JOURNAL OF AIRCRAFT
Vol. 39, No. 2, March–April 2002

Optimization of Helicopter Subfloor Components Under Crashworthiness Requirements Using Neural Networks

Chiara Bisagni,* Luca Lanzi,† and Sergio Ricci‡
Politecnico di Milano, 20156 Milan, Italy

An optimization procedure is developed for the design of structural components under crashworthiness requirements. The optimization procedure consists of first developing a system of parallel neural networks able to reproduce the structural behavior during the crash phenomena. Training and test sets are generated by finite element analyses. Then, to find the optimal configuration, both sequential quadratic programming and genetic algorithms are implemented to evaluate the objective and constraint functions using the response surfaces generated by the neural networks. The procedure is applied for the optimization of helicopter subfloor components. In this application, the objective function is taken equal to the sum of two terms: the crush force efficiency and the specific absorbed energy. The values of maximum and mean forces are constrained to meet crashworthiness requirements, whereas other constraints are applied to define the feasibility of the structural solution. The optimal configuration, obtained measuring orders of magnitude savings in the CPU time, allows an increase of the crush force efficiency equal to 18% and a decrease of the mass equal to 8%.

Nomenclature

$F_{\max}$	=	maximum force
F_{mean}	=	mean force
$\bar{F}_M$	=	upper constraint on the maximum force
$\bar{F}_m$	=	lower constraint on the mean force
K	=	weighting parameter of the objective function
$\boldsymbol{T}$	=	vector of the normalized time
$\boldsymbol{x}$	=	vector of the normalized design variable
ε	=	strain
$\dot{\varepsilon}$	=	strain rate
ν_i	=	penalty function
σ	=	stress
$\sigma_0(\varepsilon)$	=	stress value corresponding to the strain ε obtained with quasi-static loads

Introduction

THE structural design under crashworthiness requirements has to assure the safety of the occupants within a certain range of impact conditions. Survivable accidents are defined as those in which the peaks of acceleration are below the human tolerance, and some events, such as fire or collisions of hard protruding objects, are prevented.

In the aerospace field, crashworthiness requirements are established both for military and for civil vehicles.[1,2] To obtain good energy absorption, the crash phenomena should be taken into account in the first phases of the project, and the interaction between the landing gear, the subfloor, the cockpit and the seats should be considered. Besides controlled collapse of the cabin and, in case of helicopter crashes, safe supports for the suspended masses, such as the engine and the rotor, should be guaranteed.[3,4] Therefore, the design methodology appears very complex, requiring the interaction between experimental tests and numerical analyses based on multibody and finite element models.

Until a few years ago, absorption structures were designed just to satisfy the imposed requirements, whereas, nowadays, optimization procedures are investigated also for crashworthiness considerations, even during the first phases of the project.

The definition of a structural optimization procedure, under crashworthiness requirements, unfortunately is complex and very expensive from the computational point of view. For this reason, several examples of structural optimization are applied in the aerospace field, but very few examples meet crashworthiness requirements. In fact, a well-established approach to the structural optimization is based on the substitution of the original optimization problem by a sequence of approximated problems, obtained by means of a kind of approximation of constraint and objective functions.[5,6] During the local optimizations of the approximated problems, the values of constraint and objective functions, as well as their derivatives with respect to the design variables if required, are computed directly by the approximated functions, therefore, avoiding expensive finite element analyses. After every local optimization, a refined analysis is performed to update the approximated problems, and the procedure continues until convergence. This approach does work very well when the approximated problems reproduce well the analyzed structural problem, at least during the local optimization. Unfortunately, this approach is not easily applicable to crash problems, where the design space is highly nonlinear, nonconvex, and even disjointed. The finite element models should be able to take into account the elastic–plastic behavior of the materials, the changes of the boundary conditions resulting from contacts between model parts, the failure of the material and of the joints, and other phenomena related to the high deformation rate.

In 1996, Etman et al.[7] developed a crashworthiness design optimization based on multipoint sequential linear programming and applied this procedure to the road vehicle field. In 1997, Hajela et al.[8] developed a topological crashworthiness optimization of a rotorcraft subfloor based on genetic algorithms and neural networks trained by multibody models.

The present work deals with an optimization procedure based on the response surfaces obtained by neural network systems trained by finite element analyses. Both sequential quadratic programming and genetic algorithms are implemented to find the optimal feasible configuration. The optimization procedure is applied to the design of a helicopter subfloor component.

Received 19 January 2001; revision received 5 October 2001; accepted for publication 12 December 2001.

*Assistant Professor, Department of Aerospace Engineering, Via La Masa 34; Chiara.Bisagni@polimi.it.
†Graduate Student, Department of Aerospace Engineering, Via La Masa 34; lanzi@aero.polimi.it.
‡Associate Professor, Department of Aerospace Engineering, Via La Masa 34; Sergio.Ricci@polimi.it.

Optimization Procedure

The optimization procedure here presented can be schematized into three different steps.

In the first step, the finite element model of the structural component corresponding to the initial configuration is defined. The numerical model, tuned on the basis of experimental results, constitutes the starting point for the second step of the procedure.

The second step consists of developing a system of parallel neural networks able to reproduce the crash behavior of the structural component. The training and test sets of the neural networks are generated directly by finite element analyses, by the use of the finite element model validated during the first step. After the identification of the significant design variables, a minimum number of finite element analyses are performed to generate the examples for the training and test sets of the neural networks. The trained neural networks system allows the quick evaluation of the structure behavior with respect to the design variables without any other finite element analysis, considerably reducing the total computational costs.

The third step consists in the optimization phase. After the choice of the optimization method and the definition of the objective and constraint functions, the optimization algorithms are implemented to find the optimal configuration using the response surfaces generated by the neural networks.

Before the optimization of a realistic structural component was studied, the procedure was successfully applied to few simple structures.[9] First, a very simple aluminum tube was considered: by the use of the thickness and the diameter as design variables, important considerations are deduced about the training and test sets of the neural networks. Then, the optimization procedure was applied to a riveted aluminum tube, this time using the thickness, the diameter, and the number of vertical rivets as design variables. The presence of singularities in the structure behavior, given by the rivets' failure, makes the problem quite complex and allows the evaluation of the capabilities of the developed procedure.

The optimization procedure is then applied to a typical subfloor structural component of a civil helicopter, and only this example is reported in the following pages.

Helicopter Subfloor Structural Components

The helicopter subfloor structure is very important to crashworthiness because, during vertical crashes characterized by a high vertical component of the impact velocity, it has to absorb the impact energy, to distribute the loads to the upper floor as uniformly as possible, and to collapse in a controlled manner without breaking up or decreasing excessively the cabin volume.

A typical subfloor frame is presented in Fig. 1. It consists of longitudinal keel beams and lateral bulkhead sections connected to each other by intersection elements and is covered by the outer fuselage skin and the cabin floor. Under the vertical crash loads, the structural intersection elements behave as hard-point stiff columns, causing high-deceleration peak loads. Consequently, the intersection elements optimization is very important to reduce the peak of acceleration and can be considered as the first step for a global optimization of the subfloor structure.

