Acquisition and processing of critical psycho-physiological and flight parameters in light-sport aviation

Eduardo Araújo
n°65542, MMec

1 Introduction

The current document presents the developed work in the scope of Intelligent Systems discipline, where it was required to build a predictive model which could relate the pilot's performance, the aircraft control and the flight safety.

The data available to base the model upon is composed of several psycho-physiological and aircraft variables acquired in simulated flight environment – hypobaric chamber.

The approach herein presented makes use of artificial neural networks with sequential feature selection and was implemented resorting to the MatLab Neural Network Toolbox.

2 Motivation

In the dynamic aviation environment, the pilot's well-being is a crucial and demanding factor which is directly related to his/her good performance. Due to the unpressurised and non-acclimatized aircraft cabin, light-sport pilots are exposed to a wide range of environmental and psychological conditions. In addition, distinct pilots react in different ways to the same flight conditions. Therefore, it is very difficult to establish general safety limits regarding psycho-physiological factors.

The preliminary results obtained show that under some specific conditions, the stress level may alter the physiological behaviour and the normal response to an external stimulus, increasing physiologically its intensity, and even compromise the pilot's safety, especially, if he/she is not aware of his/her physiological limits. The alert system to be developed is expected to improve the flight safety by making the pilot aware of his/her own psycho-physiological comportment, as well as assisting him/her to make decisions to improve his/her performance.

3 Problem Statement

In light-sport air-crafts pilots fly in unpressurised cabins, being constantly exposed to atmospheric pressure variations. The pressure decrease associated with the increase in altitude is responsible for the hypoxia phenomenon.

Hypoxia is defined as "a state of oxygen deficiency in the body sufficient to impair function of the brain and other organs". Altitude (hypoxic) hypoxia is a physiological concern since it impairs vision, judgement, motor control, and can result in incapacitation or, in severe cases, death. [8]

It is, therefore particularly dangerous for amateur pilots who are not well trained and are not able to recognize its symptoms, having several incidents/accidents already occurred due to this phenomenon. [6]

It is possible to experimentally determine in hypobaric chamber the Time Useful of Consciousness (TUC), a parameter defined as the amount of time an individual is able to perform flying duties efficiently in an environment of inadequate oxygen supply. [7], [9]

Although this is a serious and well documented phenomenon there is some negligence regarding the aviation physiology, since there are no requirements for ground training in flight physiology for light-sport aircraft and glider pilots. [6], [9]

The goal is then to use both the pilot's psycho-physiological and the aircraft flight data to develop a predictive model that gives an alert to the pilot whenever he may accuse initial hypoxia symptoms.

4 Acquired Data

The data available to base the model upon is composed of several psycho-physiological and aircraft variables acquired in a hypobaric chamber in order to simulate a flight of 1 hour and 14 minutes reaching a maximum altitude of 2,584.09 m (8,478 ft). The equipment recorded all data related to the flight, the pilot's cerebral oximetry and ECG, which provided information about the heart rate variations. After the data synchronization, a graph similar to the one presented in figure **??** can be obtained.

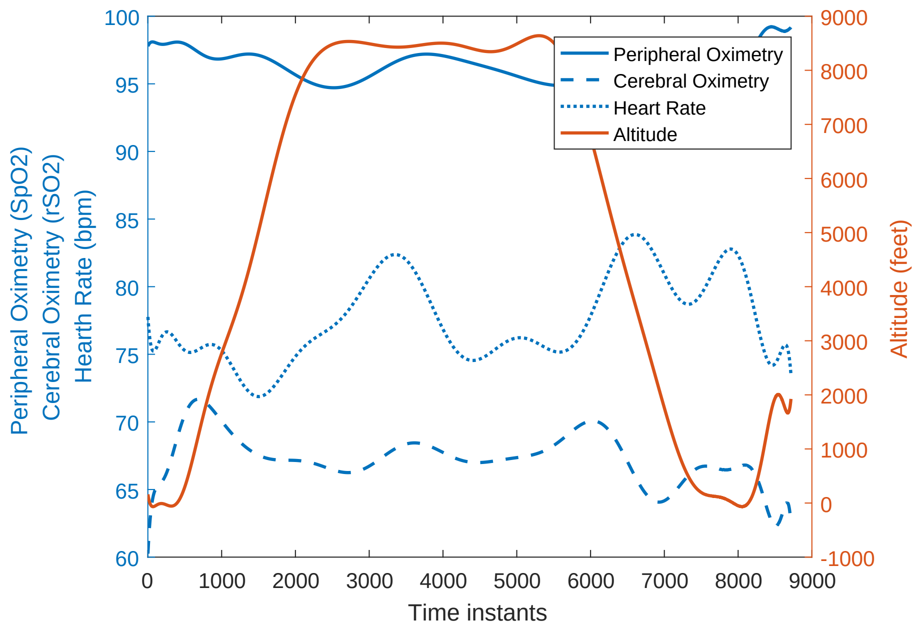


Figure 1: Data Presentation

This chapter begins with an overall presentation of the acquired data, followed by a dedicated section to data imputation and synchronization. After this a preliminary analysis on the raw data is performed. The chapter ends with a feature importance assessment, which will provide the reader with a better insight into the available data.

4.1 Data Imputation and Synchronization

The pilot was subject to 3 different flight simulations (from here on simply referred to as flight). Table 1 presents the recorded data as well as their number of samples per flight.

