FAULT TREE EVENT CLASSIFICATION BY NEURAL NETWORK ANALYSIS

Mihai Alexandru BARBELIAN [1], Casandra Venera BĂLAN (PIETREANU)[2]

Due to the growing trend of parallel processing power, it is desirable that the existing modeling systems can be linked with compatible architectures. In addition to increased processing capacity, these systems show flexibility, adaptability and are able to predict events through a learning process. The application can be run on parallel processors.

The Fault tree provides a linear, rigid analysis. In addition, Neural networks could estimate nonlinear influence and deal with more events. Therefore, the analysis of a Fault Tree by Neural networks can give results such as the setting of the main event and its associated risk, and also the total probability of risk.

Keywords: Fault tree analysis, neural network, probabilistic risk assessment.

1. Introduction

Modern technology interaction with natural phenomena influence the factors and degree of risk, whose awareness, understanding and interpretation are essential for the decisions to be taken and in order to handle risk and restore the previous situation.

Loads exerted on the structure of an aircraft can be treated and modeled as random variables in a probabilistic risk analysis, but the limitations of this analysis aims the difficulty with which one can make a prediction of the exact timing of malfunctions. Therefore, the importance of mathematical models is essential, as these are computational methods that allow analysis of the operating conditions, behavior/ system evolution, in order to estimate with a higher degree of accuracy and make precise forecasts for the occurrence of defects.

The failure probability for non-repairable/repairable components, and the constant failure probability can be calculated as follows:

$$P = \begin{cases} 1 - e^{-\lambda T} \\ \lambda\tau/(1 + \lambda\tau) \\ c \end{cases} \tag{1}$$

[1] Lecturer, Dept. of Aerospace Engineering, University POLITEHNICA of Bucharest, Romania, e-mail: barbelian_m@avianet.ro

[2] Teaching Assistant, Dept. of Aerospace Engineering, University POLITEHNICA of Bucharest, Romania, e-mail: casandra.pietreanu@yahoo.com

λ = component failure rate
T = exposure time
τ = repair time
c = constant probability

Ratings projections for different levels of the system, allow a qualitative analysis on the functioning as early as the design stage, highlighting possible weaknesses and the likelihood of malfunctions, thus improving the level of safety. In this regard, in order to maintain operational safety, the calculations allow for a possible reconfiguration and improvement of the overall technical parameters; one of the methods chosen for such a forecasting reliability analysis is redundancy. A method adapted for assessing and quantifying the probability of failure for the prediction reliability is the FTA (fault tree analysis), which brings out what lead to the failed state [1].

2. Fault Tree Analysis

An event can be represented by the occurrence of a defect after a certain time of operation. The attached probability in aviation is an indicator of the accident/ incident accounting the realization of a number of conditions under a certain criterion and it involves assigning a value that indicates the possibility of achievement to each event.

As known, the probability is a number in the [0,1] range, therefore it cannot be less than 0 (0%) or greater than 1 (100%).

$$\begin{gathered} 0 \leq P(E) \leq 1 \\ P(Impossible_{event}) = P\ (\Phi) = 0 \\ P(Certain_{event}) = P\ (\mathrm{E}) = 1 \end{gathered} \tag{2}$$

The Fault Tree is a deductive analysis which determines the failure processes that lead to the undesired event [7]. Built through a reverse logic, primary faults and then intermediates events are reconstructed in a backward way, in order to outline the scenario and study the events that preceded the emergence of the Top Event.

The decomposition of the main cause of an accident is made from the top, through the tree branches, identified as intermediate events, and by the lowest (basic) events described by material failures, human or environmental factors, etc. [6].

Based on error modeling techniques regarding certain specific system components, and combining them, the fault tree is set up as a puzzle, whose constituent parts are located in primary and intermediate classes of events that shape the built ramification.

