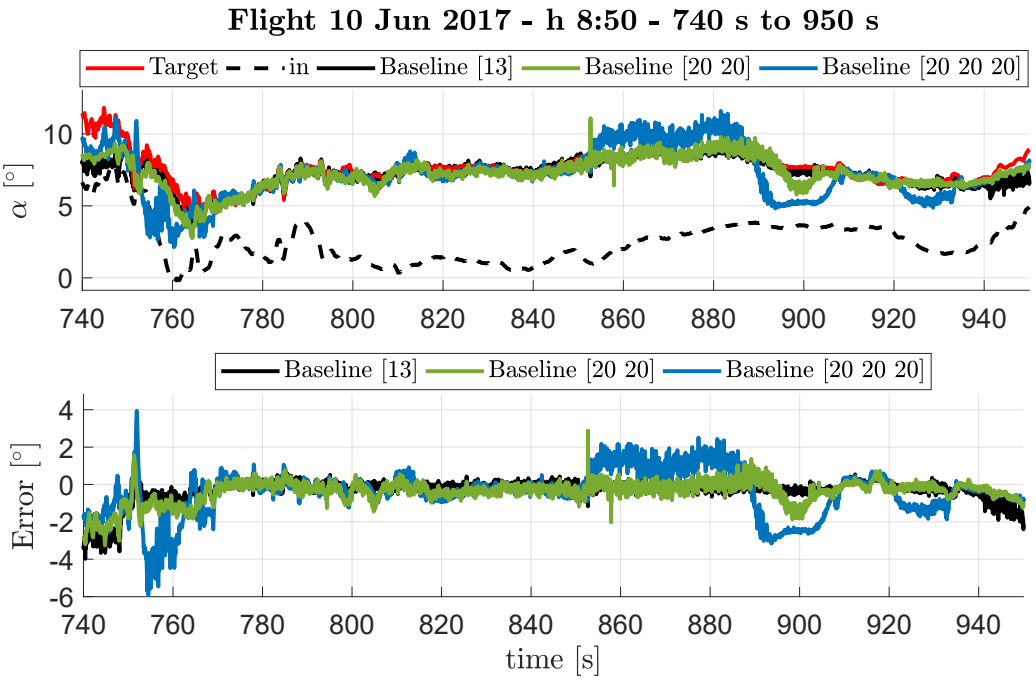


Figure 6.31: Details of the results on a particular section of a flight.

error higher than the version with less capacity, this case can again be lead back to overfitting.

The comparison between Figure 6.35 and Figure 6.36 shows that the promising manoeuver-based CV method is not able to completely reduce the error between 1920 s and 1940 s. Some outliers have also been observed during this fight.

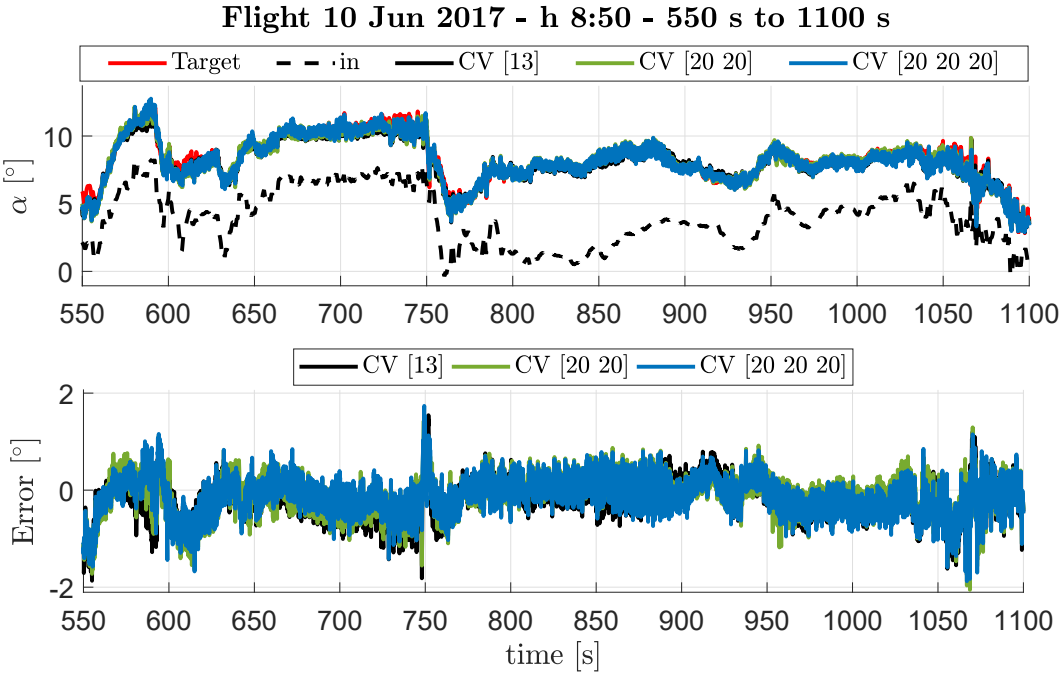


Figure 6.32: Details of the results on a particular section of a flight.

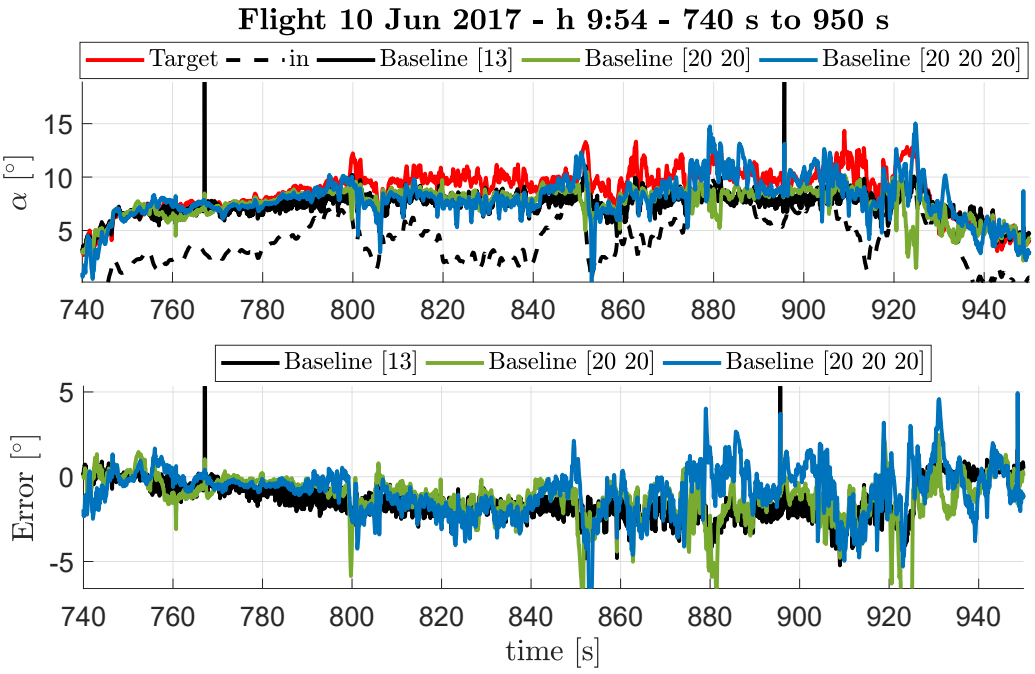


Figure 6.33: Details of the results on a particular section of a flight.

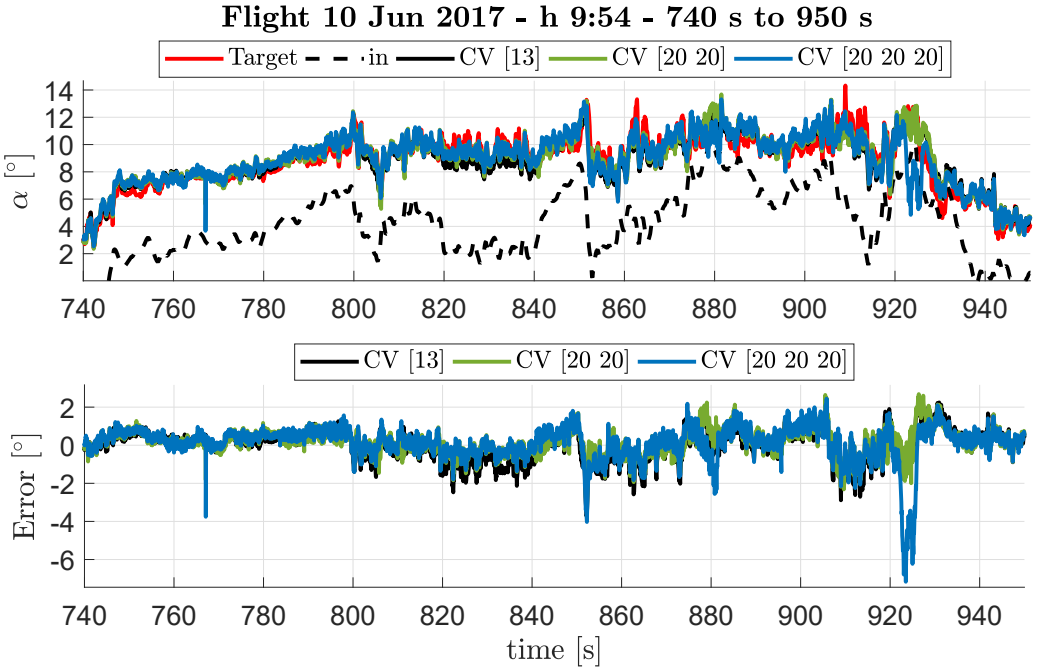


Figure 6.34: Details of the results on a particular section of a flight.

