ICAS-94-10.8.2

ARTIFICIAL INTELLIGENCE TOOLS FOR THE MAINTENANCE OF TURBOFAN ENGINES

G. TORELLA *   G. LOMBARDO**

*ITALIAN AIR FORCE ACADEMY
80078 POZZUOLI NAPOLI ITALIA

**DIPARTIMENTO DI MECCANICA ED AERONAUTICA - UNIVERSITA' DI PALERMO
PALERMO ITALIA

Summary

The paper shows the activities carried out for developing Artificial Intelligence tools for the maintenance of turbofan engines. Two different areas of Artificial Intelligence have been considered: Expert Systems and Neural Networks; computer codes have been developed for both subjects.

Some codes for Expert Systems have been set-up by an 'Event Driven' language for exalting the interactive characteristics and for allowing the constant involvement of users. Other programs have been developed by using a declarative language. Moreover particular attention has been paid to the construction of Knowledge Bases.

The Neural Networks have been built by suitable multi language packages. A high speed language has been used for evaluating the matrices of influence and the patterns for training the Networks. Successively the Neural Networks have been set-up by using again an 'Event Driven' language.

The paper shows both the code construction and some examples of their applications.

1.-INTRODUCTION

In the past, for a long time, the Artificial Intelligence has been considered a theoretical subject with scarce interest for practical applications.

In the last years, thanks to the fast development of computers and programming languages, the Artificial Intelligence has become an effective tool for many works.

Gas turbine engine maintenance is one of these activities. It is the most outstanding work carried out during the operating life of engines because the availability and the working security of engine are strictly linked to correct and effective maintenance actions.

Moreover the maintenance must be fast, effective and not expensive so there must be a continuous optimization of the employment of both human and economical resources.

Expert Systems and Neural Networks may be used for developing effective tools for supporting maintenance activity.

This paper deals with the work carried out for developing suitable Artificial Intelligence tools for turbofan engine maintenance.

The utilization of either Expert Systems and Neural Networks has been considered.

The first part of paper concerns with the possibility of setting-up suitable Expert Systems for turbofan trouble-shooting.

The second part deals with the study of items and requirements necessary for developing small Neural Networks for turbofan diagnostics.

Both parts of paper contain the analysis carried out for choosing the most effective language used during the different steps of computer code construction.

2.-THE EXPERT SYSTEMS

The Expert Systems are basically formed by two different and equally important parts: the Knowledge Base and the Inference Engine.

The Knowledge Base is the heart of Expert Systems. The reliability and the accuracy of obtained results are strictly linked to the quality and quantity of data contained in the Knowledge Base.

The information useful for developing suitable Knowledge Bases may be obtained from different sources. The main ones are the Overhaul manuals of the engines where trouble-shooting procedures are shown.

The second important source is the experience heaped-up by the personnel involved in the maintenance activity. This information is very important mainly for taking in account, facing and solving the so called *troubles out of rules*. These faults are very difficult to consider and only human experience may be the successful tool for finding the right solution of problem.

Moreover storing this information in the Knowledge Bases is the only way for preserving important and large amount of data that may be lost when the personnel leave the job.

When the Knowledge Base has been built the activities for developing the Inference Engine may start.

An effective Expert System must direct the action of maintenance. It must isolate and find the probable engine areas, where the fault is active, by using the past experiences. For this aim the Expert System may have a high degree of interactivity and must constantly use the full archive contained in the Knowledge Base. This way the maintenance intervention is fast and effective and time and human resources are saved.

This feature requires the use of suitable and effective languages for developing the computer codes. For this aim the analysis for choosing the most effective ones has been carried out.

2.1-The language selection

For developing Expert Systems codes different types of languages have been considered. In the past the different capabilities of both declarative and procedural languages have been analyzed. The declarative Prolog and Turboprolog (1), (2), are conceptually closer to the inherent nature of Expert Systems. Unfortunately for large Expert Systems the codes are difficult to write, to manage and they are not easily legible by many people (3), (4),(5).

On the other side the procedural languages, based on the 'Object Oriented' and the 'Event Driven' criteria, like C++, Visual C++ and Visual Basic professional, are truly suitable for developing fast, reliable and user-friendly codes. Moreover, even if the Visual C++ is very powerful, the inherent simplicity of Visual Basic professional is particularly useful during the development of Expert Systems with an high degree of interaction and the wide participation of user (6).

In the paper both declarative and procedural languages have been used for the development of Expert Systems.

2.2-Examples of developed Expert Systems

Here some examples of Expert Systems developed for the diagnostics and the trouble-shooting of turbofans with different configurations are presented.

The first Expert System deals with the diagnostics of a two spool mixed flow turbofan engine. This System has been developed in Turboprolog, the declarative language for Personal computer developed by Borland. Two different codes have been set-up. The first one requires only the trend of some engine parameters while the second one requires either the trend or the actual values of some engine parameters. The choice is left to the user will.

Fig. 1 shows the utilization of the first Expert System. As it is shown the computer screen is divided in different zones. One of them contains the input data that must be continuously delivered. In this example the user must give the trend of parameters one by one. When the input finishes, the Expert System consults the Knowledge Base and looks for the same trends or for the set of trends closer to the delivered one. Finally the Expert System shows the cause of fault and the suggested maintenance action, fig. 2. The information contained in the Knowledge Base, used by this Expert System, have been obtained from the maintenance manual of the engine (7).