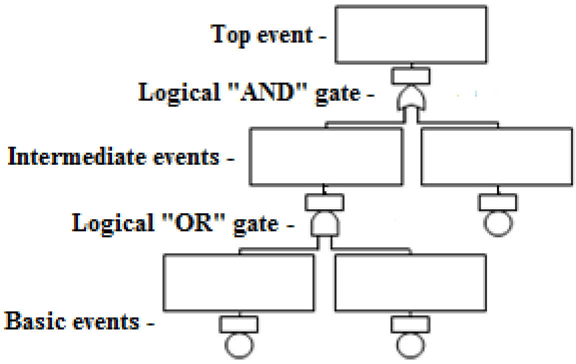

Fig. 1. The structure of a Fault Tree

Constructing a FT is an iterative process in which, after establishing the correct top event, every iteration generates a necessary and sufficient cause, respecting the limitations and taking into account the existing data through an amply analysis.

The minimum cut sets establish types of independent items that produce immediate faults or even the crash of the entire system [5]. This optimized structure presumes that a basic event is not taken into account several times, so this is not a redundant analysis. The minimum cut sets generates reliable results, they are an indicator of the vulnerability of a system; a vulnerable system will require small MCS, or even a few of them.

Using the notations:

TE = Top event,

MCS = Minimum cut set,

E = Basic event,

the probability of Top event and minimum cut sets will be calculated as follows:

$$P(TE) = P\,(MCS_1 + MCS_2 + \dots + MCS_n)$$
$$P(TE) = \Sigma\, P(MCS_k) \tag{3}$$
$$P(MCS) = P(E_1)P(E_2)\dots P(E_{SMD})$$

The probability of the Top event is calculated as the sum of the probabilities of minimum cut sets, which is the product of basic (primary) events probability.

As mentioned, the qualitative analysis of the FT is conducted using data from the minimum cut set and the probability is given by the product probability of its basic events.

The fault tree in the example below refers to the inability to control an UAV, due to propulsion, aeroelastic and environmental factors.

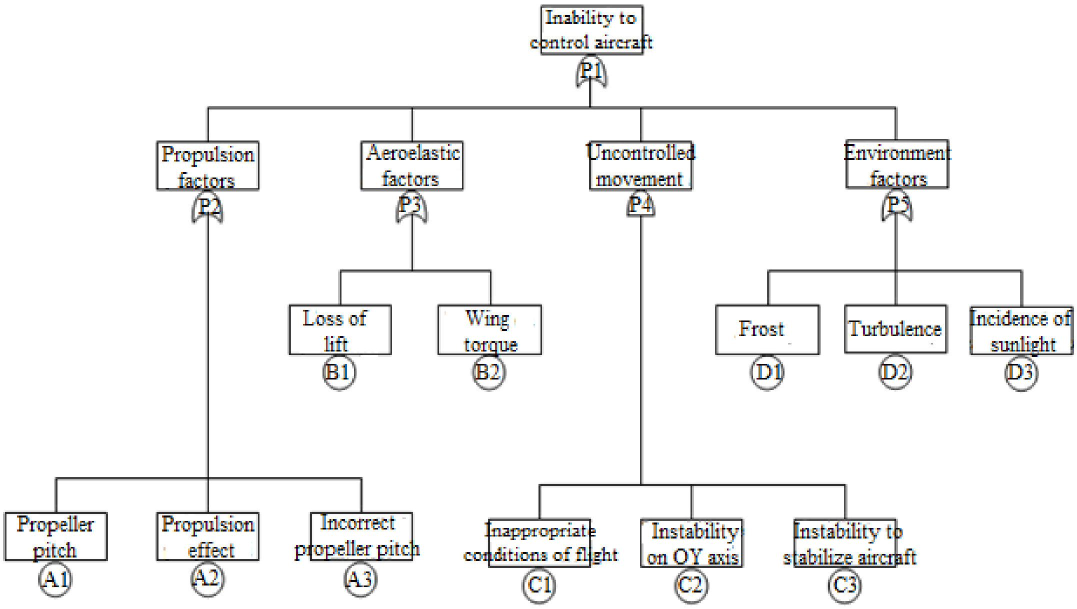

Fig. 2. Example of a Fault Tree Analysis for a UAV

If we consider the next probabilities for the basic events of the fault tree developed above:

$P(A1) = 1 \cdot 10^{-5}$; $P(A2) = 2 \cdot 10^{-4}$; $P(A3) = 2 \cdot 10^{-5}$
$P(B1) = 1 \cdot 10^{-3}$; $P(B2) = 3 \cdot 10^{-4}$;
$P(C1) = 1 \cdot 10^{-2}$; $P(C2) = 2 \cdot 10^{-3}$; $P(C3) = 1 \cdot 10^{-4}$
$P(D1) = 2 \cdot 10^{-3}$; $P(D2) = 2 \cdot 10^{-4}$; $P(D3) = 3 \cdot 10^{-5}$

Knowing the relations stated next, we can calculate the probabilities of the gates and finally of the top event.