The consequence of having used different equipment to record data was that sampled time series were collected in a misaligned uneven fashion, which required alignment correction into a regular time series template.

Technically, time series are said to be misaligned when their samples are not recorded with the same sampling time. This misalignment may occur in two ways: evenly or unevenly. [5]

Although the aforementioned variables were not acquired at the same time rate they are equally spaced in time, i.e. they are misaligned but evenly sampled [5], which greatly simplifies the problem of synchronizing data, since it only takes simple interpolation to mitigate the misalignment.

Table 1: Number of samples of each variable per flight

	Altitude	Cerebral Oximetry	Peripheral Oximetry	Heart Rate
Flight 01	7643	108	13	13
Flight 02	7463	111	14	14
Flight 03	8716	202	14	14

Data imputation was performed initially resorting to two different methods: making use of a zero-order-hold (ZOH) and doing a simple linear interpolation. However, as will be discussed in the next chapter, there was some problems caused by discontinuities, which would be mitigated by means of a polynomial interpolation, emulating a pre-filtering process.

4.2 Preliminary Analysis

From figure **??** it is possible to infer some dependency of some variables, the most apparent of which is the change in heart rate (HR) and peripheral oximetry (SpO2) with the change in altitude. On the other hand, it looks as if there is no connection between the change in altitude and the Cerebral Oximetry (rSO2) – the variable to be estimated, as will be discussed in chapter **??**.

In order to have a more quantitative insight it was performed a correlation analysis followed by a feature importance assessment.

4.2.1 Correlation Analysis

Figure **??** shows the Correlation Matrix for the variables available.

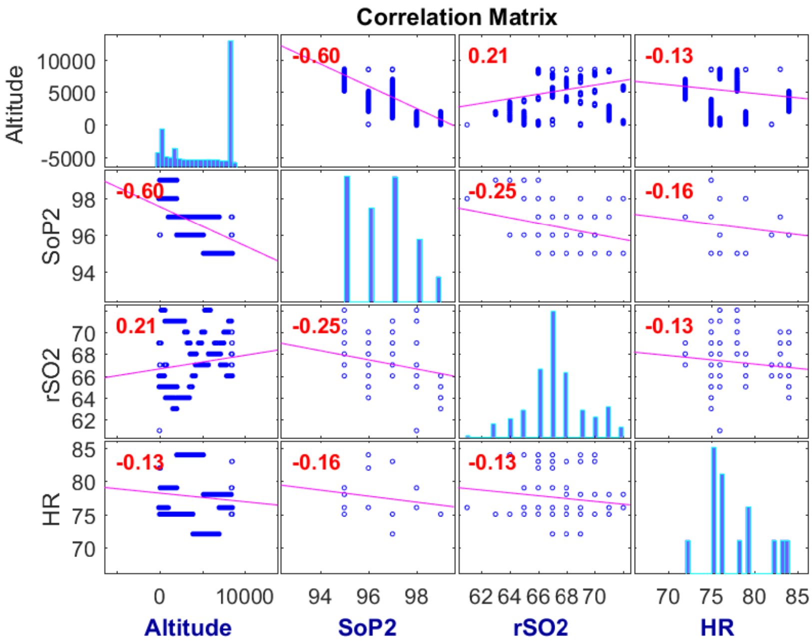


Figure 2: Correlation Matrix - Pearson's coefficient

Since the computed values are very similar and specially because there is a non-linear relationship between variables, the calculations were repeated using different coefficients. Table 2 shows a summary of the results.

Table 2: Summary of Correlation Values

	Altitude	SoP2	rSO2	HR
	Pearson	-0.60	0.21	-0.13
Altitude	*Spearman*	-0.53	0.24	-0.03
	Kendall	-0.39	0.15	-0.02
		Pearson	-0.25	-0.16
SoP2	...	*Spearman*	-0.18	-0.20
		Kendall	-0.14	-0.14
			Pearson	-0.13
rSO2	...	...	*Spearman*	-0.18
			Kendall	-0.13
				Pearson
HR	...	...	...	*Spearman*
				Kendall

4.2.2 Feature Importance Assessment

Based on the previous results it is possible to sort variables according to the degree of dependence as shown in table 3.

Table 3: Variables sorted by degrees of dependency

Pearson	Spearman	Kendall
Alt – SpO2	Alt – SpO2	Alt – SpO2
rSO2 – SpO2, Alt	rSO2 – Alt, SpO2	rSO2 – Alt,SpO2
HR – SpO2, Alt, rSO2	HR – SpO2, rSO2	HR – SpO2, rSO2

Making use of the MatLab built-in functions, it is quite easy to perform a different, more standard feature importance assessment. Namely, using the **relieff** function it is possible to compute the importance of each attribute.

As an example, this method was performed in order to assess the variables' importance to estimate the rSO2's value. Which turned out to be as follows: rSo2 – SpO2, HR, Alt.

Taking into account the correlation analysis, this result is not surprising, but the fact that two different methods produce the same conclusion shows the possibility of existence of an underlying physical mechanism that can and should be exploited for estimation purposes.

5 Modelling

As stated in chapter 3, the fundamental goal is to develop a model that alerts the pilot in case of hypoxia.

From a Control Systems point of view this can be easily transformed into an identification problem where the goal is to estimate the output variable ($\hat{y}$) knowing the input variables (u). Once $\hat{y}$ is known it is possible to compare it with the previous data acquired during on-land training and give an alert according to a previously determined threshold of safety. Figure **??** illustrates the proposed system.

