POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Tesi di Laurea Magistrale:

Neural network data analysis

for virtual air data sensors

Relatori:
Professor Piero Gili
MSc Eng Alberto Brandl

Candidato:
Angelo Scacciavillani

Luglio 2018

"Artificial intelligence (AI) is not some Asimovian fantasy,
nor an extravagance best left to starch-smocked scientists
clinking beakers together in an underground laboratory.
AI is an opportunity to create tools that save money,
save lives and improve life in ways that can't be measured."

— Colin Wood: Grounding AI, Government Technology, January 20, 2016

Chapter 1

Abstract

Over the years it is getting more present the use of machine learning techniques applied to many different disciplines. In aeronautic field they are used as a substitute or as an enrichment of flight systems in order to have a better production of flight data.

The main three aspects studied in neural networks, which are one of the many machine learning techniques and the one used in this thesis, are the network architecture, the training phase and the testing phase. There are other very important aspects such as the various methods for carrying out these operations. The knowledge of the latter is very important in order to obtain very precise networks in the purpose that is set. This thesis is in fact aimed at finding a method for the improvement of one of these aspects.

The focus of this project is on the analysis of input data, that is the training and, in a small part, which impact they do have on neural network quality, that is the testing phase, which constitutes the output.

In this specific case the neural network is used to simulate an angle of attack sensor. The main advantage of a NN is that it is possible to model, even if having an approximation, the complex behavior of a given system, without knowing equations and relations which govern the system. In this case the advantage is quite clear: knowing some flight data it is possible to find others. In this case the plane, which is a system in which the many caracteristics (ie. accelerations or rotation rates along the axes) are embedded inside the neural network.

This techinque is opposed to the complete modeling of flight dynamics equations for the given aircraft. This thechinque involves the knowledge of every characteristics of the plane. A novel approach as Neural Network modeling gives a fast tool to avoid complex modeling with a fast and reliable method to be implemented

Characteristics of the input data will be evaluated via statistics and a data reduction technique will be used in order to select only meaningful data points with the usage of k-means clustering algorithm. In previous studies an encouraging method was found to exploit neural networks in order to simulate the angle of attack sensor. However, certain of the complexity of neural networks, it is an improvable method. Therefore this thesis has the role of proposing an improvement of a previously used method.

This study has been done as a sequence of studies previously carried on Ing. Nando Groppo ULM aircrafts, and flight data utilized in this thesis are from a Groppo G70.

Contents

Chapter 2

Introduction

2.1 Air Data perspective

The air data computer (ADC) is an avionic device that is deputed to determine several flight characteristics. This system can determine:

- angle of attack
- sideslip angle
- indicated, calibrated or equivalent air speed
- Mach number
- altitude
- altitude variation

The ADC has several inputs coming from respective sensors:

- static pressure - static pressure port
- dynamic pressure - total pressure port
- temperature - thermometer
- angle of attack/side slip - differential ports, multiple holed probes, angle vanes

These air data sensors are connected together with an Air Data Computer (ADC) which elaborates signals ad enroute them to a Flight Control System (FCS).

All the knowledge for the calculation of the quantities listed above were already known in the second half of past century. Analog systems were used for those measurement on every kind of plane and they are already used in a great variety of airplanes. Just in the past ten years a big evolution has been witnessed in these techniques due to the digital revolution. Today the state of the art in air data evalution are digital instruments which compute sensors signals.

The transition from analog to digital was particularly favourable considering that, to complex and unmaneageable analog systems, electronic boards and codes were replaced to extract data from the measurements of different sensors or "sensors fusion". This is even more important considering the possibility of implementing redundant and precise systems at a very low cost and with a reasonable weight and size. In fact, many more markets are now opened to these complex systems such as the ultralight (ULM), as in this thesis, or that of autonomous aircrafts (UAV), where these systems are of primary importance for navigation.

As previously expressed, this thesis focuses on the not easy aim of measuring the angle of attack of the aircraft. This measurement is commonly performed by "vanes" for both subsonic and supersonic applications, or with differential pressure taps on the sides of a Pitot tube.

The AOA vane or alpha vane, a name due to the Greek letter generally used to indicate the angle of attack, consists of a vane of various shapes and dimensions, connected to a rotary position sensor. This kind of sensors has very high angular

resolutions and allows a very accurate measurement of the angle of attack. This method is also very simple since there is a direct measurement of the angle.

This kind of sensor is affected by two main errors. The first is due to the non-instantiation of the measurement, that is, there are some inertial effects that put the alignment of the wind vane with the current slightly delayed. The second is a position error. The aerodynamics of the aircraft can in fact influence the relative direction of the current in the position of the sensor. There are two main types of installation.

The first possibility is an installation on an air data boom. This air data boom includes static and total pressure and two ninety degrees offset vanes: one for measuring the angle of attack and one for measuring the angle of skidding. This type of installation is typical of several applications such as many fighter aircrafts and the air data boom is placed on the nose of the aircraft, where it lays in an aerodynamically undisturbed position. The main problem in this case is the flexibility of the boom that protrudes for several centimetres forward. The oscillations and resonance phenomena can in some way disturb the measurement of the angles.

The second type of installation is typical of the liner. The AOA vanes are in fact installed on the side of the nose, because an installation on the tip of the nose would be impossible given the radar behind it. The installation on the side of the nose causes the wind vane to be in an aerodynamically disturbed area from the surface of the aircraft itself. Generally, with an accurate study, it is possible to generate corrective coefficients that can bind local angle of attack with the real angle of attack.

The second technique is the use of multi-hole pressure taps. Generally they are presented as Pitot tubes with a truncated-conical head. On the flat front side there is a total pressure intake which, together with the static pressure taps on the tube shank, provide the indication of the speed. In the intermediate area, on the other hand, there are a certain number of holes (typically three to seven) connected to single pressure transducers. With a current offset with respect to the axis of the socket, a different pressure is obtained on these holes, typically a greater pressure in those oriented in the direction of the current and lower in those in the shade. It is possible to bind pressure measurements on these holes through non-linear relationships in order to obtain information on the angles of attack and the drift.

The non-linearity of the relationships and the fact that it isn't a direct measurement has been one of the major obstacles to the use of these probes and an approximation factor (in fact these relationships are found experimentally during probe calibration). A second demerit factor is the delay of the pressure signal that crosses the pneumatic line that leads from the holes to the pressure transducers. In the last applications this problem has been solved by inserting miniaturized transducers directly into the holes, in order to have instantaneous measurements.
The installation of these probes takes place on a boom generally on the nose of the aircraft. As in the previous case, the stiffness of the boom is a crucial factor for measurement accuracy. We must consider how this system is sensitive and with a considerable cost, both as regards the probe, which must have a very high production precision, and both with regard to the transducers.

The third method is the one dealt with in this thesis that is a completely indirect method. Knowing the dynamic characteristics of the aircraft it is possible to obtain the angle of attack by knowing other parameters that are easier to measure. The greatest difficulty in applying this method lies in the influence that many factors (such as the payload position, for example) have on the knowledge of the dynamic characteristics of the aircraft. Other problems lie in the difficulty of a quick calculation with the on-board computers and the approximation that one would have with this method. On the other hand, the advantages are obvious: greater aerodynamic cleanliness can be achieved compared to the two previous systems, and an absence of radar track. Furthermore redundant systems can be implemented simply by doubling the calculation system and / or the sensors. Lastly, this system would be based on sensors that are very accurate and, at the same time, cheaper like accelerometers, therefore with considerable savings [10].

2.2 Machine learning perspective

The aim set by computer science is, from the early days, to allow machines to easily solve complex problems for humans. Generally, this type of problem is mathematical or logical, based on mathematical rules. The machine learning disciplines offer a solution to this kind of problem. Machine learning is closer to the way of human reasoning, meaning machines can literally learn how a system works. The definition given in 1959 by Arthur Samuel of "machine learning" is: "Machine learning is a field of computer science that gives computer systems the ability to "learn" (ie, progressively improve performance on a specific task) with data, without being explicitly programmed. "Generally, machine learning is applied when the definition of specific problem modeling algorithms is difficult or impossible [21]. This is done by studying how a complex problem can be divided into smaller and simpler problems. The most important power of machine learning is the fact that the operator does not have to know in advance the equations that govern the complex system but the machine itself will determine them.

There are numerous approaches to machine learning. The one used in this thesis are the artificial neural networks or ANN. The name derives from the fact that this kind of technique is vaguely inspired by the functioning of the brain, that has neurons as "centres of calculation" of the brain, and of the synapses, or the connections between the latter. It is understandable how, the greater the number of neurons and the greater the complexity that can be modeled. This kind of techniques has in fact developed considerably in recent years thanks to the lower costs of very powerful computers.

As previously mentioned, there are usually two phases in the use of machine learning: a training phase and a testing phase. The training phase is the phase in which the data is fed to the network and on those the network will learn, that is, it will find linear or non-linear relationships that link the incoming data. The second phase, that one of testing, consists instead of inserting data other than training data and evaluating the output of the network. By comparing this output with reference data it is possible to find the error coming from the extimation made from neural net-

work, so it is possible to understand how the network is able to model that given problem.

The first step in creating today's neural networks was the creation of a network model based on the threshold logic by Warren McCulloch and Walter Pitts [12].
The second step was that of Dr. Hebb [6] in which he considered the hypothesis that the brain was plastic and it could be improved. This type of approach, unsupervised, is called Hebbian learning.

In 1958, Rosenblatt [19] created an algorithm for pattern recognition called perceptron. The first multi-layer network was created by Lapa and Ivakhnenko in 1965 [7].

Some research in 1969 [13] focused on two major problems in one of the neural networks. The first was the inability of a network to replicate the truth table of the xor and the second that there were no available computational resources sufficient for this purpose. Because of these problems, research in these areas experienced a period of stagnation.

In 1975 the research resumed actively following the invention by Werbro [27] of the backpropagation algorithm, which led, not only to the resolution of the xor problem, but also to a significant improvement in the training phase.

In 1992 there were the first applications of neural networks for object recognition. [24] [25] [26]

From the early 2000s to today, neural networks are being studied to fulfil increasingly complex tasks of object recognition, data analysis and more.

In 2010, Ciresan [3] demonstrated the feasibility of a hardware-based neural network. The reason for this choice lies in the speed of calculation, which for very large networks is enormously advantageous.

2.3 Previous approaches

The approach of using virtual sensors as a replacement for physical ones is not new. This has already been used in the past in order to manage redundancies. Normally these systems are based on "model based techinques" or modeling the system in advance. The difficulties encountered have always been those of making the systems sufficiently robust and immune to external disturbances. As early as 2000, Napolitano et al. [14], Oosterom and Babuska [16] studied systems useful for failure management using soft computing techniques.

In particular, the object of this thesis is the use of neural networks in place of model based techniques to model these virtual sensors. The use of neural networks for this purpose is not entirely new. In fact Rohloff et al. [28] , Samy and Green [22] have already described the modeling of virtual systems based on neural networks.

These techniques consist of the reconstruction of some data using only static pressure taps on the surface of the aircraft.

Neural networks have been applied on several different problems involving flight. One of them is to control the plane autonomously by using the neural network to convert navigation instructions into control surfaces actuators inputs [2] [23] [9].

Estimation of attitude coefficients from flight data has been made in previous application using machine learning. In a way, this application is much similar to the one treated in this thesis, as it starts from flight data acquisition. These practices started as early as 1993 demonstrating that, with modest computational resources, it is possible to find a solution to an otherwise complex problem [33] [11].

Regarding the measurement of the angles of attack and sideslip, mention must be made of the application on the Boeing X45A of a model-based system patented by Wise [29]. This makes use of inertial data and a Kalman filter, that is used to reduce the error in angles estimation.

Regarding other hypersonic airplanes, many studies have been made by chinese professor B Xu which recently worked on guidance, and sideslip control of a generic hypersonic plane [31] [32] [30].

This thesis is linked to the work carried out as a doctoral thesis by Angelo Lerro [10], who was involved in studying the application of neural networks for the estimation of angles of attack and sideslip for the Alenia Sky-Y drone. This work was the forefather of various other studies on the topic led by Professor Piero Gili. The following works mainly dealt with improving the accuracy of the network.

The innovative factor of this thesis is the quest for a system to select flight data to be used as network training. In this way it is possible to have less input data, and gaining performance on the training side, as will be seen below. Furthermore, it is also possible to study how samples quality can negatively influence network performance. In particular, data reduction and data analysis work has been carried out using k-means clustering, which is another machine learning technique and a common statistical application.

Chapter 3

Mathematics

In this chapter theoretical aspects of this thesis are introduced in order to give the reader a solid background suitable for understanding further evaluations.

3.1 Feed Forward Neural Networks

An artificial neural network is composed of a set of artificial neurons (simplified models of biological neurons) and artificial synapses (simplified models of biological synapses). Each artificial neuron is composed of the following parts [34]

input: the input signal to the neuron (generally a real number)

activation function: a function that activates the neuron only if the input signal is sufficiently large

weight: how much the output signal is increased if the neuron is active

bias: a quantity, generally negative, which describes how the content of the neuron must be activated so that it is activated

output: the resulting output signal

As for the interconnections between neurons, artificial neural networks are organized in layers, ie the outputs of a series of non-communicating neurons are given as input to a series of other neurons not connected to each other. All the neurons except from input layer are called hidden layers. As for connections or synapses, these are the connections between the neurons of a layer and, generally, all the neurons of the previous or next layer. This complex system can be expressed mathematically simply as a multiplication between matrices. Considering a generic neuron of a layer of a network one can write:

$$a_0^1 = \sigma(w_{0,0}a_0^{(0)} + w_{0,1}a_1^{(0)} + ... + w_{0,n}a_n^{(0)} + b_0)$$

Where:

a: they represent the various signals coming from the neurons of the previous layer

w: they are the weights that multiply the output values of the neurons of the previous layer

b: it is the bias that is added to the sum of the inputs for the weight and sets the threshold of activation of the neuron

σ: it is the activation function that multiplies to every single element of the summation.

Matrix-wise, for the complex of all the neurons of a layer one can write [5] :

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & ... & w_{0,n} \\ w_{1,0} & w_{1,1} & ... & w_{1,n} \\ ... & ... & ... & ... \\ w_{n,0} & w_{n,1} & ... & w_{n,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ ... \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ ... \\ b_n \end{bmatrix}\right)$$

or, in compact form [5] :

$$a^{(1)} = \sigma(Wa^{(0)} + b)$$

Here it is possible to analyse some very important aspects of the training phase. The first aspect is to understand how the network generates the error committed with respect to the desired output. What is used is a cost function. This specific function depends on the difference between the output value from the output nodes to the network and the expected value, more specifically it is the sum of the squares of all the differences in the single output neurons.
The cost function therefore depends indirectly on the web weights and biases. By varying the latter it is possible to reduce the cost function.
This is represented by the following graph where the cost function, dependent on the weight and bias vector, is decreased using a given method. The represented one is a "gradient based" method that is dependent on the gradient of the function cost function in the multidimensional space of the weight and bias.