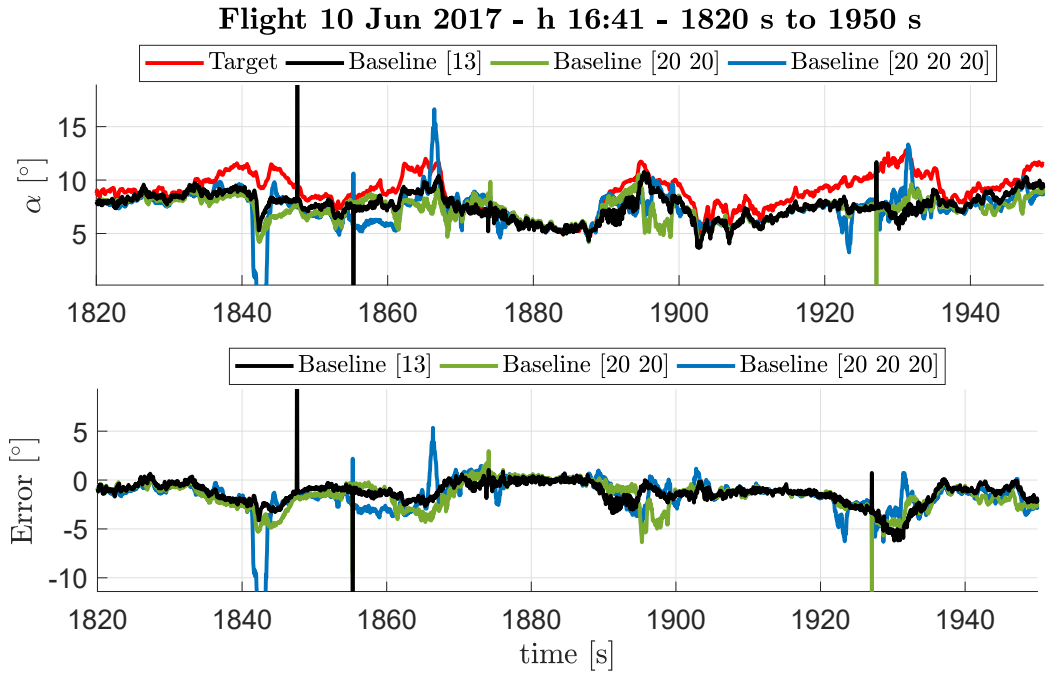


Figure 6.35: Details of the results on a particular section of a flight.

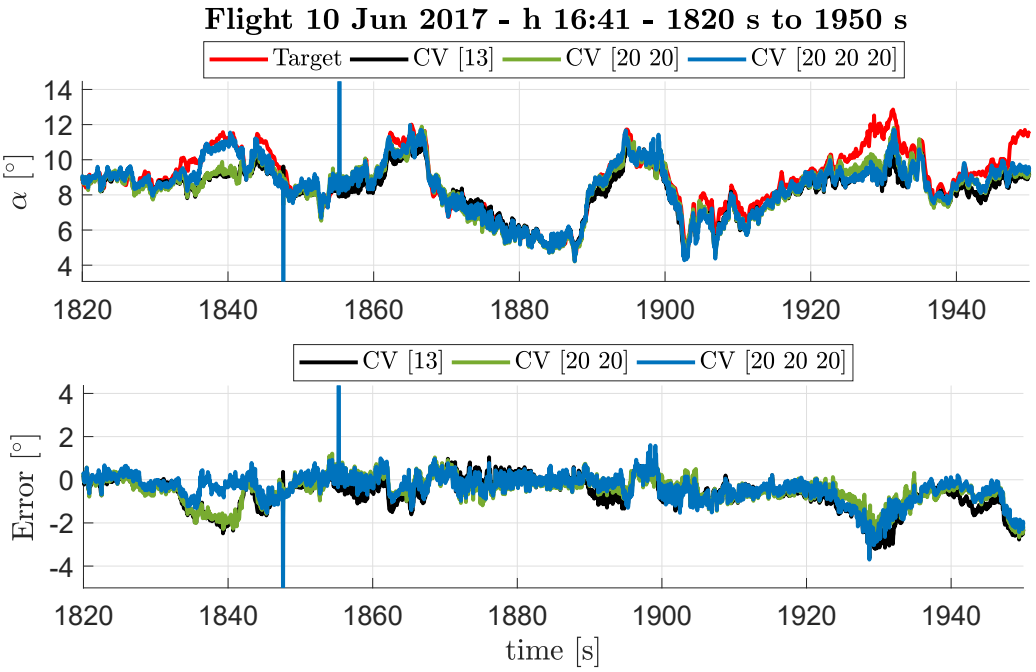


Figure 6.36: Details of the results on a particular section of a flight.

6.7 TS analysis

This last section of results concern the analysis of the TS following the methods shown in Chap. 4.

The performance of the VSs designed following the Manoeuver-based CV are considered the best among the various one. This should be confirmed by the value of the Density-Closeness function d_{nrd}, the L_C and A_C metrics. Moreover, it is interesting to compare the feature maps.

Figure 6.37 shows the distribution of the error normalized by the MSE obtained at the end of the training with respect to the value of the d_{nrd} metric assumed by the entries of the dataset. The trend confirms the thesis that increasing the d_{nrd} metric, the test error decreases quicker than linearly.

Figure 6.38 shows the value assumed by the L_C and A_C metrics on the various combinations generated by the Manoeuver-based CV method. It is interesting to notice that the subset with the minimum value of L_C and A_C is also the one that gives the best performance of the VS. Figure 6.39 shows how the TSs differ in terms of d_{nrd} values. It is shown that the best TS distributes on higher values than the other TSs. Although It does not have the higher minimum value and the TS # 8 seems to cover higher values than # 5, TS # 5 has the higher maximum value of d_{nrd}. It is hence sufficient to have some points at higher d_{nrd} to obtain better performance in training. This aspect can be explained by the fact that the relationship between the error and d_{nrd} is nonlinear. The observation in Figure 6.39 gives an evidence for the validity of the TS analysis method.

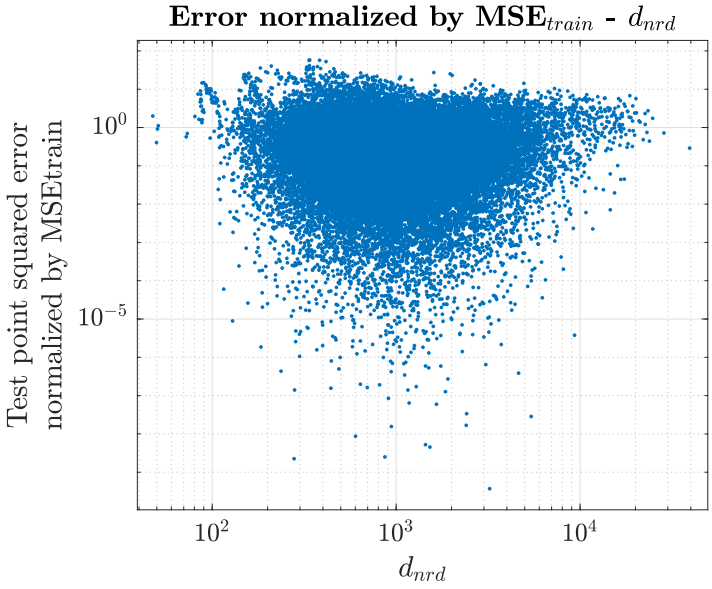


Figure 6.37: Normalized error distribution with respect to the d_{nrd} value.

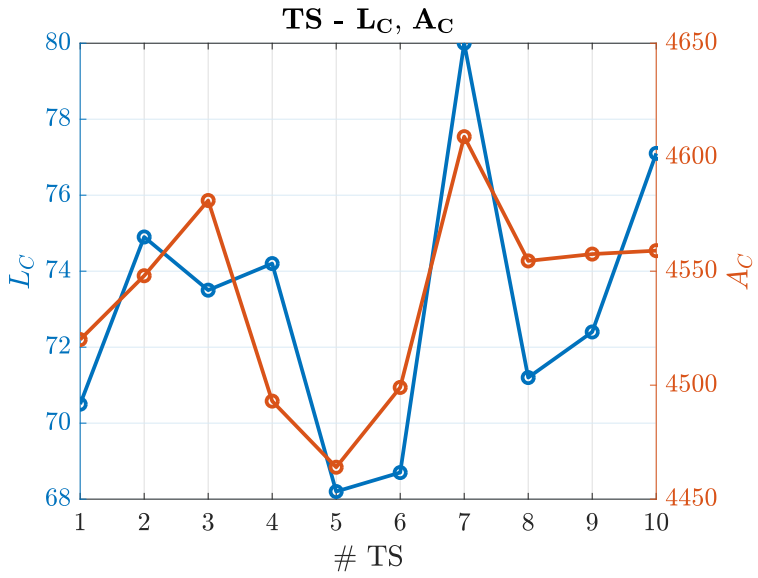


Figure 6.38: Values assumed by L_C and A_C metrics on the different combinations of TS generated during the Manoeuver-based CV.

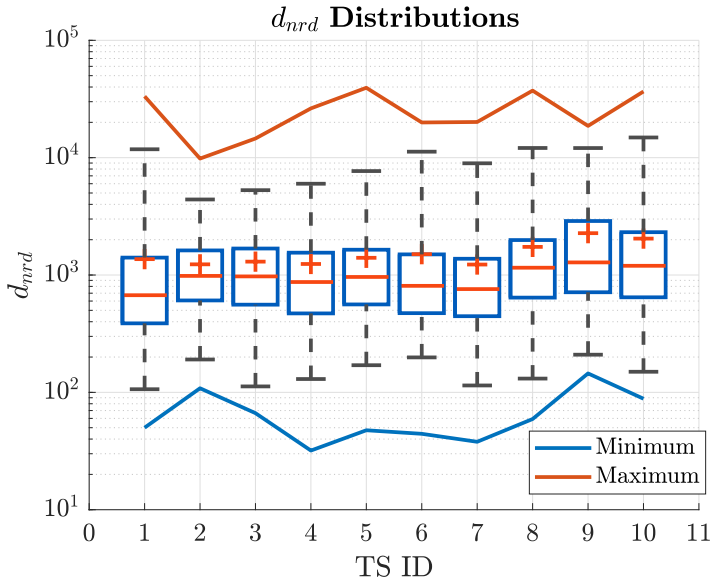


Figure 6.39: Distributions of the d_{nrd} among the different TSs.