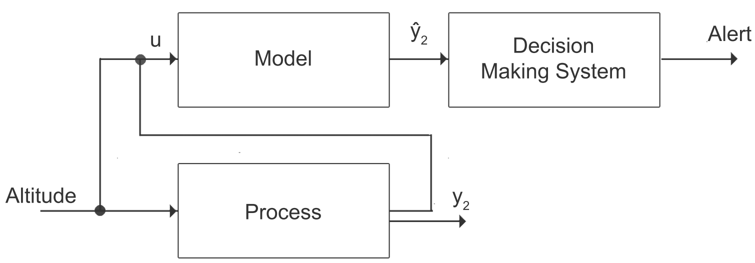


Figure 3: System's block diagram

The decision making algorithm can be as simple as the following rule:

$$R_1 : \textit{if } \hat{y} < c_t, \textit{give alert, otherwise do nothing.}$$

Clearly this solution implies some previous knowledge of the pilot's response to low partial oxygen conditions, i.e. TUC. An alternative approach would be to have standard thresholds, i.e. thresholds based on population data rather than individual data. Although the latter is more general – and consequently more error prone – it also presupposes a classification of pilots according to well established criteria.

5.1 Input and Output Variables Assignment

In figure **??** it was implicitly assumed altitude to be the sole cause of all process' outputs. It may seem a hasty decision to make, but taking into account the discussion present in chapter 3 altitude is the primordial cause of hypoxia.

Moreover, with both data and some basic biological knowledge it is easy to formulate a possible thesis on how all variables are related to each other. The rational being: the variation in altitude (Alt), causes a change in heart rate (HR), which in turn have implications on oxygenation – peripherical (SpO2) and cerebral (rSO2) oxymetry. In mathematical terms one can write this relationship as follows:

$$HR = F\,(Alt)$$
$$(SpO2, rSO2) = F\,(HR)$$

It is even possible to extend the second expression assuming a concomitant behaviour between SpO2 and rSO2, giving the following:

$$HR = F\,(Alt)$$
$$(rSO2) = F\,(HR, SpO2)$$

The point of this discussion is not to prove causation, but simply to develop a valid model, i.e. a system that is causal and respects the physical constraints encoded in data.

From the previous discussion is clearer which variables to assign to the input and output of the system:

- INPUT (u): Alt, HR, SpO2;
- OUTPU ($\hat{y}_2$): rSO2.

As a final remark on this topic, one should ponder the possibility of something more than the measured and measurable variables playing an important role in this situations, in other words, could emotions (namely stress) be an important variable to take into account? And if so, how might we be able to model this new variables?

5.2 Model Architecture

Taking into account the fact that the problem at hands so clearly implies the need for temporal processing, there are only two possible networks one can choose from the literature: [1]

- Time lagged feedforward networks (TLFNs)
- Recurrent neural networks (RNNs)

Time lagged feedforward networks (TLFNs) According to [1], TLFNs can be devided in two groups: *focused* and *distributed*. The main difference having to do with where the time-delays – which can be interpreted as *short-term memory* – are located within the network itself. The fact that in a focused TLFN delays are located at the front end of a static network limits its practical use to stationary (i.e. time-invariant) environments. In contrast, the implicit representation of time, as the name implies, is distributed throughout the network in the case of distributed TLFNs, allowing them to cope with nonstationarity (i.e. time-varying).

Recurrent neural networks (RNNs) In the neural network literature, networks with one or more feedback loops are referred to as *recurrent networks*. [1] RNNs are a powerful class of neural networks that are naturally suited to modelling time-series and other sequential data. [10]

In particular, the nonlinear autoregressive network with exogenous inputs (NARX) model is commonly used in time-series modelling applications [4], since its output depends, at each time step, on previous inputs and past computations, allowing the network not only to develop a memory of previous events, but also to be capable of approximating arbitrary non-linear dynamical systems. [10] On the other hand, the very feature that makes this model so versatile can also cause harmful effects, noting that a system that is original stable can become unstable by simply applying a feedback. [1]

Not being able to decide based on theoretical grounds which one would be the best solution for the problem herein presented, one had to investigate and decide based on experimental results.

In a nutshell, two particular architectures were subject to experimental analysis:

- Distributed Delay Neural Network
- NARX Feedback Neural Network

The following section will focus on the adopted methodology.

5.3 Methodology

5.3.1 Data partitioning

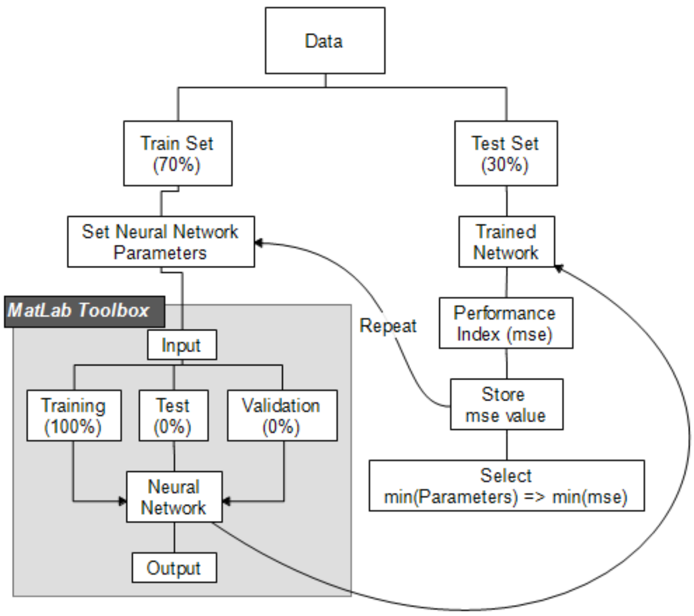
Two different methods of data partitioning were employed to create the train, test and validation sets.