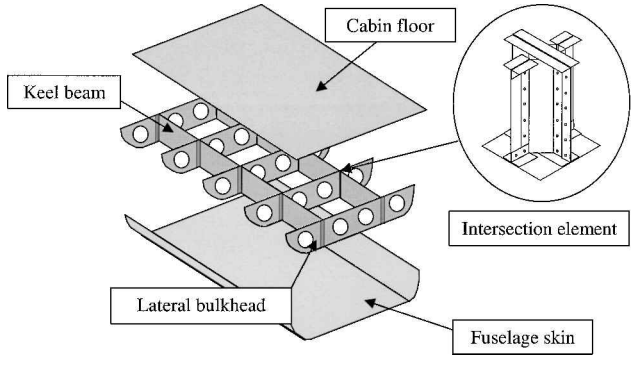


Fig. 1 Typical helicopter subfloor structure.

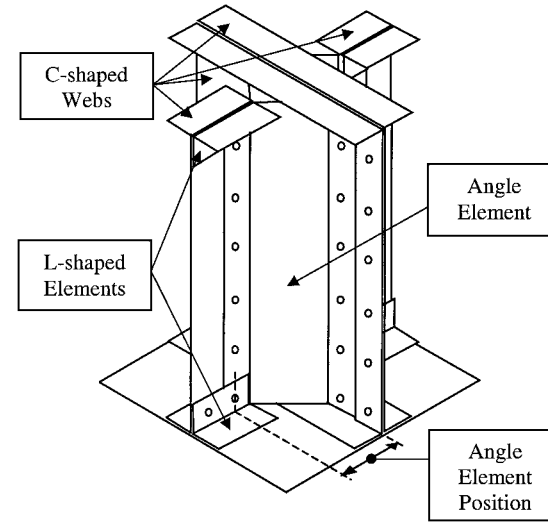


Fig. 2 Intersection element of helicopter subfloor.

The intersection element configuration considered as the baseline design is directly taken from a typical civil helicopter and is represented in Fig. 2. It is 197 mm high and consists of two vertical C-shaped webs and four angle elements, which produces a close square section with a diagonal delimited by the two webs. At the top and at the bottom, there are two bases, riveted to the vertical components and to eight small L-shaped elements. The thickness of all of the panels is equal to 0.81 mm, except for the bases, where the thickness is equal to 1.27 mm. The intersection element is made of aluminum alloy 2024 T3 (Ref. 10) and is fastened using typical aeronautical rivets vertically and blind rivets on the bases.

Finite Element Model

The finite element model[11] of the intersection element initial configuration is used in the optimization procedure as a design instrument to generate the examples for the training and test sets of the neural networks. The validation of the finite element models by experimental results is, therefore, very important before starting the neural networks generation and training. The finite element analyses are performed using the commercial explicit code PAMCRASH.[12]

The intersection element is modeled by 4-node shells, whose dimensions are about 3×3 mm. The rivets and the blind rivets are represented by beam-type elements with nonlinear behavior. The finite element model of the initial configuration consists of 21,372 shell elements and 172 beam elements.

The aluminum alloy is modeled as an isotropic elastic–plastic material, and a strain rate law is defined according to the Cowper–Symonds formulation as

$$\sigma(\varepsilon, \dot{\varepsilon}) = \sigma_0(\varepsilon) \cdot [1 + (\dot{\varepsilon}/D)^{1/P}] \tag{1}$$

where P and D are constant properties of the material.[13] When it is supposed that the collapse of the structure mainly involves the failure of the rivets, a failure criteria based on the maximum displacement is adopted for the beam elements only.[14–16]

Three kinds of contact algorithms and several contact surfaces are defined to avoid the penetrations between different parts. The contact algorithms consist in the panel contact with itself, in the surface-to-surface contact between two different panels, and in the surface-to-edge contact when the edge of a panel comes into contact with an edge of another, as in the case of the L-shaped panels with the C-shaped webs or in the case, less obvious in the undeformed structure, of the angle elements with the C-shaped webs.

The model was then validated by means of experimental results. The crash tests of the initial configuration of the subfloor component were performed using a drop test machine at the Department of Aerospace Engineering of Politecnico di Milano.[17] During the tests, the displacement and the acceleration of the impact mass were measured using an incremental encoder and a piezoelectric

Table 1 Comparison between experimental and numerical results of the initial configuration

Parameter	PAMCRASH analysis	Experimental results	Error, %
Maximum force, kN	51.45	52.31	1.6
Mean force, kN	24.29	22.48	8.5
Mass, kg	0.489	0.481	0.2
Specific absorbed energy, kJ/kg	4.172	3.926	6.3
Crush force efficiency	0.47	0.43	9.3

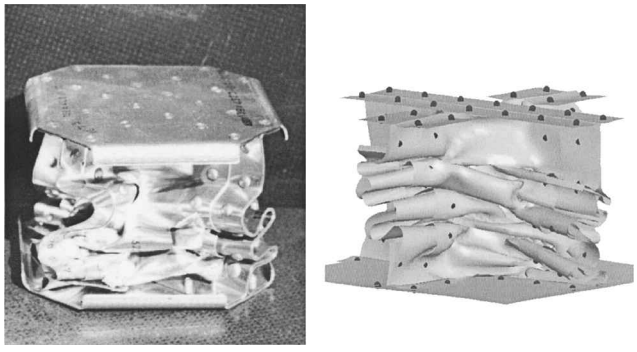


Fig. 3 Comparison between the experimental and numerical deformations of the initial configuration.

accelerometer, respectively. The tests were performed with an impact velocity of 7 m/s.

Figure 3 shows the comparison between the experimental and the numerical deformations of the initial configuration. Table 1 reports the comparison between the obtained values of the maximum force, the mean force, the specific absorbed energy, and the crush force efficiency, where the specific absorbed energy is defined as the energy absorbed during the crash over the deformed mass, and the crush force efficiency is defined as the ratio between maximum and mean force. The errors on the maximum and mean force between the numerical analyses and experimental results are 1.6 and 8.5%, respectively. They are acceptable when the complexity of the deformations and the high number of panels and contacts are considered. It seems also rational to attribute the differences on the mean force values to that no failure of the panels is considered in the numerical analyses, although the angle elements showed some failures after the experimental tests. In any case, the adopted solution seems a good compromise between the time required performing the numerical analyses and the obtained results.

Optimization Domain Definition

The complexity of the considered structure and the high number of possible design variables suggest conducting a preliminary sensitivity study to evaluate which design variables can be the most reliable for the optimization procedure. When started from the initial configuration, a sensitivity study is performed by finite element analyses. To make possible a direct comparison among different structural configurations, the impact conditions are maintained the same for all of the considered configurations, and the impact velocity is fixed and equal to 7 m/s. The sensitivity study shows the following:

1) The translation of the angle elements, toward the intersection element center, allows the increase of the mean force and the decrease of the maximum force.

2) The increase/decrease of all of the panel thicknesses allows the increase/decrease of both the mean and the maximum forces and obviously involves an increase/decrease of the structural weight.

3) The decrease of the vertical rivet number allows a decrease of both the mean and the maximum forces.