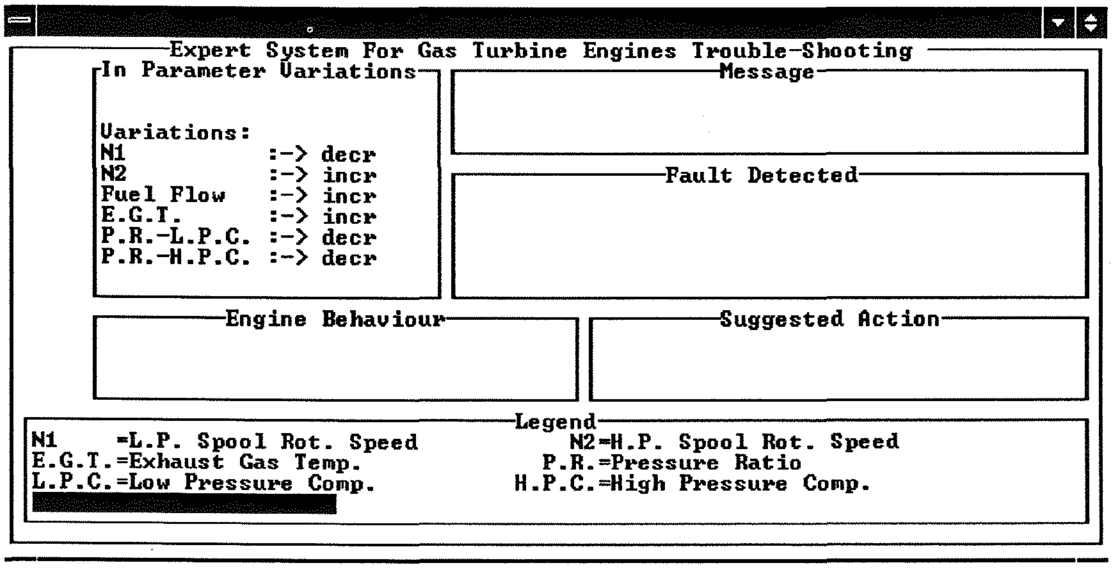

Fig. 1 Turboprolog code for Expert System using trends of engine parameters

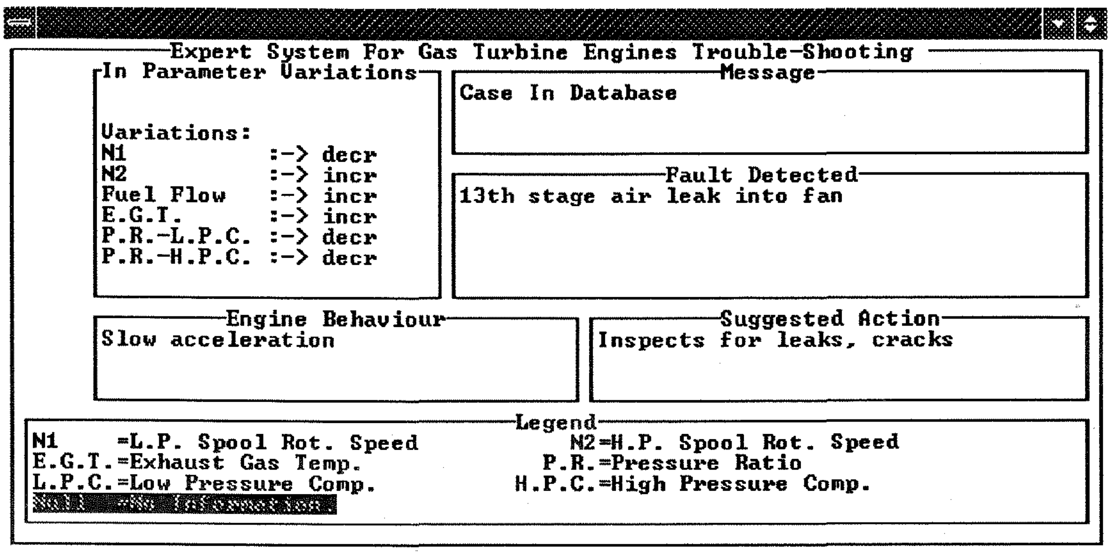

Fig. 2 Turboprolog code for Expert System using trends of engine parameters

The second Expert System allows to use both trends or actual values of some performance and thermodynamic parameters of engine. When the code starts the picture of fig.3 appears on the computer screen. The user must select the type of input he wants to use (window '**Begin**'). This time the option for delivering the actual values of some parameters has been fired (the letter '**V**'). The figure 4 shows the parameters of interest and their values.

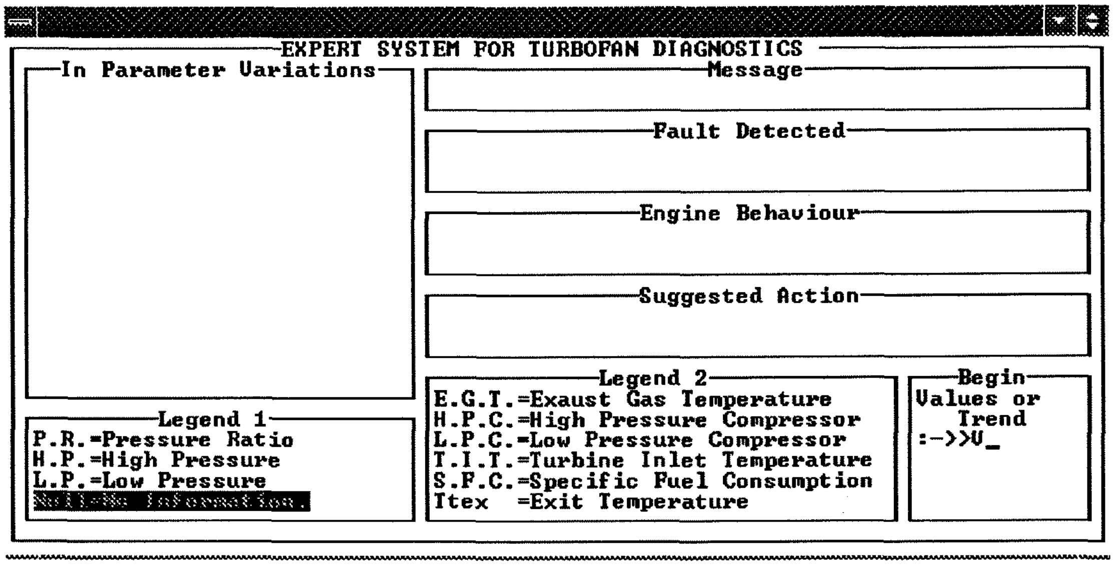

Fig. 3 Selection od input data type for Turboprolog Expert System

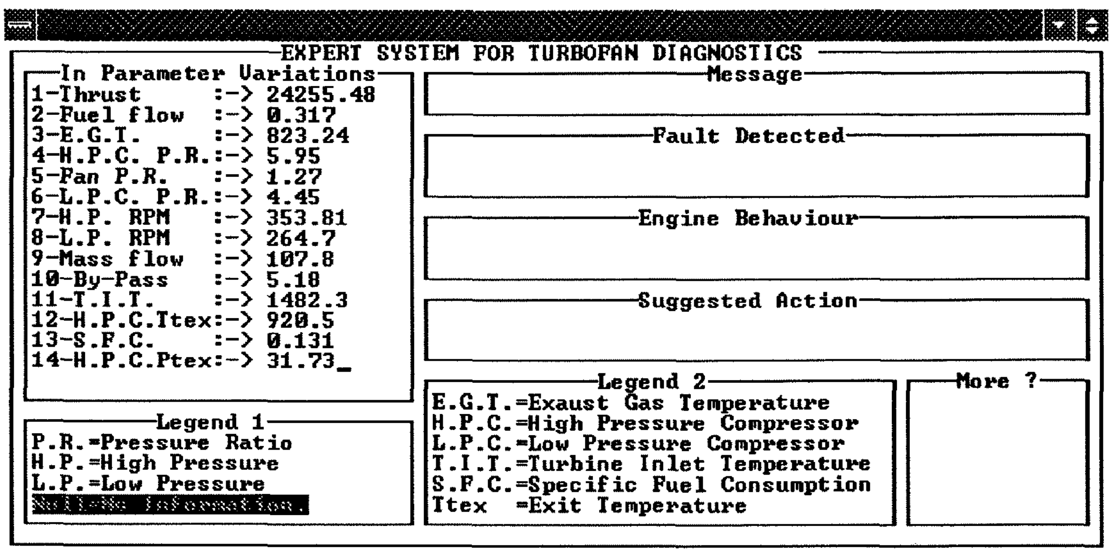

Fig. 4 Input data for the Turboprolog Expert System

When the input data is complete the Expert System searches in the Knowledge Base the situation better fitting the input data and the results are presented on the computer screen, fig. 5. Beside the cause of fault, by suitable option, (window '**Graphics?**'), the Expert System shows the behavior of main engine parameters (both performance and thermodynamic) for the variation of 1% of component decay caused by the detected fault. This time it has caused the decay of low pressure turbine efficiency (LOSS L.P.T. EFFIC.); the variation of 1% of this parameter influences the engine performance as described in the graph of fig. 6.

