

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Integration of MATLAB and ANSYS for Advanced Analysis of Vehicle Structures

A. Gauchía, B.L. Boada, M.J.L. Boada and V. Díaz

Additional information is available at the end of the chapter

1. Introduction

During vehicle structure design several parameters have to be considered, not only to pass vehicle approval tests but also to manufacture an efficient vehicle. Therefore, initial gross designs have to be optimized in order to reach the best compromise between two main antagonistic parameters: vehicle stiffness and weight. Optimization has proved to be a powerful tool in order to achieve the best possible design.

In the engineering field several programs, focused in vehicle structure analysis and based on the finite element method (FEM), are available (i.e. ANSYS, Abaqus, etc). Some of these programs allow the user to perform optimization algorithms of the vehicle structure. Even though finite element analysis of structures can also be done with MatLab it is not a FEM-focused software. However, MatLab is a powerful program that provides not only different optimization algorithms already implemented but the possibility to run a defined user optimization algorithm. Therefore, it seems that the best solution is to couple the FEM software (such as ANSYS) with MatLab in order to get the best from both while allowing total control by the user. In addition, due to the fact that the user can modify the optimization parameters as well as the vehicle geometry, the overall proposed loop has proven to be a successful tool in advanced mechanical engineering analysis. The main difference with respect to other researches that combine ANSYS and MatLab is that the proposed methodology is done completely automatically and that the developed optimization script does not require user intervention until the solution is found or stopping criteria is achieved.

The present chapter details how to carry out this coupling and how to manage the files between them. To clarify, a complex bus structure, shown in Figure 1, optimization loop is described as an example. The aim is to explain how to generate the needed files in order to establish an automatic loop that does not need user intervention. In addition, some

specific hints on ANSYS and MatLab programming are provided in order to achieve success in the final implementation.