There were available 3 datasets (Flight 01, Flight 02 and Flight 03).

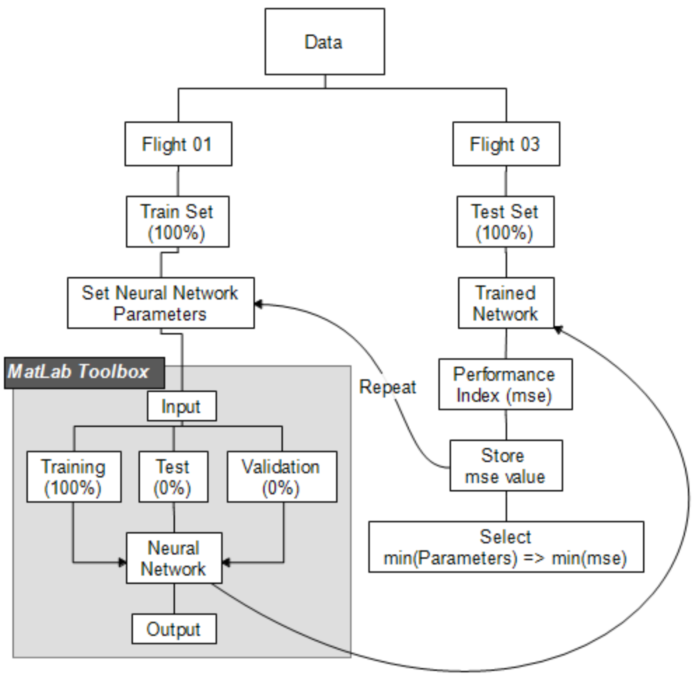
The first approach was to divide one dataset as training, test and validation sets. It was based on the fact that each flight followed the same dynamics that could be divided into 3 main sections: ascending, settling, descending. Each section was then subdivided into 2 subsections for training (70%) and testing (30%). This method had the problem of increasing the error due to discontinuities.

To mitigate this, a second method was employed making use of the whole data available. In addition, to emulate a pre-filtering process, a polynomial interpolation of the data was performed, instead of the previous methods (ZOH and linear interpolation).

An illustration of both methods can be seen in figure **??**, as well as one of the methods employed to determine the networks' parameters – to be discussed in the following section.



(a) Method I



(b) Method II

Figure 4: Employed methodology for data partitioning and network parameters determination

5.3.2 Networks' Parameters Determination

In a way of getting a general idea of what the appropriate number of neurons to choose for the problem at hands, each network was trained for different numbers of neurons, fixing the number of time-delays (td), as shown in figure **??**.

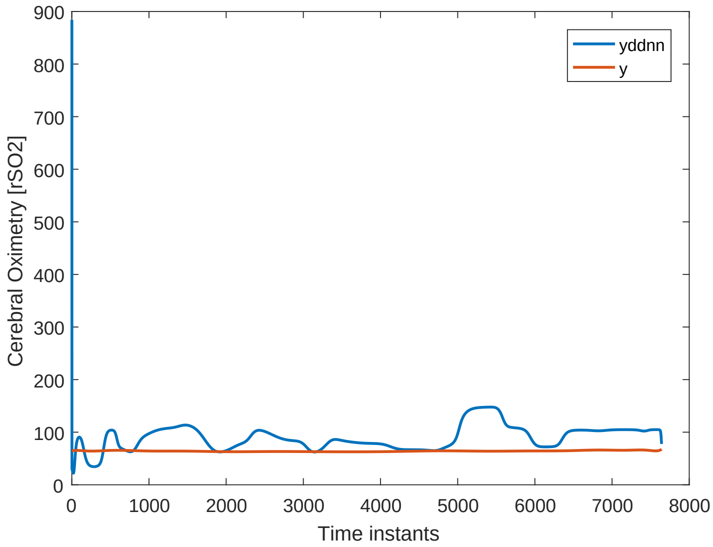
Having no clue about what would be appropriate values to this parameter, it was used 2 and 4 td.

Having determined 10 to be the best number of neurons to use the analysis focused then on which input variables would give the best results. Table **??** presents the performance results for each type of network.

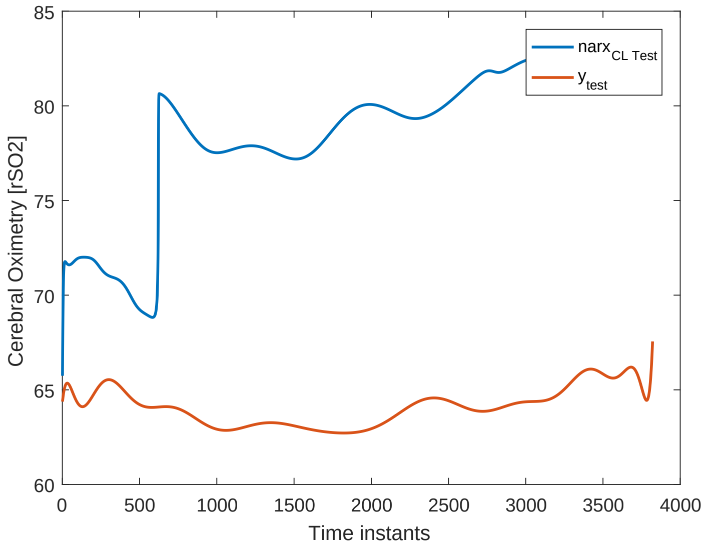
Table 4: Networks' Average Performances (mse) – *Networks' Parameters: 1 hidden layer, 10 neurons*