Nevertheless, the nonlinear behavior of the problem makes the foregoing statements true only near the examined initial configuration and not completely generalizable. However, the combination between the position and the thickness of the angle elements, the thickness of the other panels, and the number of the vertical rivets seems to allow considerable modification of the global capabilities of the intersection element. For this reason, the thickness of the webs, the thickness of the angle elements, the position of the angle elements, and the number of vertical rivets are taken as design variables. To define uniquely the design variables, the position of the angle elements that is maintained 45 deg, shaped with respect to the web, is measured as the distance of the rivets' vertical line from the intersection element center, as shown in Fig. 2. With regard to the number of the vertical rivets, only the number of the intermediate rivets is considered as a design variable because the position of the first and last rivets can not be changed as these two rivets connect together the L-shaped with the C-shaped panels. The domain values of all of the design variables, reported in Table 2, are defined around the initial configuration, taking into consideration a reasonable distance between the vertical rivets and reasonable thicknesses of aluminum sheets. During the optimization, the panels' thicknesses are treated as continuous variables, even if commercially available thicknesses have discrete values. Once the optimum values of the panels' thicknesses are found, the nearest commercially available thicknesses that are smaller or larger than the optimum values are taken in defining the final design.

Table 2 Domain values of the optimization space

Parameter	Initial value	Minimum value	Maximum value
Angular elements thickness, mm	0.81	0.64	1.3
Angular elements position, mm	40	35	65
Other panels thickness, mm	0.81	0.64	1.3
Vertical rivets number	8	4	8

Neural Networks

The idea to solve engineering problems using neural networks was developed in the 1940s in the United States.[18] The neural networks try to mimic the brain's own problem solving process. In the human brain, the transmission and processing of data are made possible by a very complicated net of neural cells, called neurons, that have the assignment to transfer electric–chemical signals from one side of the organism to the others, in the most efficient and quick way. The neural networks consist of relatively simple processing elements (nodes or units) connected by links. A unit receives the signal from the input links and computes an activation level that it is sent to the next layer along the output links.

The neural networks are developed on the hypothesis that every neuron operates giving out an output signal only if the sum of all of the input signals exceeds a certain limiting value, as in the human neural system. Another important peculiarity of the neural networks is their parallel structure. In this way, every neuron contains and represents much more information connected to more different concepts, which allows obtaining a shared out representation of quantities and concepts.[19,20] A neural network is not programmed to solve a specific problem in an applicative way. It does not use rules or equations describing the examined problem and does not include choice and classification criteria as associative rules. Just as humans apply the knowledge gained from past experience to new problems, a neural network uses previously solved examples to build a system of neurons that learn how to solve a new problem by changing the nature and the intensity of the input links. Consequently, the training phase is fundamental, and it generally consists of two main stages: the learning phase and the verification phase. In the first phase, neural networks learn to reproduce a specific problem only through the knowledge of a certain number of inputs and outputs, called a training set. In this way, the neural networks look for patterns in training sets of data, learn these patterns, and develop the ability to classify correctly a new pattern or to make forecasts and predictions. In the second phase, the neural networks capabilities are estimated in terms of the generalization ability through a certain number of verification cases that form the test sets. The choice and the settlement of the examples of the training and test sets are very

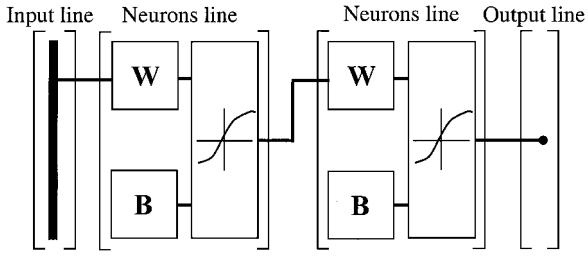


Fig. 4 Schematic representation of a MLP neural network.

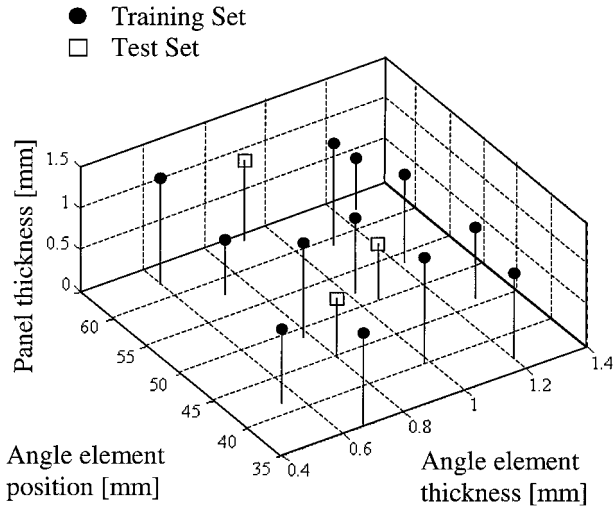


Fig. 5 Example of training and test examples location with eight vertical intermediate rivets.

important because they qualify the final performances of the neural networks.

The neural networks used in the present investigation are multilayers perceptron (MLP),[21–23] as shown in Fig. 4. The used MLP neural networks are feedforward networks because the signals always propagate from the first to the last line, whereas, in the nervous system, there are signals that come out from a neuron, cross other neurons, and return to the starting neuron, after being properly weighted.

In MLP neural networks, the learning phase is the process of adjusting only the weights of the neuron connections such that the rms error between known outputs of the training set and returned outputs is minimized.[24] The work of the most common training algorithms is based on the backpropagation learning rule.

Training and Test Sets

The training and test sets are defined into the optimization domain by changing the design variable values. To minimize the number of finite element analyses and, consequently, the total CPU time, only 45 PAMCRASH runs were performed. Among these 45 runs, 15 were made with 4 vertical rivets, 15 with 6 vertical rivets, and the last 15 with 8 vertical rivets. The values of the other design variables were chosen using a suitable algorithm, which guarantees a random and homogeneous allocation of the runs inside the optimization domain. These 45 runs were then divided into the training and the test sets. In particular, the training test consists of 36 runs, whereas the test set consists of the remaining 9 runs.

For example, Fig. 5 shows the values of the design variables corresponding to the 15 runs with 8 vertical rivets. Of these 15 runs with 8 vertical rivets, 12 runs belong to the training set and 3 to the test set. Similar runs of training and test sets are taken with 4 and 6 vertical rivets, respectively.

The need to analyze different geometrical configurations requires the implementation of an algorithm capable to generate the PAMCRASH input files in an automatic way. To do so, an ad hoc procedure is realized using MATLAB® 5.3.

Neural Networks System Definition

MLP neural networks are very adaptive and, nowadays, very efficient learning algorithms are developed based on these kind of neural networks. These properties make the MLP architecture suitable in the present application, where the main aim of the neural networks is to generate the response surfaces for a crash problem using a reduced set of examples. Then, when the attempt is made to reduce the total CPU time required to generate both training and test sets, the last ones are used in a double way, so that, when a first learning and verification phase is completed, the test examples are added to the training set, and another learning phase is started.

Because the use of only the maximum and mean force values seems a restrictive criterion to compare the performances of different intersection typologies, another neural network is designed to reproduce the whole load-time impact curve. The optimization phase requires the knowledge of the different structural responses, which are the maximum force, the mean force, and the load-time curve; moreover, the presence of rivets failure and the structural collapse make the responses very different from each other. The design of a single neural network that should be able to evaluate at the same time all of the required structural responses seems incompatible with the need to maintain the example set as limited as possible. To have better performances and to reduce the training and test sets, it was decided to use different neural network architectures.