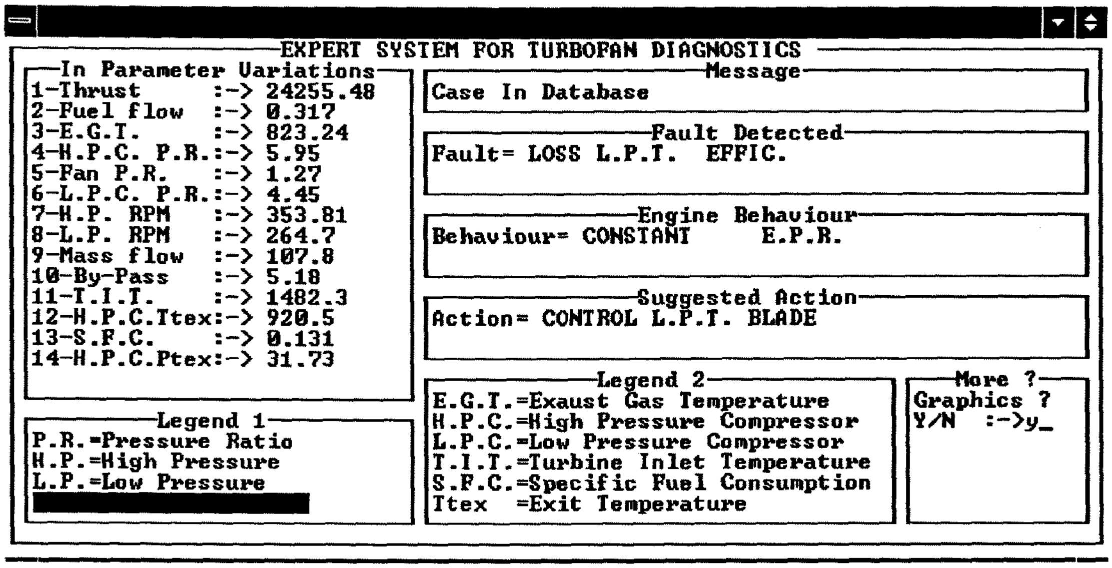

Fig. 5 Results of Turboprolog Expert System application

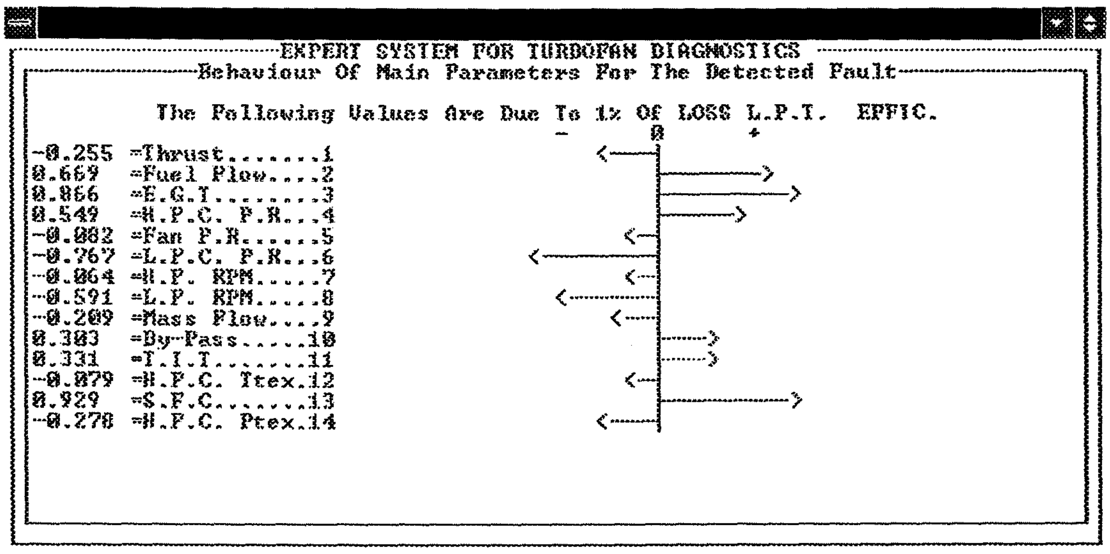

Fig. 6 Behaviour of turbofan due to the variation of decay caused by malfunction

The Knowledge Base used for this Expert System has been constructed by the massive use of simulation codes. This way the engine behavior, when there is a fault in progress causing the decay of the performance of engine component has been studied. The construction of matrices of influence for the Neural Network training will require a similar procedure.

The last type of Expert System has been developed for carrying out the trouble-shooting of a large two spool separated flow turbofan engine. The Knowledge Base has been derived from the complete trouble-shooting manual furnished by the factory.

This information has allowed the construction of the so-called fault tree for the different components and Systems of the engine. The use of this information and the continuous interaction between the user and the System allows to detect the right area of the fault as well as the cause of fault itself and the most suitable action. Moreover, it is important to underline, that a large part of the Knowledge Base is formed by information obtained from past maintenance interventions on the same engine or on other similar engines. This way the past experience is fully used and it helps to address the maintenance investigation.

In order to obtain both Expert System with high interactivity and user-friendly codes, the System have been developed in Visual Basic professional.

Fig.7-14 show an example of the Expert System utilization.

When the System starts, the form of fig.7 appears on the screen. The user must select the activity of interest. It is possible to carry out either trouble-shooting or the management of archive containing the past faults and the maintenance actions suffered by the engine.

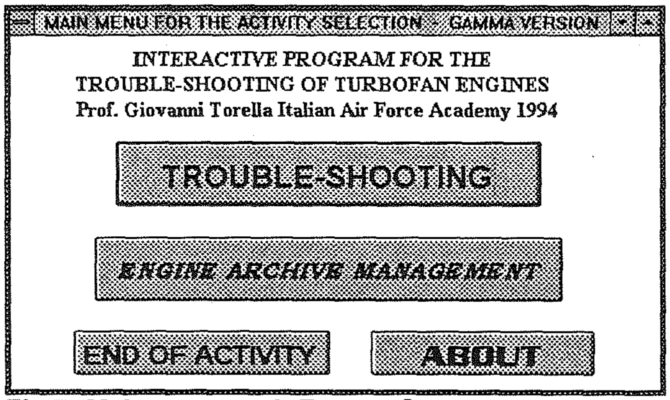

Fig.7 Main menu of Expert System for trouble-shooting

Now the option 'TROUBLE-SHOOTING' has been fired so the form of fig.8 appears on the computer screen.

The information exchange between the user and the System begins because the user must indicate the operating conditions or the events active when the trouble appeared.

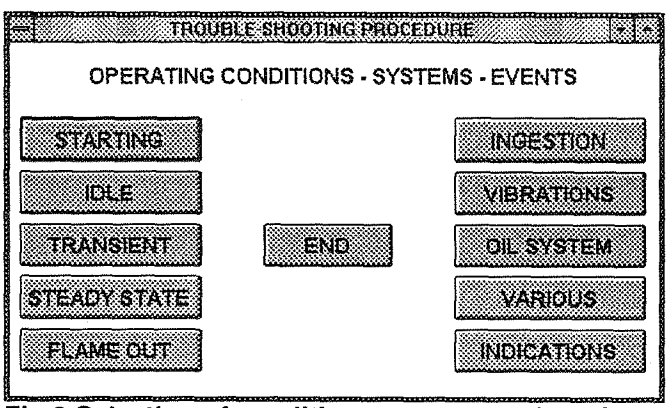

Fig.8 Selection of conditions or system of trouble

This part of paper deals only with the demonstration of Expert System capabilities, so the choice of a particular situation is quite arbitrary. This feature is very interesting because the Expert System may be used also for training purposes.

The user has supposed that the trouble appeared during starting procedure. By firing the option 'STARTING' a menu is displayed on the screen, fig. 9, and the System requires the trouble symptoms and helps the user by delivering a list of probable situations. In our demo the engine has shown "SLOW/DIFFICULT ACCELERATION ABOVE 40% N2". N2 is the rotational speed of high pressure group.

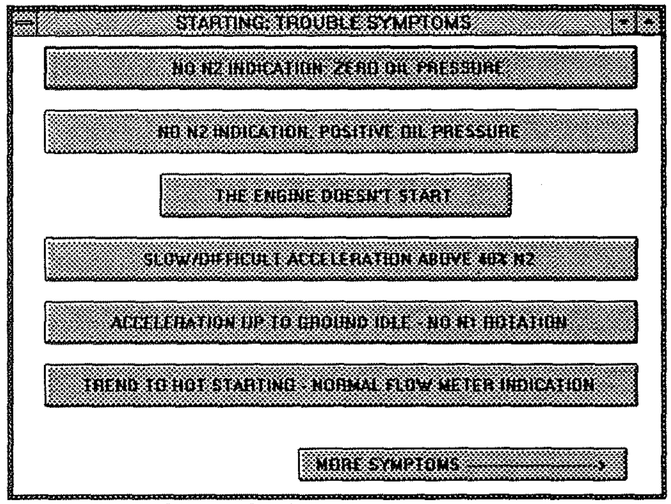

Fig. 9 Selection of fault symptoms

When the information has been furnished the System displays the summary of situation and the options for continuing the work, fig.10 One of the available opportunity is linked to the button 'STATISTICS AND ADVICE'. By firing this button the System consults the Knowledge Base and search for the past events caused by the selected trouble. The result of this research is displayed in the form of fig.11. This way the System may offer the history of the trouble and this information may be useful for addressing the research of causes and for saving time.