6.8 Design of the AOS sensor

The methods proposed in this dissertation can be used also in the design of a VS for the AOS. However, the number of graphs that should be reported here is too high. For this reason, only the final design is reported. Given the input vector

in Eq. 1.10, the MBCV method has been conducted on the same dataset applied for the AOA target. The results of the MBCV method in terms of NSSE are shown in Figure 6.40, 6.41 and 6.42. The final uncertainty on the AOS variable is higher than the one obtained for AOA. This is in accordance with the results found in literature. This can be seen in terms of estimation error in Figure 6.43 and on the uncertainty maps in Figure 6.44. Please note that in Figure 6.44c the colorbar has been cut at 20° for sake of clarity but the maximum error shows a peak at $(\alpha, \beta) = (5.26°, 0.79°)$ of 50°.

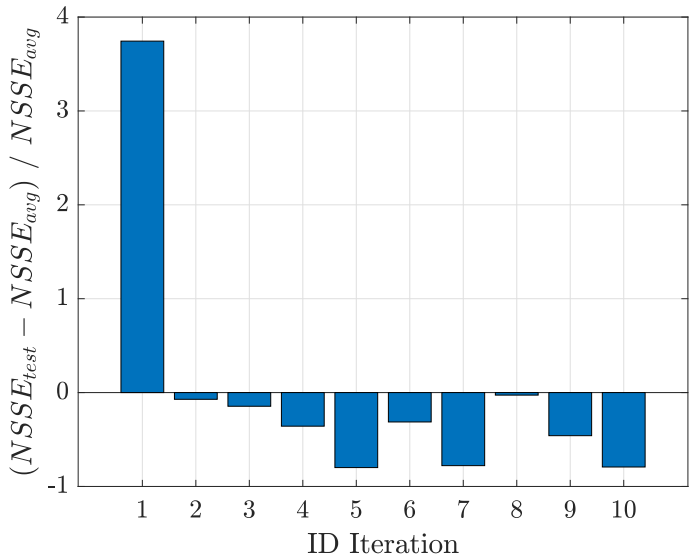


Figure 6.40: Validation NSSE values obtained with the application of the MBCV method, AOS

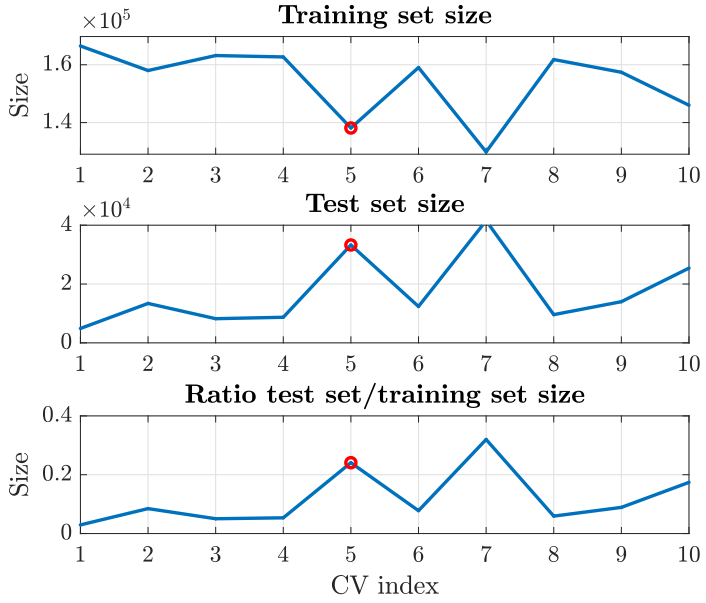


Figure 6.41: Size of the various combinations analysed during the manoeuver-based CV. The red circle points at the best partition in terms of training performance, AOS

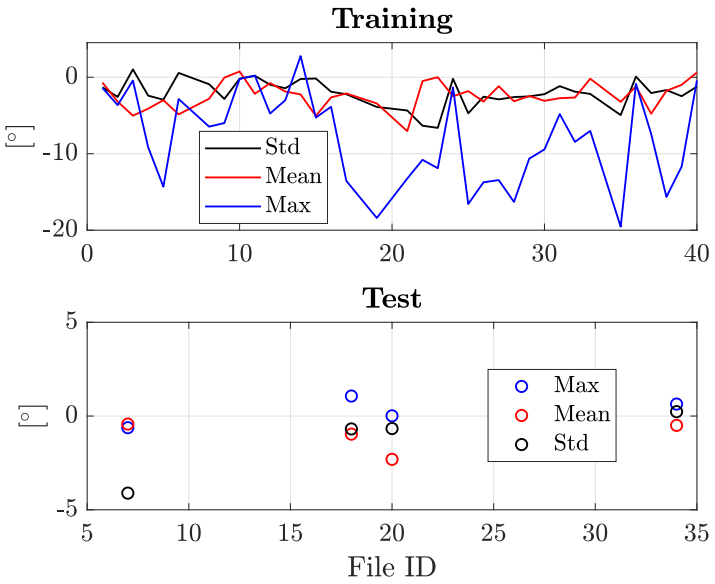


Figure 6.42: Statistics of ϵ among the various manoeuvers, AOS

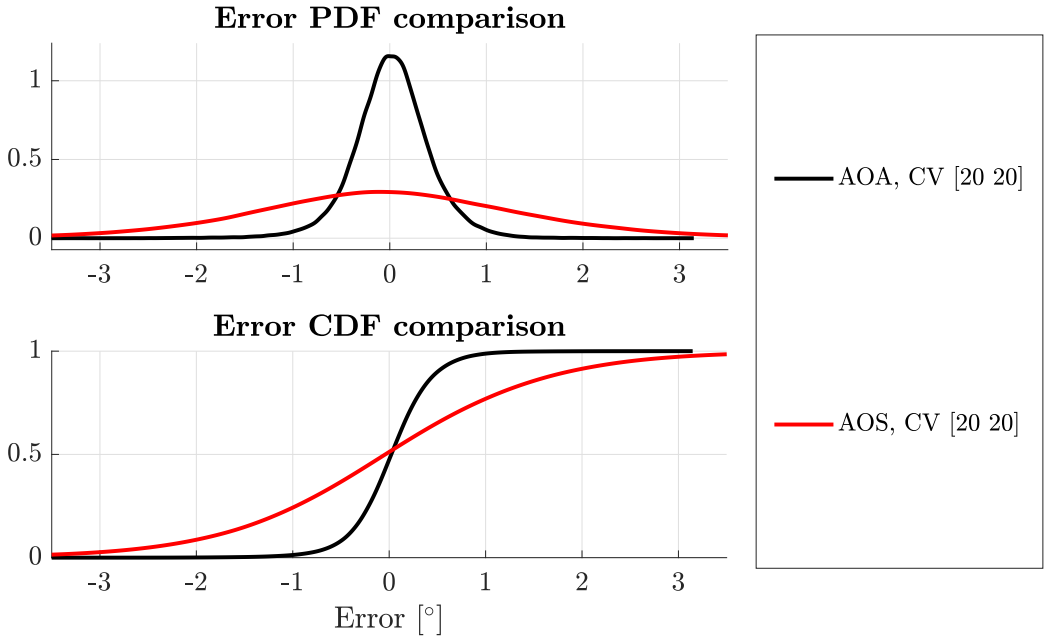
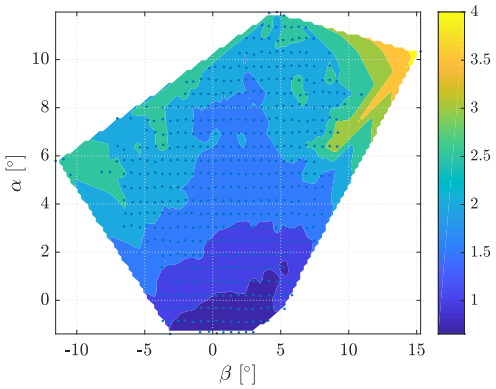
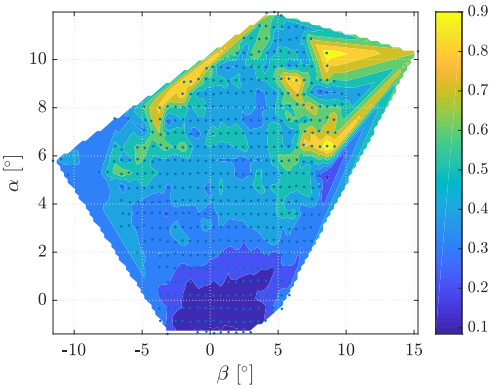


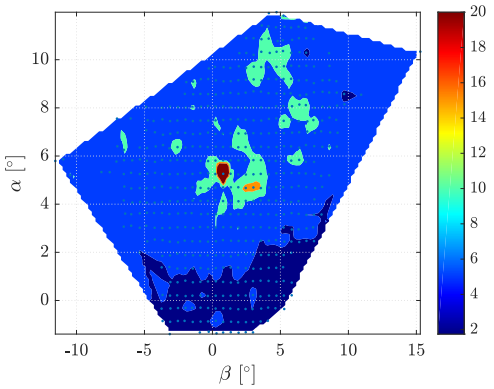
Figure 6.43: Comparison of the PDFs and CDFs obtained with MBCV using the same dataset for the estimation of AOA and AOS.



(a) 1σ values on the AOA/AOS envelope



(b) Dispersion of the 1σ values on the AOA/AOS envelope



(c) Maximum error on the AOA/AOS envelope (max value = 50°)

Figure 6.44: Uncertainty charts for the AOS VS

Chapter 7

Conclusions and future work

The aeronautical industry has recently seen two accidents with fatal results due to technical reasons on the Boeing 737 MAX 8. One of the causes was a fault in the physical sensor applied to measure the AOA. An alternative to the physical sensor can be the implementation of a synthetic sensor. Although a huge literature exists on the subject of synthetic sensors, most of the technological solutions proposed by researchers didn't go further TRL 5. Moreover, whereas for large AC the hardware redundancy can still be applied to cope with the fact that the ADS is a critical system, this is not always true on uncoming platforms as UAV or UAM.