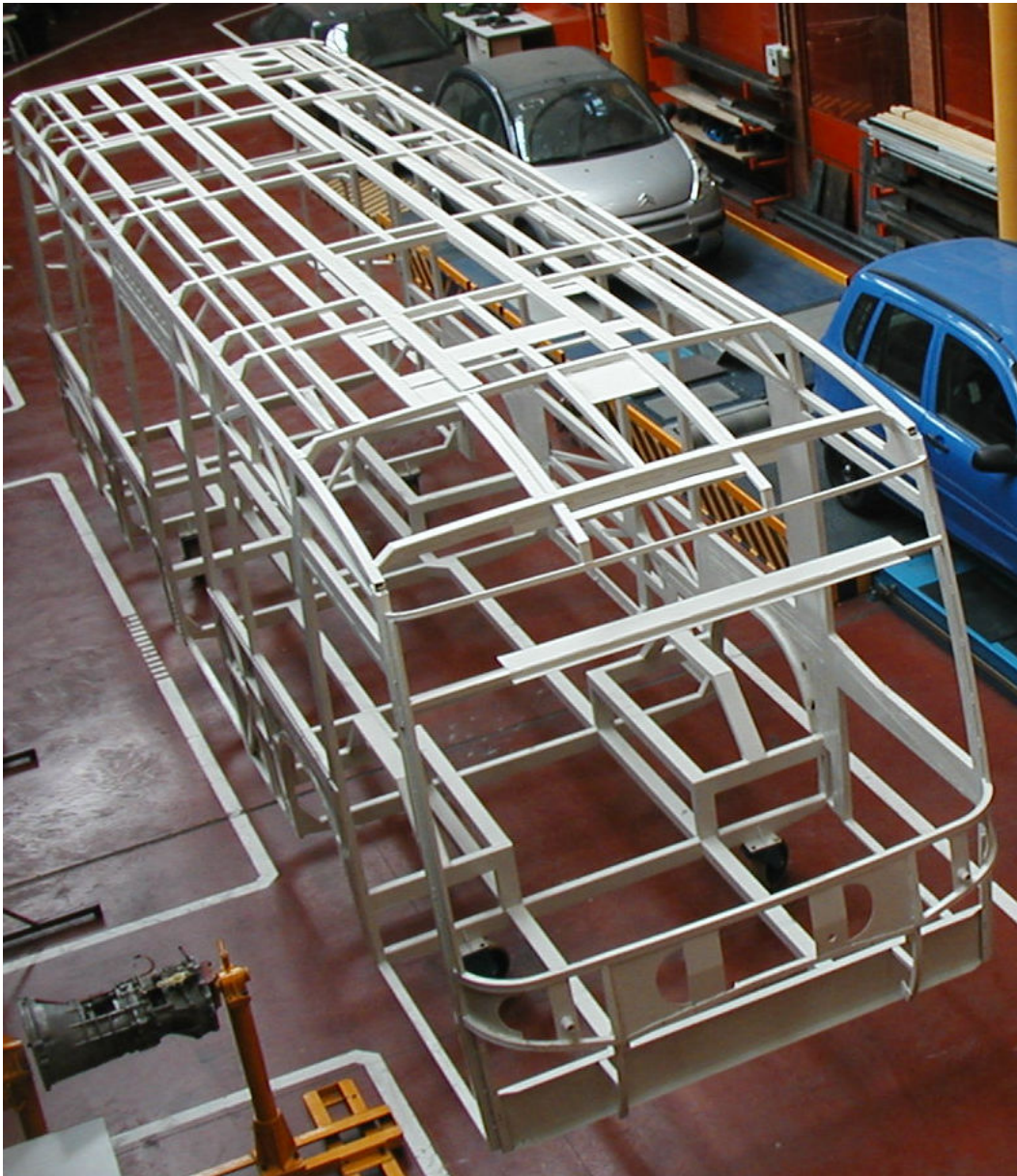


Figure 1. Real bus structure

The layout of the chapter is as follows. In section 2 the overall flow diagram is described. The aim of section 2 is to show the overall file flow which has been applied to optimize in torsion stiffness and vehicle weight a real urban bus structure. In section 3 a brief description of the bus structure model is provided as well as how to generate the output files. The genetic algorithm optimization is described in section 4. This section not only shows how the genetic algorithm is done but how to define the fitness function. In addition, this section describes and reveals how to call ANSYS from MatLab so that ANSYS is run under the operating system

(batch mode) automatically until the optimized value is achieved. Section 5 shows some of the obtained results to prove the effectiveness of the proposed advanced analysis. Finally, section 6 provides conclusions of the described methodology.

2. The overall flow diagram

The aim is to optimize a complex bus structure in weight and stiffness. The bus structure has to have a certain amount of bending and torsional stiffness, being torsion the most demanding effort [1]. Torsion loads appear during vehicle cornering, unequal bumps or potholes. A bus torsion stiff structure is desirable in terms of handling and lateral rollover dynamics [2, 3]. However, increasing vehicle torsion stiffness usually requires of adding beams to the structure, thus increasing vehicle weight. An increased vehicle sprung weight penalizes the vehicle lateral dynamics as more weight is transferred between the wheels of the same axle, thus reducing tire lateral grip. In addition, for the analyzed vehicle structure, the vehicle centre of gravity height has a massive influence in rollover. For the same torsion stiffness it is desirable that the heavier beams are placed as low as possible while being able to protect passenger from a possible rollover [4]. Therefore, for a given structure configuration and layout of beams, the optimization tool should provide the optimal structure that allows achieving a certain value of torsion stiffness by changing the thickness of the appropriate vehicle beams while keeping the vehicle weight to a minimum [5].

From the different optimization algorithms available in MatLab [6], genetic algorithms have been selected to optimize the bus structure. Genetic algorithms are being applied in many areas of the mechanical engineering field, having proved to be a suitable tool to optimize vehicle structure. In fact, genetic algorithms are particularly suitable for optimization in which a multidimensional global search with multiple local minimum is required. In addition, genetic algorithm is especially well-suited when the search space is not well known, being able to combine the best solutions yielding an even better one.

The bus structure analysis will be carried out with ANSYS [7], which is a finite element software. It is worth highlighting that any other finite element software (either licensed or opened source) may be used to be coupled with MatLab. The requirement to be fulfilled is that the software must allow programming the finite element model by means of a text file in order to be able to automate the proposed optimization methodology shown in Figure 2. The coupling between MatLab and the finite element software is done by means of line code which will be later detailed. For the particular case analyzed in this paper ANSYS has been used. The complete creation and solving of a structure may be done either by picking on the different menus or by programming in Ansys Parametric Design Language (APDL). The main advantage of creating and solving a model by means of APDL is that the model can be defined in terms of variables, thus creating a parametric model. In general, the variable that is employed to create a parametric model is beam thickness as it greatly influences vehicle stiffness and weight. Other variables such as the length of the beam could be considered, but modify the overall vehicle geometry. In terms of the optimization loop it is very convenient to create a

parametric model as an automatic loop can be created without user intervention. The values of the different variables will be changing according to the optimization loop until a minimum is reached. APDL allows, among other features, creation of files in text format to store the results of the analyzed model. These files will be used by the optimization tool in order to re-define better values for the variables of the vehicle structure.