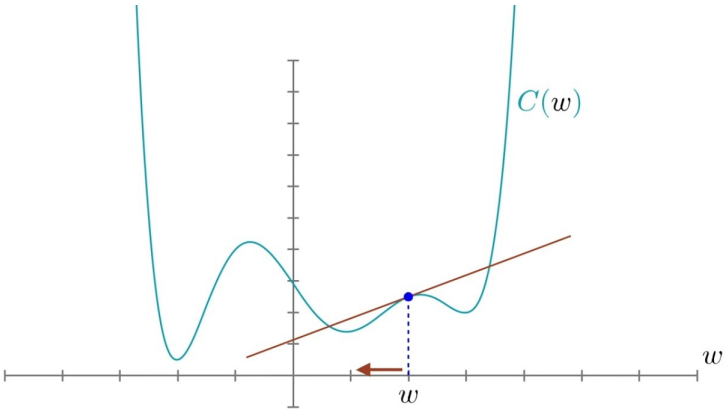


Figure 3.1: Typical gradient based minimum search, here represented in a two dimensional space.

Please notice how the weight vector tends to converge to a local minimum of the cost function and not to the absolute minimum. This depends on the way in which the vector "w" is initialized. Generally this operation happens by inserting random numbers in the vector.
The cost function minimization method is called the "learning rule" [15].
There are different learning rules or learning algorithms and they are summarized in three types: supervised, reinforcement, unsupervised and has been recently introduced the semi-supervised, a mixture of the first and the latter. In supervised learning, you have information about the quality of the network, which is a performance index on which the learning process is based. The network then compares the real output to the desired output and modifies the weights by following the backward error in the network, that is from the output layer to the input layer and modifies the weight and bias accordingly. The classic learning algorithm is Backpropagation (BP) [10].
In some cases, however, there is no detailed information on the quality of the output and therefore supervised learning can not be used. Reinforcement learning is used instead. Reinforcement learning gets feedback from the surrounding environment. More formally, the environment is modeled through an instantaneous cost distribution, an observation distribution and a transition. Together they form a Markov

chain (MC). The aim is to find a policy that minimizes the cost function.
In unsupervised learning, however, there is no feedback. The network must learn itself to correlate incoming data. The cost function therefore depends on the work that the network has to perform and every a priori assumption performed, as implicit properties of the model or of the observed variables.
Depending on the case, one of the three methods mentioned is used. In the present case, supervised learning is used, as there are specific information on the target angle of attack. Regarding the learning algorithms of supervise learning, there are three main categories that are based on as many mathematical methods:

conjugate gradient: Levenberg-Marquardt, Fletcher-Reeve, Polak-Ribiére, Powell-Beale, scaled conjugate gradient

quasi-Newton: Broyden-Fletcher-Goldfarb-Shanno, one step secant

steepest descent: variable learning rate and momentum, backpropagation

3.2 Probability and Statistics

This section illustrates a base of probability theory and statistics. Through this probability theory it is then possible to derive a series of statistical concepts useful for data analysis. The usefulness of a statistical approach in the case of the analysis of flight data is to be found in the fact that, for a large number of data, it is necessary to capture collective, and not just specific, aspects [5].

3.2.1 Probability distributions

Given a sample of data, it is useful to study the probability, or reorganize the data based on the frequency with which a data is found. This method will give rise to a function called probability distribution (of data "x"). There are several probability distributions, denoted by different control functions such as Bernoulli, Categorical, Gaussian and Laplace distributions [5].
However, these functions restitute a trend of the probability function as a curve with a single-mode. Using these functions means that there is only one peak in the probability distribution. This causes a large loss of information in the event that there is a multimodal trend in the distribution. This type of performance greatly precludes the performance of the network, rather it constitutes a discriminant in the process of data analysis, and therefore must be considered. In fact, it should be noted that, when the probability distribution chosen varies, the different aspects of the sample under analysis are highlighted.

3.2.2 Kernel Density Estimation

The analysis method chosen to characterize the probability distribution is the kernel density estimation. It was created by statisticians Manuel Parzen and Murray Rosenblatt [17] [18] [20]. This is a non-parametric method used for pattern recognition and classification through a density estimate. The algorithm allows the calculation of the probability of belonging to a class for each element, considering the density of the class in a neighborhood of the value to be classified. This neighborhood is of fixed size and is calculated based on the number of observations.

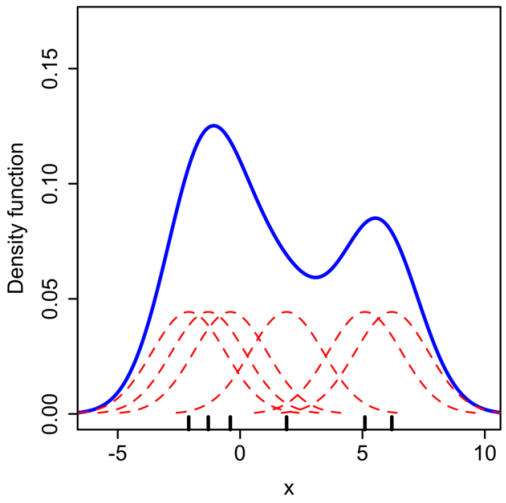


Figure 3.2: Kernel Search distribution as the sum of normal distribution for the variable "x".

A method of classification by proximity (parzen windows or k-nearest neighbors), proposes to calculate the conditional probability at a point x with the following density estimation:

$$P(x|C) = K/NV$$

where:

N is the number of estimations in the data set;

V is the volume around a certain point x

K are the elements around V, belonging to the class C.

The density kernel estimation algorithm is conceived in order to minimze the size of region V near x, depending on observations quantity N. This method is based on the idea of effectively reduce the region V, so that it is possible to have an approximation of the real estimate of the point P. At the same time it does not consider a region so minimal as to have K=0.
It is worth to evaluate a function like $K(h, P)$. It depends on the parameter h that is the scale, P_0 and its distance P. This function is must have integral equal to one on P. The method previousy described consists in assigning $P(x|C)$ in point x using the formula below:

$$P(x|C) = \frac{1}{N}P(x|C) = \frac{1}{N}\Sigma_i K(h, d(x, xi)$$

That is, the density at point x is obtained by considering the contribution as the sum of the contributions provided by the observations in the sample spread according to the law K (h, P), normalized to N.
The evaluation of h is a problem quite complex as it generally depends on the problem under examination. A thumbrule is to use the following relation: $h = O(n^{-1/5})$.
A problem with this method is due to the fixed choice of the K function. In the end, if a very small window is used, the risk of overfitting is introduced. If you use a window that is too big, you have more errors in the denser areas. For this reason a dynamic window (Algorithm k-nn) sometimes gives better results. The complete algorithm used is included in Matlab function "ksdensity" [8].

3.2.3 Geometrical aspects

Some important features of these probability distributions can be studied by some parameters:

Mean The arithmetic mean is used to resume an ensable of data with a single number. It is calculated by summing all the data set and dividing the result by the data set dimension. Mean formula is shown below:

$$M_a = \frac{1}{n}\sum_{i=1}^{n} x_i$$

The arithmetic mean can be calculated in this way if the frequency distribution is available:

$$M_a = \frac{1}{n}\sum_{j=1}^{k} x_j n_j$$

where x_j represents the j-th mode of x, K is the number of modes assumed by the set of data x, and n_j the frequency which corresponds to data x. Being then $\frac{n_j}{n} = f_j$, it follows that:

$$M_a = \sum_{j=1}^{k} x_j f_j$$

where f_j is the frequency of the j-th modality of x.
The weighted average is calculated by summing the data set values, each multiplied by weight and divided by the sum of weights. Considering this specific definition, it is possible to consider the arithmetic mean as a special weighted mean where al the weights are equal. It means that all the data have the same importance. The general formula for the weighted mean is therefore:

$$M_{a,w} = \frac{\sum_{i=1}^{n} x_i f_i}{\sum_{i=1}^{n} f_i}$$

Arithmetic mean does not provide robust statistical data because it is significantly affected by outliers, even though it is used very often. For this reason, other geometrical indice such as the median, are also often considered. These indices are less prone to accept anomalous values. A way to reduce the effect of wrong values in the calculation of the arithmetic mean is considered in the trimmed mean. This type of average is a specific way of considering the average in which only a certain quantity of the most feasible values are considered.

Mode In statistics, the mode of a frequency distribution is the value with the highest frequency, that means, it occurs more frequently. A distribution is unimodal if it has a single modal value, it is bimodal if there are two of them. The presence of many modes could be a symptom of non-homogeneus data sets. In the particular case of the normal distribution, also called Gaussian, mode coincides with the mean and the median.
The mode formula is:

$$h(i, i+1) = \frac{n(x_i, x_{i+1})}{x_{i+1} - x_i}$$

where $n(x_i, x_{i+1})$ the number of elements which falls in the class x_i, x_{i+1} and $h(i, i+1)$ is the height. The mode is a valuable characteristics because it is the only one of the central tendency indices able to synthesize qualitative characters..

Skewness The skewness index of a distribution is a value that measures its lack of symmetry. There are several different asymmetry indices. There are distributions with skewness equal to 0, therefore skewness is a sufficient but not necessary condition to symmetry. All symmetric distributions have null skewness. The commonly used asymmetry indices are based on some properties of symmetric distributions or, in particular, of the normal distribution.
The skewness is defined as:

$$\gamma_1 = \frac{m_3}{m_2^{3/2}}$$

through central moments $m_k = E[\bar{X}^k]$, that is, the expected values of the powers of the centered variable $\bar{X} = X - E[X]$.
It is now possible to consider central moments values. The first central moment is always null, the second central moment, that is the variance is null only for the distributions concentrated on a single value. The third central moment m_3 is the lowest one that can measure the asymmetry of a distribution. In certain cases the index β_1 is used instead of γ_1:

$$\beta_1 = \gamma_1^2 = \frac{m_3^2}{m_2^3}$$

In statistics the skewness calculated on a data set is: $\{x_1, ..., x_n\}$ of mean $\bar{x}$ It follow the formula:

$$g_1 = \frac{\sum_{i=1}^{n} \frac{1}{n}(x_i - \bar{x})^3}{\left(\sum_{i=1}^{n} \frac{1}{n}(x_i - \bar{x})\right)^{3/2}}$$

The next central moment m_4 is instead used to calculate the kurtosis, which measures the deviation of the distribution from the normal distribution.

Through these indices it is therefore possible to evaluate different qualities of the statistical samples considered.

3.3 K means clustering

In data science, clustering is the task of grouping the data by choosing them respecting some rules of similarity between them. Data clustering is the general activity and not a specific algorithm. Clustering is a machine learning activity as it is the ability of the machine to understand what relationships exist between the data and how to group them. There are many different methods for data clustering. The method used in this thesis to find significant flight data compared to others is called k-means clustering. K-means clustering was invented by Stuart Lloyd in 1957 for Bell, but, for reason of secrecy, it was published in 1982. The term k-means was first coined by McQueen in 1967. It is worth to mention how the same method was used independently by Forgy in 1965 therefore, sometimes k-means clustering is cited as the combination of two data scientists names, that is Lloyd-Forgy method.

The task this method operates is that of subdividing the n samples into k subgroups. For each subgroup or cluster it creates a centroid, and the elements of the subgroups are subdivided into those having a mean lower than its centroid, hence the name of the method. This can be described as the ability of the algorithm to minimize the total variance in the clusters by correctly grouping of the data. This method is iterative, and computes the initial clusters with heuristic methods, that is in a casual fashion and calculates the centroids and the variance, and then starts the cycle again, until it converges.

his sort of problems is generally computationally difficult, even though there are efficient algorithms that are generally convergent. These algorithms are based on the expectation-maximization algorithm. K-means clustering has the very important characteristic that it finds clustering that have similar shapes and dimensions. Due to the speed of convergence,this algorithm is one of the most used. It has been noted that the number of iterations it takes to converge is almost every time lower than the dimension of the data set. Arthur and Vassilvitskii studied k-means clustering in hard problems. They showed how k-means clustering takes a supernominal time to converge. Instead, Vattani has shown that the algorithm takes an exponential time to converge in the worst case. One of the main disadvantages of the algorithm is that it doesn't reach the global minimum of the problem.

The quality of the solution depends on the initial data and the number of choosen clusters. A second problem is that, in the case of naturally partitioned data, if a different number of clusters than the natural one is chosen the algorithm will generate an wrong partition. It will be necessary to manually adjust the number of clusters in order to obtain the best result. Finally, the algorithm works effectively only when n-dimensional planar clusters are detectable, where n is the size of the input data. Data N objects with attributes, modeled as vectors in an i-dimensional vector space, we define $X = X_1, X_2, ..., X_n$ as the ensemble of the other objects.

The number of searched clusters must be less than the number of data: $1 < K < N$; otherwise they would have empty clusters. It is indicated with $M = M_1, M_2, ..., M_n$ the set of K centroids. Each subdivision is identified with a matrix $U \in N^{KxN}$, with each element j belonging to the cluster i ie $u_{ij} = \{0, 1\}$. The objective function is now defined as:

$$V(U, M) = \Sigma_{i=1}^{k} \Sigma_{x_j \in P_j} ||x_j - M_i||^2$$

At this point we calculate the minimum of the objective function as follows:

1. U_v and C_v are casually generated
2. U_n is calculated that minimizes $V(U, M_v)$
3. M_n is calculated that minimizes $V(U_v, M)$
4. If the algorithm is converged the cycle is interrupted, otherwise $U_v = U_n$, $M_v = M_n$ and it goes back to second step

Typical convergence criteria are:

- No change in matrix U

- the difference between the values of the objective function in two successive iterations does not exceed a predetermined threshold

As described above, there is a need for some method to initialize the calculation. The quality of the partition will depend on this initialization.
There are two commonly used methods. The first is called the Forgy method and chooses random data from the input data and uses it as the starting center of the algorithm. The second method is that of the Random Partition in which each data is assigned randomly to one of the k clusters and this condition is used as a starting point. Generally, the first initialization is used for the standard k-means. A study by Fayyad and Bradley [1] compares a good number of initialization methods and evaluates the results.

For this work a modified method of the original k-means called k-means ++ was used. This improved form was invented in 2007 by Arthur and Vassilvitskii and aims to improve the seeding phase of the algorithm in order to ensure a faster convergence in the case of NP-hard problems. The basis of this new algorithm is the idea spreading the centroids it is bossible to get the best results. The algorithm in fact computes the initial centroids randomly, and then the following are chosen from the data with probabilities proportional to the square of the distance with the nearest centroid. The steps are listed below:

1. The algorithm choses a centroid randomly from the data
2. For each point it calculates the distance between it and the nearest centroid
3. It chooses a new centroid from the data, using a probability distribution where a proportional probability is associated with the point
4. The intermediate steps are repeated until the centroids are good quality ones
5. Then it continues as the k-means previously described

Regarding the timing of execution, this algorithm, although having a longer initial phase for the choice of centroids to start with, converges more quickly than the classical k-means, and therefore there is an overall containment of the execution time.

Chapter 4

Data Analysis

4.1 Data Acquisition

4.1.1 Rig

As explained before data sampling has been done on a Groppo G70. The method used was to apply an air data boom near the junction between wing and wing bracing. This position is favourable as it is a structurally rigid point. It is worth to mention that measurements quality is influenced by the loss of rigidity in the acquisition system, and it is out of propeller wake, that is another source of perturbation.

Figure 4.1: The picture shows the air data boom mounted outboard.