The main reason behind the difficulty on the diffusion of VS has been identified on the total absence of a shared and applied metrological theory that could support a procedural design and hence give the possibility to define a regulatory basis for this kind of system. Starting from the literature review, this dissertation focuses on NN-based VS and it describes the typical approaches applied to demonstrate the capabilities of a VS based on NN. Then, several design methods are proposed and analysed in order to provide the measurement uncertainty of a synthetic sensors with a certain level of confidence. The results have been shown for the Smart-ADAHRS, a patented method based on NN. Some of them did not show the improvements able to justify its application. In Figure 7.1 the methods proposed in this dissertation have been classified based on their applicability. DA has been excluded from the Air Data estimation group because it seems unpractical to build a model so accurate to augment the dataset. This does not exclude that it could be done. At the same time, the Jacobian and Hessian matrices are not really useful in the design of an air data VS because of the nonlinearity of the functions.

The traditional comparison of the timeseries of the output and of the target has been showed to be useful but also limited and tricky. Due to the dimension of the problem in terms of number of variables involved, the visualization on a chart of which variable is the source of error during a particular manoeuver is not always possible. Moreover, the length of the timeseries considered in this dissertation is relevant and not limited to brief flight sections as often found in literature. This

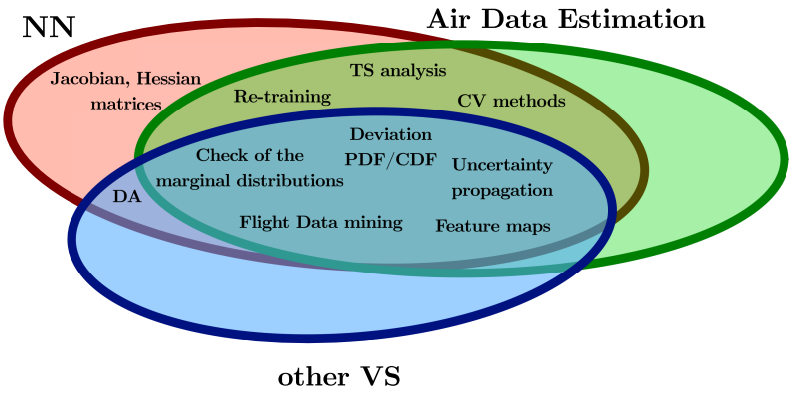


Figure 7.1: Subdivision of the methods based on the performance shown in this dissertation.

aspect increase the time needed to analyse the results once the design of a VS is concluded. Because of the engineering process of design is recursive, the time needed to design a VS increases to unacceptable level.

The majority of the proposed methods are then based on statistical analysis and data mining. Some of them analyse the data in order to provide indications on the results that can be obtained training a NN-based VS. Other analyse the results to give a global metric and they can be applied to almost any kind of VS.

To begin with, the PDF of the estimation error has been proposed, analogously to what is done for other kind of sensors. A kernel-based density estimation method has been applied to show that the MLP actually improves the initial estimation. It is interesting to notice the shape of the PDF, which is monomodal and almost simmetrical with mode close to 0°.

In terms of NN training analysis, the TS inclusivity verification has been introduced by means of hypercube coverage (following the procedure already presented by the same author in [110]). Although the marginal distributions does not provide information on the signal correlations, it still provides a first hint to the designer if the NN can asymptotically tend to the desired function thanks to the result of the UAT. If a test set contains several points outside the bounds provided by the TS, then nothing can be said a priori on the estimation error on those cases.

A modification of the CV procedure has been introduced. The manoeuver-based CV starts from the output of the automatic trim detector and it partitions a flight log based on its steady flight conditions. This method can help to find which are the manoeuvers that improve the training of a TS. The main difficulty is the ratio of number of entries applied for training and testing that must be controlled manually.

A new procedure for the analysis of a TS has been introduced. It is based on the definition of multivariate regression that is proven to be the result of an ideal training of an MLP. The analysis of the distribution of the target values given

the similar input vector is used to define three similarity measures: the Density-Closeness d_{nrd} function, the area of the cardinality of stationary points in the target conditional distribution with respect to the number of clusters A_C and the mean value of the same variable on the first n subdivision L_C. All of them have been tested both on a controlled situation based on an analytical function and on the flight data object of this dissertation. It has been empirically demonstrated that if a TS contains higher values of d_{nrd}, it results in a better training. In fact, the manoeuver-based CV converged to the same TS indicated as the better one by this analysis. The same agreement has been found studying the L_C and A_C values. The TS selected by the manoevuer-based CV is the one with minimum L_C and A_C value at the same time. This provides an evidence on the fact that if the number of stationary points quickly drops, the TS is better distributed.

Also, a data reduction has been proposed, similarly to what is done in ML. It culminates in a feature map, that shows the 84-th percentile on a 2D map related to the flight condition. On this map, the origin coincides with the symmetric steady state condition, moving along one of the two axis it is introduced asymmetry or unsteady motion. This chart helped to find a defect of the Smart-ADAHRS on the steady-state condition. In this case, the uncertainty was higher than in dynamic conditions, due to unbalancing of the TS. Thanks to this method, the problem has been identified and it was possible to act in order to increase the homogeneity in the results.

Another important aspect introduced is the sensitivity analysis of VS. The procedure traditionally applied to study the uncertainty related to nonlinear function has been used and the results have been reported on a AOA/AOS chart, as required by several aeronautical industry. This method allowed to compare the effect of the different MLP architectures and of the Manoeuver-based CV. Figures give evidence of the overfitting condition that can occour with an increased capacity and on the regularization that introduces the CV method. It is interesting that in most of the cases the ambiguity of the AOA/AOS chart is not problematic. In fact, each (α, β) point can be associated to several flight conditions, steady or unsteady. However, the dispersion of the distribution at any point is low. This is in contrast with the previous assumption of increased uncertainty in stationarity condition. However, it must be reminded that it depends of the number of points available.

Moreover, the recursive formulation of Jacobian and Hessian matrix of MLP have been analytically derived. Although these matrices can have several uses, in this dissertation they have been applied for the sensitivity analysis based on a linearization of the function. The nonlinearity of the function represented by the MLP together with the difficulty of definition of the input covariance matrix brought uncertainty values higher than those obtained with previous methods.

Data aumentation has been proposed to improve results in stationary flight condition. The additional set of data have been generated in two ways. First, a nonlinear AC model has been obtained based on the DATCOM formulation. A set

of simulations regarding the stick-fixed dynamics of a perturbed equilibrium condition has been conducted. Second, a recent paradigm of ML called GAN has been used. Unfortunately, results showed no improvement given by data augmentation. The reason has been associated to the matching of the model with the real AC. This is actually an evidence of the fact that the manual definition of a model is not a practical way of design a VS.

Starting from these methods, it is possible to define a shared metrological procedure so that a regulation on synthetic estimation can be obtained. A design flow is hence proposed as follows:

- analysis of the coverage of the α, β envelope for TS and test set
- analysis of the marginal distributions
- verification that the number of test points is at least the 15 % (better 25 %) of the TS cardinality, as common in ML, also in case of non-ML kernel
- analysis of the PDF/CDF of the estimation error
- estimation of the uncertainty propagation charts and use of this datum as comparison metrics
- application of the CV method is highly recommended

Additional architectural features can be added to the VS. A saturation of the output of the NN could avoid peaks coming from erroneous training. A NN can be trained to estimate AOA and AOS at the same time. Avoiding the application of the initial estimation $\hat{\alpha}$ as input signal to the NN reduces the sensitivity to the wind. However, those design choices can be analyzed only when the adopted design flow is able to catch the real differences. The suggested design flow can solve this ambiguity.

Future research can focus on both research and development. A natural progression of this work is the implementation of these techniques in the EU project MIDAS. The demonstration of the validity of this preliminary design flow into a project with TRL 7 could lead a transition to a shared procedure that can, optimistically, translate into a set of certification guidance notes. Moreover, further work could help a better understanding of the mechanism behind the data analysis techniques. From the applied machine learning point of view, a set of theoretically justified techniques would be of great help in the design of innovative systems. However, much effort should be done in demonstrating the validity of the approach based on similarity function, also in terms of comparison with methods that are already known in statistical analysis. At the same time, although the data augmentation techniques has showed to be not effective in this case, an advanced model identification method might help to obtain a set of augmented data that can have a

better fitting with the flight test recordings. Moreover, the quasi-steady state identification algorithm could be enhanced to mine also other flight conditions. This would ease the identification of the data for the evaluation of particular aerodynamic derivatives.

Following these techniques is considered by the author as the minimum set of operations needed in order to demonstrate the validity of a VS. Once these results are obtained, then it is possible to think to the actual application of synthetic sensor in real AC, potentially saving human lives.

Bibliography

[1] Angelo Lerro, Manuela Battipede, Piero Gili, and Alberto Brandl. "Advantages of Neural Network Based Air Data Estimation for Unmanned Aerial Vehicles". In: *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* 11.5 (2017), pp. 1016–1025. ISSN: PISSN:2010-376X, EISSN:2010-3778. DOI: 10.5281/zenodo.1130557. URL: http://waset.org/Publications?p=125.

[2] Bernard Etkin and Lloyd D. Reid. *Dynamics of flight*. Wiley New York, 1971. ISBN: 0-471-24620-4.

[3] Brian L. Stevens, Frank L. Lewis, and Eric N. Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[4] Mario Innocenti, Gianluca Carnasciali, and Francesco Nasuti. "Angle of attack guidance via robust approximate inversion". In: *Guidance, Navigation, and Control Conference and Exhibit*. Vol. 3. 1. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 1998, pp. 138–144. DOI: 10.2514/6.1998-4113. URL: http://arc.aiaa.org/doi/10.2514/6.1998-4113.

[5] Eugene A. Morelli. "Real-Time Aerodynamic Parameter Estimation Without Air Flow Angle Measurements". In: *Journal of Aircraft* 49.4 (July 2012), pp. 1064–1074. ISSN: 0021-8669. DOI: 10.2514/1.C031568. URL: http://arc.aiaa.org/doi/10.2514/1.C031568.