Figure 6 shows the adopted neural networks architecture. Neural networks are used, in numbers of two and three, to calculate the maximum force and the mean force values separately. After that, a third neural network is designed to calculate the load-time curve. The final values are obtained putting all of the outputs together into a final neural network.

BOX_1(...) denotes the function that is assigned to receive as input the vector $\boldsymbol{x}$, containing the normalized design variables and in particular the normalized values of the angle element thickness, the angle element position, the thickness of the other panels, and the number of vertical rivets. BOX_1(...) uses two neural networks, net1_a and net1_b, and returns the normalized maximum force. Both net1_a and net1_b use nine neurons divided into three different lines, respectively, with tansigmoid, tansigmoid, and linear functions.[23] Their learning is performed using the Levenberg–Marquardt backpropagation method (see Refs. 25 and 26) implemented with MATLAB 5.3 (Ref. 23). Table 3 reports the number of neurons in every line. The maximum error between the value of the maximum force obtained by finite element analysis and by neural networks is about 7.2% and is obtained for a training set case, as shown in the bar graph of Fig. 7.

BOX_2(...) denotes the function that is assigned to receive $\boldsymbol{x}$ as input and to return the normalized mean force value. To obtain the behavior of the mean force response surface as regular as possible and to avoid overfitting phenomena, or training memorization,

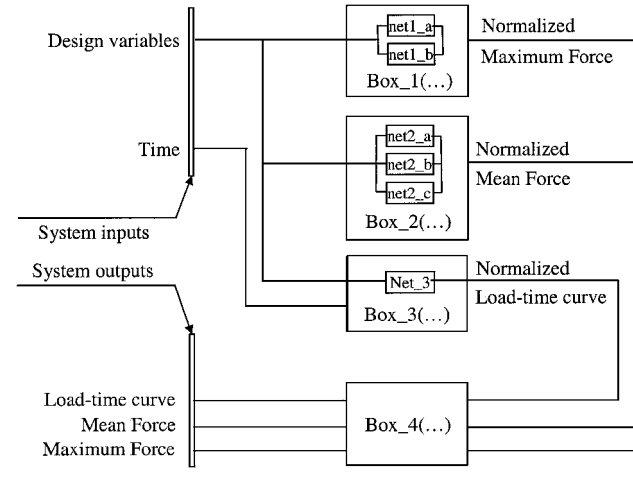


Fig. 6 Adopted neural networks system.

three different neural networks, net2_a, net2_b, and net2_c, are used. Both net2_a and net2_b use 9 neurons, whereas net2_c uses 11 neurons, as reported in Table 3. Three different lines, respectively, with tansigmoid, tansigmoid, and linear function form the nets. Their learning is performed using the Levenberg–Marquardt backpropagation method. The BOX_2(. . .) subsystem also allows good generalization capabilities. The maximum error between the mean force value obtained by finite element analysis and by the neural network is about 6.8% and is obtained for a training set case as shown in Fig. 7.

BOX_3(. . .) receives as input two normalized column vectors, $\boldsymbol{x}$ and $\boldsymbol{T}$, where $\boldsymbol{T}$ contains the time values used to describe the points of the load-time curve. The time vector is normalized and is 60 points long, which is considered sufficient to describe with a good approximation crash phenomena that are about 0.012 s long. A neural network, net3, is used with three lines of 8, 15, and 1 neurons, respectively, as reported in Table 3, with tansigmoid, tansigmoid, and linear functions. The BOX_3 learning phase is rather heavy from a computational point of view because there are 175 weights and 24 bases. Consequently, there are 199 independent variables for the rms error minimization. On every iteration, it is necessary to evaluate the error obtained for the 60 points received as input and for the 36 cases used in the training set. However, the learning phase is several orders of magnitude less expensive than a finite element model simulation. The neural networks learning is performed using the Levenberg–Marquardt backpropagation method. The dimensions of the net and of the training set make the training almost independent from the initial conditions. The maximum errors between the maximum and mean forces obtained by the finite element analysis and by neural network are about 7.3 and 8.6% respectively, as shown in Fig. 7.

To improve the neural network performances, BOX_1(. . .), BOX_2(. . .), and BOX_3(. . .) are used together. Their responses are then passed as inputs to BOX_4(. . .). The algorithm of BOX_4(. . .) evaluates, first, the maximum and mean forces calculated on the load-time curve returned by BOX_3(. . .); then, it compares the new obtained values to the values returned by BOX_1(. . .) and BOX_2(. . .), respectively, and finds the final values as a weighted average. After this, the load-time curve is redrawn.

The capabilities of the complete neural networks system seem able to reproduce very accurately the crash behavior. Figures 8 show the comparison of load-time curves obtained by numerical analysis and by neural networks system for a training set case and a test set case, respectively.

Table 3 Number of neurons and architecture of the used neural networks

	Neurons		
Network	First line	Second line	Third line
net1_a	4	4	1
net1_b	5	3	1
net2_a	4	4	1
net2_b	5	3	1
net2_c	6	4	1
net3	8	15	1

Considerations About Response Surfaces

Different groups of response surfaces for both maximum and mean forces are obtained using the neural networks system. Figure 9 shows some examples of response surfaces, obtained fixing two normalized design variables. The response surfaces demonstrate highly nonlinear behavior and present opposed behavior, so that only few general considerations can be pointed out.

Figure 9a, obtained by fixing the number of intermediate rivets and the position of the angle elements, shows that the maximum force value increases when the angle element thickness or the other panels thickness is incremented. In particular, a variation of the