Figure 2 resumes the optimization loop. In the depicted figure the output of each of the steps is also shown. It can be seen that the information between ANSYS and MatLab is based on text files which are overwritten in each loop. All of the created files must be placed on the same folder.

The optimization loop will flow through the following steps:

- The file "OptiBus.m" creates the file "parameters.inp" which stores the value of the thickness of the beams. This file is the one that runs the genetic algorithm, which needs to evaluate the fitness function.
- Once the file "parameters.inp" stores the assigned values of the thickness of the beams the finite element model can be run. The finite element model is programmed, by means of APDL, in a text file ("BusModel.txt"). This text file retrieves the values of the beam thickness from file "parameters.inp". Once the model is solved, the results ("Displacement.txt", "ReactionForce.txt", "Stress.txt" and "Weight.txt") are stored in different files. These results will be used to evaluate the fitness function, which represents the objective function that has to be minimized.
- Next, the genetic algorithm mutates the value of the thickness so as to achieve a minimum in the fitness function. The new values for the beam thickness are stored and overwritten in "parameters.inp".
- The loop will continue until a stopping criteria or minimum is achieved.

Although an optimization loop has been presented in this chapter as an example of how to couple MatLab and ANSYS, any other loop vehicle calculation could be performed.

It is worth highlighting that prior to any optimization, a sensitivity analysis of the vehicle structure must be performed [8]. The aim of the sensitivity analysis is to find the beams of the vehicle structure so that a change of beam thickness is highly sensitive to weight and torsion stiffness. Therefore, the optimization process will be focused on the beams whose change in beam thickness generates great changes on weight and torsion stiffness. Thus, the optimization loop will only be applied to certain beams of the bus structure. For the particular case analyzed in the current chapter, the main optimized beams belong to the bus chassis, as shown in Figure 3. The sensitivity analysis is performed by measuring the change of weight and torsion stiffness when 1 mm increase is applied to the beams of the vehicle structure. By computing the ratio variation of weight to variation of thickness (sensitivity to weight) and comparing this result for the different beams the most sensitive beams to weight can be selected. In addition, by computing the ratio of the variation of torsion stiffness to the variation of thickness (sensitivity to torsion stiffness) the most sensitive beams to torsion stiffness can be found. Due to the fact that the aim is to select the beams that are most sensitive to both parameters at the same time,