Table 1

The calculus of probabilities for "OR" and "AND" gates

Type of Gate	Number of inputs	Probability
OR	2	$P(A) + P(B) - P(A)P(B)$
OR	3	$\big(P(A) + P(B) + P(C)\big) - \big(P(AB) + P(AC) + P(BC)\big) + P(ABC)$
OR	4	$\big(P(A) + P(B) + P(C) + P(D)\big) - \big(P(AB) + P(AC) + P(AD) + P(BC) + P(BD) + P(CD)\big) + \big(P(ABC) + P(ABD) + P(BCD) + P(ACD)\big) - P(ABCF)$
AND	2	$P(A) \bullet P(B) = P(A)P(B)$
AND	3	$P(A) \bullet P(B) \bullet P(C) = P(A)P(B)P(C)$

***a*.**) $P(P2) = \big(P(A1) + P(A2) + P(A3)\big) - \big(P(A1A2) + P(A1A3) + P(A2A3)\big) + P(A1A2A3) = 2.2999380004 \cdot 10^{-4}$

***b*.**) $P(P3) = P(B1) + P(B2) - P(B1)P(B2) = 1.2997 \cdot 10^{-3}$

***c*.**) $P(P4) = P(C1)P(C2)P(C3) = 2 \cdot 10^{-9}$

***d*.**) $P(P5) = \big(P(D1) + P(D2) + P(D3)\big) - \big(P(D1D2) + P(D1D3) + P(D2D3)\big) + P(D1D2D3) = 2.229534012 \cdot 10^{-3}$

***e*.**) $P(P1) = \big(P(P2) + P(P3) + P(P4) + P(P5)\big) - \big(P(P2P3) + P(P2P4) + P(P2P5) + P(P3P4) + P(P3P5) + P(P4P5)\big) + \big(P(P2P3P4) + P(P2P3P5) + P(P3P4P5) + P(P2P4P5)\big) - P(P2P3P4P5) = 0.7555210437000111845087275313 1 \cdot 10^{-3}$

So, the calculated probability of the Top event is:

$$P(P1) = 3.75552104370001118450872753131 \cdot 10^{-3} \cong 3.8 \cdot 10^{-3}$$

3. Fault Tree Neural Network Analysis

The feed forward propagation neural networks allow the emulation of an input-output behavior through neurons activation internal functions. This behavior can be controlled through drive algorithms, the result being according to the chosen algorithm, an exact input-output correlation or just at a stochastic level by minimizing a cost function given by the square error between input and output [8].

Since feed forward propagation neural networks have a great opening to the integration with programming environments with existing parallel processing [9], the fault tree will be modeled with this type of network.

A fault tree modeling is necessary in order to obtain the required training data sets.