[6] Austin Murch. "A Flight Control System Architecture for the NASA AirSTAR Flight Test Infrastructure". In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. August. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 2008, pp. 1–8. ISBN: 978-1-60086-999-0. DOI: 10.2514/6.2008-6990. URL: http://arc.aiaa.org/doi/10.2514/6.2008-6990.

[7] Matthew Rhudy, Trenton Larrabee, Haiyang Chao, Yu Gu, and Marcello R Napolitano. "UAV Attitude , Heading , and Wind Estimation Using GPS / INS and an Air Data System". In: June 2014 (2013). DOI: 10.2514/6.2013-5201.

[8] Matthew B Rhudy, Yu Gu, Jason N. Gross, and Haiyang Chao. "Onboard Wind Velocity Estimation Comparison for Unmanned Aircraft Systems". In: *IEEE Transactions on Aerospace and Electronic Systems* 53.1 (Feb. 2017), pp. 55–66. ISSN: 0018-9251. DOI: 10.1109/TAES.2017.2649218. URL: http://ieeexplore.ieee.org/document/7807266/.

[9] *Aircraft Accident Investigation Report - Final - KNKT.18.10.35.04*. Tech. rep. Komite Nasional Keselamatan Transportasi, Oct. 2019.

[10] Ihab Samy, Ian Postlethwaite, and Da Wei Gu. "Survey and application of sensor fault detection and isolation schemes". In: *Control Engineering Practice* 19.7 (2011), pp. 658–674. ISSN: 09670661. DOI: 10.1016/j.conengprac.2011.03.002. URL: http://dx.doi.org/10.1016/j.conengprac.2011.03.002.

[11] K. C. Wong. "Aerospace Industry Opportunities in Australia - Unmanned Aerial Vehicles (UAVs)". In: August 1997 (2006).

[12] *Unmanned Aerial Vehicle (UAV) Market by Vertical, Class, System, Industry (Defense & Security, Agriculture, Construction & Mining, Media & Entertainment), Type, Mode of Operation, Range, Point of Sale, MTOW and Region - Global Forecast to 2025*. Feb. 2019. URL: www.researchandmarkets.com.

[13] Congress of the United States of America. *FAA Modernization and Reform Act of 2012*. 2012. URL: https://www.gpo.gov/fdsys/pkg/CRPT-112hrpt381/pdf/CRPT-112hrpt381.pdf.

[14] Claudia Stöcker, Rohan Bennett, Francesco Nex, Markus Gerke, and Jaap Zevenbergen. "Review of the Current State of UAV Regulations". In: *Remote Sensing* 9.5 (2017). ISSN: 2072-4292. DOI: 10.3390/rs9050459. URL: http://www.mdpi.com/2072-4292/9/5/459.

[15] Paul Freeman, Peter Seiler, and Gary J. Balas. "Air data system fault modeling and detection". In: *Control Engineering Practice* 21.10 (Oct. 2013), pp. 1290–1301. ISSN: 09670661. DOI: 10.1016/j.conengprac.2013.05.007. URL: https://doi.org/10.1016/j.conengprac.2013.05.007%20http://linkinghub.elsevier.com/retrieve/pii/S0967066113000993.

[16] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter. "Model-based fault diagnosis for aerospace systems: a survey". In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 226.10 (Oct. 2012), pp. 1329–1360. ISSN: 0954-4100. DOI: 10.1177/0954410011421717. URL: http://pig.sagepub.com/lookup/doi/10.1177/0954410011421717%20http://journals.sagepub.com/doi/10.1177/0954410011421717.

[17] Janos J. Gertler. "Survey of Model-Based Failure Detection and Isolation in Complex Plants". In: *IEEE Control Systems Magazine* 8.6 (1988), pp. 3–11. ISSN: 0272-1708. DOI: 10.1109/37.9163.

[18] Rolf Isermann. "Model-based fault-detection and diagnosis - Status and applications". In: *Annual Reviews in Control* 29.1 (2005), pp. 71–85. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2004.12.002. URL: http://www.sciencedirect.com/science/article/pii/S1367578805000052.

[19] Rolf K. Isermann and Peter Ballé. "Trends in the application of model-based fault detection and diagnosis of technical processes". In: *Control Engineering Practice* 5.5 (1997), pp. 709–719. ISSN: 0967-0661. DOI: https://doi.org/10.1016/S0967-0661(97)00053-1. URL: http://www.sciencedirect.com/science/article/pii/S0967066197000531.

[20] Ihab Samy and Da-Wei Gu. *Lecture Notes in Control and Information Sciences*. Springer-Verlag Berlin Heidelberg, 2011. ISBN: 9783540239505. DOI: 10.1007/978-3-642-24052-2.

[21] José Carlos M. Oliveira, Karen V. Pontes, Isabel Sartori, and Marcelo Embiruçu. "Fault Detection and Diagnosis in dynamic systems using Weightless Neural Networks". In: *Expert Systems with Applications* 84.Supplement C (2017), pp. 200–219. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.05.020. URL: http://www.sciencedirect.com/science/article/pii/S0957417417303366.

[22] Ryan Eubank, Ella Atkins, and Stephanie Ogura. "Fault Detection and Fail-Safe Operation with a Multiple-Redundancy Air-Data System". In: *AIAA Guidance, Navigation, and Control Conference*. Aug. 2010. ISBN: 978-1-60086-962-4.

[23] Peng Lu, Laurens Van Eykeren, Erik-Jan Van Kampen, and Q Chu. "Air Data Sensor Fault Detection and Diagnosis with Application to Real Flight Data". In: (Jan. 2015). DOI: https://doi.org/10.2514/6.2015-1311.

[24] Maha Kooli and Giorgio Di Natale. "A survey on simulation-based fault injection tools for complex systems". In: *2014 9th IEEE International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS)*. May 2014, pp. 1–6. DOI: 10.1109/DTIS.2014.6850649.

[25] Alfredo Garro and Andrea Tundis. "System reliability analysis: a Model-Based approach and a case study in the avionics industry". In: *Proceedings of the 3rd Air and Space International Conference (CEAS)*. 2011, pp. 24–28.

[26] Alfredo Garro and Andrea Tundis. "A model-based method for system reliability analysis". In: *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium*. Society for Computer Simulation International. 2012, p. 2.

[27] Haissam Ziade, Rafic Ayoubi, and Raoul Velazco. "A Survey on Fault Injection Techniques". In: *The International Arab Journal of Information Technology* 1.2 (2004), pp. 171–186. ISSN: 03600300. DOI: 10.1.1.167.966.

[28] L. Sankaralingam and C. Ramprasadh. "A comprehensive survey on the methods of angle of attack measurement and estimation in UAVs". In: *Chinese Journal of Aeronautics* November (Nov. 2019). ISSN: 10009361. DOI: 10.1016/j.cja.2019.11.003. URL: https://doi.org/10.1016/j.cja.2019.11.003%20https://linkinghub.elsevier.com/retrieve/pii/S1000936119304078.

[29] M. A. Menzies. "Integrated air data sensors". In: *The Aeronautical Journal (1968)* 105.1046 (2001), pp. 223–229. DOI: 10.1017/S000192400002546X.

[30] Floyd W. Hagen and Harald Seidel. "Deutsche airbus flight test of Rosemount smart probe for distributed air data systems". In: *[1993 Proceedings] AIAA/IEEE Digital Avionics Systems Conference*. Oct. 1993, pp. 110–117. DOI: 10.1109/DASC.1993.283561.

[31] Georges Hardier, Cédric Seren, and Pierre Ezerzere. "Model–Based Techniques for Virtual Sensing of Longitudinal Flight Parameters". In: *International Journal of Applied Mathematics and Computer Science* 25.1 (Mar. 2015), pp. 23–38. ISSN: 2083-8492. DOI: 10.1515/amcs-2015-0002. URL: http://content.sciendo.com/view/journals/amcs/25/1/article-p23.xml.

[32] *Heated Probe, Miniature, Pitot or Pitot-Static*. 4207. Rev. K. SpaceAge Control, Inc. Nov. 2006.

[33] David Allerton and Huamin Jia. "A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems". In: 58 (Sept. 2005), pp. 405–417.

[34] Bruno Tranchero and Cosimo Latorre. "Neural Network-Based Virtual Sensors In Flight Control Systems". In: *IFAC Proceedings Volumes* 34.15 (Sept. 2001), pp. 416–421. ISSN: 14746670. DOI: 10.1016/S1474-6670(17)40763-4. URL: http://linkinghub.elsevier.com/retrieve/pii/S1474667017407634%20https://linkinghub.elsevier.com/retrieve/pii/S1474667017407634.

[35] G. Palma, O. Scognamiglio, and M. Lavorgna. "Low Cost Virtual Pressure Sensor". In: *SAE Technical Paper*. SAE International, Mar. 2004, pp. 1–15. DOI: 10.4271/2004-01-1367. URL: http://dx.doi.org/10.4271/2004-01-1367.

[36] Yuebin Yu, Denchai Woradechjumroen, and Daihong Yu. "Virtual surface temperature sensor for multi-zone commercial buildings". In: *Energy Procedia* 61 (2014), pp. 21–24. ISSN: 18766102. DOI: 10.1016/j.egypro.2014.11.896. URL: http://dx.doi.org/10.1016/j.egypro.2014.11.896.

[37] Philippe Goupil, Rémy Dayre, and Patrice Brot. "From theory to flight tests: Airbus Flight Control System TRL5 achievements". In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 10562–10567. ISSN: 1474-6670. DOI: http://dx.doi.org/10.3182/20140824-6-ZA-1003.02547. URL: http://www.sciencedirect.com/science/article/pii/S147466701643291X.

[38] *EASA airworthiness directive ad no.: 2013-0068*. Tech. rep. 29 March. EASA, 2013.

[39] *EASA airworthiness directive ad no.: 2015-0135*. Tech. rep. 15 July. EASA, 2015.

[40] *Magnetic angle of attack sensor*. Tech. rep. wO 01/77622 A2. 2001.