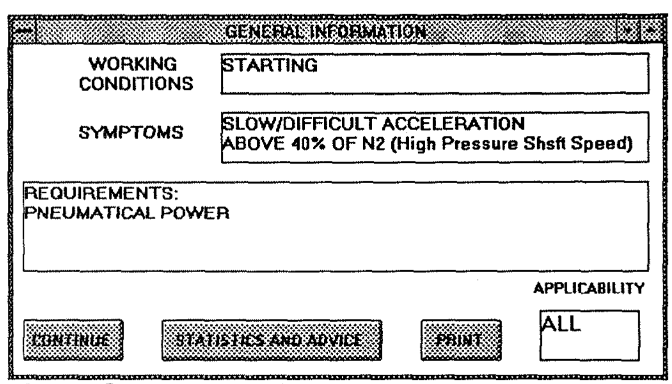

Fig. 10 Summary of situation

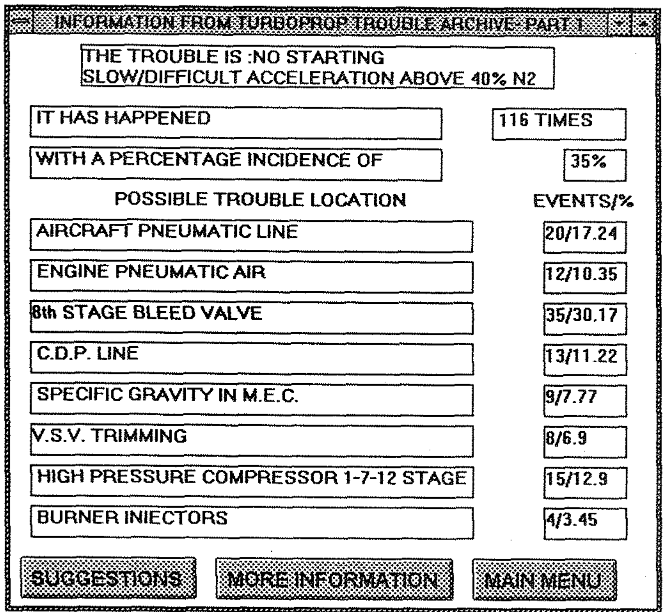

Fig.11 History of the selected fault

By pressing the button 'SUGGESTIONS' of fig.11, the form of fig.12 appears. The System shows the most probable area of trouble on the base of past experience. The user may follow the suggestion and carry out either a simplified trouble-shooting procedure or may apply the full procedure. The simplified procedure considers only the most probable area of malfunction, i.e. the first one shown in the suggestion list of fig.12.

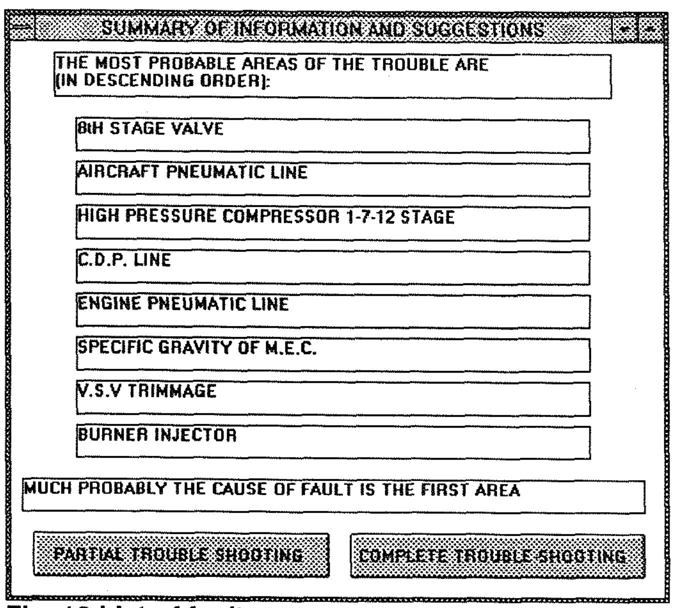

Fig. 12 List of fault area

Now the user has selected the option of partial analysis and another highly interactive phase of trouble-shooting procedure starts. The System asks for some information and, owing to the user replays, it follows the fault tree up to the detection of right fault. The various steps of this procedure are shown in fig.13 and 14.

When the fault has been detected, the System asks for information necessary for improve the archive. This way the just made experience is not lost and may be useful for the future activities.

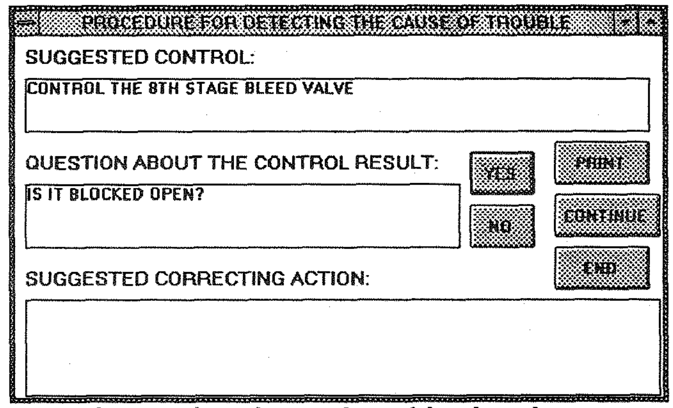

Fig.13 Interactive phase of trouble-shooting

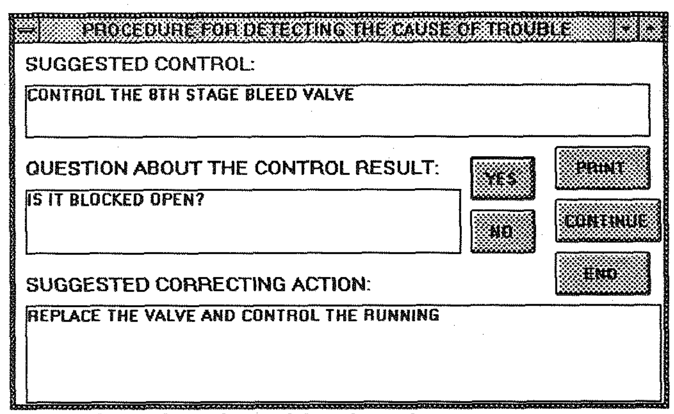

Fig.14 Results of trouble-shooting

As it has been shown the System is very powerful, user friendly and, if correctly used, it allows to save time and money during engine maintenance. Moreover it may be profitably used for training the personnel just involved in maintenance activities.

3.-NEURAL NETWORK

This part of paper deals with the activities carried out for developing small Neural Networks for turbofan diagnostics. Owing to their inherent nature and to the characteristics of used information the Neural Networks have demonstred their suitability to solve diagnosis problem (8).

A Neural Network is a computer code composed by a large number of processing elements organized and connected in different ways.

The development of a typical Neural Network is formed by different steps ranging from the selection of Network type and structure to the choice of patters for eventual training, the eventual training, the testing and, finally, the applications of Neural Network.

There are different types of Neural Networks; here multilayers Back Propagation Error Networks are considered (9). These Networks are formed by a variable number of units arranged in different layers. There is one input layer, some hidden layers and one output layer. The number of units forming each layer depends upon the problem considered. Moreover the number of input and output units depends upon the structure of both input and output data.

Each unit of a layer is fully linked to the units of the previous and following layers. There are no connections among the units of the same layer.