The air data boom arrangement is as follows:

- a total pressure port is located on boom top that allows total pressure measurement

- on port stem flush holes are present, for static pressure measurement
- going to the back there is a vane, on the side of the boom, for angle of attack measurement
- on boom stem there is another vane, 90 degrees out of phase from the previous one, for side slip angle measurement

On board an inertial platform is located right down the seats, close to the center of gravity for acceleration measurement, a gps for altitude and geographic coordinates recording and a barometer for altitude evaluation.

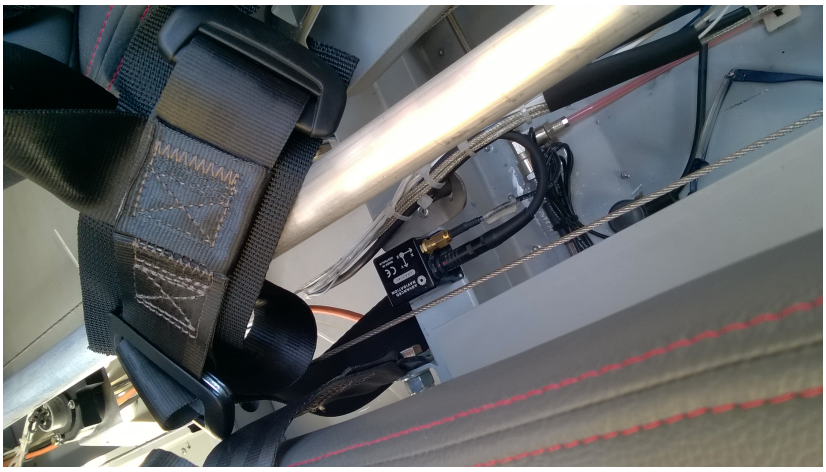


Figure 4.2: The picture shows the Inertial Measurement Unit in place between the seats.

The complete list of signal sampled is the following:

from GPS: V_{north}, V_{east}, V_{down}, $altitude_{GPS}$, latitude, longitude

from clock: time

gyroscopes: p, q, r

from barometer: $altitude_{barometer}$

from Pitot probe: qc

from α and β vanes: α, β

from accelerometers: n_x, n_y, n_z

from magnetometer: Roll, Pitch, Yaw

4.1.2 Overview of samples

As a training set for the neural network 7 samplings have been chosen test maneuvers. Samplings have been runt during 10th, 11th, and 17th of june 2017.

Data	Hour	Sampling duration
10 June 2017	8:50	2220s
10 June 2017	9:54	1970s
10 June 2017	14:35	420s
10 June 2017	15:37	1900s
10 June 2017	16:41	480s
11 June 2017	16:35	900s
17 June 2017	10:11	2010s

4.2 Maneuvers

Numerous aircraft testing maneuvers have been carried out to ensure that all aspects of the flight are covered, whether they are maneuvers or conditions, and for the completion of the flight envelope. The flight tests were carried out by several pilots who reported, for each flight, take-off and landing times, take-off weights, consumption and finally the maneuvers carried out with detailed descriptions.
These maneuvers are typical of the testing of an aircraft, according to well-defined procedures, which however are not the subject of this study, and therefore the motivations inherent to the use of such maneuvers will not be explored.
The maneuvers performed were as follows:

- Sawtooth glide at various speeds
- Sawtooth climb at various speeds
- Dutch roll triggering at various speeds
- Phugoid triggering with both fixed and free stick
- Longitudinal static stability testing at various altitudes
- Equilibrator doublet
- Stall idle
- Speed stability at different speeds
- Wind speed triangle
- Steady handling sideslip

The data coming from the maneuvers indicated and the remaining parts of the flight are contained in the acquired data which will then be used for this study.

4.3 Analysis Environment

It is appropriate now to describe the environment in which data analysis has been done.

4.3.1 Computational Performace

The analysis has been performed on MATLAB®program from MathWorks, for the pre- and post-processing phase on a pc and for the processing phase, which requires consistent computational resources, it has been runt on Hactar HPC, the supercomputer of Politecnico di Torino. First topic to face is just the one afore mentioned: the computational resources needed for the analysis. Pre-processing phase mainly consists of signals re-sampling all to the same frequency, as the sensors outputs have typically different sampling frequencies.

After that re-sampled and re-synchronized data are saved into a suitable data structure, ready for further analysis. It must be written as this part has been developed using scripts written by Alberto Brandl, who coordinated this thesis, for his PhD work. This first phase doesn't need neither big temporary storage space (RAM), nor long time for computing: the PC can handle these operations on large data structures (in the order of 10^5 matrix rows) in less than a minute.

The second phase, that is processing, is the most demanding from a computational point of view. It needs indeed a large amount of calculations, whom time depends on CPU performances. So, even if K means Clustering is meant to be a fast and practical method for data partitioning, it must be considered that for this activity a specific infrastructure is needed.
Third phase, that is post-processing, is partially an onerous phase, in which detailed plots are built as a main computational activity.

4.3.2 Analysis domain

A second aspect of analysis are the troubles on data interpretation. Man lives in a three-dimensional space in time domain, and so we tend to refer to an analysis that is in some degree amenable to these conditions. This specific analysis is done in a multidimensional non-time dependent space instead. Neural Network training data haven't got a temporal logic, as the order in which they are used doesn't affects training.

Another abstraction factor is the large multidimensionality. This factor is of special importance in data visualization. A big problem that is generally found in this discipline is to manage data belonging to a n-dimensional space with n bigger than 3, and to manage a very large amount of data. If we add that in computer graphics third dimension is simulated and not real, it is possible to understand how shrinked is a computer screen for this sort of job. As a matter of facts a great attention has been paid to data visualization, in order to obtain easy to read and meaningful plots.

4.4 Statistical approach

Due to the large number of data managed an approach that evaluates data individually is completely not feasible. So the need arises of using a statistical approach. First it must be considered flight data as a set of samples all referred to a coordinate of interest that is flight angle of attack. According to this quantity it is possible to calculate the probability distribution (PDF) for all cluster population or evaluating

the most likely value or mode inside the cluster regarding $\Delta\alpha$.
This analysis gives important qualitative informations on cluster.
Following parameters have been evaluated.

Multimodality The multimodality of the PDF was computed using the derivative centred and evaluating the number of maxima present. In a early analysis the PDF on clusters was evaluated and it has been noticed how a large number of clusters had a multimodal trend. It is a clear sign that the cluster itself tends to describe more than a $\delta\alpha$ state. This factor can be considered as an marker of how "good" the cluster is. From a mathematical point of view, a number of peaks greater than one means the cluster is constituted of two different points of accumulation for the minimization of the variance in the cluster, index of an incorrect partitioning of the data.

Variance The variance reflects how the population of a cluster is distributed in a small or large neighbourhood of the centroid. An ideal cluster would be the one with the whole population close to the centroid, that is with a low variance. This means that the centroid "resumes" the state of the population it represents properly.

Skewness Skewness is an indication of how much the probability function is symmetric, that is that data points with values bigger or lower than the centroid must have the same distribution around it.

The order in which these features have been illustrated is not accidental: they have in fact been placed in order of importance for the analysis. The most important feature is indeed the multimodality of the PDF. A high-quality cluster must only describe one $\delta\alpha$ state at a time. Secondly, there's the distribution of the population, even if in this case a quality cluster does not have a scattered population. Finally there is a second indication on how the clusters are centralized towards the centroid. Clustering on flight data proved to be of poor quality immediately, so the second and third aspects are secondary to the first.

In theory we should consider a "0" aspect, which disregards all the others, that is the number of data in the cluster. If a cluster has a small population, a statistical approach would not be correct. The population of the clusters is therefore another aspect evaluated in the analysis.

4.5 Clustering Strategy

As explained in the previous chapter, the algorithm of K-means clustering works by giving the desired number of clusters as inputs. This factor is particularly disadvantageous in the case of naturally partitioned data. For example, imagine throwing a deck of cards on a table. A chart is created for illustrating how cards faces are oriented on the table: the possible states are clearly 2: with the back up or with the back down.

However, if K-means clustering is used to partition the deck of cards with respect to the orientation of the faces in three clusters, this would obviously generate an error. More specifically, in the case in question, it is necessary to consider that the number of accumulation points is not known a priori by the algorithm, that is the number of minima for the variance of distances from centroids. This is a considerable difficulty in deciding the number of clusters to choose from. However,it is possible to identify the first discriminating factor for a raw skimming of the number of suitable clusters.

On the one hand it is not suitable to use a low number of clusters since this would not be descriptive of the various possible states identified during the flight. At the extreme opposite of the spectrum having a large number of clusters is not feasible because it would have too low data populations and therefore it could not carry out a statistical analysis on the data. It is also evident that a cluster contending a small population of data represents a very specific aspect of the flight, if not even of the wrong data of the acquisition.

The average number of peaks in the clusters was chosen as the cost function to be minimized for analysis. Conceptually the variance of the PDF on clusters could also be considered, since this is the variable resolved by the K means, but a greater amplitude of the data immediately appears as a secondary factor with respect to the description of several states simultaneously. Since the complete calculation of all possible partitions would be very time-consuming, two separate analysis of the cost function were carried out. The first is from a macroscopic point of view of its trend and a second from a microscopic point of view. Clustering from 100 to 50100 clusters every 500 clusters (ie 100, 600, ..., 49600, 50100) has noticed a very evident behavior:

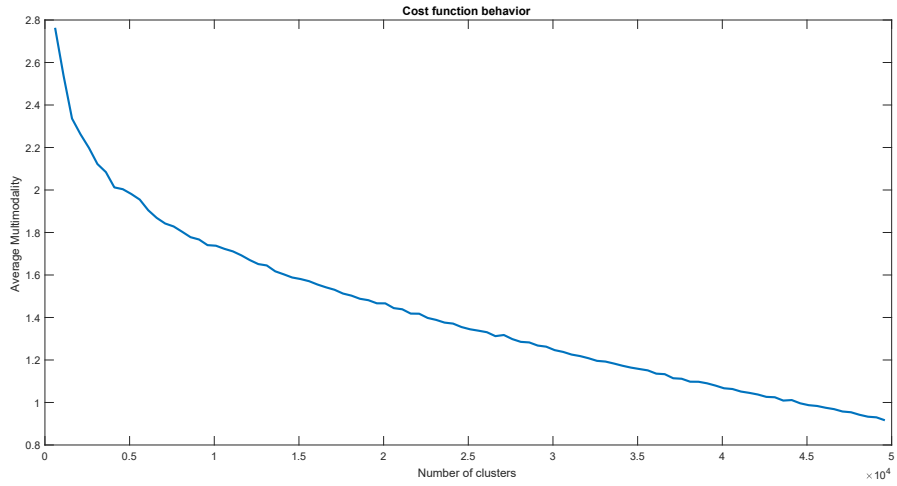


Figure 4.3: Average peak numbers for clustering between 100 and 50100 Clusters every 500 clusters.

Where average multimodality is defined as:

$$average\ multimodality = \frac{\sum number\ of\ peaks\ in\ each\ centroid}{number\ of\ centroids}$$

In figure 4.3 the cost function has a left side having a very high slope. Note that

this, however, is not a true vertical asymptote: for a partition with a single cluster, the average number of peaks, which corresponds to the number of peaks in the PDF of the single cluster, depends on the distribution of the points. Having a series of data distributed appropriately, it would be possible to obtain a single-mode PDF, obtaining a value of the exact function of zero.

In theory in the zero point, that is, not partitioning the data it would be possible to find a number of infinite peaks in the PDF, as there are infinite distances between the points and therefore infinite peaks in the distributions. In reality this case is not possible therefore the calculation inevitably starts from 1. The right branch decreases instead with a sublinear slope. The gradient here is about -0.232.

Because of the excessively high computational requests, the complete mapping of the cost function was not possible. With a macroscopic observation of this kind one could think that, with a simple algorithm of descent we can identify the minimum of the function, being monotonically decreasing, at least for the considered interval.

One observation that has been carried out is one with a reduced sample of data. In this way it was possible to carry out a large number of observations in a rather easy way. The clustering of the first 4000 data points has therefore been evaluated. Mean multimodality and variance were observed. The test was repeated for other samples of data points and the results were similar, so they were not reported.

The first graph that is shown is the one related to the variance.

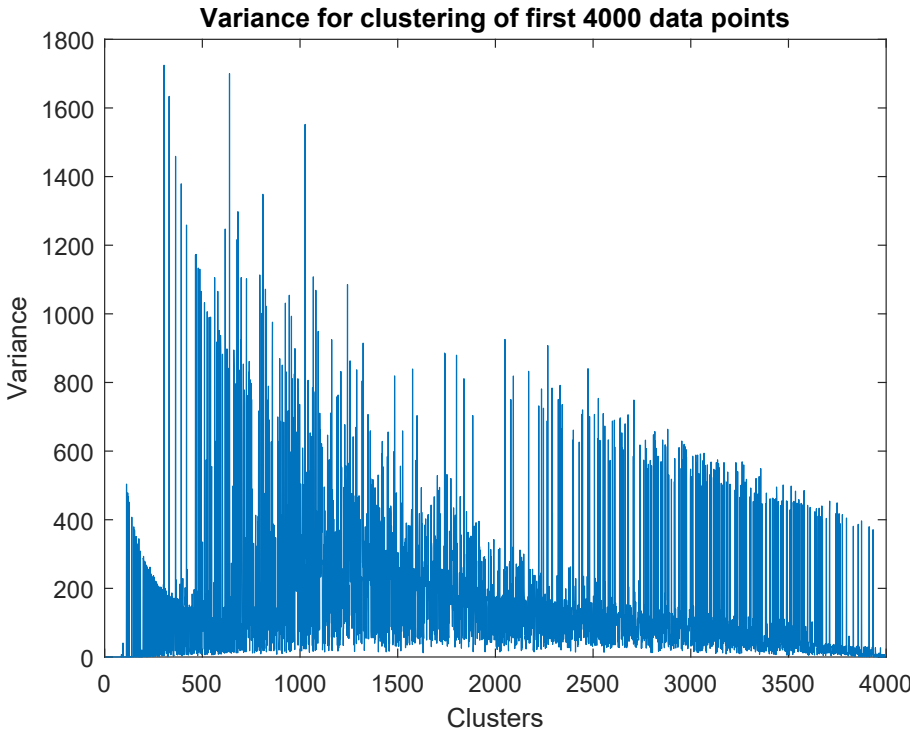


Figure 4.4: Average variance for clustering between 1 and 4000 for reduced dataset.

In figure 4.4 it is possible to immediately observe a noisy trend of the function. There are numerous spurious peaks due to clusters that have extremely high vari-

ances. However, the trends described by the peaks are quite clearly recognizable. These trends are like 1 / x. This is due to the mediation that is made of the variances of the various clusters, which are on the x of the graph. Therefore these changes are due to this factor. The fact that they are different is due to the population with respect to the data vertical position, they are divided into zones with similar variances. Less evident is the trend of the minima.

In fact there is a trend that approaches zero for some cases with a low number of clusters, which is maximized in the middle of the interval and then returns to minimum values towards a unit subdivision, ie with a cluster number close to or equal to the number of data points. This trend is very different from what was expected, as it was expected a minimum value for a number of clusters different from the minimum or maximum.

The average multimodality of the clusters is now analyzed.