[41] D. H. Lenschow. "Vanes for Sensing Incidence Angles of the Air from an Aircraft". In: *Journal of Applied Meteorology* 10.6 (Dec. 1971), pp. 1339–1343. ISSN: 0021-8952. DOI: 10.1175/1520-0450(1971)010<1339:VFSIAO>2.0.CO;2. URL: http://journals.ametsoc.org/doi/abs/10.1175/1520-0450%7B%5C%%7D281971%7B%5C%%7D29010%7B%5C%%7D3C1339%7B%5C%%7D3AVFSIAO%7B%5C%%7D3E2.0.CO%7B%5C%%7D3B2.

[42] *Multi-function air data sensing probe having an angle of attack vane*. Tech. rep. uS 6941805 B2. 2005.

[43] *Outside Air Temperature (OAT) Sensor Series 0129*. Tech. rep. Burnsville, USA. UTC Aerospace Systems, 2005.

[44] *Angle of attack (AOA) sensors*. Tech. rep. Burnsville, USA. UTC Aerospace Systems, 2005.

[45] *State-of-the-art air data products solution guide*. Tech. rep. Palmdale, USA. SpaceAge Control.

[46] William Denson, Greg Chandler, William Crowell, and Rick Wanner. *Non-electronic parts reliability data 1991*. Tech. rep. Reliability Analysis Center Griffiss AFB NY, 1991.

[47] Sergio Chiesa, Sara Cresto Aleina, Giovanni Antonio Di Meo, Roberta Fusaro, and Nicole Viola. "Autonomous take-off and landing for unmanned aircraft system: Risk and safety analysis". In: *29th Congress of the International Council of the Aeronautical Sciences, ICAS 2014*.

[48] Daniel A. Reasor Jr., Keerti K. Bhamidipati, and Reagan K. Woolf. "Numerical Predictions of Static-Pressure-Error Corrections for a Modified T-38C Aircraft". In: *Journal of Aircraft* 52.4 (2015), pp. 1326–1335. DOI: https://doi.org/10.2514/1.C032925.

[49] Jehanzeb Masud, Shakil Sheikh Adnan Latif, and Khalid Parvez. "Robust Design of an Aerodynamic Compensation Pitot-Static Tube for Supersonic Aircraft". In: *Journal of Aircraft* 44.1 (2007), pp. 163–169. DOI: https://doi.org/10.2514/1.22775.

[50] EASA. *Definitions and abbreviations used in Certification Specifications for products, parts and appliances*. 2007.

[51] Department of Defense of the USA. *Definitions of Terms for Reliability and Maintainability*. MIL-STD721C. 1981.

[52] NASA. *Safety and Mission Assurance Acronyms, Abbreviations, and Definitions*. NASA-STD 8709.22 w/ Change 1. 2011.

[53] National Aeronautics and Space Administration. *NASA Reliability and Maintainability (R&M) Standard for Spaceflight and Support Systems*. NASA-STD-8729.1A. 2017.

[54] International Organization for Standardization. *Condition monitoring and diagnostics of machines - Vocabulary*. ISO13372-2012. 2012.

[55] J.B. (Sperry Flight Systems Division) Dendy and K.G. (Sperry Flight Systems Division) Transier. *Angle-of-Attack Computation Study (AFFDL-TR-69-93)*. Tech. rep. 1969.

[56] Francesco De Vivo, Manuela Battipede, Piero Gili, and Alberto Brandl. "Ill-conditioned problems improvement adapting Joseph covariance formula to non-linear Bayesian filters". In: *WSEAS Transactions on Electronics*. Ed. by WSEAS Press. Vol. 7. 2016, pp. 18–25. DOI: 10.13140/RG.2.1.3027.0960.

[57] Francesco De Vivo, Alberto Brandl, Manuela Battipede, and Piero Gili. "Joseph covariance formula adaptation to Square-Root Sigma-Point Kalman filters". In: *Nonlinear Dynamics* 88.3 (2017), pp. 1969–1986. ISSN: 0924-090X. DOI: 10.1007/s11071-017-3356-x. URL: https://doi.org/10.1007/s11071-017-3356-x.

[58] Peter McCullagh. "What is a statistical model?" In: *Annals of Statistics* 30.5 (2002), pp. 1225–1267. ISSN: 00905364. DOI: 10.1214/aos/1035844977.

[59] Duane B Freeman. *Angle of Attack Computation System*. Tech. rep. AFFDL-TR-73-89. Air Force Flight Dynamics Laboratory, 1973.

[60] Joseph Zeis, Jr., Heather Lambert, Robert Calico, and Daniel Gleason. "Angle of attack estimation using an inertial reference platform". In: *15th Atmospheric Flight Mechanics Conference*. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 1988, pp. 180–190. DOI: 10.2514/6.1988-4351. URL: http://arc.aiaa.org/doi/10.2514/6.1988-4351.

[61] Joseph E. Zeis. "Angle of attack estimation using an inertial reference platform". School of Engineering of the Air Force Institute of Technology, 1988.

[62] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (1960), p. 35. ISSN: 00219223. DOI: 10.1115/1.3662552.

[63] Leonard A. McGee and Stanley F. Schmidt. "Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry". In: *NASA Technical Memorandum* November (1985), p. 21. URL: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19860003843%7B%5C_%7D1986003843.pdf.

[64] Thomas J. Rohloff, Stephen A. Whitmore, and Ivan Catton. "Air data sensing from surface pressure measurements using a neural network method". In: *AIAA Journal* 36.11 (1998), pp. 2094–2101. ISSN: 00011452. DOI: 10.2514/2.312.

[65] Richard Colgren, Michael Frye, and Wayne Olson. "A proposed system architecture for estimation of angle-of-attack and sideslip angle". In: *Guidance, Navigation, and Control Conference and Exhibit*. c. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 1999, pp. 743–750. DOI: 10.2514/6.1999-4078. URL: http://arc.aiaa.org/doi/10.2514/6.1999-4078.

[66] Richard Colgren and Keith Martin. "Flight test validation of sideslip estimation using inertial accelerations". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. August. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 2000. ISBN: 978-1-62410-301-8. DOI: 10.2514/6.2000-4448. URL: http://arc.aiaa.org/doi/10.2514/6.2000-4448.

[67] Richard D. Colgren. *Method and System for Elimination and Correction of Angle of Attack and Sideslip Angle from Acceleration Measurements*. Patent No. 6,273,370 B1. 2001.

[68] C. Westhelle. "X-38 Backup Air Data System (AeroDAD)". In: *40th AIAA Aerospace Sciences Meeting & Exhibit*. January. Reston, Virigina: American Institute of Aeronautics and Astronautics, Jan. 2002. DOI: 10.2514/6.2002-7. URL: http://arc.aiaa.org/doi/10.2514/6.2013-4568%20http://arc.aiaa.org/doi/10.2514/6.2002-7.

[69] Thirumalainambi Rajkumar and Jorge Bardina. "Training data requirement for a neural network to predict aerodynamic coefficients". In: ed. by Anthony J. Bell, Mladen V. Wickerhauser, and Harold H. Szu. Apr. 2003, p. 92. DOI: 10.1117/12.486343. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.486343.

[70] Thirumalainambi Rajkumar and Jorge Bardina. "Prediction of aerodynamic coefficients using neural networks for sparse data". In: *FLAIRS*. 2002, pp. 242–246.

[71] Kevin A. Wise. *Computational Air Data System for Angle-Of-Attack and Angle-Of-Sideslip*. US 6,928,341 B2. 2005.

[72] Paraskevi A. Samara, George N. Fouskitakis, John S. Sakellariou, and Spilios Fassois. "Aircraft angle-of-attack virtual sensor design via a functional pooling narx methodology". In: *2003 European Control Conference (ECC)*. IEEE, Sept. 2003, pp. 1816–1821. ISBN: 978-3-9524173-7-9. DOI: 10.23919/ECC.2003.7085229. URL: https://ieeexplore.ieee.org/document/7085229/.

[73] Francesco Nebula, Roberto Palumbo, Gianfranco Morani, and Federico Corraro. "Virtual Air Data System Architecture for Space Reentry Applications". In: *Journal of Spacecraft and Rockets* 46.4 (July 2009), pp. 818–828. ISSN: 0022-4650. DOI: 10.2514/1.42485. URL: https://arc.aiaa.org/doi/10.2514/1.42485.

[74] Georges Hardier, Cédric Seren, Pierre Ezerzere, and Guilhem Puyou. "Aerodynamic model inversion for virtual sensing of longitudinal flight parameters". In: *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE, Oct. 2013, pp. 140–145. ISBN: 978-1-4799-2855-2. DOI: 10.1109/SysTol.2013.6693835. URL: http://ieeexplore.ieee.org/document/6693835/.

[75] F. Adhika Pradipta Lie and Demoz Gebre-Egziabher. "Synthetic Air Data System". In: *Journal of Aircraft* 50.4 (July 2013), pp. 1234–1249. ISSN: 0021-8669. DOI: 10.2514/1.C032177. URL: http://arc.aiaa.org/doi/10.2514/1.C032177.

[76] F. Adhika Pradipta Lie and Demoz Gebre-Egziabher. "Sensitivity Analysis of Model-based Synthetic Air Data Estimators". In: *AIAA Guidance, Navigation, and Control Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Jan. 2015, pp. 1–18. ISBN: 978-1-62410-339-1. DOI: 10.2514/6.2015-0081. URL: http://arc.aiaa.org/doi/10.2514/6.2015-0081.

[77] J.E. Cashman, B.D. Kelly, and B.N. Nield. *Operational use of angle of attack on modern commercial jet airplanes*. Tech. rep. 12. 2000, pp. 12–21. URL: http://www.boeing.com/commercial/aeromagazine/aero%7B%5C_%7D12/attack%7B%5C_%7Dstory.html.

[78] Mario Luca Fravolini, Matthew Rhudy, Srikanth Gururajan, Silvia Cascianelli, and Marcello Napolitano. "Experimental Evaluation of Two Pitot Free Analytical Redundancy Techniques for the Estimation of the Airspeed of an UAV". In: *SAE International Journal of Aerospace* 7.1 (Sept. 2014), pp. 2014–01–2163. ISSN: 1946-3901. DOI: 10.4271/2014-01-2163. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-84924348522%7B%5C&%7DpartnerID=tZOtx3y1%20https://www.sae.org/content/2014-01-2163/.