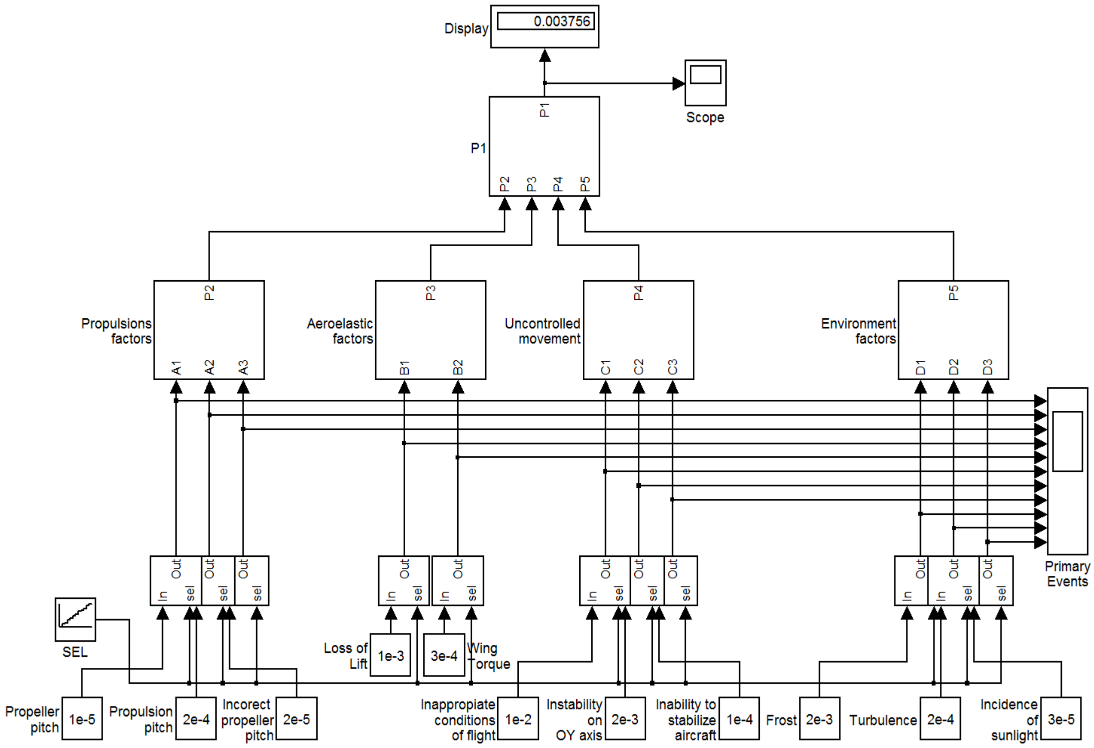

Fig. 3. The simulated model of the Fault Tree

The feed forward neural network used for fault tree modeling is defined by one input linear layer with as many neurons as primary events, one hidden tangent-sigmoidal layer with a number of neurons adapted to the best learning performance and one output layer for events and classes identification. The fault tree used for neural network training has a perturbation selector to test the network accuracy and to identify the perturbed probability for primary event and its class.

The perturbed inputs for each event is as follows:

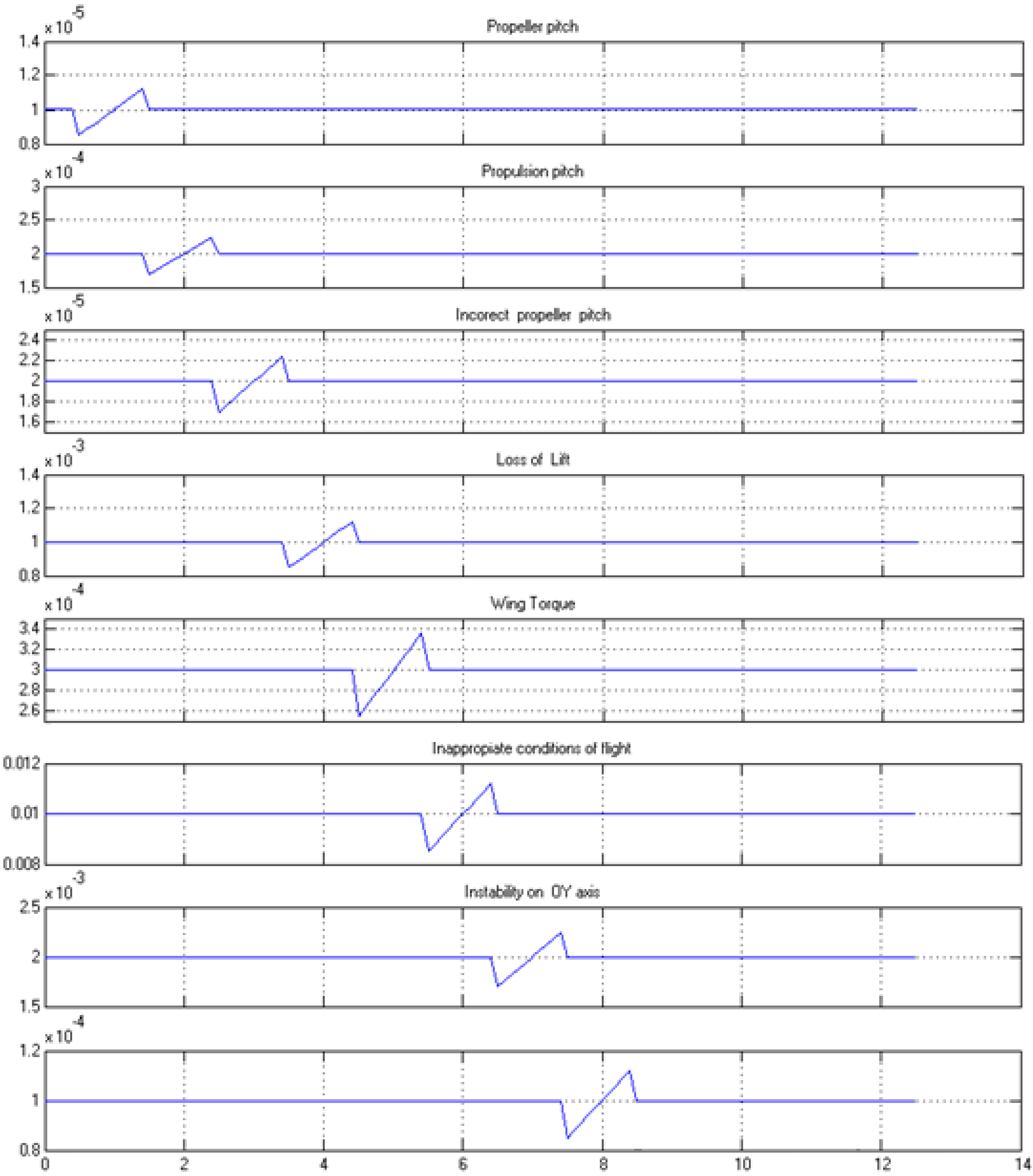

Fig. 4. The perturbed inputs for each event

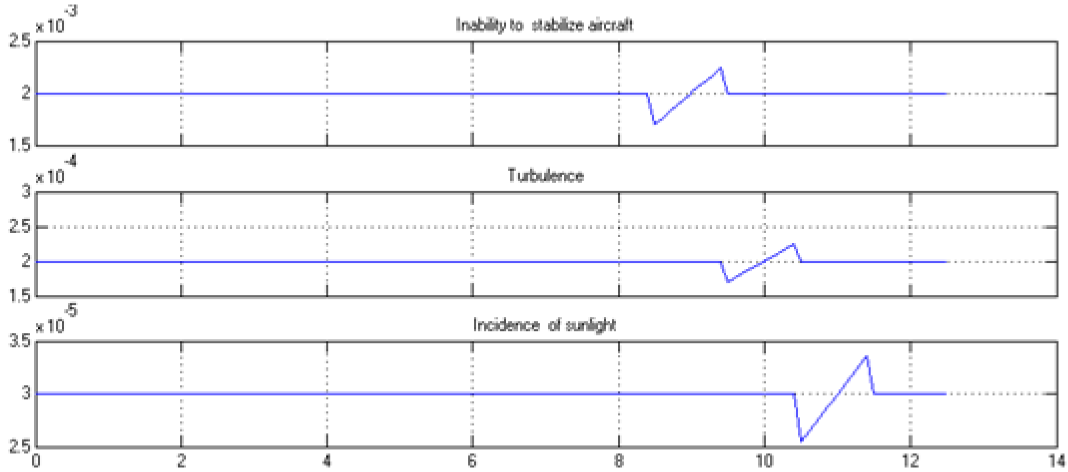

Fig. 5. The perturbed inputs for each event

The training set used for neural network is generated from the deviation in the perturbed states of primary events probability and the answer set of neural network is defined accordingly to the number of events and classes. The answer of the network is considered valid for values greater than 0.7.