	NARX		**DDNN**	
Input Variables	***2 td***	***4 td***	***2 td***	***4 td***
HR	39.9822	49.4351	465.5827	3.6002e+03
Alt, HR	52.3692	52.1761	66.1305	68.7651
Alt, SpO2, HR	55.3377	106.809	194.1043	210.1319
Alt, SpO2	65.4850	36.4821	150.1289	104.8496

If the previous table does not seem clear enough, the following graph illustrates these results.



(a) DDNN



(b) NARX

Figure 5: Illustration of a typical network response

To have a better understanding of what might be the cause of these bad results, the performance assessment process was broken into steps and was focused on the NARX network, since, although being the more complex of the two, it was the one that performed better:

- Performance assessment during training:
  - open loop performance check;
  - close loop performance check;
- Performance assessment during test.

Figure **??** shows that the problem occurs when the close loop is applied.

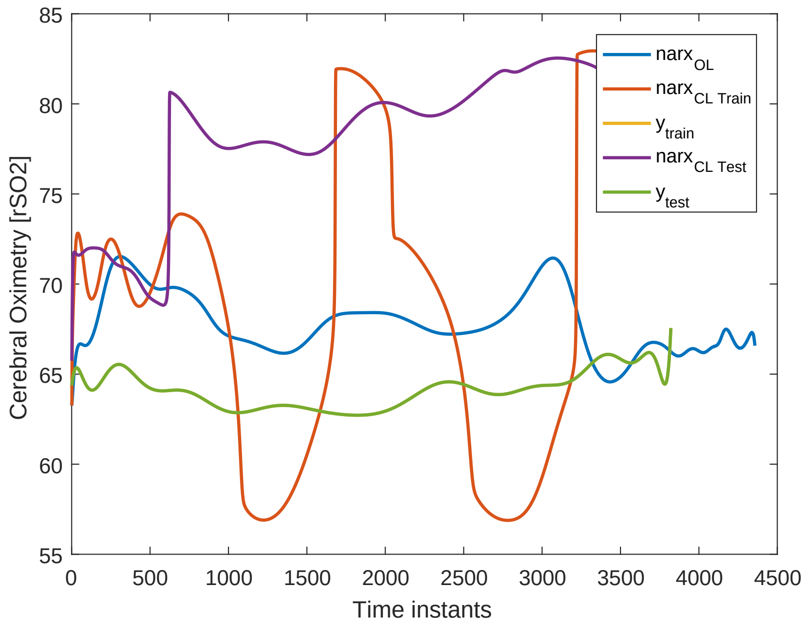


Figure 6: NARX response – detailed analysis

Recalling the discussion from section 5.2, although this kind of problem is not rare in this type of networks, there is not a standard solution for it. Hence, two methods were derived as an attempt to solve the problem:

- Training twice the network: once in open loop and another after closing the loop;
- Preprocessing the inputs, delaying each variable before entering the network.

Although the first approach was able to decrease the errors incurred by tenfold, it was still not enough to solve the problem. There will be therefore no further mention of it.

The second approach was based on the fact that the proposed networks implied the use of explicit time-delays, which in itself presupposes that variables have some relationship between them and themselves for present and past time instants. Noticing this is of major importance since it is possible to devise a method that enables one to both determine the time-delays values and drastically reduce the computation time just by pre-delaying the variables before concatenating them into the input matrix.

The determination of the time-delays (**td**) to be used was performed by redoing the correlation analysis of the previous chapter. Table 5 presents the maximum correlation between input-output variables and the corresponding **td** at they occur.

Table 5: Maximum correlation absolute values and respective time-delays

	Kendall		**Pearson**		**Spearman**	
	corr	**td**	**corr**	**td**	**corr**	**td**
Alt	0,1895	4276	0,2790	4359	0,2897	4217
SpO2	0,3774	2831	0,4220	2797	0,4476	2831
HR	0,2883	1774	0,2975	1780	0,3705	1780

This method gives equivalent results to the cross-correlation analysis usually performed in system's identification.

It is important to mention that although this method in fact solved the close loop problem of the NARX neural network (see figure **??**), as well as making computations of thousands of time-delays feasible, it resulted in over-fitting. This problem will be addressed in the following chapter.