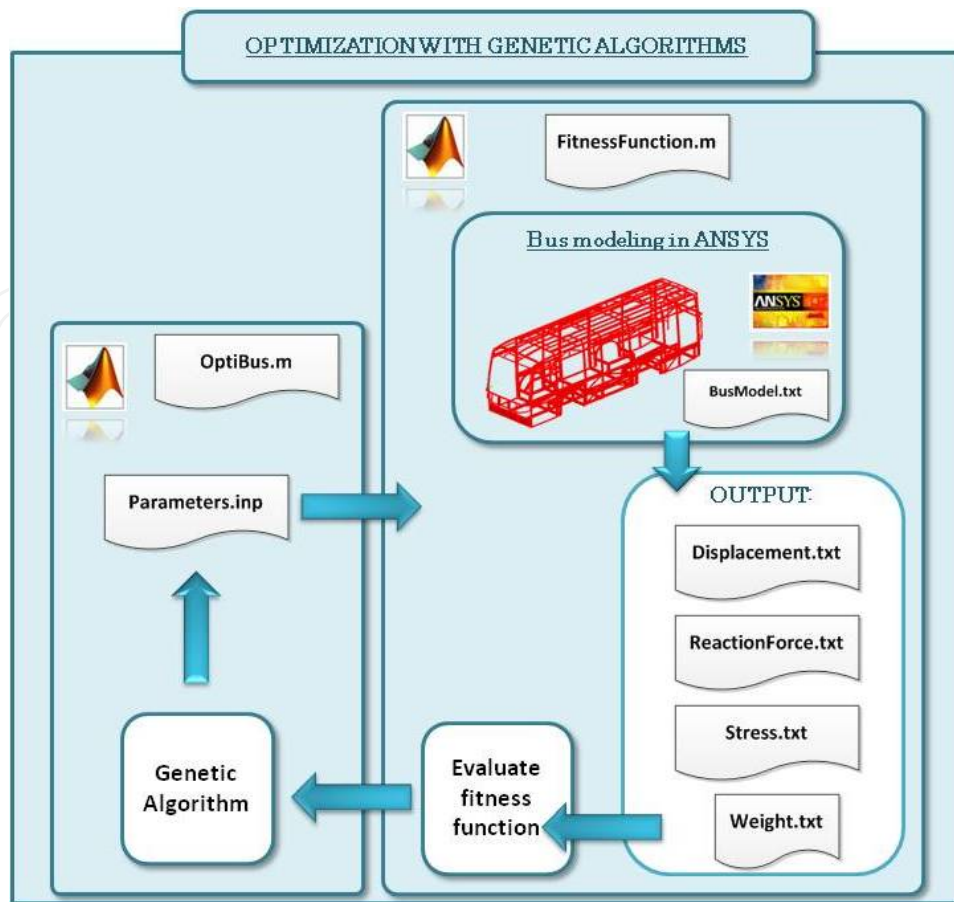


Figure 2. Optimization flow diagram

a ratio between the sensitivity to weight and the sensitivity to torsion stiffness may be defined. In addition, it is worth checking if the results of the sensitivity analysis are coherent with the expected. For the analyzed bus structure, it is expected that the beams placed at the chassis (lower part of the superstructure represented in Figure 1) are very sensitive to torsion stiffness. The reason is that these beams are the connection link between both bus axles and also to the superstructure.

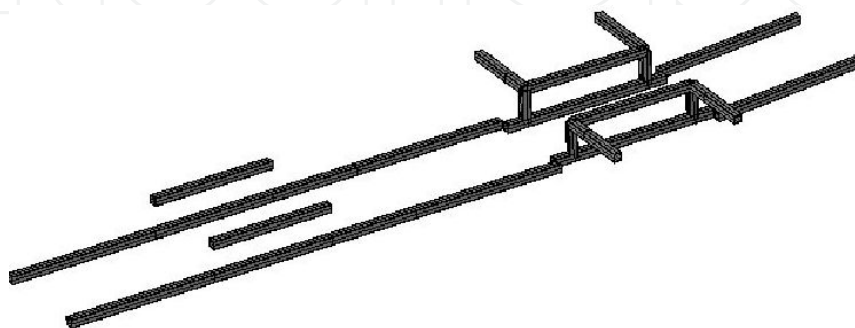


Figure 3. Beams of the bus chassis most sensitive to weight and torsion stiffness

From Figure 3 it can be highlighted that, as expected, the beams of the bus chassis have a great influence on weight and torsion stiffness. Due to the fact that input loads come from the wheels and these are transmitted, by means of suspension attachments, to the bus structure, it is expected that these beams will be heavier than other bus beams. Therefore, not only these beams will be more sensitive to bus structure torsion stiffness but also to overall bus weight.

3. Generating the bus model in ANSYS

Most of the finite element model programs allow the user to model the vehicle structure by means of a file which is written in a special code. Particularly, in the case of ANSYS the vehicle structure can be created either by picking on the various available options in the menu or by means of APDL programming. The authors propose for advance users to use the latter as it is the most suitable for parametrization. Parametric programming of the vehicle structure consists of defining variables instead of providing specific values to certain variables. Thus, in the present example it is more convenient to parametrize the thickness value of the vehicle beams. The variables that define this beam thickness will change until the final optimized value is achieved.

It is worth noting that the aim of the present chapter is not to provide detail knowledge of the optimization with genetic algorithms nor become skillful with finite element modeling, but to learn how to create the complete calculation progress by coupling ANSYS and MatLab. Therefore, in depth concepts of APDL programming are not explained in this chapter.

Modeling of a vehicle structure, and in general, of any engineering process by means of the finite element method, must follow three steps: Pre-processor, Solution and Post-processor. During the Pre-processor the geometry of the bus structure and material properties are defined. Once the geometry of the structure is generated the user must specify in the Solution step the loading condition and constraints. Finally, during the Post-processor results can be visualized.