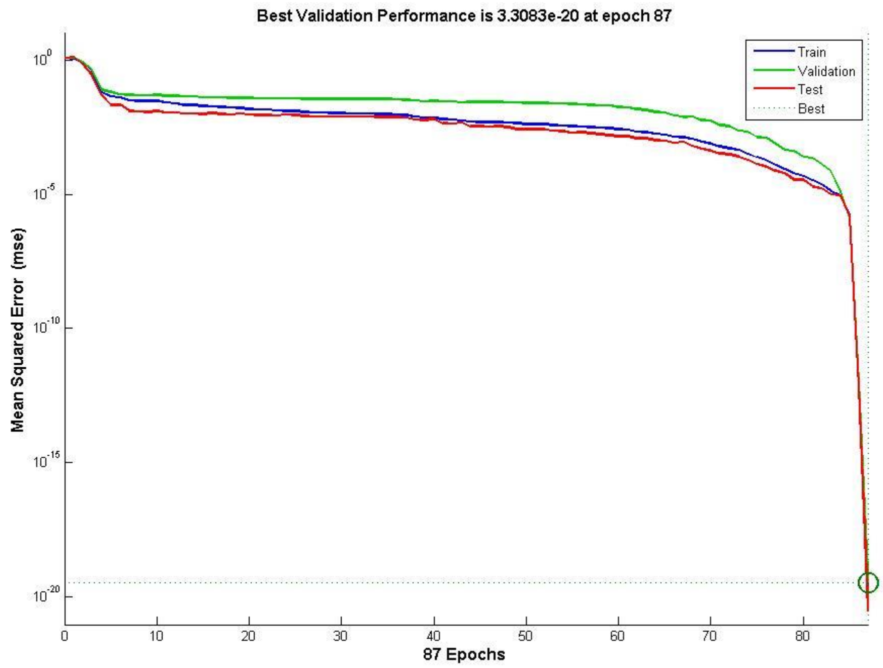

Fig. 6. Validation performance for 87 training epochs

The best performance of neural network is achieved with a minimal network architecture size by a triple number of neurons in the hidden layer relative to the number of events. Best validation performance value is $3.3e^{-20}$ and is obtained after 87 epochs.

The training algorithm used for training is Levenberg-Marquadt and the performance goal is defined by the mean square error.

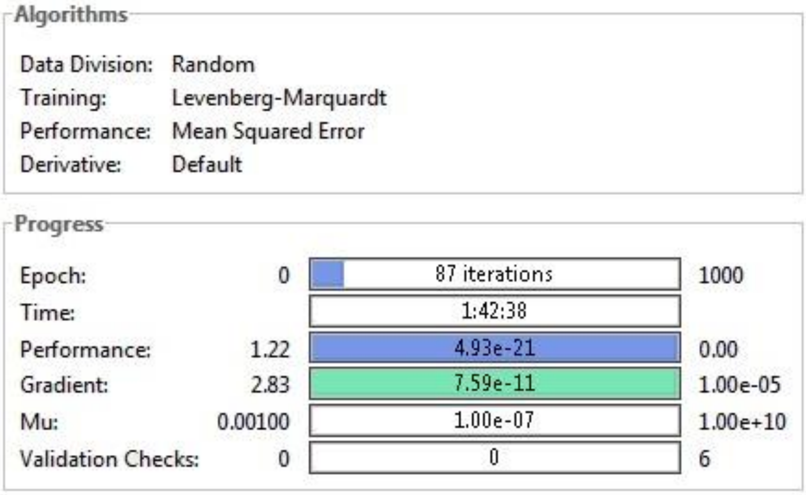

Fig. 7. Training parameters for Levenberg-Marquadt algorithm

The answer of the network is shown in the following two figures.