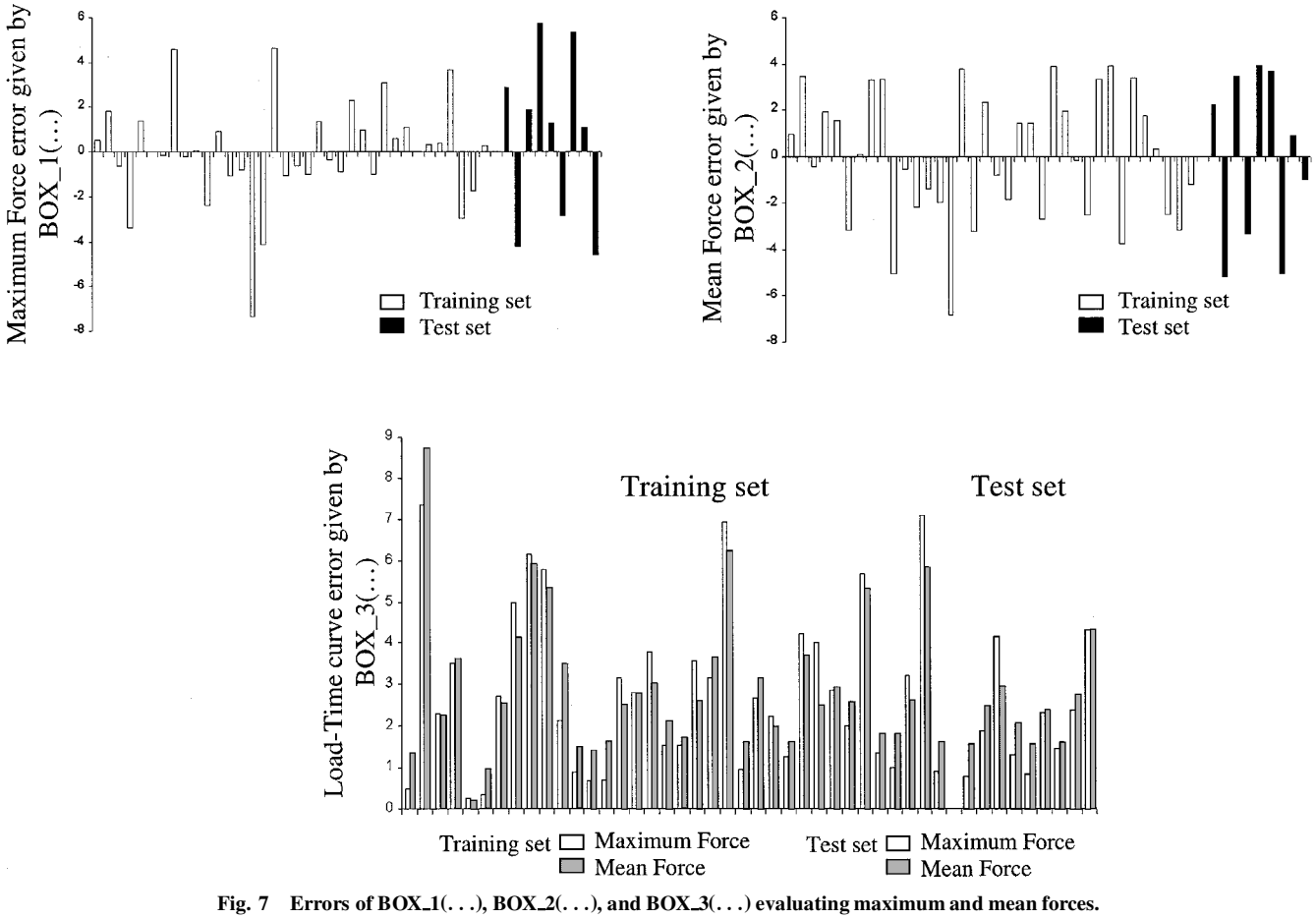
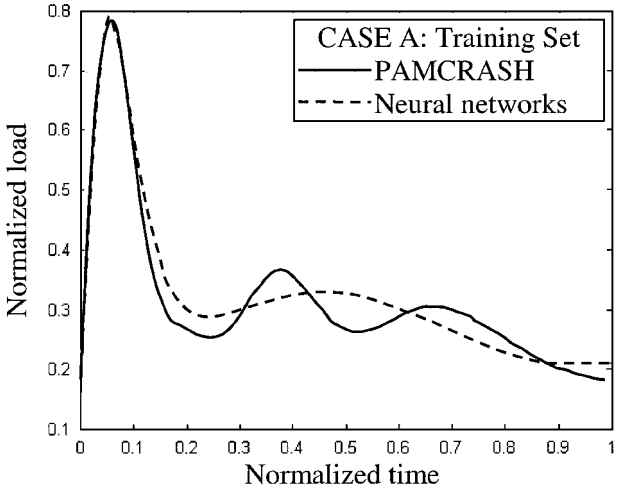
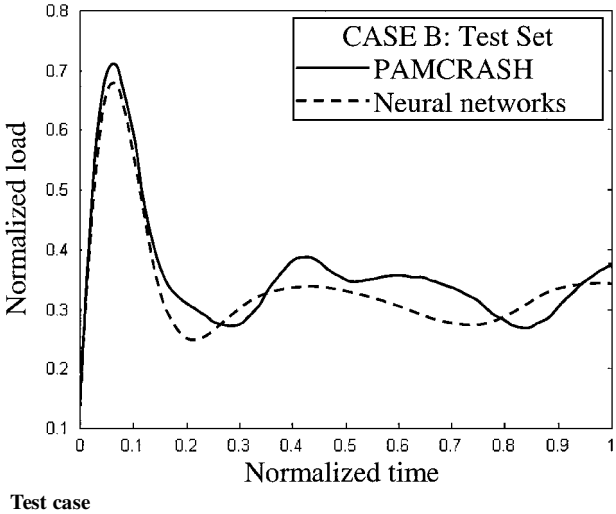


Fig. 7 Errors of BOX_1(. . .), BOX_2(. . .), and BOX_3(. . .) evaluating maximum and mean forces.



Training case



Test case

Fig. 8 Load-time curves obtained by PAMCRASH and by the neural networks.

angle element thickness produces a greater effect than a variation of the other panel's thickness. Experimental results confirm this consideration, which is also true for the mean force. The failure and the deformations of the angle elements are, in fact, more significant than the ones observed in the other panels.

On the other hand, the maximum force behavior appears irregular and not completely foreseeable when the rivets number and the angle elements position are changed, as shown in Fig. 9b, obtained by fixing both the thickness of the angle elements and of the other panels. A partial interpretation of the maximum force behavior could be given thinking that, during the first phases of the crash phenomena, the angle elements are neither immediately or directly compressed, because they are spaced out from the basis, and that their role is to make the structures stiffer. It seems reasonable that the angle elements' collapse happens by destabilization in the horizontal direction and not in the vertical direction, so that there is an ideal angle elements configuration depending on the thickness, on the position, and on the points in which stresses are concentrated, namely, on the vertical rivets number.

The behavior of the mean force appears complex and irregular. Consequently, a direct comparison between different surfaces does not allow general considerations, as shown in Figs. 9c and 9d, both generated by fixing two different values for the thickness of the angle elements and of the other panels. However, the position and the thickness of the angle elements and the number of the vertical rivets are able to modify highly the intersection element capabilities.

Optimization Problem Formulation

The objective function is chosen to describe well the two most desirable characteristics of the helicopter subfloor intersection elements during crash phenomena, that is, high crush force efficiency and high mass specific absorbed energy. The high crush force efficiency allows the minimization of the difference between the mean force and the maximum force, limiting the peak of acceleration. At the same time, for the same value of the mass, the mean force must be as high as possible. In the considered problem, because the impact velocity is chosen to remain constant, the mean force and the mass specific absorbed energy are equivalent: the impact velocity is, therefore, constant and equal to 7 m/s. Upper and lower constraints are defined on the maximum and the mean force values,