3.1. Pre-processor

Regarding the Pre-processor, the bus structure is defined by means of beam elements. Prior to generate the beam elements the structure geometry is built with lines and keypoints. In addition, the user must define the cross section of all of the beams of the bus structure, which are 2445 beams. During the meshing process each beam is defined with its corresponding cross section. More than 50 different beam cross sections were defined. Due to the fact that the entire described bus model is done by programming, instead of providing a specific value for the thickness a variable ("T") was given. In fact, each of the thicknesses of the each beam is defined as T_i (view Figure 4).

The bus beam cross sections are made of hollow rectangular steel beams. The material of the bus structure is steel with the material properties shown in Table 1.

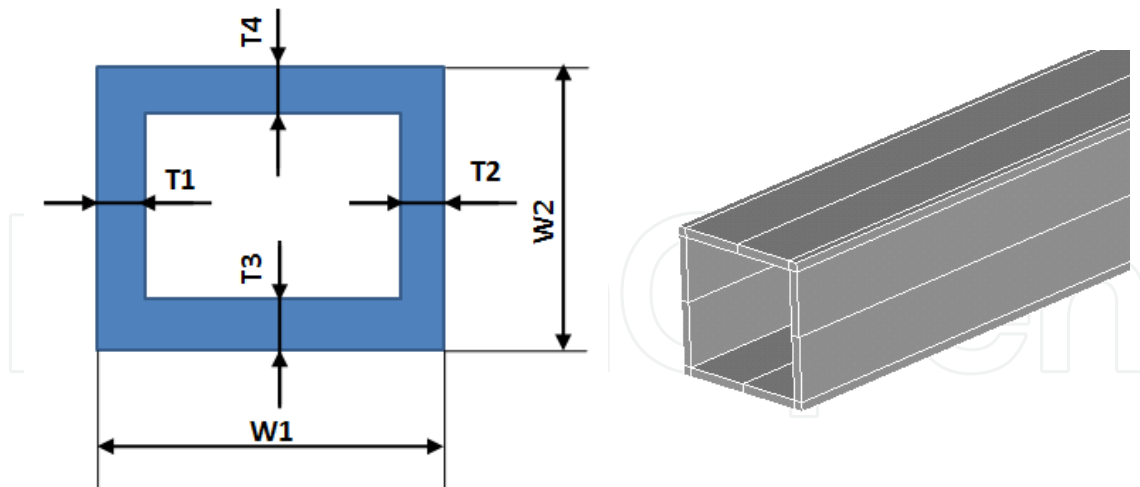


Figure 4. A parametric beam cross section (left) and the corresponding beam of the bus structure (right).

Material	Modeled as	Elasticity modulus (GPa)	Poisson ratio	Density (kg/m ³)
Steel	Linear elastic	200	0.3	7850

Table 1. Bus structure material properties

The bus structure is modeled as linear elastic. All of the applied loads are intended to be computed in the linear range of the material. The reader must recall that the loading conditions must provide the bus structure torsion stiffness and the weight, thus, none of these loads requires information regarding its plastic behavior. Figure 5 depicts the geometry of the complete bus structure.