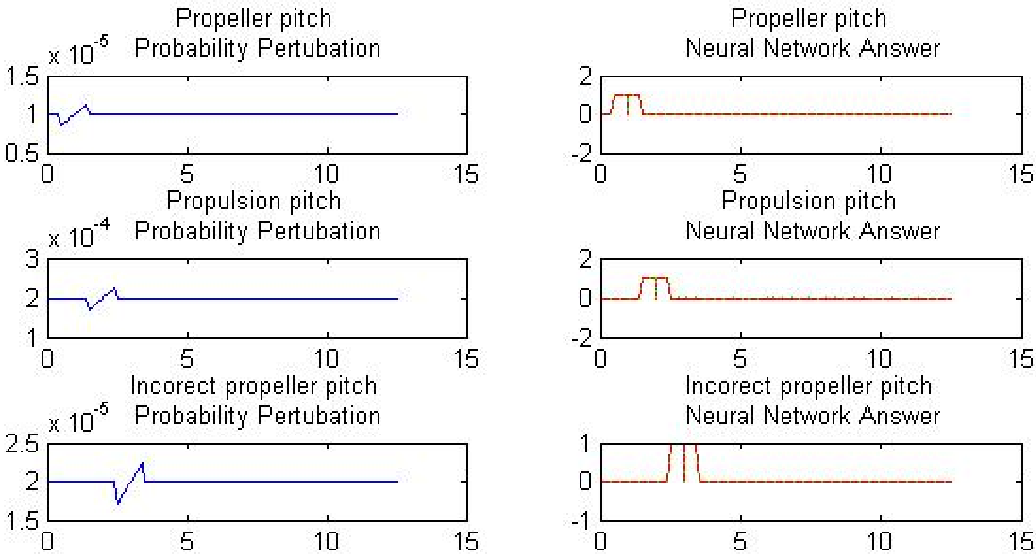

Fig. 8. Perturbations on probability and the desired neural network answer (continuous-red) and actual (dotted – green)