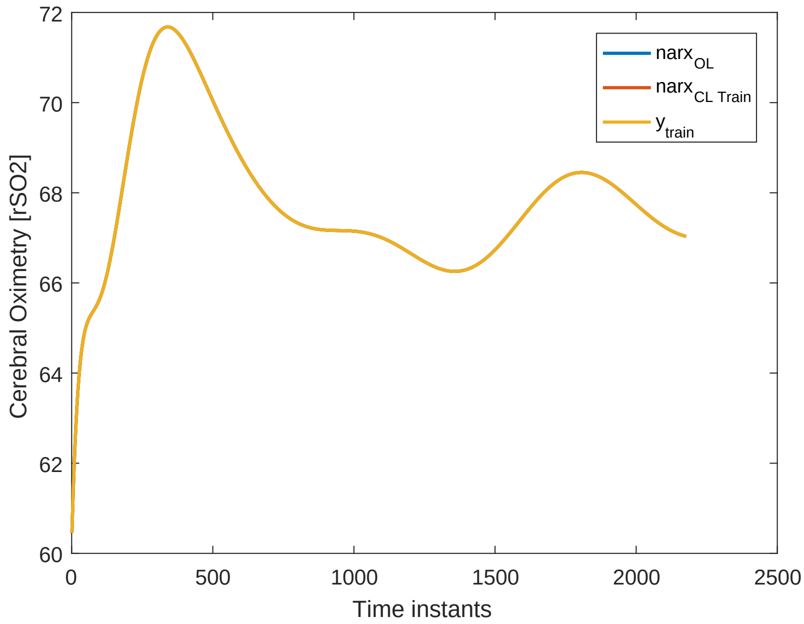


Figure 7: NARX train response

6 Results

6.1 Performance Index

In order to be able to compare the performance of different networks it was important to choose a suitable index.

It is true that mean squared error (mse) is a popular choice in regression problems. However, it fails to characterize the behaviour of the system response.

$$mse = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

On the other hand, variance accounted for (VAF) [3] is very susceptible to variance, resulting in a poor performance assessment for cases in which the model output has a higher variance than the process it is trying to model.

$$VAF = 100 \left[1 - \frac{cov(y - \hat{y})}{cov(y)} \right]$$

Hence, it was imperative to develop a custom performance index that was able to cope with the aforementioned problems.

The solution was to develop an error index composed of two parts: average error percentage (*aep*), and maximum error (*me*). The tuplet is defined as follows:

$$AME = (aep, me)$$

$$aep = \frac{100}{m \times me} \sum_{i=1}^{m} (y_i - \hat{y}_i)$$

$$me = max(|y_i - \hat{y}_i|)$$

where AME stands for average and maximum error.

Taking into account the previous definition, it is clear that a good performance depends heavily on the *me* incurred, which should be the first part to consider during performance assessment, while *aep* gives information about the overall error of the model's response.

6.2 DDNN

In order to choose the adequate number of neurons to be used, it was used the same methodology described in section 5.3.2, making use of the developed performance index.

Table 6 and figure **??** present the various results for different numbers of neurons and different values for the delay located after the hidden layer, **d2**.

Table 6: AME for different numbers of neurons – DDNN

	d2 = 0:1		**d2 = 1:2**	
Number of Neurons	**error [%]**	**max error**	**error [%]**	**max error**
1	62,49776	6,328088	62,40458	6,308045
2	32,69868	287,8368	16,35863	67,03509
3	34,79809	170,1574	5,973575	176,0658
4	48,80156	30,4155	21,81901	39,32132
5	53,01024	21,82395	37,8715	38,65638
6	28,13215	32,6676	29,59692	31,89933
7	38,08438	35,57443	33,90857	25,74781
8	41,44748	21,89998	40,04338	14,16025
9	44,62011	10,07174	47,59059	12,27194
10	54,16081	19,86164	45,59098	14,93374
11	41,81556	21,75418	46,45516	13,76487
12	46,99439	13,66663	33,02526	16,54577
13	43,36947	16,70053	35,89004	28,82357

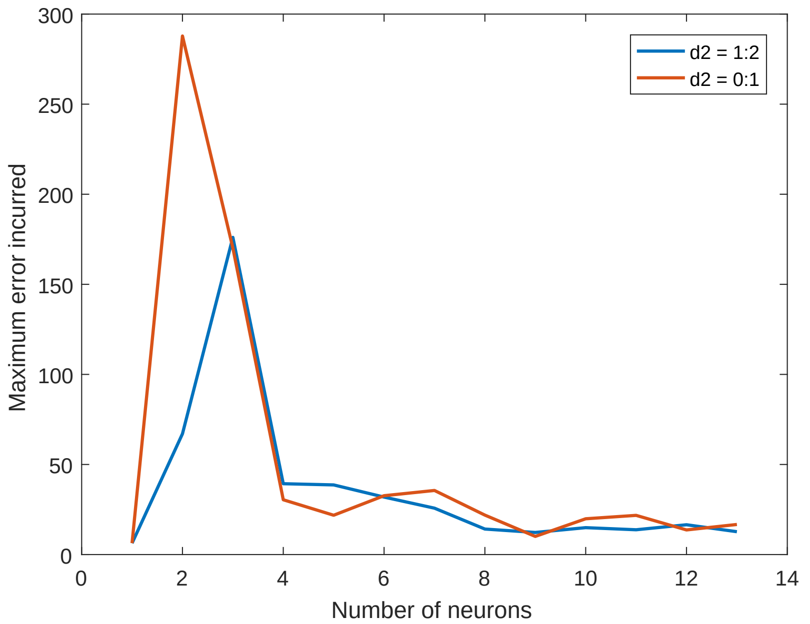


Figure 8: AME for different numbers of neurons – DDNN

As can be seen, the number of neurons that minimizes both the average and maximum errors incurred is 9.

Figure **??** and table 7 show the performance of the developed model, where can be seen that it has an accuracy of 67%, approximately, with a registered maximum error inferior to 8.