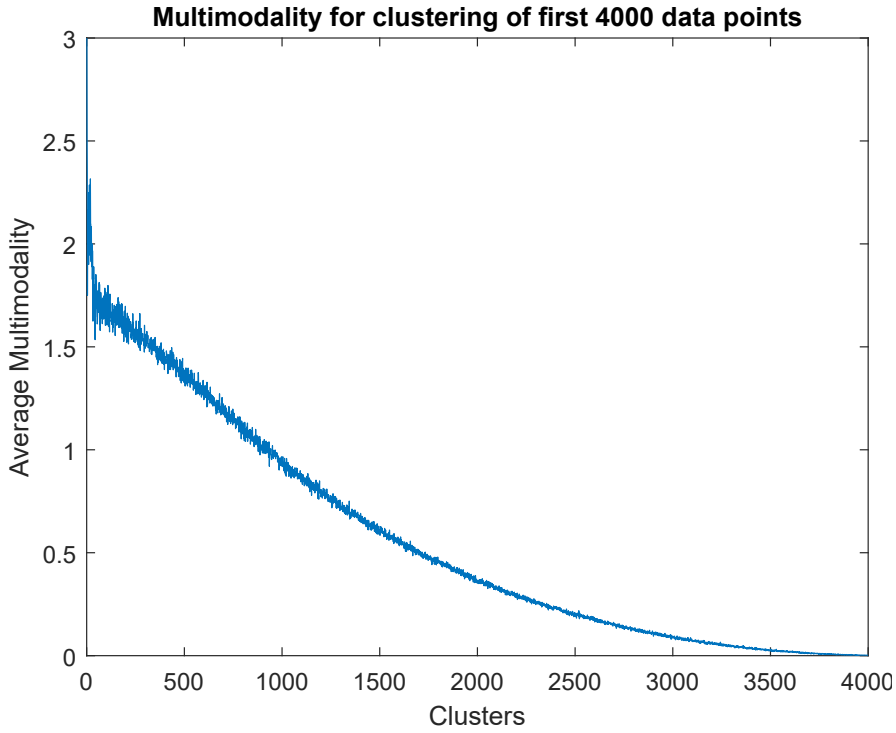


Figure 4.5: Average multimodality for clustering between 1 and 4000 for reduced dataset.

Also figure 4.5 presents a very noisy trend with a noise that decreases with increasing the partitioning. In this case one would have expected to find a minimum value along the interval, instead it has a monotone trend.
Since there are no particular minima in theory all types of research of the global minimum would be useless. However, it might be that with different flight data there are cases in which a minimization of a characteristic is necessary or useful. Therefore we continue in research independently of this discovery. Furthermore, it is not certain that the overall trend of data clustering is exactly equal to a part of it. By the way this observation is still indicative of some specific behavior and so it is worth to do it.

By carrying out a more detailed analysis in a small range of clusters it was possible

to observe what is the real obstacle in the analysis. Clustering has been performed for every cluster number from 490 to 510 clusters in order to understand better the local morphology of the cost function chosen. Calculating the average number of peaks for each partition from 490 to 510 clusters a trend has been noticed:

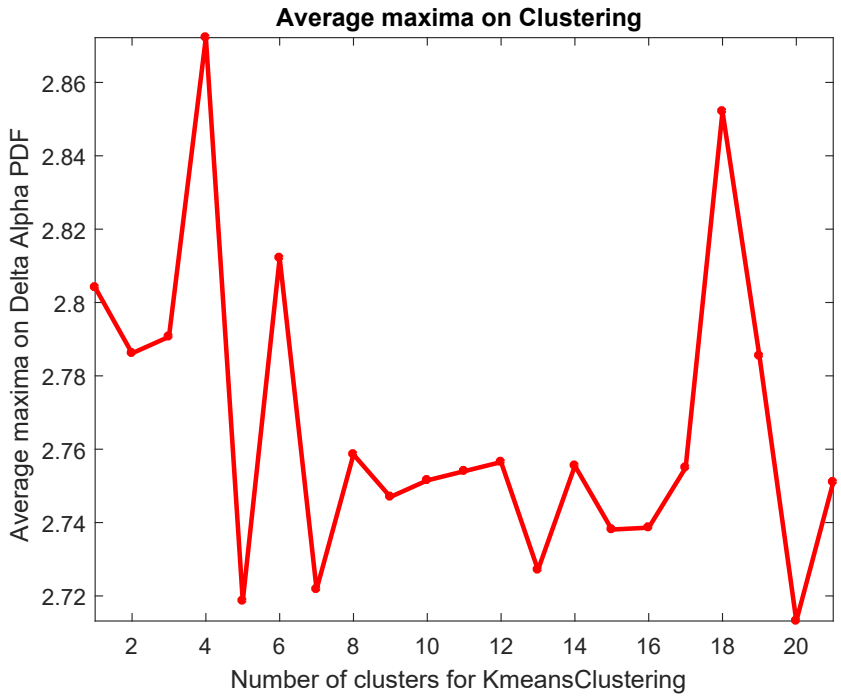


Figure 4.6: Average peak numbers for clustering between 490 and 510 Clusters.

Note in figure 4.6 how this trend is strongly discontinuous and how a small number of partitions have been used in order to reduce calculation time. Here we highlight a notable problem: it is analytically complex to find the minimum of the cost function. Notice how there is a large number of minima even for a few partition values (in this case 21). Therefore, various possibilities for approaching the problem arise.

4.5.1 Brutal approach

The first possible approach is to evaluate the cost function for each of the possible cases, taken in a range of reasonable values. This approach is what guarantees the possibility of finding the absolute minimum of the problem in 100% of cases. However, this approach is enormously expensive from a temporal and computational point of view. In fact, consider that the algorithm of K means clustering takes time and RAM superlinearly, which means that for a partition with many clusters it is possible to experience a very high calculation time. This approach proves to be infeasible because one of the secondary purposes of this research is to find a quick and effective method to find the network training points.

4.5.2 Standard approach

A second type of approach could be to evaluate a certain quantity (to say 20) of data partitions in a given reasonable range. This method has the advantage of being

able to give an overview of how the partitioning affects the cost function, but we can not know if, for a given partition, there is not a much lower value of the cost function in the adjacent partitions, given the jagged trend of the cost function when the partition changes. This approach is also not recommended.

4.5.3 Genetic Algorithms approach

At this point it could be imposed to use some algorithm to find the absolute minimum of the cost function. One of the possible methods to find the minimum of a non-smooth function is genetic algorithms. This type of approach has been tested by the author but it has proved to be very slow since it requires a considerable number of evaluations of the function, albeit a smaller number than the brutal approach.

4.5.4 Optimized approach

Now a compromise could be evaluated. It is possible to evaluate a standard approach, but somehow find the local minimum close to the various initial assumptions. This guarantees to have a computationally light method but at the same time more accurate than the standard approach. This constitutes the method chosen.

4.5.5 Chosen Method

In this subsection the chosen method is described in more detail. In order to find the local minimum, a simple gradient descent algorithm has been implemented in the MATLAB®code. This was written in order to guarantee a restricted amount of function evaluations and to work on functions from N to R. The code is as follows:

```
x=n_cluster;                          %first guess partition
    i=0;                              %counter initialization
    max_it=10;                        %number of maximum iterations
    sx=FastClustering(x-1);           %evaluation of left point
    cx=FastClustering(x);             %evaluation of initial guess
    dx=FastClustering(x+1);           %evaluation of right point
    d_dx=dx-cx;                       %right derivative calculation
    d_sx=cx-sx;                       %left derivative calculation
    d=dx-sx;                          %centered derivative calculation

    while i<max_it

        i=i+1;

        if d_dx*d_sx<0 && d_dx>0      %evaluates if current point is a minimum
           minimum=x;
           break
        else                          %evaluates direction of descent
            if d>0
              x=x-1;
              dx=cx;
              cx=sx;
              sx=FastClustering(x-1);
              d_dx=dx-cx;
              d_sx=cx-sx;

            elseif d<0
                  x=x+1;
                  sx=cx;
                  cx=dx;
                  dx=FastClustering(x+1);
                  d_dx=dx-cx;
                  d_sx=cx-sx;
                  d=dx-sx;

            else
                minimum=x;
                break
            end
        end
    end

    Clusters=KmeansClustering(minimum); %evaluates full clustering for local minimum
```

In the code it is possible to notice how, at the beginning of it, there are three evaluations of the Fast Clustering function. This is the minimum cost that this method

has for the optimization of the number of clusters. The computational cost of this optimization algorithm for the search of the local minimum is therefore at least 3 times higher than that of the Standard Approach. Also note how this function differs from the final function K means Clustering: the first one only calculates the information necessary for the evaluation of the cost function, while the second one calculates a whole series of parameters that will be illustrated in the next section. This was done in order to lighten the code and make the Fast Clustering function faster to perform. From subsequent tests it was found that the number of maximum iterations set was never reached. As the algorithm is written there is also only one other function evaluation per cycle.
After the first function evaluations in order to find the right, left and centered derivatives of the cost function, one enters the cycle. The first "if" evaluates if it already starts in a minimum, that is if the right and left derivatives differ in sign and if the right derivative is less than zero (without this second condition it isn't possible to distinguish the minima from the maxima). If this condition is not verified, it continues. The sign of the centered derivative is evaluated and the "slope descent" direction is then evaluated. It moves one unit left or right and reassign the derivatives of the previous step and calculate the missing extreme derivative.
At this point the cycle repeats itself.

4.6 Analysis Workflow

Here are listed all the operations done in MATLAB®code which resumes the workflow used in the analysis. It must be said that optimization loop is not described. In optimization loop K-means fast does not include all the features listed below. This following workflow is the basic one, valid for a single evaluation.

Clustering K means clustering ++ algorithm is performed given the number of partitions desired.

Main Calculations Raw data are assigned to a data structure which holds all data in analysis. Cluster populations and cluster coordinates are assigned to proper structure sites.

$\Delta\alpha$ Calculations $\Delta\alpha$ is calculated for each data point and it is also calculated for each cluter by integral averaging $\Delta\alpha$ of cluster population.

Radii Calculation Euclidean distance centroid and its population is computed. Minimum, medium and maximum radii are stored in the structure.

Statistics Calculations normal PDF of each cluster is computed and several different aspects of it are evaluated:

- multimodality
- mode
- position of most likely value
- skewness
- variance

These data are saved in the data structure.

Statistics check Values pertaining to the statistical analysis are evaluated and it is noted which clusters exceed the thresholds present on multimodality, variance, skewness and cluster size.

Failed check Clusters that exceed in all previous cases are reported as "failed clusters".

Quality Check Evaluates quality of the entire partition and stores indices of clusters in order of ascending quality.

Position plotting The population of the failed cluster data for each dimension is plotted.

PDF plotting The probability density function of the failed clusters is plotted.

Radii plotting All information relating to the rays of the data present in the failed clusters is plotted.

Quality check A quality function is used to compute quality for each cluster, then they are sorted in order of quality.

Data Storing The data are saved in different structures. These structures are in the input format to the neural network. Cluster-failed coordinates are saved separately. The coordinates of healthy clusters are saved.

Training network Training of the network with chosen clusters takes place.

Testing network Network is tested both on training and on test data.

4.7 Quality Check

Once a data partition has been generated, it is necessary to find a selective method for evaluating the obtained clusters. Based on the previously obtained statistical evaluations it was possible to have a basis on which to compare the clusters.

4.7.1 Failed Check

A first check is to evaluate some aspects and to set thresholds such that, if all are exceeded, the cluster is marked as 'failed'. In this way it will be then possible to carry out the analysis of these to understand how they could influence the training of the neural network. The aspects considered with the relative thresholds are:

Multimodality 1 (expresses whether or not the cluster is multimodal)

Dimension of cluster Population 10% of maximum cluster

Variance 0.01

Skewness 0.01

This means that all clusters with Variance and Skewness above 0.01 are marked. All the clusters smaller than 10% compared to the average population of the clusters, calculated as the total population normalized on the number of clusters, is marked.

4.7.2 Quality Ranking

For a high partition it has been noted that there are no clusters marked as 'failed', so a secondary mechanism of choosing low quality clusters is needed. This mechanism is always based on the evaluation of the statistical qualities of the various clusters, but without indicating the thresholds.
First, the variance, multimodality, population and symmetry of each cluster are collected. The first three are ordered from top to bottom, as having for example a high variance is a demerit factor for the cluster, while the last one is ordered in ascending order, since a high cluster population is a factor of about. The trend of one minus the quantities collected in the previous step was then plotted. For example, there is a trend like this:

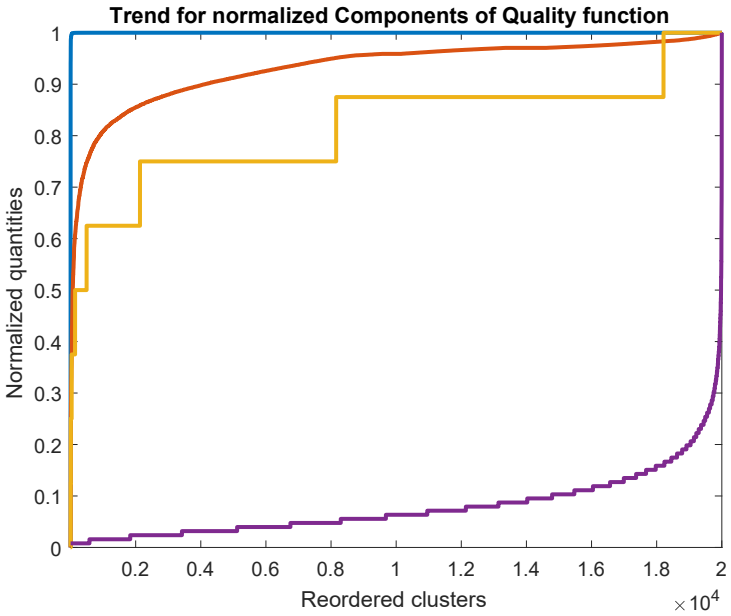


Figure 4.7: Blue: variance, Red: Skewness, Yellow: Multimodality, Purple: Population.

Here follows the explanation of figure 4.7. First of all, note the variance curve with just a few clusters with high variance (on the graph are those on the left) and then a whole series of clusters with low variance values on the rest of the blue curve. Bear in mind that the curves represents the reciprocal of the defined normalized variance. Also remember how variance is the factor minimized by the k-means algorithm. This means that there are a whole series of well-characterized clusters and a small part of clusters with scattered data. Skewness has a less steep macroscopic pattern than variance. Notice how the line is less marked. This is because, by enlarging the curve, it is characterized by a broken pattern. Once again, skewness is a secondary aspect of the analysis. Being characterized by natural numbers, multimodality is presented as a stepped function. Finally there is the population of clusters that has a different course from all those previously considered. In fact, there is a very flat trend for low quality clusters, which means that many clusters have low and similar populations and few clusters have high populations. The stepped character is also surprising, since the number of points per cluster is in all natural numbers, but it is also true that the possible range is very high. At this point, a function appropriately called "Quality Function" is created, given by the sum of these four factors analyzed and then normalized again.