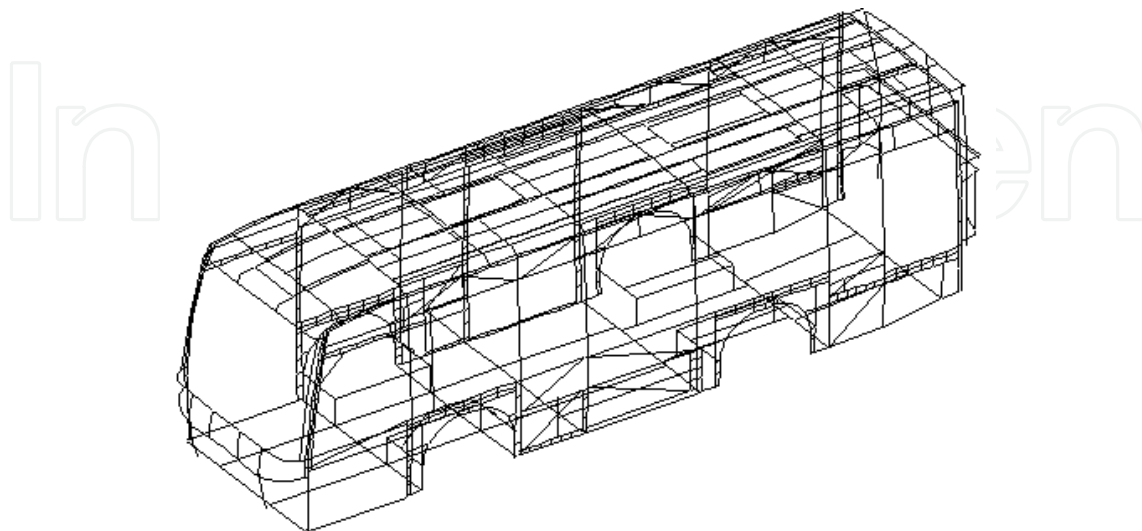


Figure 5. Finite element of the bus structure.

The complete bus structure does not include side panels nor windows, seats, etc. However, they may be included in the designer needs them to be considered. Figure 6 shows an example of how the bus structure geometry by means of APDL programming, is defined. It can be observed that a file “parameters.inp” is needed as an input to generate the bus structure as it contains the values of the thickness of the beams. Afterwards, the type of element is created (Beam 44) and the material properties are provided. Next, the geometry of each of the cross sections is generated. It can be observed that cross section 1 thickness has been parametrized with T1, T2, T3 and T4 (see Figure 4). The “parameters.inp” just assigns values for each thickness variable by stating for example that T1=1.5. The file “parameters.inp” is the output of the genetic algorithm implemented in MatLab, which will be described later on the chapter.

```

/inp,parameters,inp

/PREP7

/TITLE,BUS-4: BUS STRUCTURE OPTIMIZATION

ET,3,BEAM44
ET,4,MPC184
KEYOPT,4,1,1
MP,EX,1,210000
MP,NUXY,1,0.3
MP,DENS,1,7.85e-9

SECTYPE, 1, BEAM, HREC, 6P303015, 0
SECOFFSET, USER, 15, 15
SECDATA,30,30,T1,T2,T3,T4,0,0,0,0

SECTYPE, 2, BEAM, HREC, 252515, 0
SECOFFSET, CENT
SECDATA,25,25,1.5,1.5,1.5,1.5,0,0,0,0

```

Figure 6. APDL programming to generate the bus structure (“Busmodel.txt”)

In Figure 7 an example of the “parameters.inp” is shown.

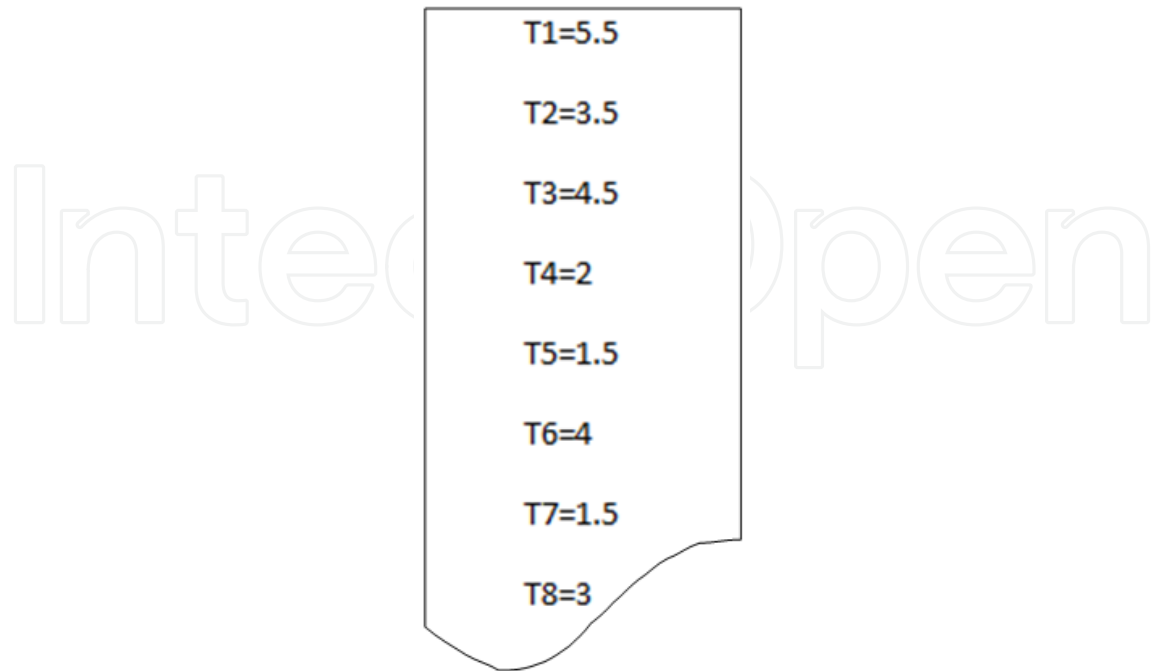


Figure 7. Example of value of thickness parameters (“parameters.inp”)