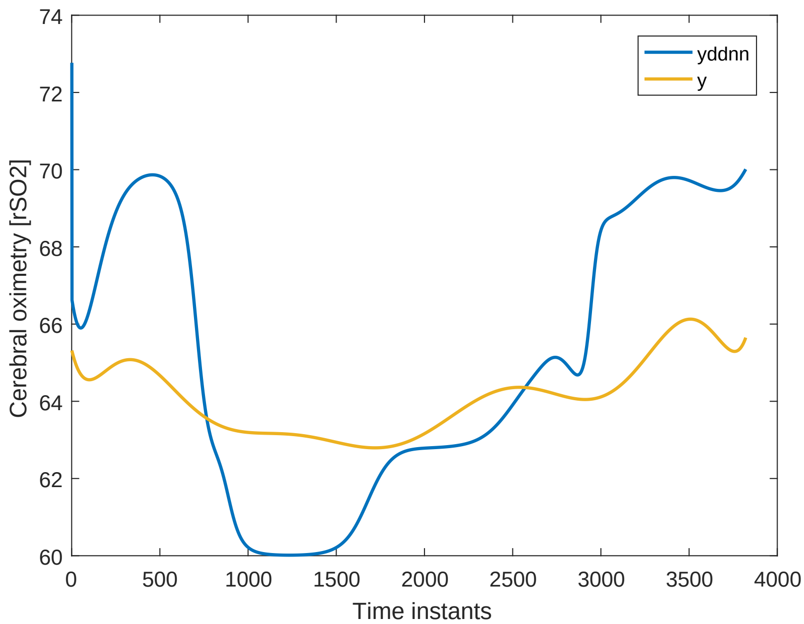


Figure 9: DDNN Predictive Model

Table 7: AME for DDNN Predictive Model

error [%]	**max error**
33,2528	7,4530

6.3 NARX

Repeating the process for the NARX network yields the following results:

Table 8: AME for different numbers of neurons – NARX

Number of Neurons	Train - OL error [%]	Train - OL max error	Train - CL error [%]	Train - CL max error	Test - CL error [%]	Test - CL max error
1	18,29797	0,041918	48,6018	4,792709	85,13294	7,568471
2	10,50716	0,04415	35,02798	7,116281	80,27564	63,25574
3	12,01514	0,017383	30,68822	4,90599	45,39421	12,01329
4	15,99715	0,004547	30,51203	3,343641	67,09306	15,00869
5	12,53198	0,001797	27,98126	0,013341	63,03437	10,48885
6	14,94015	0,000605	18,4988	3,556515	55,10212	17,09496
7	11,70091	0,000367	27,59707	3,953492	63,07873	15,14618
8	10,5104	0,000238	26,0354	2,272281	51,63548	13,39542
9	14,22263	0,000176	22,81024	6,774208	59,61047	10,11153
10	11,3166	0,000435	16,04734	10,79611	56,22948	9,174857
11	9,423539	0,000267	26,23092	5,115362	63,98118	13,32476
12	10,70215	0,000207	24,60525	9,02562	44,81085	11,75009
13	11,41234	0,000176	10,25006	1,304201	47,25039	7,85678

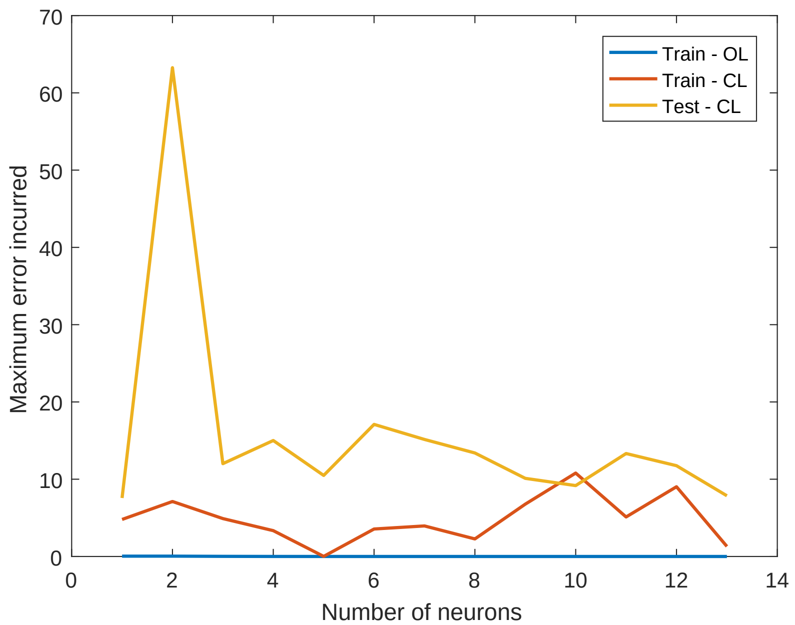


Figure 10: AME for different numbers of neurons – NARX

Given the previous information, the number of neurons that minimizes both the maximum and average errors incurred during test is 13.

Figure **??** and table 9 show the performance of the developed model.