[79] Matthew B. Rhudy, Mario L. Fravolini, Yu Gu, Marcello R. Napolitano, Srikanth Gururajan, and Haiyang Chao. "Aircraft model-independent airspeed estimation without pitot tube measurements". In: *IEEE Transactions on Aerospace and Electronic Systems* 51.3 (July 2015), pp. 1980–1995. ISSN: 0018-9251. DOI: 10.1109/TAES.2015.130631. URL: http://ieeexplore.ieee.org/document/7272846/.

[80] Tor A. Johansen, Andrea Cristofaro, Kim Sorensen, Jakob M. Hansen, and Thor I. Fossen. "On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors". In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, June 2015, pp. 510–519. ISBN: 978-1-4799-6010-1. DOI: 10.1109/ICUAS.2015.7152330. URL: http://ieeexplore.ieee.org/document/7152330/.

[81] P. Lu, L. Van Eykeren, E. van Kampen, C. C. de Visser, and Q. P. Chu. "Adaptive Three-Step Kalman Filter for Air Data Sensor Fault Detection and Diagnosis". In: *Journal of Guidance, Control, and Dynamics* 39.3 (Mar. 2016), pp. 590–604. ISSN: 0731-5090. DOI: 10.2514/1.G001313. URL: http://arc.aiaa.org/doi/10.2514/1.G001313.

[82] Francesco Schettini, Gianpietro Di Rito, Roberto Galatolo, and E. Denti. "Sensor fusion approach for aircraft state estimation using inertial and air-data systems". In: *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*. 1. IEEE, June 2016, pp. 624–629. ISBN: 978-1-4673-8292-2. DOI: 10.1109/MetroAeroSpace.2016.7573289. URL: http://ieeexplore.ieee.org/document/7573289/.

[83] G. Alcalay, C. Seren, G. Hardier, M. Delporte, and P. Goupil. "Development of virtual sensors to estimate critical aircraft flight parameters". In: *IFAC-PapersOnLine* 50.1 (July 2017), pp. 14174–14179. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.2083. URL: https://doi.org/10.1016/j.ifacol.2017.08.2083%20https://linkinghub.elsevier.com/retrieve/pii/S240589631732743X.

[84] Francesco Schettini, Gianpietro Di Rito, and Eugenio Denti. "Aircraft flow angles calibration via observed-based wind estimation". In: *Aircraft Engineering and Aerospace Technology* 91.7 (July 2019), pp. 1033–1038. ISSN: 1748-8842. DOI: 10.1108/AEAT-06-2017-0145. URL: https://www.emeraldinsight.com/doi/10.1108/AEAT-06-2017-0145.

[85] Chen Lu, Rong-bing Li, Jian-ye Liu, and Ting-wan Lei. "Air Data Estimation by Fusing Navigation System and Flight Control System". In: *Journal of Navigation* 71.5 (Sept. 2018), pp. 1231–1246. ISSN: 0373-4633. DOI: 10.1017/S037346331800022X. URL: https://www.cambridge.org/core/product/identifier/S037346331800022X/type/journal%7B%5C_%7Darticle.

[86] Nicola De Divitiis. "Wind Estimation on a Lightweight Vertical-Takeoff-and-Landing Uninhabited Vehicle". In: *Journal of Aircraft* 40.4 (July 2003), pp. 759–767. ISSN: 0021-8669. DOI: 10.2514/2.3155. URL: http://arc.aiaa.org/doi/10.2514/2.3155.

[87] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Andrew Mathers, and Sami Haddadin. "The flying anemometer: Unified estimation of wind velocity from aerodynamic power and wrenches". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 1637–1644. ISBN: 978-1-5090-3762-9. DOI: 10.1109/IROS.2016.7759264. URL: http://ieeexplore.ieee.org/document/7759264/.

[88] Am Cho, Jihoon Kim, Sanghyo Lee, and Changdon Kee. "Wind Estimation and Airspeed Calibration using a UAV with a Single-Antenna GPS Receiver and Pitot Tube". In: *IEEE Transactions on Aerospace and Electronic Systems* 47.1 (Jan. 2011), pp. 109–117. ISSN: 0018-9251. DOI: 10.1109/TAES.2011.5705663. URL: http://ieeexplore.ieee.org/document/5705663/.

[89] Jack W. Langelaan, Nicholas Alley, and James Neidhoefer. "Wind Field Estimation for Small Unmanned Aerial Vehicles". In: *Journal of Guidance, Control, and Dynamics* 34.4 (July 2011), pp. 1016–1030. ISSN: 0731-5090. DOI: 10.2514/1.52532. URL: http://arc.aiaa.org/doi/10.2514/1.52532.

[90] H. Ryu and S. Park. "Vision-based wind and position estimation with fixed-wing unmanned aerial vehicle". In: *Journal of Guidance, Control, and Dynamics* 41.10 (2018), pp. 2281–2290. ISSN: 07315090. DOI: 10.2514/1.G003646.

[91] Kerry Sun, Christopher D. Regan, and Demoz Gebre Egziabher. "GNSS/INS based estimation of air data and wind vector using flight maneuvers". In: *2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 - Proceedings* (2018), pp. 838–849. DOI: 10.1109/PLANS.2018.8373461.

[92] Rongbing Li, Chen Lu, Jianye Liu, and Tingwan Lei. "Air Data Estimation Algorithm under Unknown Wind Based on Information Fusion". In: *Journal of Aerospace Engineering* 31.5 (Sept. 2018), p. 04018072. ISSN: 0893-1321. DOI: 10.1061/(ASCE)AS.1943-5525.0000889. URL: http://ascelibrary.org/doi/10.1061/%7B%5C%%7D28ASCE%7B%5C%%7D29AS.1943-5525.0000889.

[93] Andreas Wenz, Tor Arne Johansen, and Andrea Cristofaro. "Combining model-free and model-based angle of attack estimation for small fixed-wing UAVs using a standard sensor suite". In: *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, June 2016, pp. 624–632. ISBN: 978-1-4673-9334-8. DOI: 10.1109/ICUAS.2016.7502583. URL: http://ieeexplore.ieee.org/document/7502583/.

[94] Angelo Lerro, Manuela Battipede, and Piero Gili. *System and process for measuring and evaluating air and inertial data*. Patent No. EP3022565A2. 2013.

[95] Manuela Battipede, Piero Gili, and Angelo Lerro. "Neural Networks for Air Data Estimation: Test of Neural Network Simulating Real Flight Instruments". In: *Engineering Applications of Neural Networks: 13th International Conference, EANN 2012, London, UK, September 20-23, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 282–294. ISBN: 978-3-642-32909-8. DOI: 10.1007/978-3-642-32909-8_29. URL: http://dx.doi.org/10.1007/978-3-642-32909-8%5C_29.

[96] Feb. 2020. URL: https://cordis.europa.eu/project/id/821140.

[97] Jürgen Schmidhuber. "Deep Learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2014.09.003. eprint: 1404.7828. URL: http://dx.doi.org/10.1016/j.neunet.2014.09.003.

[98] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review". In: *International Journal of Automation and Computing* 14.5 (Oct. 2017), pp. 503–519. ISSN: 1751-8520. DOI: 10.1007/s11633-017-1054-2. URL: https://doi.org/10.1007/s11633-017-1054-2.

[99] Kenneth Levenberg. "A method for the solution of certain non-linear problems in least squares". In: *Quarterly of applied mathematics* 2.2 (1944), pp. 164–168.

[100] Donald W Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441.

[101] Martin Riedmiller and Heinrich Braun. "A direct adaptive method for faster backpropagation learning: The RPROP algorithm". In: *IEEE international conference on neural networks*. IEEE. 1993, pp. 586–591.

[102] AR Webb, David Lowe, and Mark D Bedworth. *A comparison of nonlinear optimisation strategies for feed-forward adaptive layered networks*. Tech. rep. Royal Signals and Radar Establishment Malvern (UK), 1988.

[103] Manuela Battipede, Piero Gili, Angelo Lerro, Silvio Caselle, and Paolo Gianardi. "Development of Neural Networks for Air Data Estimation: Training of Neural Network Using Noise-Corrupted Data". In: *3rd CEAS Air & Space Conference, 21st AIDAA Congress*. 2011, pp. 1–10. ISBN: 9788896427187.

[104] Manuela Battipede, Mario Cassaro, Piero Gili, and Angelo Lerro. "Novel Neural Architecture for Air Data Angle Estimation". In: *Engineering Applications of Neural Networks: 14th International Conference, EANN 2013, Halkidiki, Greece, September 13-16, 2013 Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 313–322. ISBN: 978-3-642-41013-0. DOI: 10.1007/978-3-642-41013-0_32. URL: http://dx.doi.org/10.1007/978-3-642-41013-0%5C_32.

[105] Angelo Lerro, Manuela Battipede, Piero Gili, and Alberto Brandl. "Survey on a Neural Network for Non Linear Estimation of Aerodynamic Angles". In: *Proceedings of the 2017 Intelligent Systems conference (IntelliSys)*. Vol. 1. IEEE, 2017, pp. 929–935. ISBN: 978-1-5090-6435-9. DOI: 10.1109/IntelliSys.2017.8324240.

[106] Angelo Lerro, Manuela Battipede, Alberto Brandl, Piero Gili, Alberto Luigi Michele Rolando, and Lorenzo Trainelli. "Test in Operative Environment of an Artificial Neural Network for Aerodynamic Angles Estimation". In: *28th Society of Flight Test Engineers European Chapter Symposium (SFTE-EC 2017)*. 2017, pp. 1–12.

[107] Alberto Brandl, Manuela Battipede, Piero Gili, and Angelo Lerro. "Sensitivity Analysis of a Neural Network based Avionic System by Simulated Fault and Noise Injection". In: *2018 AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics, 2018. DOI: doi:10.2514/6.2018-0122. URL: https://doi.org/10.2514/6.2018-0122.