The heart of problem is to find the values of the importance of each connection (the weights). For the Neural Networks considered here the weight calculation is performed by the Back Propagation Error procedure. The weights are important because each unit of each layer fires when the sum of arriving inputs, multiplied for the weights, overcomes a fixed threshold value. For the Networks developed here a sigmoid law, for unit activation, is considered. The sigmoid law of each units is trimmed by using a suitable bias value (9).

The development of Back Propagation Error Neural Network similar to the ones developed here has different steps:

-the acquisition of suitable patterns for the training of Network;
-the training of Network;
-the use of Network for recognizing general patters.

The different steps for constructing the Neural Networks are described in the following sections. First of all the language selection problem is considered.

3.1-The language selection

The main characteristic of Neural Network development is the necessity to carry-out a lot of calculation in the shortest time possible. For large Neural Networks the parallel processing is the most suitable to follow. This way many microprocessors contemporary work and, even if the units of Networks are very many, the processing time is reduced (10). When parallel computers are not available, suitable languages allow to simulate parallel working on sequential machines.

The development of Neural Network for turbofan diagnostics permits the use of sequential machines owing to the low number of involving units. Moreover, the main aim of this study is to evaluate the possibility to develop Neural Networks of limited size running on personal computers. These codes might be widely used and they should be powerful tools during the maintenance activities.

During this study the use of different languages, for the phases of Neural Network development, has been very useful for reducing the computer time.

FORTRAN codes have been used for evaluating the patters for Neural Network training.

Again FORTRAN programs have been developed for the training Neural Networks.

Finally, when, the training has been completed, a high interactive, flexible language has been used: the Visual Basic. This language, developed by Microsoft, is 'Event Driven' and allows the construction of 'User Friendly' codes. This way the Neural Networks may be used by everyone also with scarce experience in computer usage.

3.2-The construction of training patterns

The patterns for training Neural Networks have been obtained by the so called "matrices of influence". These matrices are files where the variation of engine parameters (power, fuel flow, rotational speed, exhaust gas temperature, pressure ratio, etc.) versus engine component performance variations (efficiencies, capacities, bleed air, etc.) are stored in matrix form. They are obtained by the massive use of codes for the turbofan simulation.

The basic idea for using the 'matrices of influence' is justified by the following considerations.

Any trouble in a component of engine (compressor, combustor, turbine, etc.) is responsible of the decay of one or more performance of component itself (efficiency, capacity, pressure ratio, etc.). The amount of decay is not the same for the different performance. So it is reasonable to think that each trouble is the cause of the decay of a performance and has poor influence on the other.

Now each row of a matrix of influence represents the replay of the engine to the decay of a particular component performance. In other words the values of each row may be considered as the symptoms the engine show when a fault or a trouble is active.

Therefore each row of matrix represents an input pattern for Neural Network training to which an exact output pattern corresponds.

The exact output is a vector of 13 components. All values must be equal 0 except one that must be 1. The position of the value 1 in the vector indicates the cause of the malfunction. The complete set of exact output vectors is presented in table.1

Finally a sigmoid bias controlled law (9) has been used as activation law of units.

TABLE 1

STRUCTURE OF EXACT OUTPUT VECTORS

STRUCTURE OF OUTPUT VECTOR	CAUSE OF MALFUNCTION
1 0 0 0 0 0 0 0 0 0 0 0 0	**Δηfan**
0 1 0 0 0 0 0 0 0 0 0 0 0	**ΔQfan**
0 0 1 0 0 0 0 0 0 0 0 0 0	**ΔηL.P.C.**
0 0 0 1 0 0 0 0 0 0 0 0 0	**ΔQL.P.C.**
0 0 0 0 1 0 0 0 0 0 0 0 0	**ΔηH.P.C.**
0 0 0 0 0 1 0 0 0 0 0 0 0	**ΔQH.P.C.**
0 0 0 0 0 0 1 0 0 0 0 0 0	**Δηburner**
0 0 0 0 0 0 0 1 0 0 0 0 0	**ΔηH.P.T.**
0 0 0 0 0 0 0 0 1 0 0 0 0	**ΔQH.P.T.**
0 0 0 0 0 0 0 0 0 1 0 0 0	**ΔηL.P.T.**
0 0 0 0 0 0 0 0 0 0 1 0 0	**ΔQL.P.T.**
0 0 0 0 0 0 0 0 0 0 0 1 0	**ΔA hot exit**
0 0 0 0 0 0 0 0 0 0 0 0 1	**ΔA cold exit**

where: η is efficiency, Q is capacity,
L.P.C. is Low Pressure Compressor, **H.P.C.** is High Pressure Compressor, **H.P.T.** if High Pressure Turbine and **L.P.T.** is Low Pressure Turbine, **A** is area.

When the training ends the presentation of a general pattern, formed by the values of the same parameters contained in the input patters and related to an engine with a fault in progress, must lead the Neural Network to detect the fault. Moreover, owing to its inherent characteristics, the Neural Network must be able to find the right fault cause even if wrong or incomplete data is used.

3.3-The training of Neural Network

The patterns developed as shown in the previous section are used for the training of Network.

During this step, after having defined the structure of Network, the values of weights of connections among the units of layers are evaluated.

The training is carried out by the so called Back Propagation procedure. It works in the following way:
a)- The weights of the connessions among the units, bias included, are initialized to small, random values;
b)- The first pattern is delivered to the input units.
c)-The output vector is evaluated;
d)-the errors among each component of actual output vector and exact vector are computed;
e)-the sum of square of errors is evaluated;

f)-by the values of errors, the weights of connections arriving and leaving each unit are changed;
g)-the steps b-f are repeated until the value of error calculated in step 'e' is lower than a fixed tolerance:
h)- the second, the third, and all successive patterns are presented one by one and for each pattern the step c-g are repeated;
i)-the iterative process starts again with the first pattern and the steps b-h are repeated.

The global iterative process stops when a unique set of weights, minimizing the differences among the values of real and exact output vectors, is evaluated.

Now the training step may stop and the evaluated weights are the input data for the Neural Network.

3.4-The developed Neural Network

This study deals with Networks formed by three layers. The number of input units is equal to 15 because this is the number of turbofan thermodynamic and performance parameters used for monitoring the health level of engine plus an input unit used as bias.

Four Neural Networks with different number of the units of hidden layers, including the bias unit, has been considered (8, 15, 29, 43). In the following the Networks will be indicated respectively with capitol letter A, B, C, D.

The global training iterations for each Neural Network are shown in the table 2.

TABLE 2
GLOBAL ITERATION CYCLES OF DEVELOPED NEURAL NETWORKS

NETWORK A	151215 CYCLES
NETWORK B	80264 CYCLES
NETWORK C	59231 CYCLES
NETWORK D	51586 CYCLES

The complete training of Network A has required about 45 minutes using a 80486 microprocessor with 66 MHz clock.

3.5-Testing and application of Neural Network

The evaluated weights have been stored in suitable matrices and have been the main input data of Neural Networks. An example of the typical utilization of code is presented in fig.15-29

When the code starts the form of fig.15 is shown and by pressing the button 'START' the utilization of Neural Network begins and the menu of fig.16 appears on the computer screen.