Below it is possible to see what form it takes:

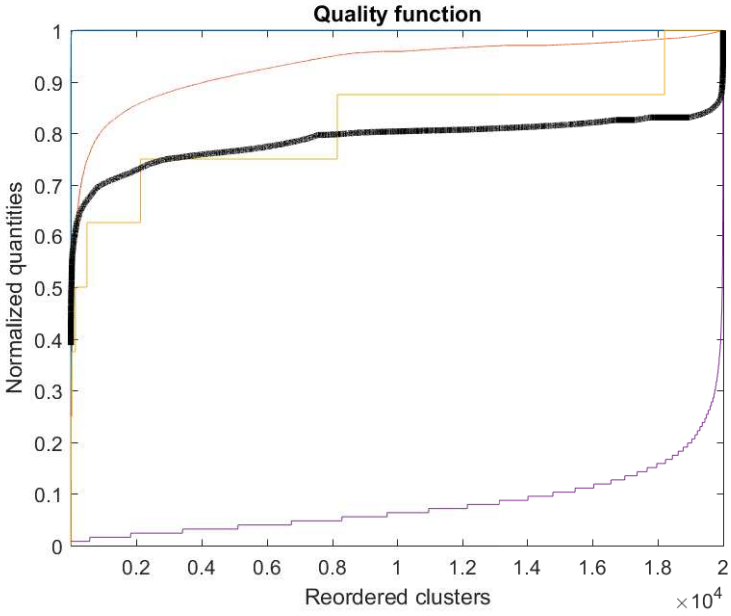


Figure 4.8: Quality function shown in black

This quality function is calculated for each cluster. The quality funtion is formulation is shown below:

$$Q = \left(1 - \frac{\sum Variance}{n.clusters}\right)\left(1 - \frac{\sum Skewness}{n.clusters}\right)\left(1 - \frac{\sum Multimodality}{n.clusters}\right)\left(\frac{\sum Population}{n.clusters}\right)$$

The 'S' shape of the Quality Function in figure 4.8 allows to find clusters of the highest quality (those present at the right side of the graph) and clusters of low quality (those present at the left side of the graph). In the middle, instead, there are medium quality clusters. For simplicity of analysis, if there were no failed clusters, the 10 clusters with the worst quality are analyzed. We must also consider how the 'failed' clusters are not necessarily those with the worst quality. This is mainly due to the fact that the calculation methods are different. As explained previously, the 'Quality function' is built giving equal importance to all the factors while the thresholds are fixed bounds. This means that a cluster having three very bad characteristics, for example Population, Skewness and Multimodality, but having an excellent Variance means that the cluster is not considered "failed" but has a very low quality. These differences must be kept in mind in the subsequent analysis.

4.8 Data Visualization

As explained above, the visualization of data for this kind of problems is particularly difficult. In the course of the thesis work, the typology of the plots has been changed several times in order to contain the salient information and to be easy to read. Finally we chose to plot only the data related to the clusters marked as "failed" so as not to accumulate a disproportionate amount of plots to be analysed. In fact, we are interested in investigating only the characteristics of the clusters to be improved rather than those of the valid clusters, which however we would like to be.

PDF visualization The plotting of the normal probability distribution is the basis of the post-processing analysis and it serves to evaluate the number of peaks and the shape of the distribution. On the plot there is the curve of the PDF in red with the probability on the y axis and the $\Delta\alpha$ on the x axis. In addition there are the remaining important information related to the cluster, that is the cluster population, the skewness and the variance.

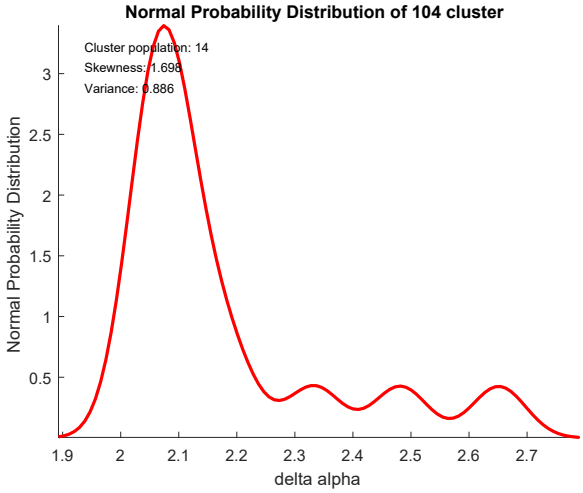


Figure 4.9: Probability Distribution plot for Cluster 104 in 1000 clusters partitioning (failed cluster).

Position Visualization Another important aspect to display is the geometric position of the clusters. Since the multidimensional space is the only way to display is to display one dimension at a time compared to the reference coordinate which is the $\Delta\alpha$. What you see is a cloud of points. The cluster population is highlighted in red while the centroid coordinate is represented by a blue circle. In this way it is possible to have an idea of the positioning of the cluster. Note that numerically this display could also be made numerically, evaluating the ranges with respect to the dimension to be analysed.

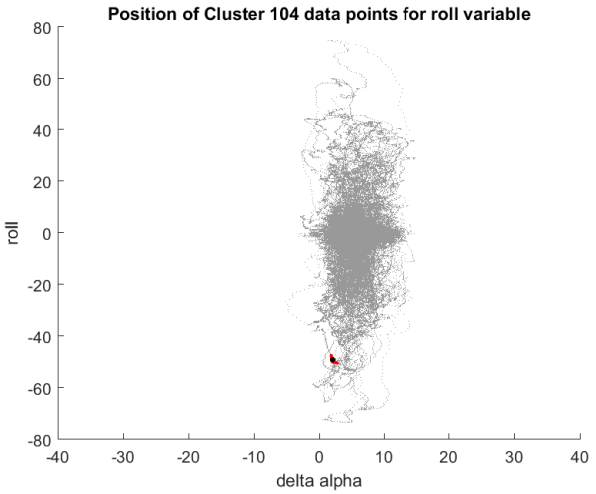


Figure 4.10: Roll data cloud plot for Cluster 104 in 1000 clusters partitioning (failed cluster).

Radii Visualization The radii display is the last type of plot that is used for this analysis. It is a circular-shaped plot in which the maximum radius and the minimum radius of the data are indicated by two black circles, while the medium radius is indicated by a red circle. The cluster population is then ordered from the datum with minimum radius to the datum with maximum radius. Remember that by radius the Euclidean distance in the multidimensional data space is indicated.
The n-dimensional Euclidean distance is computed as follows:

$$D = \sqrt{\sum_{k+1}^{n}(p_k - q_k)^2}$$

where p_k and q_k are the coordinates of the points in the k-th dimension out of n dimensions. The maximum, average and minimum radii are also written literally.

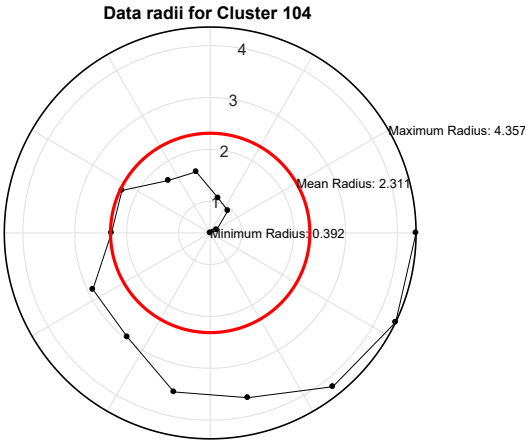


Figure 4.11: Radii plot for Cluster 104 in 1000 clusters partitioning (failed cluster).

Chapter 5

Comparative Analysis

In this chapter a comprehensive discussion of clustering quality will be illustrated.

The previously explained tools are now used to perform an analysis of the acquired flight data. The analysis is comparative in that it needs to be evaluated the differences between a subdivision in a few and in many clusters and how these differences affect the training of the neural network.

In this way it is possible to cover partially all the different possibilities of clusters dimension. It is importanto to notice that, not having a unique and solid way to determine which nuber of clusters to choose, both ways would be possible.

Fist of all choice of clusters dimension will be discussed. It will follow a description of population characteristics and then a final review of what found.

5.1 Cluster Dimensions choice

The initial points chosen for the optimized method are 20000 and 200 clusters. This choice is not random. Below a graph pictures where the two cases are positioned and which value they have with respect to the cost function.

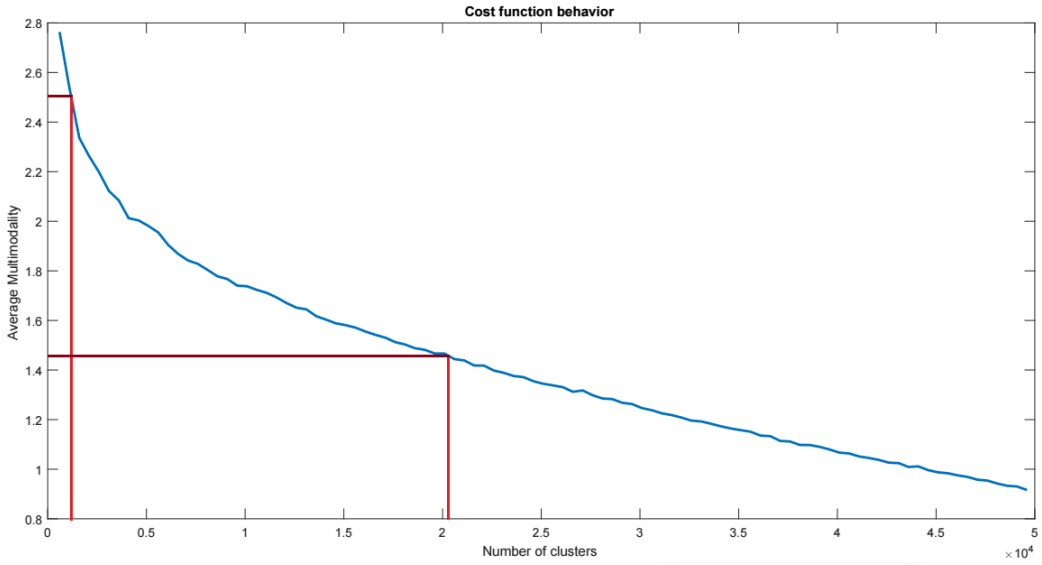


Figure 5.1: Two case choiched indicated on cost function behavior curve.

20000 clusters is placed at the value of the cost function, as can be seen in figure 5.1. Then 200 clusters had been chosen because it is exactly two orders of magnitude from the first one.

The simple choice is one of the may possibilities however it well represent two opposite conditions. In the study will follow a first part of the analysis on the 200 clusters and its comparison with the 20,000, which we will call respectively: case 1 and case 2.

5.2 Cluster Processing

First we analyze how the optimized method is performed on the 200 clusters. In the figure it is possible to observe how, with only 4 evaluations of the function, it reaches the desired result, that is the search for the nearest local minimum.

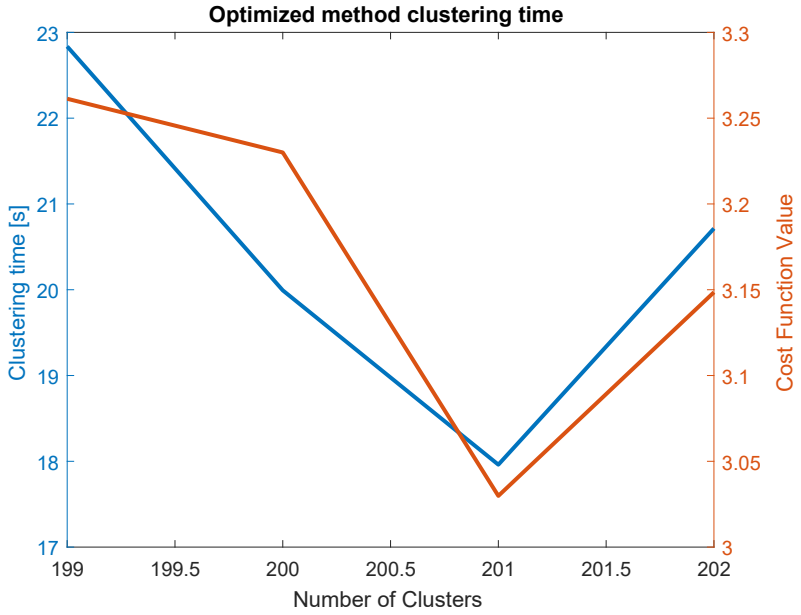


Figure 5.2: Optimized method starting from 200 clusters.

In figure 5.2 it is interesting to note that, at a lower value of the cost function, a lower calculation time is also associated. It would have been expected that the calculation time would scale as the number of clusters required. This trend, however, could be explained as the fact that the k-means algorithm, which was set on Matlab in order to have a maximum of 1000 iterations to improve the cluster position, converges well before the maximum limit in the case of the minimum local. The minimum local reached is 201 clusters.

However it is worth to mention that k-means clustering uses an heuristic method for initializing clusters, as explained in chapter 3. So to be completely sure that 201 represents a proper minimum the process of clustering should be done numerous times in order to ensure a valid selection. As this approach is computationally demanding for this study the authore preferred not to investigate how initialization affected local cost function behavior.

Again it needs to be considered that a different initialization of clusers could lead to a different local situazion. It can be estimated that the global condition would rest similar.

For case 2, there is a different situation.

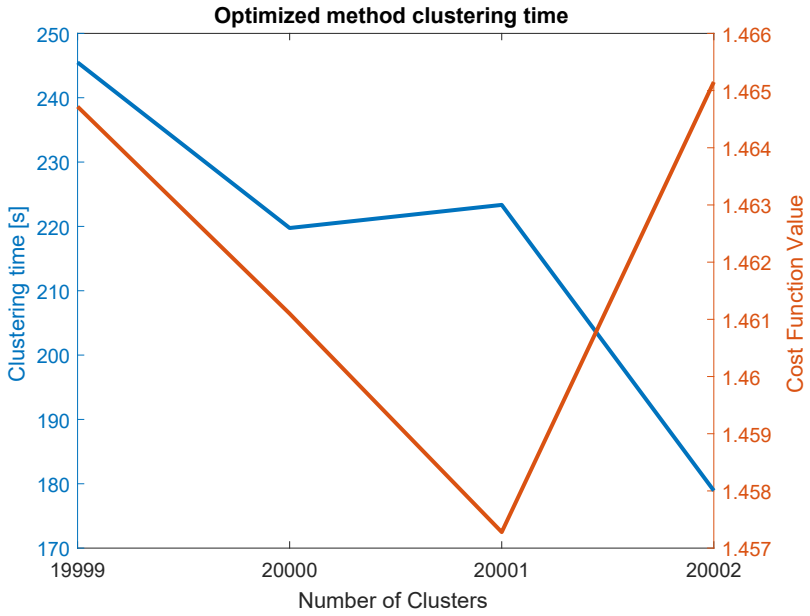


Figure 5.3: Optimized method starting from 20000 clusters.

In figure 5.3 the number of function evaluations required is always 4 but the time spent on the account is unexpected. The calculation time is inversely proportional to the number of clusters. However, this could only be a local condition or dependent on the scheduling of the supercomputer used rather than by a particular condition. The local minimum stands at 20001 clusters.

It should be noted by comparison how much the difference between case 1 and 2 is: there is a difference of more than one order of magnitude in the calculation time with the same computational conditions. Also note how the cost function is equally fragmented for both a small and a large subdivision.

5.3 Cluster Post-processing

The post-processing of the two cases begins with the evaluation of the quality of the computed clusters.