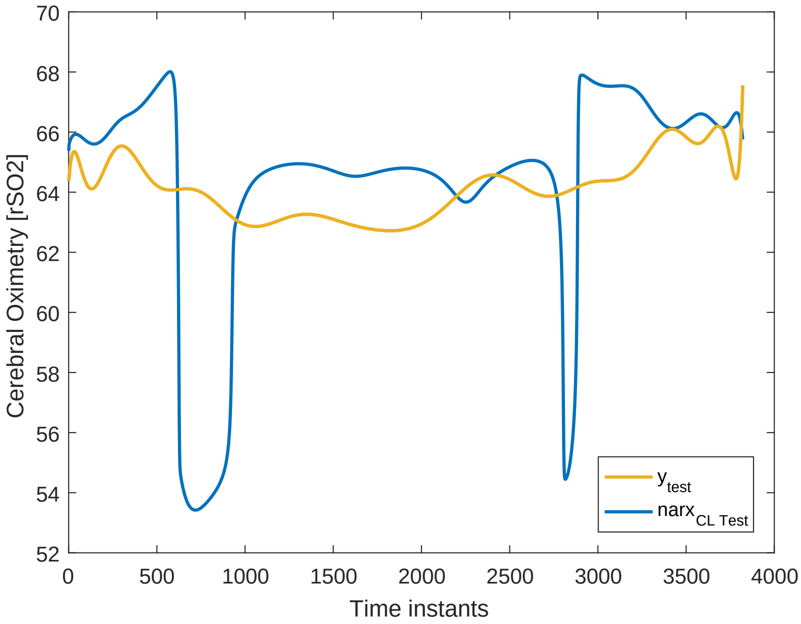


Figure 11: NARX Predictive Model

Table 9: AME for NARX Predictive Model

Train – OL		Train – CL		Test – CL	
error [%]	**max error**	**error [%]**	**max erro**	**error [%]**	**max error**
10,3165	0,0002	5,6216	4,9631	21,1620	10,6613

As can be seen, the developed model presents a 79% accuracy, approximately, with a registered maximum error inferior to 11.

7 Conclusion

The fundamental goal was to develop a predictive model able to alert the pilot in case of hypoxia. However the available data was relative only to normal situations, i.e. situations in which the hypoxia phenomenon did not occur. Hence, in this circumstances the approach devised was to turn this into a systems identification problem, where it was necessary to estimate the variable of interest based on the remaining ones.

For this, two types neural networks were tested – distributed delay (DDNN) and nonlinear autoregressive network with exogenous inputs (NARX).

Table 10 presents a summary of the results.

Table 10: Results Summary

	error [%]	**max error**	**Performance [%]**
DDNN	33,2528	7,4530	**66,7472**
NARX	21,1620	10,6613	**78,8380**

Although reasonable results were obtained using the NARX network it can not be concluded the problem was solved. Rather, the results only suggest this methodology is suitable to achieve that goal

8 Future Work

It would be interesting to repeat the same analysis using a fuzzy modelling analysis. Not only to have a means of comparison, but also in order to gain a deeper insight into the process being studied. Since fuzzy modelling has the advantage of interpretability, it would be the most suitable framework to approach the causality problem.

An independent analysis using system identification tools would also be of some advantage.

Focusing on the NARX neural network approach herein proposed, it would be of fundamental importance to employ an optimum control analysis on the system (i.e. the NARX model), in order to gain some insight regarding its stability.

It would be also interesting to extend the neural network analysis and employ long short-term memory (LSTM) units.

Since the data available to derive the model upon was based on normal situations, where the hypoxia phenomenon did not occur, one of two things could be done:

- gather data representative of the hypoxia phenomenon;
- approach the problem in an inverse way, turning it into a classification problem.

The first suggestion is as easy as subjecting the pilot to test in hypobaric chamber, forcing him to experience low partial oxygen levels.

The second, would require to perform a completely different analysis from what has been discussed here, where the goal would rather be to determine in which of the 3 stages of flight the aircraft is, based on psycho-physiological data.

References

[1] Simon O. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 1999

[2] Andries P. Engelbrecht *Computational Intelligence: An Introduction*. John Wiley, Chichester, 2002

[3] Robert Babuska, *Fuzzy Modeling for Control*. 1998 Kluwer Academic Publishers

[4] Neural Network Toolbox,
`https://www.mathworks.com/products/neural-network.html`

[5] Cismondi, F., Fialho, A., Vieira, S., Reti, S., Sousa, J. and Finkelstein, S. (2013). *Missing data in medical databases: Impute, delete or classify?*. Artificial Intelligence in Medicine, 58(1), pp.63-72.

[6] Silva, Jorge; Zorro, Sara; Patrão, Luís; Moutinho, Alexandra; Vieira, Susana; Fonseca, Agostinho; Gabriel, Joaquim; Amorim, Pedro (2016). *Accquisition and processing of critical psychophysiological and flight parameters in light-sport aviation*. CEM 2016 – Mechanical Engineering Conference

[7] Yoneda I., Tomoda M., Tokumaru O., Sato T., Watanabe Y. (2000). *Time of useful consciousness determination in aircrew members with reference to prior altitude chamber experience and age.*. Aviat Space Environ Med. 2000 Jan;71(1):72-6.

[8] Carla A. Hackworth, Linda M. Peterson, Dan G. Jack, Clara A. Williams, Blake E. Hodges. *Examining Hypoxia: A Survey of Pilots' Experiences and Perspectives on Altitude Training*. Federal Aviation Administration 2003.

[9] da Rocha, L. (2017). *Desempenho de Pilotos e Segurança de Voo O Caso da Hipoxia em Aviação Desportiva*. Master's Degree. UNIVERSIDADE DA BEIRA INTERIOR.

[10] Filippo Maria Bianchi, Enrico Maiorino, Michael C. Kampffmeyer, Antonello Rizzi, Robert Jenssen. *An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting*. arXiv:1705.04378v1 [cs.NE] 11 May 2017.