[108] Alberto Brandl, Angelo Lerro, Manuela Battipede, and Piero Gili. "Air Data Virtual Sensor: a Data-Driven Approach to Identify Flight Test Data Suitable for the Learning Process". In: *5th CEAS Conference on Guidance, Navigation and Control*. 2019, pp. 1–16.

[109] Angelo Lerro, Manuela Battipede, Piero Gili, and Alberto Brandl. "Aerodynamic angle estimation: comparison between numerical results and operative environment data". In: *CEAS Aeronautical Journal* (Sept. 2019). ISSN: 1869-5590. DOI: 10.1007/s13272-019-00417-x. URL: https://doi.org/10.1007/s13272-019-00417-x.

[110] Angelo Lerro, Manuela Battipede, Piero Gili, and Alberto Brandl. "Comparison Between Numerical Results and Operative Environment Data on Neural Network for Air Data Estimation". In: *PROCEEDINGS of the 6th CEAS Air and Space Conference Aerospace Europe 2017*. Vol. Unico. 2017, pp. 1–20.

[111] Kristina Olausson and Christopher Skogh. "Preparing for Ground Test of the First Gripen E Test Aircraft: Efficient Testing in Rigs and Simulators". In: *Proceedings of the 28th Society of Flight Test Engineers (SFTE) European Chapter Symposium*. 2017.

[112] *IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros*. 2008.

[113] Alberto Brandl, Manuela Battipede, and Piero Gili. "An Approach for the Analysis and Creation of an Effective Training Set". In: *Submitted for acceptance* (2019).

[114] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press Oxford, 1995. ISBN: 978-0198538646.

[115] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006. ISBN: 978-0-387-31073-2.

[116] David Kriesel. *A Brief Introduction to Neural Networks*. 2007. URL: http://www.dkriesel.com/en/science/neural%7B%5C_%7Dnetworks.

[117] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274. URL: https://doi.org/10.1007/BF02551274.

[118] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(91)90009-T. URL: http://www.sciencedirect.com/science/article/pii/089360809190009T.

[119] Jean-Gabriel Attali and Gilles Pagès. "Approximations of Functions by a Multilayer Perceptron: a New Approach". In: *Neural Networks* 10.6 (Aug. 1997), pp. 1069–1081. ISSN: 08936080. DOI: 10.1016/S0893-6080(97)00010-5. URL: http://linkinghub.elsevier.com/retrieve/pii/S0893608097000105.

[120] J.L. Castro, C.J. Mantas, and J.M. Benítez. "Neural networks with a continuous squashing function in the output are universal approximators". In: *Neural Networks* 13.6 (July 2000), pp. 561–563. ISSN: 08936080. DOI: 10.1016/S0893-6080(00)00031-9. URL: http://linkinghub.elsevier.com/retrieve/pii/S0893608000000319.

[121] Sylvain Arlot and Alain Celisse. "A survey of cross-validation procedures for model selection". In: *Statist. Surv.* 4 (2010), pp. 40–79. DOI: 10.1214/09-SS054. URL: https://doi.org/10.1214/09-SS054.

[122] Andrew R. Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information Theory* 39.3 (1993), pp. 930–945. ISSN: 00189448. DOI: 10.1109/18.256500.

[123] Andrew R. Barron. "Approximation and estimation bounds for artificial neural networks". In: *Machine Learning* 14.1 (Jan. 1994), pp. 115–133. ISSN: 0885-6125. DOI: 10.1007/BF00993164. URL: http://link.springer.com/10.1007/BF00993164.

[124] European Commission. *Part 19 - Commission Decision C(2014)4995*. 2014.

[125] Lorenzo Trainelli, Alberto Rolando, Giovanni Bonaita, and Paolo Chimetto. "Experiences in academic flight testing education". In: *Aircraft Engineering and Aerospace Technology* 86.1 (Dec. 2013), pp. 56–66. ISSN: 0002-2667. DOI: 10.1108/AEAT-10-2012-0178. URL: https://www.engineeringvillage.com/blog/document.url?mid=cpx%7B%5C_%7DM282a52e2143729812faM5bfa2061377553%7B%5C&%7Ddatabase=cpx%20https://www.emerald.com/insight/content/doi/10.1108/AEAT-10-2012-0178/full/html.

[126] Saverio Oldani. "Comprehensive Flight Testing and Data Analysis for a New Ultralight Aircraft". Politecnico di Milano, 2017.

[127] Francesco Battaini. "Identification of the basic aerodynamic model of an ultralight aircraft from flight test data". MA thesis. Politecnico di Milano, 2017.

[128] Alberto Rolando, Federico Rossi, Tommaso Castelletti, and Federico Reghenzani. "Mnemosine Mark-V: The Fifth Generation of an Ultra Light Machine-Dedicated FTI System". In: *27th Annual Society of Flight Test Engineers European Chapter Symposium*. 2016, pp. 1–12.

[129] Lorenzo Trainelli and Alberto Rolando. "Reliable and Cost-Effective Flight Testing of Ultralight Aircraft". In: *Journal of Aircraft* 48.4 (July 2011), pp. 1342–1350. ISSN: 0021-8669. DOI: 10.2514/1.C031277. URL: http://arc.aiaa.org/doi/10.2514/1.C031277%20https://arc.aiaa.org/doi/10.2514/1.C031277.

[130] *Spatial reference manual*. Tech. rep. Advanced Navigation, 2015.

[131] *Air data unit reference manual*. Tech. rep. Advanced Navigation, 2015.

[132] Samson B. Cooper and Dario DiMaio. "Static load estimation using artificial neural network: Application on a wing rib". In: *Advances in Engineering Software* (2018). ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2018.01.007. URL: http://www.sciencedirect.com/science/article/pii/S0965997817301916.

[133] Tim Mueller, Aaron Gilad Kusne, and Rampi Ramprasad. "Machine learning in materials science: Recent progress and emerging applications". In: *Reviews in Computational Chemistry* 29 (2016), pp. 186–273.

[134] Laura Mainini and Karen E. Willcox. "Data to decisions: Real-time structural assessment from sparse measurements affected by uncertainty". In: *Computers & Structures* 182 (2017), pp. 296–312. ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2016.12.007. URL: http://www.sciencedirect.com/science/article/pii/S0045794916308197.

[135] Roberto Lopez Gonzalez. "Neural Networks for Variational Problems in Engineering". PhD thesis. Technical University of Catalonia, Department of Computer Languages and Systems, 2008.

[136] Sheldon M. Ross. *Introduction to probability and statistics for engineers and scientists*. Academic Press, 2014.

[137] Robert McGill, John W. Tukey, and Wayne A. Larsen. "Variations of box plots". In: *The American Statistician* 32.1 (1978), pp. 12–16.

[138] J.S. Sánchez, R. Barandela, A.I. Marqués, R. Alejo, and J. Badenas. "Analysis of new techniques to obtain quality training sets". In: *Pattern Recognition Letters* 24.7 (2003), pp. 1015–1022. ISSN: 0167-8655. DOI: https://doi.org/10.1016/S0167-8655(02)00225-8. URL: http://www.sciencedirect.com/science/article/pii/S0167865502002258.

[139] Victoria Hodge and Jim Austin. "A Survey of Outlier Detection Methodologies". In: *Artificial Intelligence Review* 22.2 (Oct. 2004), pp. 85–126. ISSN: 1573-7462. DOI: 10.1023/B:AIRE.0000045502.10941.a9. URL: https://doi.org/10.1023/B:AIRE.0000045502.10941.a9.

[140] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection". In: *Data Mining and Knowledge Discovery* 28.1 (Jan. 2014), pp. 190–237. ISSN: 1573-756X. DOI: 10.1007/s10618-012-0300-z. URL: https://doi.org/10.1007/s10618-012-0300-z.

[141] Angelo Lerro, Alberto Brandl, Manuela Battipede, and Piero Gili. "A Data-Driven Approach to Identify Flight Test Data Suitable to Design Angle of Attack Synthetic Sensor for Flight Control Systems". In: *Aerospace* 7.5 (May 2020), p. 63. DOI: 10.3390/aerospace7050063. URL: https://doi.org/10.3390%5C%2Faerospace7050063.

[142] Marco Borri and Lorenzo Trainelli. "A Simple Framework for the Study of Airplane Trim and Stability". In: *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. 2003, pp. 5620–5644.

[143] Robert S. Swanson and Stewart M. Crandall. *Analysis of Available Data on the Effectiveness of Ailerons Without Exposed Overhang Balance*. Tech. rep. Advance Confidential Report L4E01. NACA, 1944.

[144] Jan Roskam and Chuan-Tau Edward Lan. *Airplane aerodynamics and performance*. DARcorporation, 1997.

[145] Mark Rauw. *FDC 1.2 - A Simulink Toolbox for Flight Dynamics and Control Analysis*. Tech. rep. 2001.

[146] John E. Williams and Steven R. Vukelich. *The USAF Stability and Control Digital Datcom*. Tech. rep. AFFDL-TR-79-3032. Air Force Flight Dynamics Laboratory, Apr. 1979.

[147] R. D. Finck. *USAF Stability and Control Datcom*. Tech. rep. AFWAL-TR-83-3048. USAF, Apr. 1978.

[148] Rotax. *Operators Manual for Rotax Engine Type 912 Series*. Tech. rep. Nov. 2016.

[149] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27* (2014), pp. 2672–2680. ISSN: 10495258. DOI: 10.1017/CBO9781139058452. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[150] Andrew Y. Ng and Michael I. Jordan. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in Neural Information Processing Systems 14 (NIPS 2001)*. Vancouver, B.C., CA, 2001, pp. 841–848.

[151] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[152] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs". In: *30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*. 2016, pp. 1–9. ISBN: 0924-6495. DOI: arXiv:1504.01391. eprint: 1606.03498.

[153] Ian J. Goodfellow. "NIPS 2016 Tutorial: Generative Adversarial Networks". In: abs/1701.00160 (2017). arXiv: 1701.00160. URL: http://arxiv.org/abs/1701.00160.

[154] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". In: ().

[155] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. "Improved Training of Wasserstein GANs". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5767–5777.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.