5.3.1 Failed Clusters checking

Regarding case 1, the results of the failed cluster analysis are as follows:

Failed Variance: 103 (51%)

Failed Skewness: 198 (98%)

Multimodal: 173 (86%)

Unconformal Dimension: 5 (2.5%)

Failed Clusters: 2 (1%)

As far as case 2 is concerned, the situation is as follows:

Failed Variance: 19241 (96%)

Failed Skewness: 19772 (98%)

Multimodal: 8157 (40%)

Unconformal Dimension: 580 (3%)

Failed Clusters: 0

At this point it is very important to carry out an analysis of these collected data. First of all, note how, in case 1, there are high percentages of clusters with three of the four high characteristics and the fourth the most restrictive one. In fact, out of 5 clusters with non-compliant dimensions, only two are defective overall.
Notice how the percentage of non-compliant dimensions between the two cases is almost the same. In the second case it is particularly interesting how, despite the high number of clusters with non-conforming features, there is not an intersection of all four sets. It is also important to note that the percentage of multimodality is halved but Variance and Skewness worsen considerably. This could be the first visible effect of the choice of multimodality as a cost function.
Another thing ti say is that flight data were acquired with high quality instrumentation, so it was expectable how the number of failed clusters, which is directly linked to data acquisition. This is also a good feedback on data quality and resampling itself.

5.3.2 Cluster Quality checking

The secondary method for evaluating clusters is by observing quality and its function.
First of all, the low quality clusters were selected and below are the identifiers of the 10 clusters with the lowest quality and the respective quality for the two cases:

CASE 1		CASE 2	
Cluster ID	quality	Cluster ID	quality
127	0.2616	10922	0.3877
5	0.2902	7844	0.4253
52	0.4614	4046	0.4460
120	0.4987	10922	0.3877
132	0.5261	5835	0.4520
19	0.5500	4619	0.4598
74	0.5562	16613	0.4662
149	0.5612	18154	0.4666
148	0.5614	13039	0.4785
111	0.5689	4876	0.4797

Being difficult to interpret below is presented a chart containing the quality values compared:

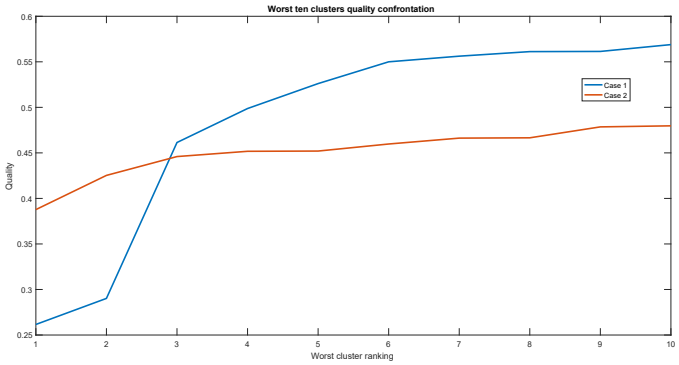


Figure 5.4: Worst clusters ranking confrontation for case 1 and 2.

In figure 5.4 it is interesting to note that the minimum quality is half for the first case but immediately the quality goes up, while in the second case it remains higher, but also more constant. This is primarily due to the complete distribution of the Quality Function. Being less clusters in the first case, there will certainly be a faster growth.
The comparison between the two Quality Functions is shown below:

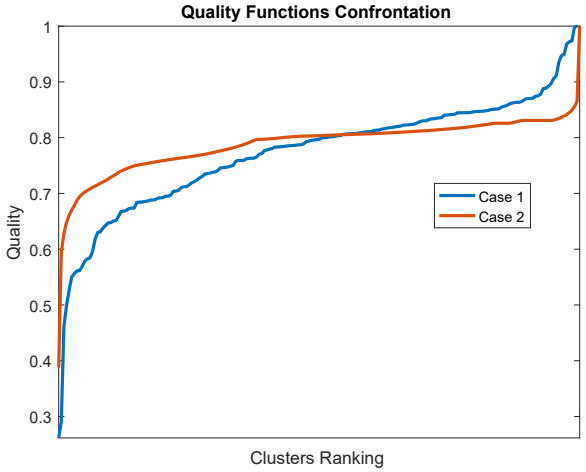


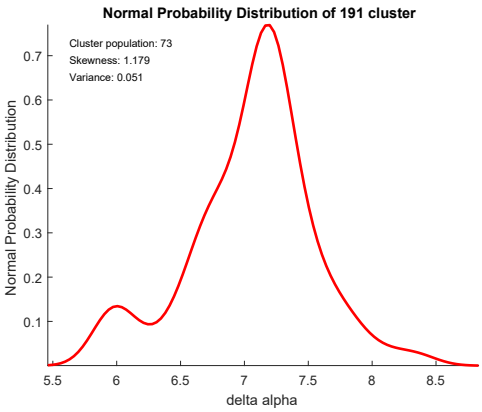
Figure 5.5: Quality function confrontation for case 1 and 2.

In figure 5.5 notice how the global trend remembers the local trend. In case 2, there is a greater discrepancy between the three cases of high, low and medium quality clusters. In case 1, also note how there are more fluctuations in the value. This trend is mainly due to the reduced number of clusters compared to case 2.

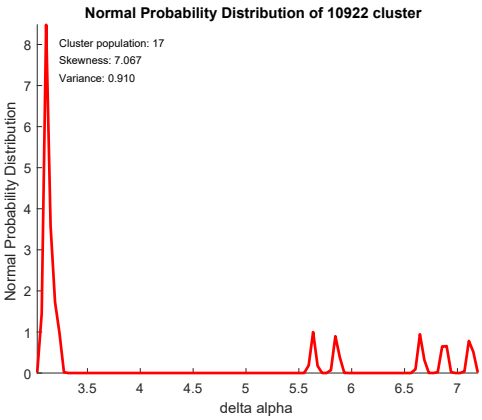
5.3.3 Clusters Content evaluation

At this point it is possible to carry out the actual analysis of the clusters marked as 'failed'. The graphs generated automatically as a report from the Matlab script used are analyzed. In this section three cases will be compared: one of the two clusters 'failed' for the first case, the worst cluster in the ranking of the second case and then the cluster with higher quality of the first and second case.

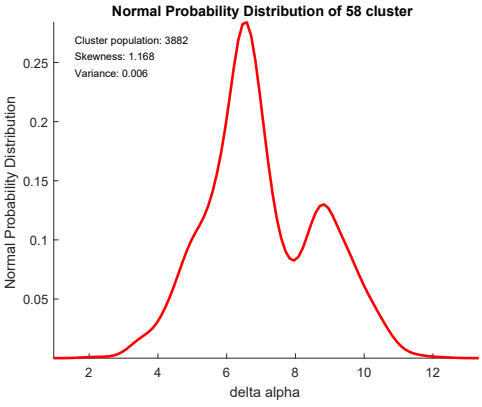
Probability Distribution The graphs below show the distribution of the Probability Desity Function or PDF. From the three graphs we can see that in general the trend is multimodal, just as it can be found in the cost function, which is more than 0. Not exceeding any of the cases the multimodal distribution, is considered a good cluster one with as few peaks as possible , with them very close together and with a good coverage of the cases contained among the maximums.



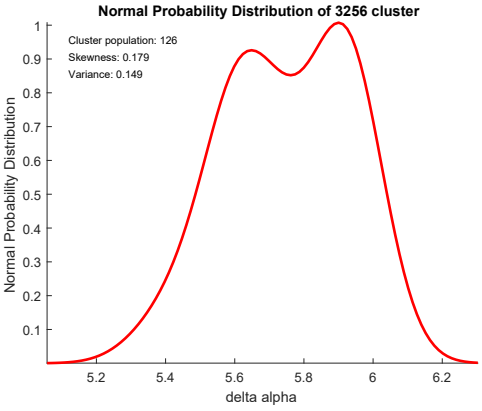
Cluster 191 is small and has a bimodal distribution. However, one of the two peaks is higher than the other.



The cluster consists of a population of only 17 points and poorly distributed. Notice how there are as many as 6 peaks in the graph. This trend certainly denotes a problematic cluster.

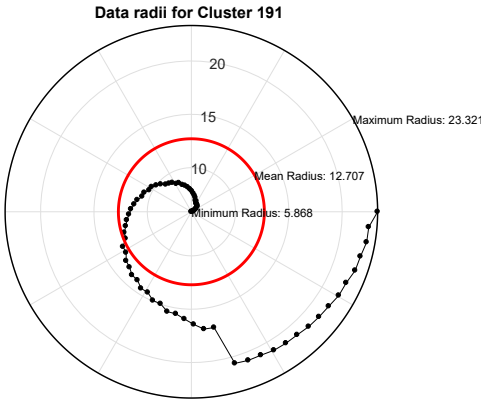


Again a bimodal trend, however the curve is rather narrow and the two peaks are quite similar. The population of this cluster is very high (almost four thousand points).

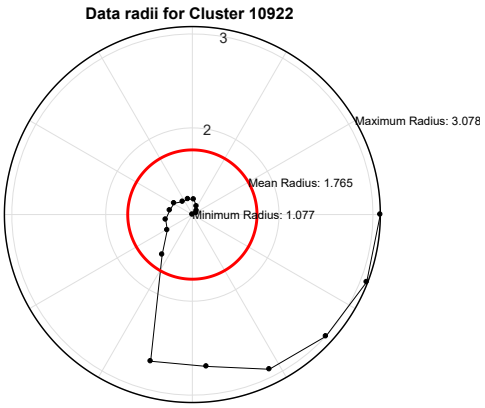


The 3256 cluster has a bimodal trend but with very close peaks and a well-matched function between them. The bell is also sufficiently narrow. The population is sufficiently high.

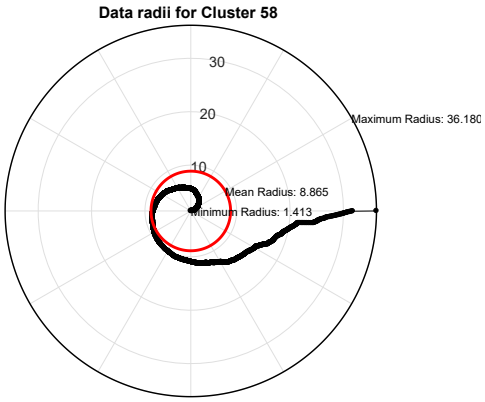
Distances between points and centroids he graphs below show the distribution of distances between the cluster population and its centroid. It is to be considered a good cluster that having the population contained in a well-circumscribed and small area, which would therefore have rays not exceeding a certain small amount.



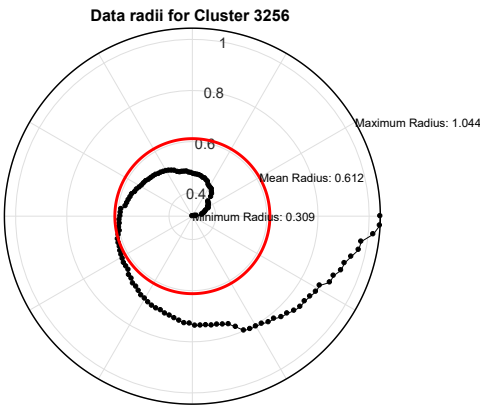
In this graph it is clear to note, for cluster 191, that there are two distinct series of radii and that both the medium radius and the maximum radius are very high.



As in the previous case there are rays divided into two blocks. Notice how, compared to the adjacent case, how the radii are smaller. This is due to the fact that having more clusters there are distances from the lower centroid.

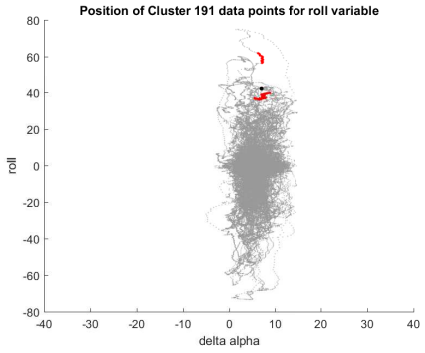


For cluster 58 there is a fairly unequal distribution of rays. This cluster also has lower average harness than cluster 191. Note also that the maximum radius is only high due to the fact that one point is more distant from the centroid than the others.

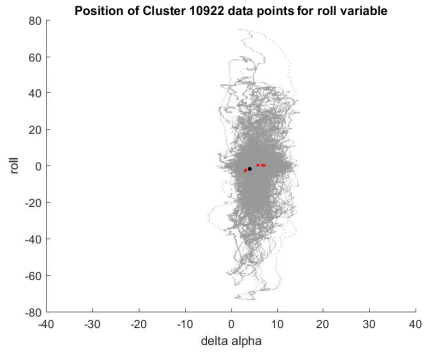


In this case there is an extremely compact distribution of the population and, in fact, the average radius is remarkably low. This represents quite well what one would expect from an ideal cluster.

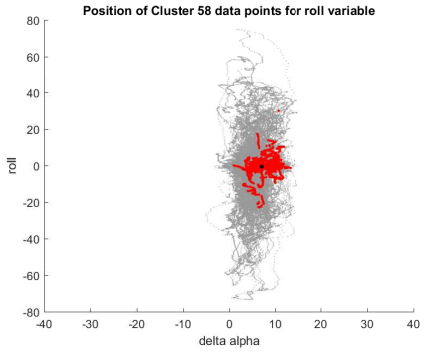
Roll angle .



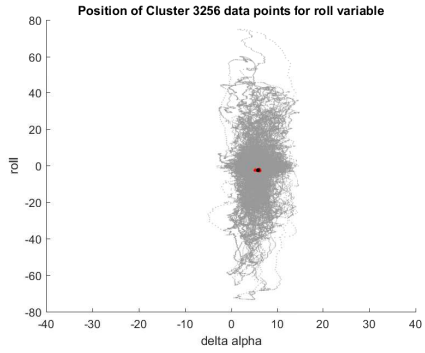
Notice how there are two groups of points equidistant from the center but very separate from each other. This behavior divided into two groups is also evident in other plots of the same cluster.



In this plot you immediately notice how small the cluster population is. The cluster is also positioned in a very dense area of points.

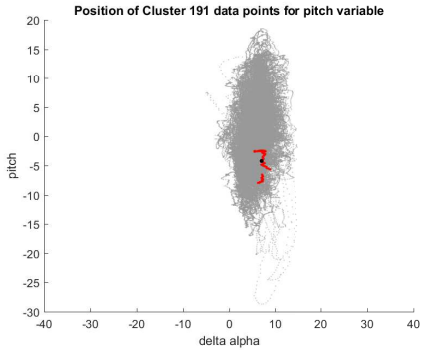


You can immediately see the remarkable extension of this cluster that covers a large part of the graph. The population is sufficiently well distributed.

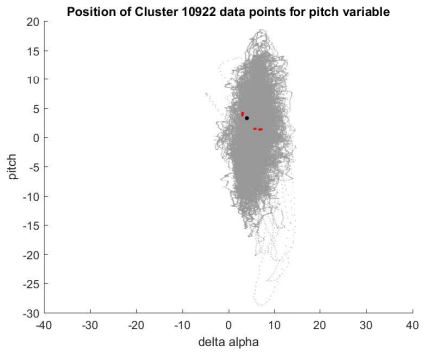


The graph does not notice how much the population of this cluster is concentrated, at least for this coordinate in a very narrow area.