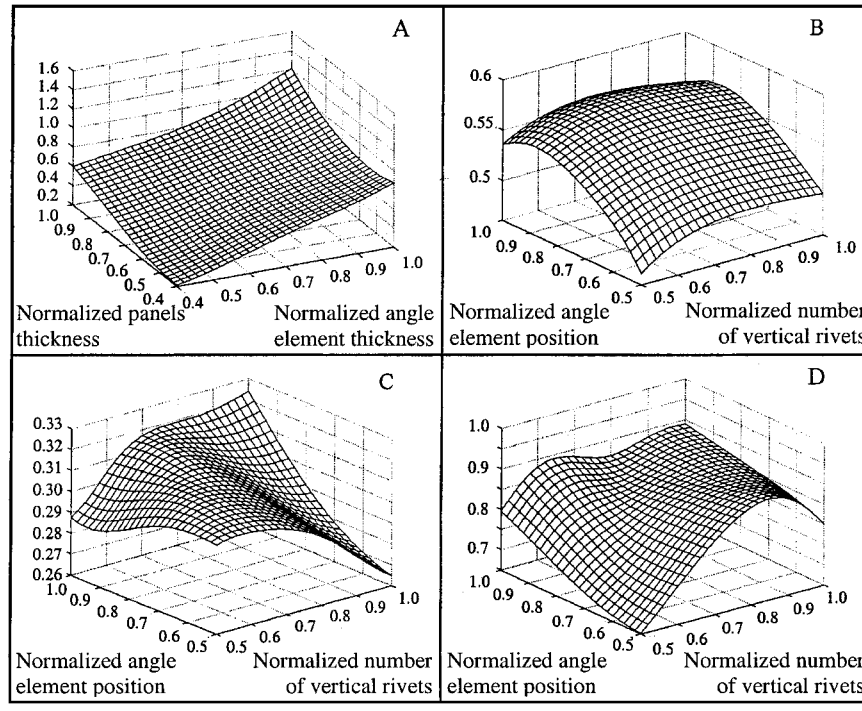


Fig. 9 Examples of response surfaces for different values of normalized design variables: a) maximum force response surface, b) maximum force response surface, c) mean force response surface, and d) mean force response surface.

representing the human tolerance limit, $\bar{F}_M$, and the structural efficiency, $\bar{F}_m$, respectively.

The optimization problem is expressed, in terms of the objective and constraint functions, by

$$\text{obj} = -\left\{\left(\frac{F_{\text{mean}}(\boldsymbol{x})}{F_{\max}(\boldsymbol{x})}\right) + K \cdot \left(\frac{F_{\text{mean}}(\boldsymbol{x})}{\text{mass}(\boldsymbol{x})}\right)\right\}$$

$$F_{\text{mean}} \geq \bar{F}_m, \qquad F_{\max} \leq \bar{F}_M \tag{2}$$

The vector $\boldsymbol{x}$ is given as input to the neural networks system that returns as outputs the mean force value, the maximum force value, and the load-time curve. The obtained values are then passed to the optimization algorithm.

An important aspect concerns the possibility to give a different weight to the crush force efficiency or to the mass specific absorbed energy. For this reason a weighting parameter K is introduced. The optimization is performed using two different algorithms: a sequential quadratic programming (SQP) algorithm and a genetic one.

SQP Algorithm

A possible strategy in solving constrained nonlinear optimizations is to transform the original problem into several easier subproblems that can be solved using an iterative process. In particular, the SQP algorithm may represent the state-of-art in nonlinear optimization programming methods.[27,28]

This method allows solving constrained optimization, just as it is done for unconstrained optimization. The main idea is to formulate a quadratic subproblem based on the quadratic approximation of the Lagrangian function, where nonlinear constraints are linearized and bound constraints are expressed as inequality constraints. The implementation used consists of three main steps:

1) Update the Hessian matrix of the Lagrangian function. At every iteration, a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function is performed using the Broyden-Fletcher-Goldfarb and Shanno (BFGS) methods.[29]

2) Solve the quadratic programming problem. The quadratic programming subproblem is obtained by linearizing the nonlinear constraints and, in the implementation used, is solved using an active set strategy. The solution procedure involves two different phases. The first phase requires the calculation of a feasible point, whereas the second phase involves the generation of a sequence of points that converges to the solution.

3) Conduct a line search and merit function calculation. The step length is determined to produce a suitable decrease in the merit function.

The SQP optimization is applied using the algorithms implemented in MATLAB 5.3.[30] The SQP optimization algorithms can be penalized by the presence of an integer design variable and by irregular and opposite behaviors of the mean and the maximum forces. In this case, the SQP methods could find, as the optimum point, a local minimum instead of the global minimum. Like in many others gradient-based optimization methods, the optimal design could depend on the initial design.

Genetic Algorithm

The process of evolution in biological population motivates this algorithm. The evolutionary process allows an increase of the population surviving capabilities because, during the process, the genetic information stored in chromosomal strings (DNA) evolves over generations to adapt favorably the population.[31] This approach is performed by defining a fitness function or a measure that indicates the goodness of a member of the population in a given generation during the evolution process.

The genetic algorithms allow finding the maximum points of the fitness function, and this characteristic comes directly from the definition of evolution. For this reason, the objective function, defined earlier in Eq. (2), must be changed in sign and, because the fitness function $F(\text{DNA})$ is the only way to evaluate the performance of every member of the population, the design constraints must be indirectly considered using penalty functions during the fitness evaluation. The penalty function definition is often a difficult task because it could modify the convergence rate and the global performances. In the genetic algorithm, gradient computation is not required, which offers an important advantage during the optimization process. Consequently, the presence of integer variables and the absence of derivable functions do not represent a difficulty yet.

In the present work, a single binary string 51 bits long codifies the genetic information of any member of the population in a given generation. In particular, three groups, each one composed of 16 bits, are used for the angle element thickness, for the angle element position, and for the other panels thickness, respectively, whereas the remaining 3 bits are used for the vertical rivet codification. The codification of the vertical rivet number is a little bit complex. This design variable can assume only integer values between 4 and 8, according to the optimization space domain. A 3-bit codification allows the vertical rivets number to assume all of the integer values between 1 and 8. A specific penalty function gives a very low fitness function to the members of population that have 1, 2, or 3 vertical rivets to obtain premature extinction of this kind of population member. On the other hand, with the adopted strategy, a certain number of population members have low fitness values, especially in the first phases of the evolutionary problem, and the convergence rate can be low.

The introduction of the penalty functions into the objective function expression gives a new fitness function, defined as

$$\text{obj} = v_1 \cdot v_2 \cdot v_3 \cdot \left\{\left(\frac{F_{\text{mean}}(\boldsymbol{x})}{F_{\max}(\boldsymbol{x})}\right) + K \cdot \left(\frac{F_{\text{mean}}(\boldsymbol{x})}{\text{mass}(\boldsymbol{x})}\right)\right\} \tag{3}$$

where v_1, v_2, and v_3 describe the maximum force, the mean force, and the vertical rivets number constraints, respectively.

The genetic algorithm used in the present investigation is the one designed by Andrew Potvin in 1993 and implemented in MATLAB 5.3.[30] The algorithm uses only three basic genetic operations, selection, crossover, and mutation, but the performances seem suitable enough to solve the considered problem.

Results

Different optimizations were performed changing the constraint values and the value of the weighting parameter K into the objective function. Only the two most interesting solutions are described here.

First Optimization

The first optimization was obtained making $K = 0$, to search the optimal solution characterized by the maximum crush force efficiency value. In this way, the second term of the objective function, representing the mass specific absorbed energy, was not considered. The acceptable value for the maximum force was fixed at 120 kN, which is greater than the value returned by the initial design, which allowed a search for the optimal configuration in a large space of the defined domain. The mean force was lower constrained to 25 kN, that is, a value close to the initial configuration value. Both the SQP optimization and the genetic algorithm were used to find the optimal solution.

Several SQP optimizations were performed changing the starting points and finding, as the optimal point, the configuration described by the following values of the design variables: angle elements thickness, 1.245 mm; angle elements position, 35 mm; other panels thickness, 0.812 mm; and vertical rivets number, 6.88. In the final design, the vertical rivets number, considered as continuous variable during the optimization, was rounded to the nearest integer, 7. When started from the initial configuration, the SQP algorithm evaluated the objective function in 107 different points using the behavior responses given by the neural networks system.