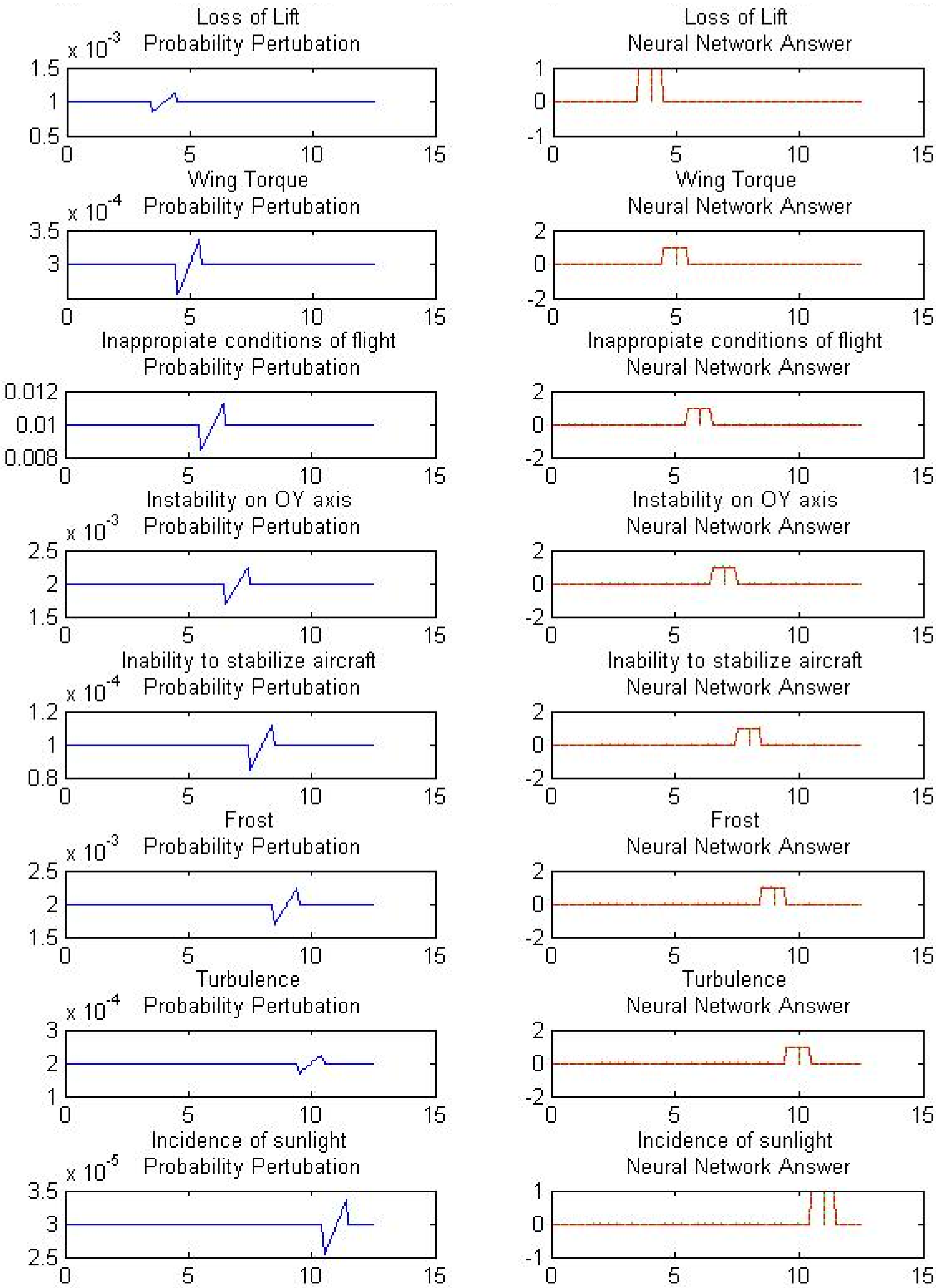

Fig. 9. Perturbations on probability and the desired neural network answer (continuous-red) and actual (dotted – green)

4. Conclusions

From the depicted answer in the previous graphs of the neural network, the results achieved are considered accurate, this assuring a good detection for values that are not in the training set. Also, for an imposed limit on the probability, one can detect if we have an increase or an inaccurate value of the predicted state. This is useful for a better analysis the of data, in order to take into consideration only accurate values, and to provide a platform for parallel processing in case of real time determination of predicted probability of failure.

R E F E R E N C E S

[1]. *C. Bălan (Pietreanu)*, Boolean Algebra Application in Analysis of Flight Accidents, INCAS BULLETIN, Volume 7, Issue 4/ 2015.

[2]. *M. Stamatelatos*, Fault Tree Handbook with Aerospace Applications, Prepared for NASA Office of Safety and Mission Assurance, NASA Headquarters, Washington, DC 20546, August, 2002.

[3]. *W.E. Vesely, F.F. Goldberg, N.H. Roberts and D.F. Haasl*, Fault Tree Handbook, U.S. Nuclear regulatory Commission, Systems and Reliability Research, Office of Nuclear Regulatory Research, NUREG-0492, Washington, D.C. 20555, 1981.

[4]. *J. Marshall*, An Introduction to Fault Tree Analysis (FTA), The University of Warwick, Peuss 2011/2012.

[5]. *D. Kececioglu*, Reliability Engineering Handbook, Volume 2, Prentice Hall, Inc., New Jersey, 1991.

[6]. *David Blockely*, Engineering safety, McGRAW-HILL Book Company Europe, 1992.

[7]. *Bill Vesely*, Fault Tree Analysis (FTA): Concepts and Applications, NASA HQ.

[8]. *M.A. Barbelian*, Studiu asupra unui sistem integrat de management al funcționării aparaturii de bord, Phd. Thesis, 2015.

[9]. *Amir Abolfazl Suratgar*, *Mohammad Bagher Tavakoli*, *and Abbas Hoseinabadi*, Modified Levenberg-Marquardt Method for Neural Networks Training, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:6, 2007.

[10]. *Udo Seiffert*, Artificial neural networks on massively parallel computer hardware, Neurocomputing 57 (2004): 135-150

[11]. *A. Doshi*, Aircraft Position Prediction Using Neural Networks, B. Sc. Th., Massachusetts Institute of Technology, 2005.

[12]. *C. Paleologu, R. Mihnea, A. Udrea, A. Enescu,* Prelucrarea numerică a semnalelor, Ed. Electronica 2000, 2004.
[13]. *Şt. Gh. Pentiuc,* Aplicații ale recunoaşterii formelor în diagnosticul automat, Ed. Tehnică, 1997.
[14]. *M. B. Wilamowski*, *L. Torvik*, Modification of gradient computation in the back propagation algorithm, Artificial Neural Network in Engineering, Nov. St. Louis, Missouri 1993.