3.2. Solution

During the solution step the loading condition and constraints must be applied. In this particular case, the aim is to optimize the vehicle structure in torsion stiffness and weight. Therefore, two loading conditions must be defined:

- Load condition to compute bus structure weight: Gravity ($g=9810 \text{ N/mm}^2$) must be applied to the body structure and the four suspension supports must be constrained. It must be highlighted that while at the front axle there is one suspension support per wheel, at the rear axle there are two suspension attachments per wheel, thus, the whole bus structure sprung mass is supported by six suspension supports. During the post-processor step, the sum of the reactions at these supports will provide the bus structure weight.
- Load condition to compute torsion stiffness: A specific vertical displacement (Δz) is applied at one of the suspension supports while the others supports are constrained. Torsion stiffness is computed by means of the following equation:

$$K_T = \frac{M_T}{\Delta\phi} = \frac{F_z \cdot B}{\tan^{-1}\left(\frac{\Delta z}{B}\right)} \quad [\text{N}\cdot\text{m}/\text{deg}] \quad (1)$$

where F_z is the vertical force measured at the suspension support at which the vertical displacement Δz is applied and B is the wheeltrack. In addition to the loads, constraints must be applied to the model. In Figure 8 the loads and constraints applied to the bus structure for the torsion loading condition are depicted.