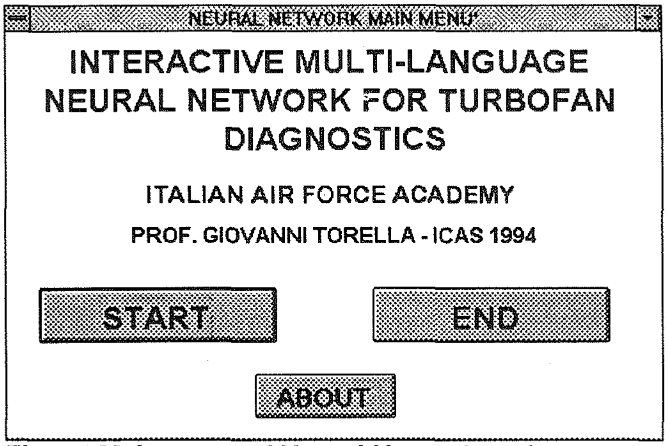

Fig. 15 Main menu of Neural Network code

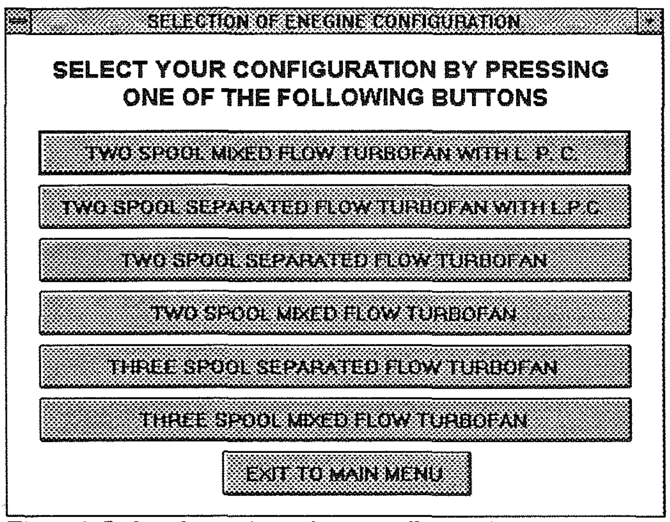

Fig. 16 Selection of engine configuration

The user must select the turbofan configuration he wants to study. Now a two spool separated flow turbofan with low pressure compressor has been selected. The considered engine is a turbofan of 20 KN. thrust class with a by-pass ratio about 5. Nevertheless different engines may be considered.

After the selection, the form of fig.17 is shown. It depicts a draft of Neural Network with different layers and it is useful for selecting the dimensions and the structure of Neural Networks. The user must provide the number of 'OUTPUT UNITS', 'HIDDEN LAYERS', 'HIDDEN UNITS' for each hidden layer and 'INPUT UNITS'. This time the input values allow to construct a Neural Network with one input layer with 15 units, 1 hidden layer with 29 units and one output layer with 13 units.

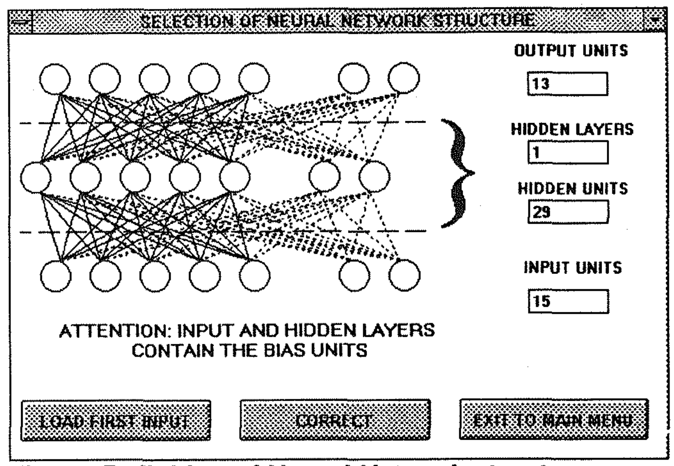

Fig. 17 Definition of Neural Network structure

It is possible to correct an eventual error by clicking the button 'CORRECT'.

If the input data about the Network structure is exact, by firing the button 'LOAD FIRST INPUT', the code proceeds and the form of fig.18 appears on the screen.

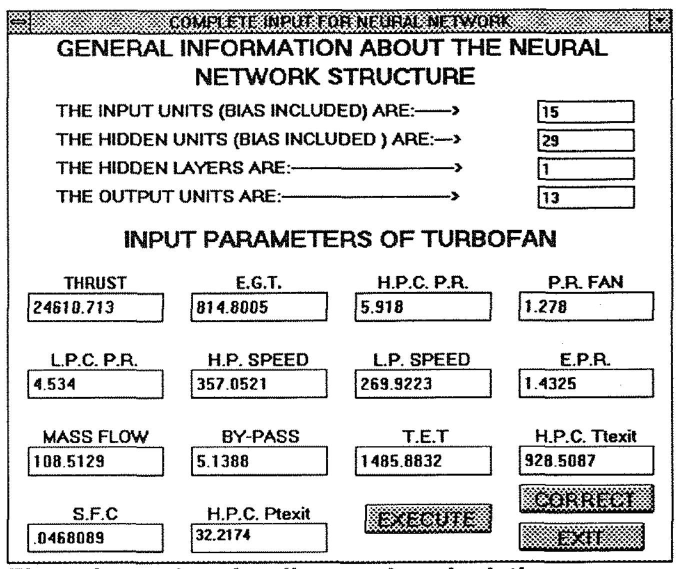

Fig.18 Input data for diagnostic calculations

The first part of form contains the summary of Network structure while the second part contains the actual values of some engine parameters. Their utilization, together with the distributed knowledge, the already computed weights, allows to detect the possible fault and its amount.

Again it is possible to correct eventual mistakes, button 'CORRECT', and the Neural Network starts to operate by firing the 'EXECUTE' button.

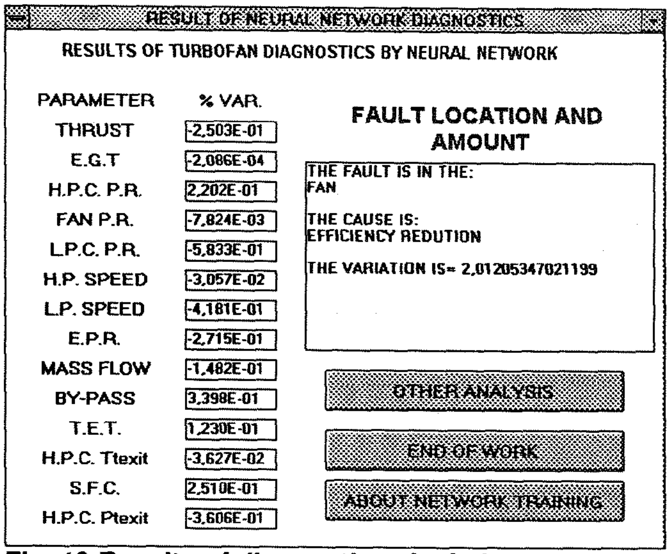

Fig. 19 Results of diagnostic calculations

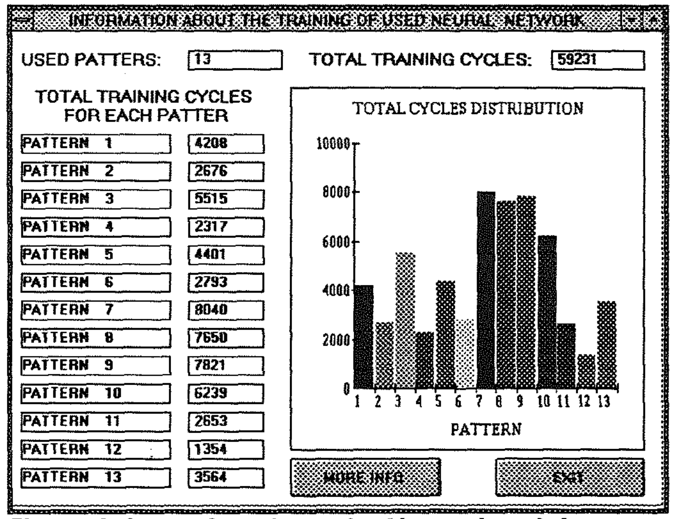

Fig. 20 Information about the Network training

In real time the form of fig.19 is displayed. It contains the value of engine parameter deviations with respect to the baseline values of an 'Healthy' engine (% VAR.). Moreover the fault location and its amount are displayed. The test is successful because the values, delivered by the form of fig. 18, have been obtained from the turbofan codes just by simulating a decay of fan efficiency of about 2%.