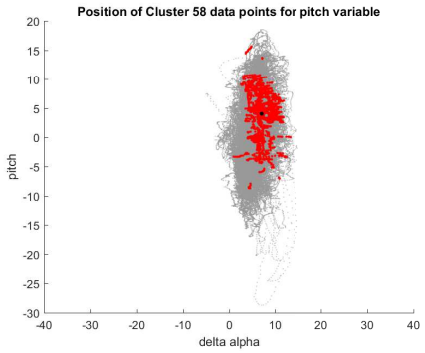
Pitch angle .



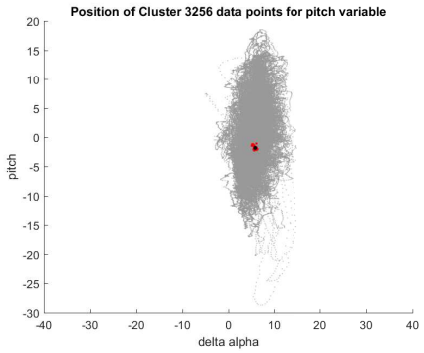
Here a noticeable vertical dispersion of the points can be noted. This means that the centroid of this cluster is attempting to represent many pitch values with a single $\Delta\alpha$ value. This is not necessarily a sign of poor cluster quality.



The horizontal dispersion presented in the plot is instead a sign of poor quality of the cluster as it seeks to summarize many $\Delta\alpha$ values with a single value.

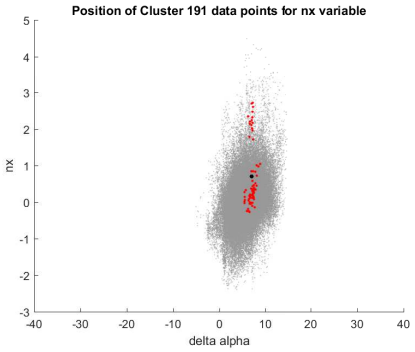


The dispersion of the points in the graph should not be misleading as the population of this cluster is very high.

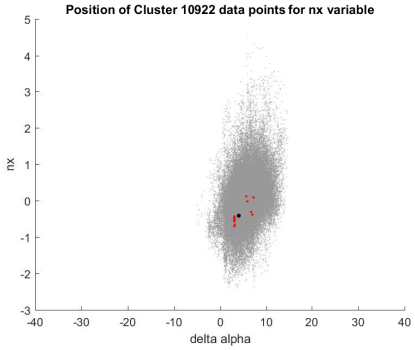


Once again the 3256 cluster has a very compact population.

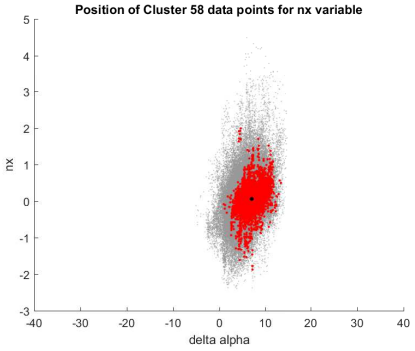
Longitudinal acceleration .



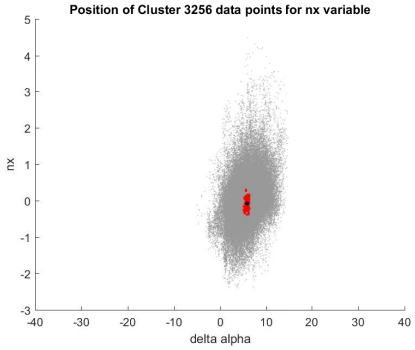
In this plot we notice a high vertical scatter of the points and once again a sharp division into two groups.



or this plot an extremely high dispersion of the points is observable.

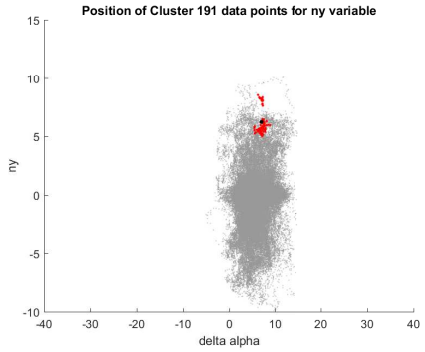


No particular specific trends are noted.

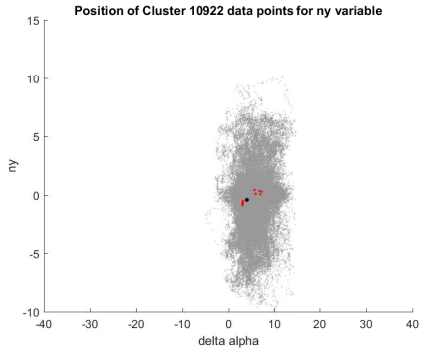


Strangely enough, for the 3256 cluster there is a slight verifiable dispersion.

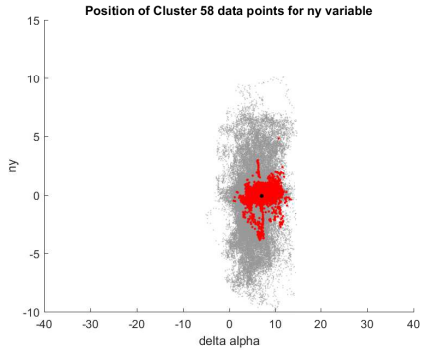
Lateral acceleration .



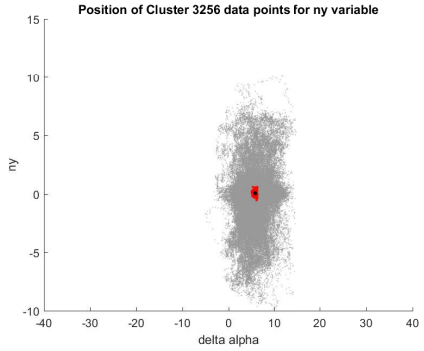
As in the previous plots, we note the division into two groups.



Also in this plot we notice a fairly distinct subdivision into two groups.

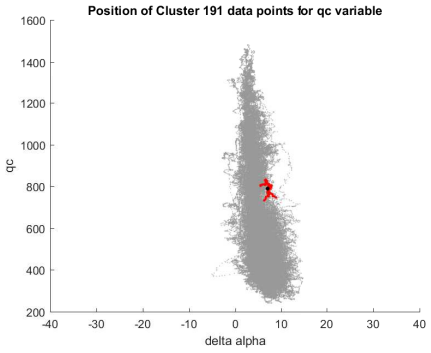


For cluster 58 we have a rather curious trend, or a fair number of points with the same value of $\Delta\alpha$ (8 degrees) but with very different lateral acceleration.

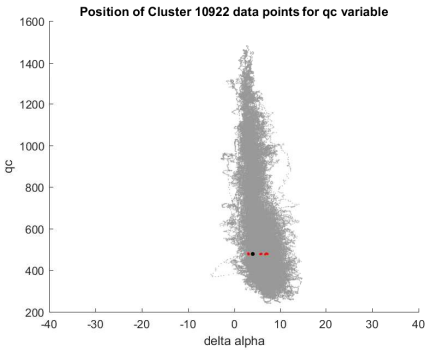


A slightly elongated distribution is noted.

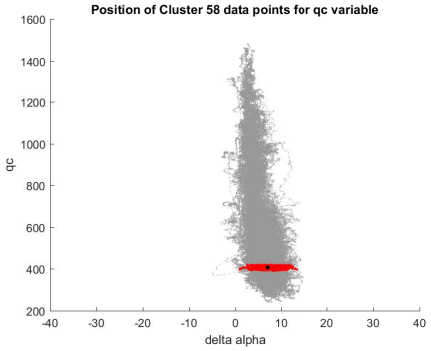
Dynamic pressure .



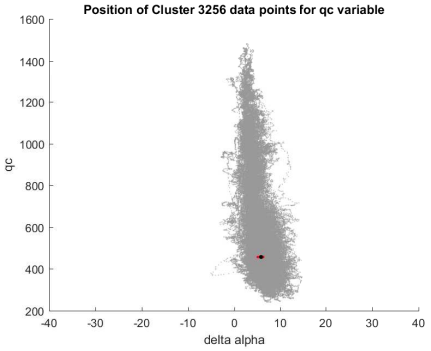
Dynamic pressure contains speed information. In this case cluster 191 represents a fair speed range for a fairly small alpha delta.



A horizontal scatter is observed in the cluster population distribution.

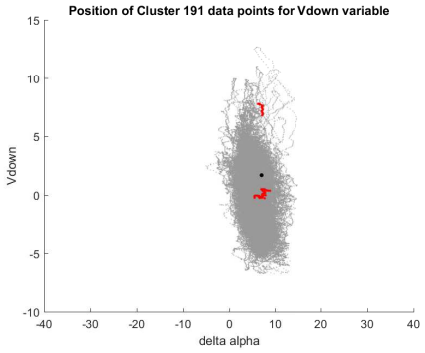


For cluster 58 there is an extremely strong trend, ie all points are part of maneuvers carried out in not very narrow range of speeds (low). This cluster proves to be well characterized as it summarizes the behavior of a series of well-defined states.

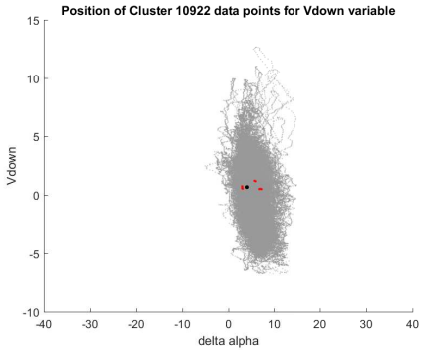


No particular specific trends are noted.

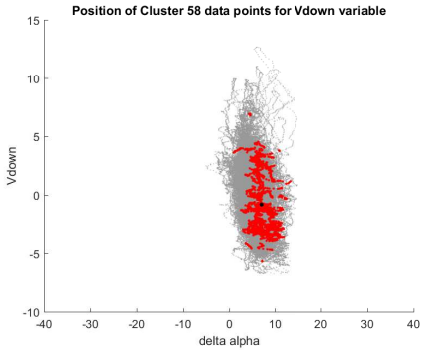
Vertical speed .



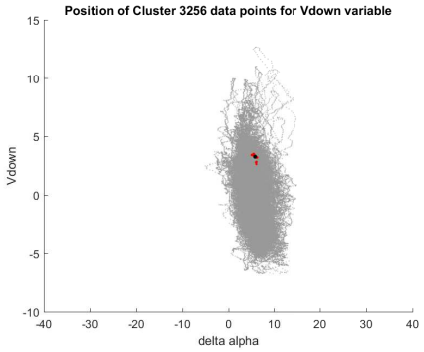
Once again the cluster 191 is divided into two distinct and extremely distant two zones.



A scatter is present in the points.



Cluster 58 has an extremely large population distribution with regard to vertical speed.



No particular specific trends are noted.

5.4 Conclusions for comparative analysis

The comparative analysis showed interesting differences between a clustering with few and many centroids. These differences are not limited to low-quality clusters, but are also evident in high-quality clusters.

However, the choice of the strategy to follow is not trivial. Surely the number of clusters has a significant influence on the speed of the analysis of the latter and in the calculation. We must not forget how the centroids are used to train the neural network. A large number of clusters can give overfitting problems and does not allow the use of training methods like Levenberg-Marquardt which are more computationally more expensive. A low number of clusters may fail to correctly summarize all phases of the flight.

Since the discrepancies in the number of clusters are not the source of enormous overall differences, a useful method could be that of clustering with a few indicated clusters and discarding the failed ones, carefully adjusting the thresholds used for selection. However, the complete answer to the problem must be amened to testing results. The next chapter is about studying how these strategies influence clustering.

Chapter 6

Neural Network Training

Once the data to be inserted in the network has been analyzed, it is now necessary to take care of the training and testing of the neural network. First of all, the characteristics of the network will be illustrated.

6.1 Neural Network Architecture

The chosen neural network presents only one hidden layer with 11 neurons each. Notice how the input contains exactly 11 inputs. The output layer instead consists of only 1 neuron. The network will convert incoming signals (which are p, q, r, Roll, Pitch, Yaw, Nx, Ny, Nz, qc and Vdown) in an estimated $\Delta\alpha$.
The type of network is of the "Feed Forward" type. This type of networks consists of one or more layers connected in a single direction. Each layer only has connections with the next layer.
The universal approximation theorem for neural networks says that each continuous function can be approximated with a feed forward network having only one layer providing enough hidden units. [4] Another variable in neural network architecture is the activation function. Precisely in compliance with this law, and wishing to carry out tests on a very simple network to minimize the calculation time, a single layer architecture was chosen. Even in a possible future view of the use of the on-board network, it is important to have a very simple network, which has very short calculation times and which can be performed with simple and light systems. In this regard, the hyperbolic tangent sigmoid is used as an activation function in this neural network.

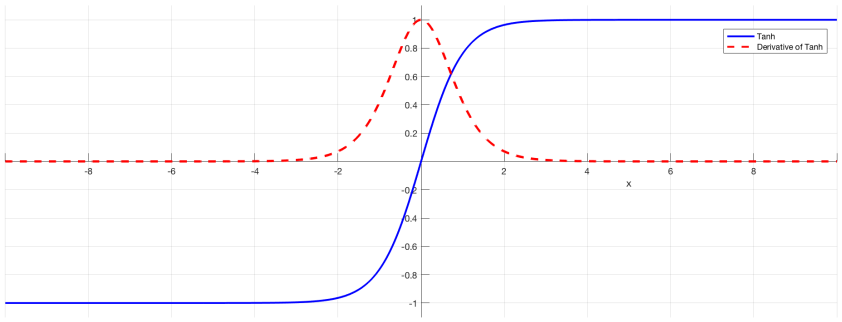


Figure 6.1: Hyperbolic Tangent sigmoid function and its derivative.

The transfer function is:

$$tanhsig(x) = tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

6.2 Neural Network Training

The methodology chosen for the training is that of Levenberg-Marquardt as it is the one that allows the minimization of the objective function in a smaller number of steps. A drawback is that it might be computationally more demanding than other methods.
The maximum number of iterations in the training is 1000. The maximum number

of validations is 10.
It was also decided to re-train the same network 100 times because the quality of the network is strongly dependent on the initial conditions chosen. In fact, at the beginning of the training there is an heuristic initialization of neurons. It is therefore a good practice to re-run the training in order to obtain a good fitting on the data provided. We must consider how the training performed by MATLAB®follows a very precise algorithm according to which not all the data related to the training are used for this purpose but some are used for checking the quality of the network. In fact, it is good practice to divide the input data into three subset.

The first is the training set, with which the real training of the network takes place. The second is the validation set and the error committed in evaluating this subset is monitored. In the first iterations of the process of calculating weight and bias this error drops. However when the network starts to overfit the data this error increases. The structure of the network is then saved to iteration with the minimum of this error. The last subset is that of the test. Normally, during the training the testing error is also calculated. This error is not used for training but is used to calculate the error to compare different models. If this error does not reach a minimum during validation, this may indicate a bad characterization of the network. The division of data between training and testing presents different strategies, including the random one used by default.
The standard Matlab settings are:

- Training set: 75% of the data
- Test set: 25% of the data
- Validation set: 25% of the data

Compared to the work carried out it would be conceptually wrong not to consider part of the clusters in the training phase as they are subtracting the salient information of the flight to the network. It would therefore be better to cancel both the validation and the test during the training phase in order to also consider overfitting. On the other hand, using MATLAB®'s training algorithm it could be possible to find better networks than those evaluated in this test. Another one could be cross-validation, that will be tested in future studies. A third strategy could be to alter the selection of data to be inserted in the validation and in the test set in order to effectively carry out the testing of the network in one step.