With regard to the genetic algorithm, the first population was generated randomly. The optimization procedure used populations of 200 members, and the crossover and mutation probability were fixed to 0.7 and 0.03, respectively. The convergence of the optimization procedure was obtained after 42 generations and required 8400 fitness function evaluations. The optimal solution showed an integer

Table 4 Outputs of the two optimizations compared to the PAMCRASH analysis results

Parameter	Neural networks	PAMCRASH	Difference, %
First optimization			
Maximum force, kN	78.614	77.189	1.8
Mean force, kN	47.374	45.194	4.8
Mass, kg	0.530	0.530	——
Specific absorbed energy, kJ/kg	7.508	7.163	4.8
Crush force efficiency	0.602	0.590	2
Second optimization			
Maximum force, kN	49.084	46.289	6
Mean force, kN	26.050	25.795	0.9
Mass, kg	0.449	0.449	——
Specific absorbed energy, kJ/kg	4.870	4.824	0.9
Crush force efficiency	0.523	0.557	6.1

Table 5 Geometry and outputs of the two optimal configurations compared to the initial configuration

Parameter	Initial configuration	First optimal configuration	Second optimal configuration
Angular elements thickness, mm	0.81	1.25	0.82
Angular elements position, mm	40	36	37
Panels thickness, mm	0.81	0.79	0.72
No. of intermediate vertical rivets	8	7	6
Maximum force, kN	51.45	77.19	46.29
Mean force, kN	24.29	45.19	25.79
Mass, kg	0.489	0.530	0.449
Specific absorbed energy, kJ/kg	4.172	7.16	4.824
Crush force efficiency	0.47	0.59	0.56

number of vertical rivets and was very similar to the one obtained by the SQP algorithm: angle elements thickness, 1.25 mm; angle elements position, 36 mm; other panels thickness, 0.791 mm; and vertical rivets number, 7.

The obtained solutions of both the SQP and the genetic algorithm are almost coincident, confirming the reliability of both the solutions.

The obtained solution was verified by means of a finite element analysis using PAMCRASH. The outputs of the optimization are compared to the results of the PAMCRASH analysis in Table 4, where it is possible to see that the maximum difference is obtained on the mean force and is equal to 4.8%.

Although a direct comparison between initial and final solutions does not seem particularly important, because the maximum force value of the final configuration is greater than that of the initial configuration, it appears that the optimization procedure allows an increase of the crash efficiency equal to 25%. Indeed, the numerical simulation of the initial configuration returned a crush force efficiency equal to 0.47, whereas the numerical simulation of the optimized solution returned a crush force efficiency equal to 0.59, as reported in Table 5.

An important consideration, with regard to the neural network system capabilities, can be pointed out. The obtained optimal configuration shows 7 vertical rivets, whereas the neural networks are trained using only cases with 4, 6, and 8 intermediate vertical rivets, thus showing the generalization capability of the neural networks. At the same time, the opportunity to interpolate the behavior of complex phenomena also under the presence of integer variables, such as the rivet number, points out the general capabilities of the developed procedure.

Second Optimization

A second optimization was obtained choosing a value for K so that the mass specific absorbed energy also was taken into consideration and both the two objective function terms were of the same order of magnitude. As acceptable maximum force, the value of 50 kN was considered, that is, a value slightly below the initial configuration maximum force, to obtain a solution that has a lower maximum force. The mean force was lowered, constrained to 25 kN, as in the first optimization. This choice of the constraint values made it possible to have a direct comparison between the initial and the optimal configurations.

Both the SQP gradient optimization and the genetic algorithm were used to find the constrained solution. Several SQP optimizations were performed changing the starting points to verify the global nature of the final optimal point. The SQP algorithm found, as the optimal solution, the configuration characterized by the following values of the design variables: angle elements thickness, 0.82 mm; angle elements position, 35 mm; other panels thickness, 0.71 mm; and vertical rivets number, 6.349. Even in this case, the vertical rivets number was rounded to the nearest integer, 6. When started from the initial configuration, the SQP algorithm evaluated the objective function in 111 different points using the behavior responses given by the neural networks system.

With regard to the genetic algorithm, the first population was generated randomly. The optimization procedure used populations of 200 members, and the crossover and mutation probability were fixed to 0.7 and 0.03, respectively, as in the first optimization. The convergence was reached after 62 generations and required 12,400 fitness function evaluations. Even in this case, the obtained solution showed an integer number of vertical rivets and was very close to the one obtained by the SQP algorithm: angle elements thickness, 0.82 mm; angle elements position, 37 mm; other panels thickness, 0.72 mm; and vertical rivets number, 6. The obtained solutions of both SQP and genetic algorithm are almost coincident, confirming the reliability of both optimal designs.

Even in this case, as reported in Table 4, the obtained solution was verified by means of a finite element analysis using PAMCRASH. A direct comparison between the initial configuration and the founded optimal configuration, reported in Table 5, shows the following points:

1) There is an increase of the mean force equal to 6% and a decrease of the maximum force equal to 10%.

2) There is an increase of the crush force efficiency equal to 18%. Indeed, the finite element model of the starting configuration returns a crush force efficiency equal to 0.47, whereas the crush force efficiency of the final configuration, obtained by the finite element analysis, is equal to 0.56.

3) There is a decrease of the mass equal to 8%. The weight reduction of the whole subfloor structure could be even more considerable, considering that the studied intersection elements were present in quite a number and considering that the panels were disposed along all of the subfloor structure and their thickness was reduced from 0.81 to 0.72 mm.

Another important aspect to take into consideration is the computational time necessary for the adopted procedure. In this application, to find the optimal configuration of the subfloor intersection element, 45 finite element analyses were performed. Consequently, the total CPU time required was about 416 h because any analysis requires about 9.5 h on a V-Class Hewlett–Packard 4 PARisc CPU parallel computer, whereas all of the neural networks training processes and all of the optimization runs are comparable to a single finite elements analysis. In comparison, a direct optimization that uses only finite element analyses to evaluate the objective function and uses an SQP algorithm for the optimum search would have required about 110 different simulations, which means about 1045 h of CPU time.

This methodology is also profitable to allow more optimization processes to run using different methods and changing both constraints or objective functions, without any other numerical simulations and, consequently, without any increase of the requested CPU time.

Conclusions

This paper proposes a possible procedure for the structural optimization under crashworthiness requirements. The total time

required for the optimization is reduced to become acceptable by replacing the finite element analyses with the response surfaces of the problem, obtained by means of MLP neural networks. They can be considered as black boxes that are able to learn a certain number of examples combining every input with the corresponding output and that are also capable of generalization, giving back an acceptable output when an unknown input is received. The use of accurately trained neural networks makes it possible to introduce, during the optimization, constraint functions not limited to specific structural response indices, that is, mean and maximum crash force, but also on the whole load-time response curve.