It is possible to carry out another analysis for a different case; by pressing the button 'OTHER CASE' the menu of turbofan configuration is shown again.

The code may furnish also information about the training of Neural Network and the history of weight calculation. By pressing the button 'ABOUT NETWORK TRAINING' the form of fig.20 appears. Detailed information about the number of total cycles, for exactly learning each patter, are displayed. Moreover a bar graph with the same information is shown. At last the total cycles (59231), for all training activity, are shown.

With the button 'MORE INFO' more details about the history of each pattern training may be seen.

The selection may be performed by a combo box. The fig.21-25 show the history of the training of pattern 4, 7, 8, 9 and 12.

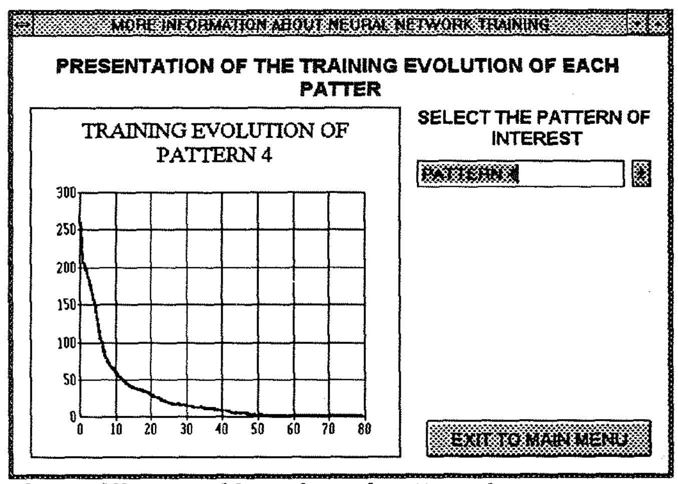

Fig. 21 History of learning of pattern 4

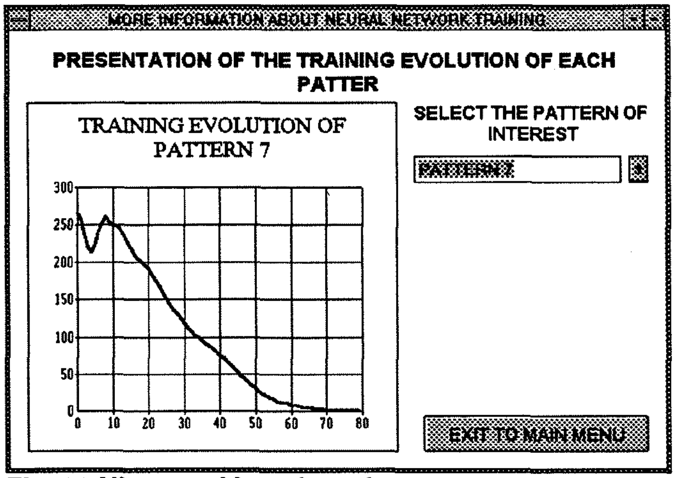

Fig. 22 History of learning of pattern 7

These pattern have been selected because pattern 12 has been learned in the least number of cycles, pattern 4 is the second one while pattern 7, 8 and 9 are the patterns requiring the higher number of total iterations. The detail of each cycle had been previously shown in the form of fig.20

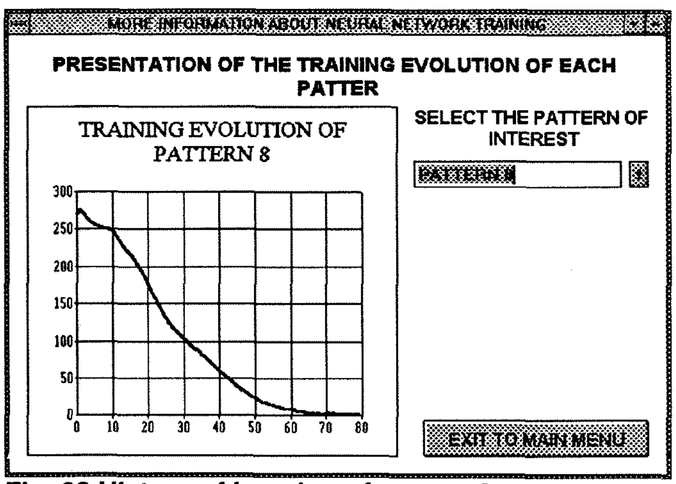

Fig. 23 History of learning of pattern 8

As shown in previous figures the first set of calculations has dealt with the test of Neural Network in order to evaluate the reliability of results. The input patterns, different from the ones used during training, have been obtained by turbofan codes by suitably simulating the decay of some component performance. As shown the codes detect both the right cause of fault and the right amount of decay.

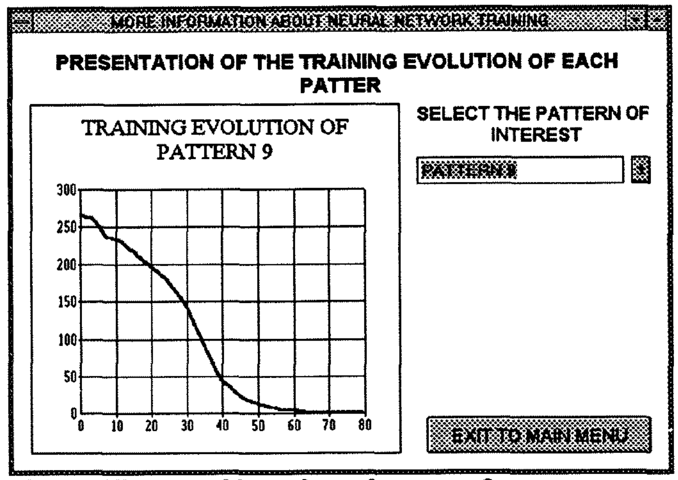

Fig. 24 History of learning of pattern 9

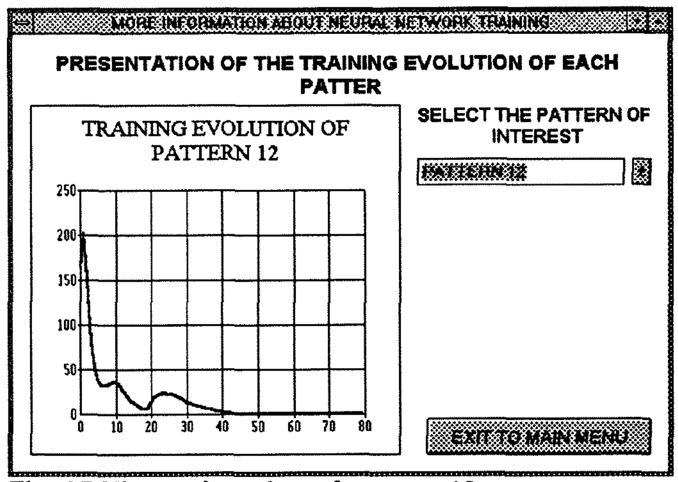

Fig. 25 History learning of pattern 12

The second set of tests has concerned the study of Neural Network capability to detect the right cause also when the input data are wrong or not all available.

Fig.26-27 shows an example of code utilization with wrong data, while fig.29-30 show example with no information about some parameters.

When the code requires the input data, fig. 26, the user has given a very wrong value of Low Pressure compressor pressure ratio (L.P.C.P.R.=**33!**). Nevertheless fig. 27 shows that the Neural Network detects the right cause.