6.3 Neural Network Testing

Network testing was carried out both on the same sample of data used for training, which is here called self-test, and on a sample of flight data not used for training. Flight data not belonging to the training are data acquisition carried out together with data from the testing data. From these data the data belonging to taxiing on the runway, take-offs and landings have been subtracted.

6.4 Analysis choice

For this analysis 9 neural networks were instructed starting from different input data.
As a benchmark, a network was drawn up, always of the architecture described above, with all the data related to the training. This is clearly the simplest method of network instruction. Two negative factors of considerable importance must be considered in this methodology. The first is that the time and the computational resources necessary for training starting from complete data are very large. And this is precisely the reason for the thesis: that is to find a method of data reduction to make the training phase practicable.

The second reason, always connected to the fundamental reason of the thesis, is that of underfitting: that is when too much data is given to the network and this produces a underfitting error, ie the network is not able to estimate the behavior of cases for which has not been instructed. Also in this case a data reduction should be able to reduce this problem. Starting from clustering for 201 and 20001 clusters, the coordinates of the centroids have been extrapolated according to the following logic: the quality function has been divided into three zones, respectively low, medium and high quality, and the clusters have been used to train the network. An analysis was then carried out with all the clusters. The two failed clusters were removed from the partition with 201 clusters.
Of each network have been recorded, as previously written, 100 workouts, being 9 networks has reached a total of 900 workouts. Of these the average value of the error in the test and in the self-test and the best network was evaluated. Schematically:

Benchmark	All data from flight recordings
201-all	All clusters centroids
201-best	Best quality centroids
201-medium	Medium quality centroids
201-worst	Worst quality centroids
20001-all	All clusters centroids
20001-best	Best quality centroids
20001-medium	Medium quality centroids
20001-worst	Worst quality centroids

The complete test was performed once again on the HPC Hactar of the Polytechnic University of Turin. It must be said that the only true calculation that needs such computational resources is the "benchmark".

The comparison of the networks is done by calculating the error in the case of the self-test in the following way:

$$Error\,on\,training\,=\,\frac{\sum(|training\,data\,-\,net\,output|)}{number\,of\,training\,points}$$

While in the case of the actual test the formula used is the following:

$$Error\,on\,test = \frac{\sum(|test\,data\ -\ net\,output|)}{number\,of\,testing\,points}$$

Notice how to calculate the absolute value: this is due to the fact that it does not matter if the network error is due to excess or defect.

6.5 Test Results

Average error values are presented for the nine networks analyzed and the minimum errors committed by the networks. The first table contains errors on the network test on training data, while the second table contains those related to errors during testing.
The numbers that are read in the table are actually the average errors on the $\Delta\alpha$. The table is therefore to be read as: the Benchmark network makes a mean error of 0.3769 degrees on the delta alpha.

Auto-test		
NN Name	Best Res. [deg]	Average Res. [deg]
Benchmark	0.3712	0.3769
201-all	0.7028	2.695
201-best	1.1667	1.5684
201-medium	1.4385	2.2195
201-worst	1.6939	2.7384
20001-all	0.3876	0.3966
20001-best	0.3936	0.4061
20001-medium	0.3910	0.4024
20001-worst	0.3953	0.4064

The benchmark stands at the lowest values in this test. The obvious conclusion is that the network trained with all data has a minimal error on the test based on the data itself. Notice how there is very little difference between the best network and the average of the error on the 100 networks tested. Another important consideration is to notice that clusters-trained networks have errors in the tenth grade order compared to the non-clustered network. This is to be considered a small confirmation of the validity of the method. That is, starting from summary data it is possible to analyse many different network architectures with much lighter computational training, without encountering considerable errors.

As for the data coming from the 201 clusters, there are very marked differences between the four cases. The difference between the average values and the best networks is very important compared to the best one. The network trained with the highest quality clusters presents the best data. Also note how training with all clusters approximates the data set better. This may be due to the fact that, being few clusters, eliminating two thirds at a time of these are lost vital information to a good training of the network. As for the data coming from the 20001 clusters,

one immediately notices how the results are rather compact around similar values. In general this can be explained by the fact that all the networks, including that of Benchmark, have undergone all the same degree of overfitting.
It is interesting to note that the network including all the clusters is better in this test than the others belonging to the 20001 clusters. However, we must consider how the difference is not so important as to suggest specific conclusions. Comparing the data from the 201 and 20001 clusters, it is immediately clear how the latter obtained the best overall results in the analysis phase on the training data.

Test		
NN Name	Best Res. [deg]	Average Res. [deg]
Benchmark	1.6456	2.0835
201-all	1.7157	4.2266
201-best	1.7730	2.3142
201-medium	2.0267	2.8691
201-worst	2.2223	3.1811
20001-all	1.7610	2.1821
20001-best	1.5812	1.9644
20001-medium	1.6424	1.9308
20001-worst	1.7565	2.1756

For the test, it is observed that the benchmark network stands on an average error of about 2 deg on the delta alpha and about 1.6 deg as the best result. From now on, we can see how this is a noticeable difference between the best result and the average result.
As for the 201 clusters, we note that the differences are rather limited compared to the 20001 clusters. The average values are very high, however the minimum values are comparable. Notice how the "201 all" network has an average error value much higher than the others, in the order of double.
As for the networks trained on the 20001 clusters, it is noted that the error on the average of the 100 trainings knows minimal regarding the "medium" network, that is the one with the clusters with quality media. As for the results of the best networks there is a minimum error of 1.58 degrees for the "best" network. This is a confirmation of how the method is right, that is, the division into clusters and the use of only the best clusters makes the results slightly better. The gap between the best network of 201 and the best of 20001 is contained in the tenth of a degree.
bviously another aspect to consider are the timing. Consider how for the training of a single network with all the input data it takes about 4 minutes, while for a network deriving from the 20001 clusters just under 1 and, lastly, about 10 seconds for the 201 clusters network. Obviously these times are completely reasonable compared to the clustering phase, but we must consider how the training was carried out on very simple networks. In the case of more complex networks and a greater amount of data, the times will be much longer.

Chapter 7

Conclusions and Future Developments

First, a recapitulation of the method presented and discussed in this thesis is presented.

- The flight data are taken, the outputs of the various sensors are synchronized and resampled.
- The data created are treated with the k-means clustering algorithm in order to extract the salient points of the flight data.
- A "steepest descent" optimization method is used to find the local minimum of the cost function.
- The clusters found by means of statistics are processed in order to evaluate their quality.
- These clusters are used for the training of the neural network responsible for finding the angle of attack.

Looking back over the analysis presented in this text, an assessment of the work must be done and the differences between expected data and data found. First of all, the computational resources necessary to carry out the clustering of the sample are very large and must be taken into account for a possible future application of the method. Secondly, the use of k-means clustering also has considerable disadvantages, for example that of having to impose a priori the number of sample clusters. A possible evaluation of other clustering methods such as the "Gaussian Mixture", where the number of clusters is determined by the method could be considered. Note how much of the work done is due to the choice of the number of clusters and the census of their quality. Unfortunately, in order to test the validity of the method, the author had to face the whole pipeline from raw data to network testing, leaving little room to evaluate alternatives to the method. It reserves to those who will continue the research on the processing of input data in a neural network for the processing of flight data to deepen the possibility of finding alternatives.

As written, multimodality as a cost function was used to find the best clustering. Unfortunately, the strong irregularity of the function has been very decisive and unexpected in the search for the local minimum. Probably the study of a more refined cost function, which takes into account several factors, could have a smoother trend and therefore could facilitate the search for the appropriate number of clusters. It needs to be considered how the flight data are a particularly complex problem to analyse and, at least from the theoretical point of view, there may not exist a partition that can effectively summarize the various flight states.

Going forward in the analysis one must consider how the quality function has influenced the selection of the clusters. Also in this case the quality function has been elaborated in a very simple way so as to alter the result of the analysis as little as possible. A more refined selection of clusters could greatly improve the overall result of the network. The division of clusters for training in three sections is also a notable simplification to the method, which could provide a more complex methodology for the extraction of quality clusters for training.

In retrospect, the author probably would have given greater importance to the population of individual clusters, as this is a real watershed in the analysis. From the

training and testing of the networks it has come out how the networks trained with the best quality clusters, which are almost always the ones with bigger dimensions, have the best behavior on the others, albeit only slightly. It also must be considered how a defect that underlies this procedure, namely the fact that clustering causes the network to reach centroids that represent different populations, this means that a cluster with hundreds of points and a cluster with some points have the same weight in training. This is conceptually incorrect. A clustering with a high number of points allows to limit this problem as there are no clusters with excessively high populations.

Finally there is the neural network, which is the heart of this analysis. First of all it needs to be shown how this is initialized with a heuristic method, just like the centroides of k-means clustering. A good practice in these cases is to train different networks from different data in order to find the absolute minimum, or at least a very low local minimum. The same procedure should have been carried out for clustering. However, being an extremely expensive procedure from a computational point of view, it was not possible to carry out the analysis more than once. Not being the data of the networks trained with 201 and with 20001 clusters enormously different, probably choose a number of clusters not very high. It therefore allows the repeatability of the operation for a number of times to comfort those who perform it on reaching a local minimum consistent, or the absolute minimum of the problem, it could lead to important overall results.

The architecture of the network itself also affects the overall result. A mean error on the delta alpha of one and a half degrees for the best network is a disappointing result, considering that this network should replace or perform a back-up of a sensor with the precision in the order of tenths of a degree . By testing different network architectures, it is possible to reduce the error made by the network.

The method of calculating the error is not a Mean Square Error, as is usually done by expressing the error committed by a network. The fact of not carrying out the square means that the differences between the networks are rather small. Using the MSE could have more substantial discrepancies between the networks.

The merits of this method are certainly the speed in data reduction, and the possibility of having an extra control variable or data reduction, in the improvement of a network. As mentioned, this method is not complete and tested in all its variables, as it would have been a job well above the possibilities of a master thesis. This work as a possible forerunner for a technique that certainly has a promising future. In addition, the work has been set up from the point of view of the implemented MATLAB®scripts, so as to be easy to use to deepen all the aspects discussed. Also in terms of definition of the procedure, of the data structures and of the nomenclature, this work stands as a solid base from which to start.

Lastly, the author is extremely satisfied with the progress made on understanding the functioning of neural networks, on data analysis and visualization abilities, and on the use of software. These last aspects are not absolutly of secondary importance, as the thesis develops, in the life of a university student, a moment of semi-guided study that is important for providing tools in working life.

Bibliography

[1] Bradley, P. S. and Fayyad, U. M. (1998). Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer.

[2] Calise, A. J. and Rysdyk, R. T. (1998). Nonlinear adaptive flight control using neural networks. *IEEE control systems*, 18(6):14–25.

[3] Cireşan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220.

[4] Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48.

[5] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*, volume 1. MIT press Cambridge.

[6] Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

[7] Ivakhnenko, A. G. and Lapa, V. G. (1967). *Cybernetics and forecasting techniques*. North-Holland.

[8] Ivanka, H., Jan, K., and Jiri, Z. (2012). *Kernel Smoothing in MATLAB: theory and practice of kernel smoothing*. World scientific.

[9] Lee, T. and Kim, Y. (2001). Nonlinear adaptive flight control using backstepping and neural networks controller. *Journal of Guidance, Control, and Dynamics*, 24(4):675–682.

[10] Lerro, A. (2012). *Development and Evaluation of Neural Network-Based Virtual Air Data Sensor for Estimation of Aerodynamic Angles*. PhD thesis, Politecnico di Torino.

[11] Linse, D. J. and Stengel, R. F. (1993). Identification of aerodynamic coefficients using computational neural networks. *Journal of Guidance, Control, and Dynamics*, 16(6):1018–1025.

[12] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[13] Minsky, M., Papert, S. A., and Bottou, L. (2017). *Perceptrons: An introduction to computational geometry*. MIT press.

[14] Napolitano, M. R., An, Y., and Seanor, B. A. (2000). A fault tolerant flight control system for sensor and actuator failures using neural networks. *Aircraft Design*, 3(2):103–128.

[15] Ojha, V. K., Abraham, A., and Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60:97–116.

[16] Oosterom, M. and Babuska, R. (2000). Virtual sensor for fault detection and isolation in flight control systems-fuzzy modeling approach. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 3, pages 2645–2650. IEEE.

[17] Parzen, E. (1960). *Modern probability theory and its applications*. John Wiley & Sons, Incorporated.

[18] Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

[19] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[20] Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.

[21] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229.

[22] Samy, I., Postlethwaite, I., Gu, D.-W., and Green, J. (2010). Neural-network-based flush air data sensing system demonstrated on a mini air vehicle. *Journal of aircraft*, 47(1):18–31.

[23] Shin, D.-H. and Kim, Y. (2004). Reconfigurable flight control system design using adaptive neural networks. *IEEE Transactions on Control Systems Technology*, 12(1):87–100.

[24] Weng, J., Ahuja, N., and Huang, T. S. (1992). Cresceptron: a self-organizing neural network which grows adaptively. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 1, pages 576–581. IEEE.

[25] Weng, J. J., Ahuja, N., and Huang, T. S. (1993). Learning recognition and segmentation of 3-d objects from 2-d images. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 121–128. IEEE.

[26] Weng, J. J., Ahuja, N., and Huang, T. S. (1997). Learning recognition and segmentation using the cresceptron. *International Journal of Computer Vision*, 25(2):109–143.

[27] Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavior science. *Unpublished Doctoral Dissertation, Harvard University*.

[28] Whitmore, S. A. and Ellsworth, J. C. (2008). Simulation of a flush air-data system for transatmospheric vehicles. *Journal of Spacecraft and Rockets*, 45(4):716–732.

[29] Wise, K. A. (2005). Computational air data system for angle-of-attack and angle-of-sideslip. US Patent 6,928,341.

[30] Xu, B., Gao, D., and Wang, S. (2011a). Adaptive neural control based on hgo for hypersonic flight vehicles. *Science China Information Sciences*, 54(3):511–520.

[31] Xu, B., Sun, F., Yang, C., Gao, D., and Ren, J. (2011b). Adaptive discrete-time controller design with neural network for hypersonic flight vehicle via back-stepping. *International Journal of Control*, 84(9):1543–1552.

[32] Xu, H., Mirmirani, M. D., and Ioannou, P. A. (2004). Adaptive sliding mode control design for a hypersonic flight vehicle. *Journal of guidance, control, and dynamics*, 27(5):829–838.

[33] Youssef, H. and Juang, J.-C. (1993). Estimation of aerodynamic coefficients using neural networks. *Flight Simulation and Technologies*, page 3639.

[34] Zell, A. (1994). *Simulation neuronaler netze*, volume 1. Addison-Wesley Bonn.

.

.

.

Computational resources provided by hpc@polito,which is a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://hpc.polito.it).