The approach proposed here offers the advantage of a complete separation between the system modeling and optimization problems. Indeed, the choice to afford a local optimization, where only a component of a whole structure is optimized, is only the first step of the research, aimed at the complete optimization of the subfloor structure. The procedure described here is very general, allowing the possibility to run more optimization processes, changing the minimization algorithms and the constraints values. It is also possible to change the objective function formulation, without any other numerical simulations, or even to start a multiobjective optimization in which a vector containing more objectives is defined and a Pareto optimality solution is reached.

The application of the proposed procedure to the intersection element of a typical aluminum alloy helicopter subfloor required the generation of 45 examples, of which 36 were used during the neural networks learning phase and the other 9 as verification examples. On the other hand, if the starting point were to be chosen as the initial configuration, an SQP optimization algorithm would have required the evaluation of about 110 different configurations, whereas a genetic algorithm would have required about 10,000 different configurations. The use of neural networks seems a promising means to overcome the main difficulty typical of the optimization problems under crashworthiness requirements, represented by the high number of complete finite element simulations necessary to reach a suitable optimum. For the application example reported here, the rotorcraft subfloor intersection elements are of concern. The procedure presented allowed an increase of the mean force equal to 6%, a decrease of the maximum force equal to 10%, an increase of the crush force efficiency equal to 18%, and a decrease of mass equal to 8%.

The next step of the research will be a global optimization of the subfloor structure, considering both size (thickness, number of rivets, etc.) and topological (relative positions of the parts) design variables of the structural elements, aimed at the maximization of the global crashworthiness performances.

References

[1]"Light Fixed Wing and Rotary Wing Aircraft Crash Resistance," MIL-STD-1290A (AV), Department of Defense, Washington, DC, 1988.

[2]"Aircraft Crash Survival Design Guide," US Army Aviation System COMmand. TR 89-D-22A-E, Vol. I–IV, 1989.

[3]Giavotto, V., Caprile, C., and Sala, G., "The Design of Helicopter Crashworthiness; Energy Absorption of Aircraft Structures as an Aspect of the Crashworthiness," *Proceedings of the AGARD 66th Meeting of the Structures and Material Panel*, AGARD CP-443, Luxembourg, 1988; pp. 6.1–6.9.

[4]Kindervater, C., Kohlgruber, D., and Johnson, A., "Composite Vehicle Structural Crashworthiness: a Status of Design Methodology and Numerical Simulation Techniques," *Proceedings of the International Crashworthiness Conference*, edited by Chirwa, E. C. and Viano, D. C., Dearborn, Michigan, USA; 1998, pp. 444–460.

[5]Haftka, R. T., and Gürdal, Z., *Elements of Structural Optimization*, 3rd ed., Kluwer Academic, Dordrecht, The Netherlands, 1992.

[6]Bindolino, G., Mantegazza, P., and Ricci, S., "Integrated Servostructural Optimization in the Design of Aerospace Systems," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 167–175.

[7]Etman, L. F. P., Adrieans, J. M. T. A., Van Slagmaat, M. T. P., and Schoofs, A. J. G., "Crashworthiness Design Optimization Using Multipoint Sequenzial Linear Programming," *Structural Optimization*, Vol. 12, No. 4, 1996, pp. 222–228.

[8]Hajela, P., and Lee, E., "Topological Optimization of Rotorcraft Subfloor Structures for Crasworthiness Consideration," *Computers and Structures*, Vol. 64, No. 2, 1997, pp. 65–76.

[9]Lanzi, L., "Modellazione mediante Reti Neuronali ed Ottimizzazione Multiobiettivo di Strutture Elicotteristiche per la Sicurezza Passiva," M.S. Thesis, Dept. of Aerospace Engineering, Politecnico di Milano, Milan, Oct. 2000 (in Italian).

[10]"Metallic Material and Elements for Aerospace Vehicle Structures," MIL-HDBK-5, Dept. of Defense, Washington, DC, 1983, pp. 3.63–3.92.

[11]Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1982, p. 735.

[12]"PAMCRASH Users Manuals, Pam Generis Reference Manual and Pamview Reference Manual," Ver. 2000, ESI/PSI, Inc., Paris, March 2000.

[13]Blazinsky, T. Z., *Materials at High Strain Rates*, Elsevier Applied Science, London, 1987, pp. 133–185.

[14]Bisagni, C., "Energy Absorption of Riveted Structures," *International Journal of Crashworthiness*, Vol. 4, No. 2, 1999, pp. 199–212.

[15]Juanicotena, A., and Malherbe, B., "On the Modelling of Rivets Joints by Beam Type Spring Element in RADIOSS CRASH Code," *Proceedings of the International RADIOSS Users Conference and International RADIOSS Users Meeting*, MECALOG, Sophia Antiopolis, France; TP1999-133, 1999, pp. 113.1–113.10.

[16]Langrand, B., Patronelli, L., Deletombe, E., Markiewicz, E., and Drazetic, P., "FE Database for Riveted Joint Models and Airframe Crashworthiness," *Proceedings of the International Crashworthiness Conference, ICRASH2000*, edited by Chirwa, E. C. and Viano, D. C., London, UK; 2000, pp. 663–674.

[17]Bisagni, C., "Crashworthiness of Helicopter Subfloor Structural Components," *Aircraft Engineering and Aerospace Technology*, Vol. 7, No. 1, 1999, pp. 6–11.

[18]McCulloch, W. S., and Pitts, W. H., "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics, Selected Papers on Optical Neural Networks*, Vol. 96, No. 5 1943, pp. 115–133.

[19]Cammarata, S., *Reti Neuronali: dal Perceptron alle Reti Caotiche e Neuro-Fuzzy*, Etas libri, Milan, 1997 (in Italian).

[20]Rosemnblatt, F., *Principles of Neurodynamics*, Spartan, Washington, DC, 1961 p. 181.

[21]McClelland, J., Rumelhart, D., and the Parallel Distributed Processing Research Group, *Parallel Distributed Processing, Foundations*, Vol. 1, MIT Press, Cambridge, MA, 1986 p. 576.

[22]McClelland, J., Rumelhart, D., and the Parallel Distributed Processing Group, *Parallel Distributed Processing, Psychological and Biological Model*, Vol. 2, MIT Press, Cambridge, MA, 1986 pp. 216–268.

[23]Demuth, H., and Beale, M., "Neural Network TOOLBOX," User's Guide, MATLAB 5.3, MathWorks, Natick, MA, Jan. 1999.

[24]Rummelhart, D. F., Hinton, G. E., and William, R. J., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318–362.

[25]Hagan, M. T., and Menhay, M., "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, 1994, pp. 989–993.

[26]Greenman, R. M., and Roth, K. R., "Minimizing Computational Data Requirements for Multi-Element Airfoils Using Neural Networks," *Journal of Aircraft*, Vol. 36, No. 5, 1999, pp. 777–784.

[27]Fletcher, R., "Practical Methods of Optimization," *Constrained Optimization*, Vol. 2, Wiley, New York, 1980 p. 450.

[28]Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, London, 1981 p. 401.

[29]Powell, M. J. D., "A Fast Algorithms for Nonlinearly Constrained Optimization Calculations," *Numerical Analysis*, Vol. 630, Lecture Notes in Mathematics, Springer-Verlag, Hannover, Germany, 1978 pp. 144–157.

[30]"Optimization Toolbox," User's Guide, MATLAB 5.3, MathWorks, Natick, MA, Jan. 1999.

[31]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, London, 1989 p. 412.