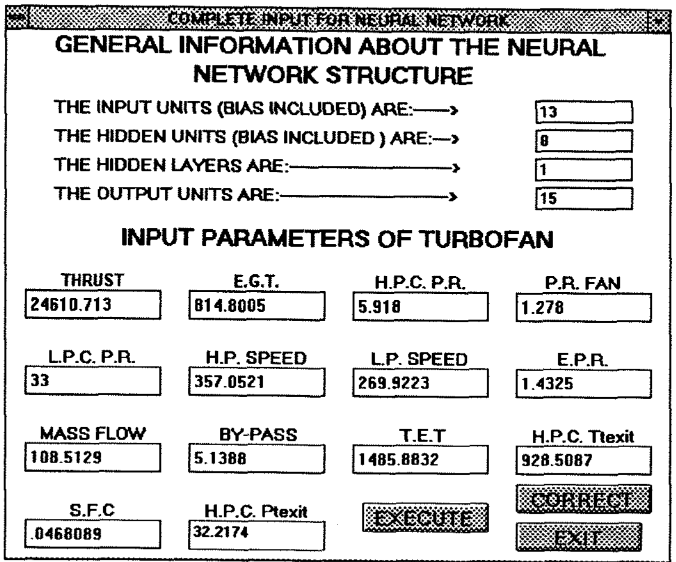

Fig. 26 Input with wrong data

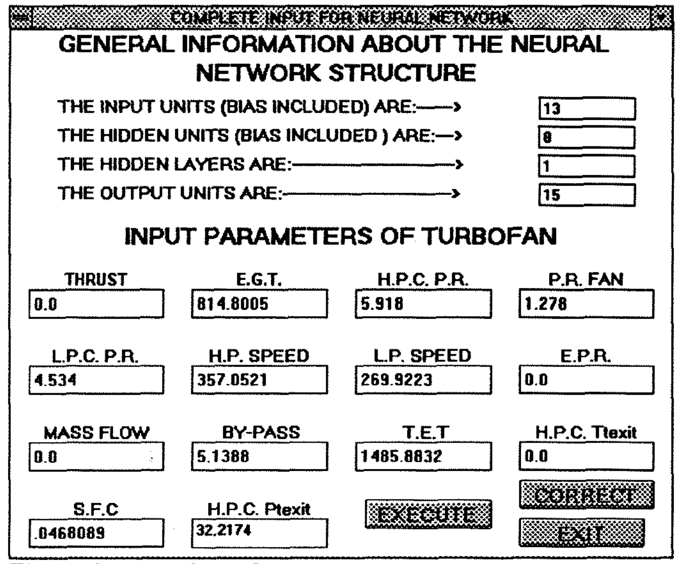

Fig.28 Incomplete data

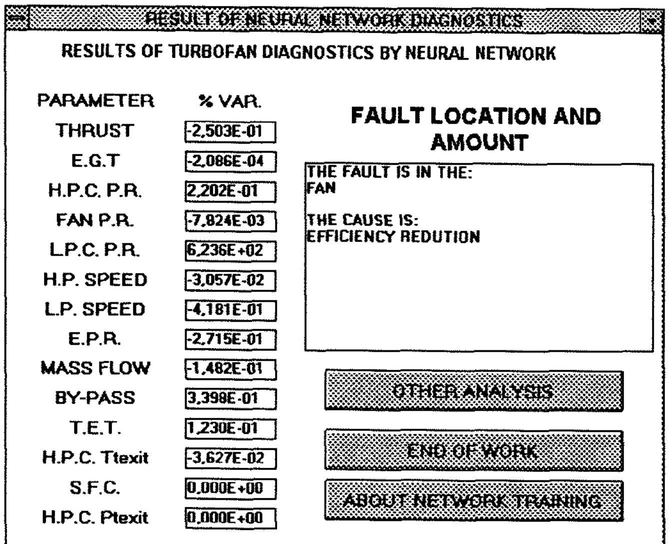

Fig.27 results of diagnostic calculations

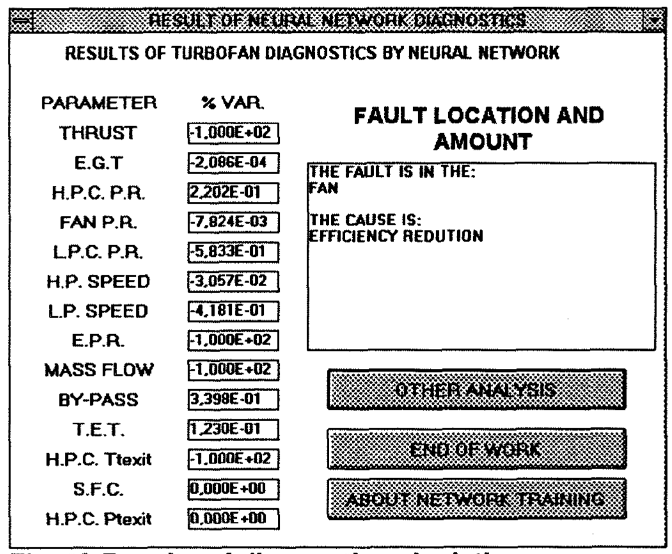

Fig. 29 Results of diagnostic calculations

Fig.28 shows the very sad situation of loss of four data corresponding to the values of thrust, engine pressure ratio (E.P.R.), inlet mass flow rate and high pressure compressor exit temperature (H.P.C. Ttexit). The Neural Network is able to detect the right fault cause, fig.29.

The calculations of both examples have been carried out by Network D. Anyway the situation is quite similar and the faults are rightly detected also with the other Networks. This means that the number of hidden layers (limitedly to the considered situation) does not affect the results.

This way it is possible to use Neural Network with hidden layer formed by units equal to the half of input units. So it possible to save both computing time and code memory allocation.

These are only preliminary results. The aim of authors is to investigate carefully by changing either the number of hidden layers and the neuron activation law. Moreover different type of Neural Networks will be considered.

4-.CONCLUSIONS

The study has proved that Expert Systems with both procedural and declarative languages may be successfully developed for the diagnostics and trouble-shooting of turbofan engines.

The study has shown that the Knowledge Base is the heart of Expert System and it's the only way the save the expertise of personnel.

The developed Expert Systems are fast, reliable and run on personal computers. Therefore they are powerful tools for both maintenance activities and the training of personnel.

At the present works are in progress for developing improved Expert Systems. These Systems require the values of performance and thermodynamic parameters obtained from test bed and, step by step, guide the user to detection of both right fault and maintenance action.

Moreover the study has shown that it is possible to develop small Neural Network for Turbofan diagnostics. The patterns for the training may be obtained from matrices of influence or from suitable codes for turbofan simulation. The Network codes are reliable, fast and user-friendly and are capable to detect the right fault even when the input data are not complete or the values are wrong.

At the present works are in progress for developing other type of Neural Networks with different structure and organizations.

5.-BIBLIOGRAPHY

1)-C. Townsend "Introduction to TURBO PROLOG" Sybex Edition 1986

2)-N. Goldenthal "TURBO PROLOG" Weber Systems, Inc. 1987

3)-G.TORELLA "Expert System for the Simulation of Gas Turbine Engines" ASME Paper 92-GT-408 Cologne, Germany June 1-4, 1992

4)-G.TORELLA "Expert System for the Trouble-Shooting and the Diagnostics of Engines" 28th AIAA/SAE/ASME/ASEE Joint Propulsion Conference July 6-8, 1992 Nashville, TN

5)-G.TORELLA "Expert Systems for the Simulation of Turbofan Engines" - XI International Symposium on Airbreathing Engines ISABE - Tokyo 1993

6)-G.ENTSMINGER "Secrets of the VISUAL BASIC for Windows Masters" SAMS Publishing 1992

7)-PRATT & WHITNEY AIRCRAFT "JT8 Engine Manual" 1983

8)-T.SORSA, H.N. KOIVO "Application of Artificial Neural Networks in the Process Fault Diagnosis" Automatica, vol. 19 No.4 pp843-849, 1993

9)-J.A.FREEMAN, D.M.SKAPURA "Neural Networks Algoritms, Applications and Programming Techniques" Addison-Wesley Publishing Company 1991

10)-M.McCORD NELSON, W.T.ILLINGWORTH " A Practical Guide to Neural Nets" Addison-Wesley Publishing Company 1991