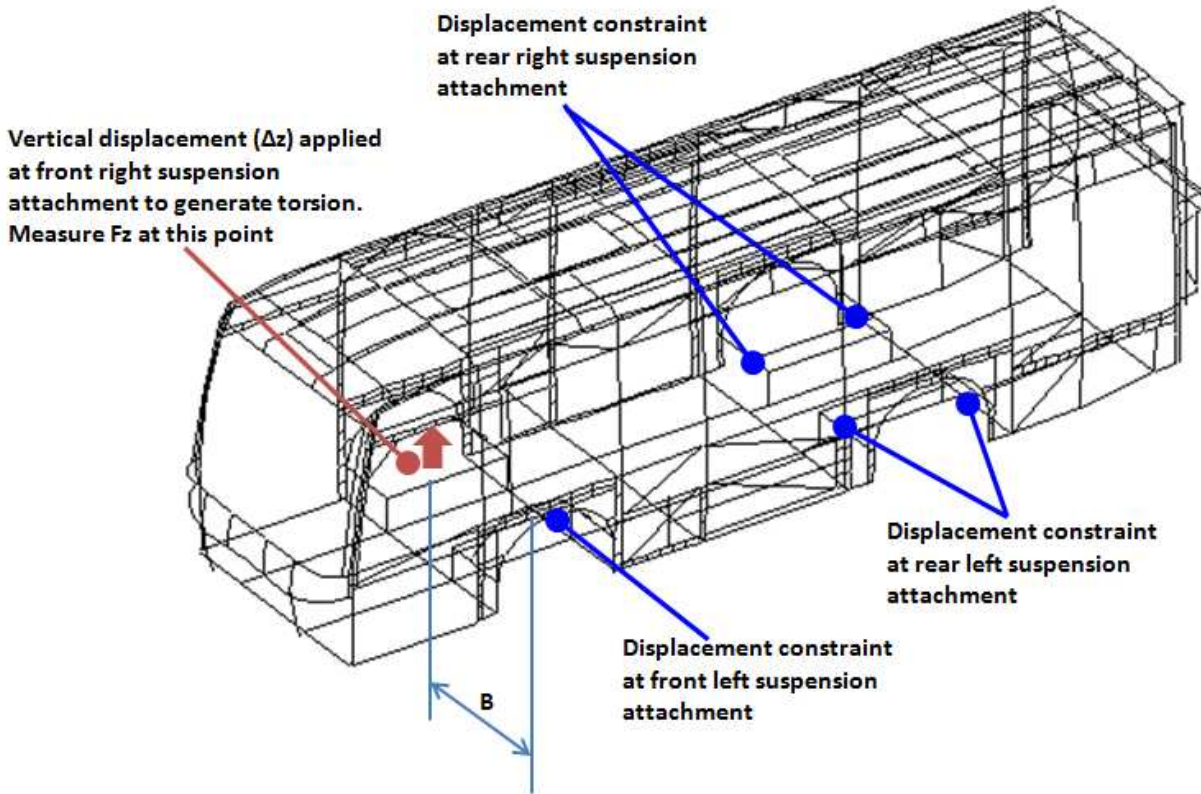


Figure 8. Constraints and loads for the torsion load case

The rotation angle generated in the structure (φ) due to the applied vertical displacement is measured by means of the vertical displacement (Δz) of the lifted suspension strut and the wheeltrack (B). It is worth highlighting that if a massive vertical displacement is applied to the bus structure it would reach permanent deformation, shown in Figure 9, which is a scenario not desired to be achieved. The reason is that the bus structure can be considered to behave as a torsion spring under torsion loads. Therefore, the value of the measured torsion stiffness should be measured during the elastic linear range of the bus structure material; otherwise, it would not represent the elastic bus structure performance correctly. As shown in Figure 9, for the applied vertical displacement in one of the suspension struts the maximum stress at the bus structure must be below the yield stress of the bus structure material, which in the particular example shown in this chapter must be below 250 MPa. Therefore, during the post-processor the maximum stress measured in the bus structure must be below the yield stress during the torsion load condition.

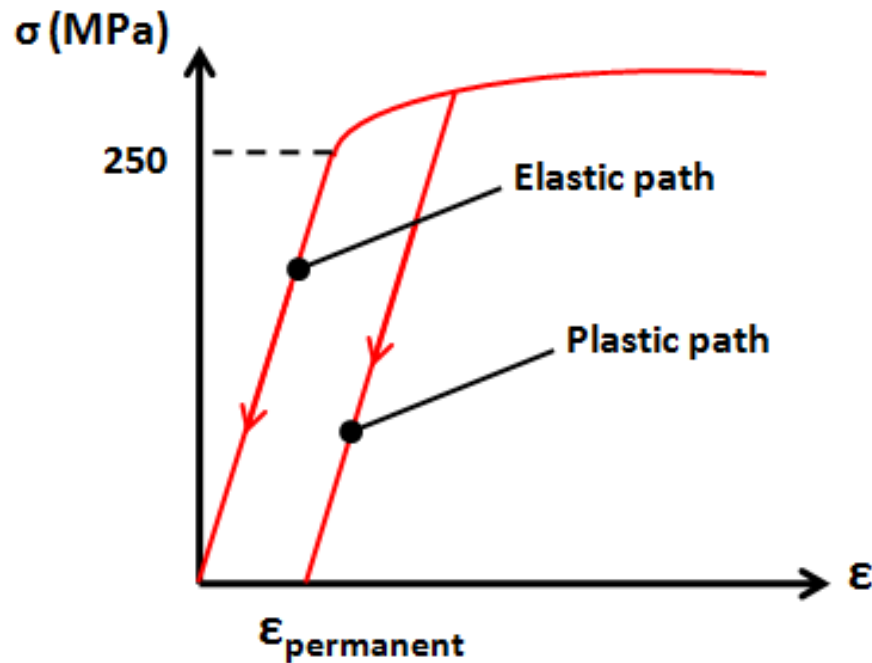


Figure 9. Bus material behaviour

Due to the fact that the value of the torsion stiffness is computed in the elastic range its value is independent of the applied vertical displacement. A bigger applied vertical displacement will provide the same value for the bus structure stiffness, as the value of the measured vertical force (F_z) will change to provide the same amount of stiffness. More complex models that account for a damping factor for the bus structure can be considered, but the frequency of the applied torsion loads is usually small, thus the torsion spring is the main activated component. Most of the vertical displacement is absorbed by the bus pneumatic suspension travel. Bigger torsion loads take place during very fast cornering or large potholes, being for the former mass longitudinal distribution an important parameter.

3.3. Post-processor

Finally, results can be viewed and stored in files for further usage. The value of the weight of the bus structure is done in the APDL file. The outputs are four files: "Displacement.txt", "ReactionForce.txt", "Stress.txt" and "Weight.txt". These files will be used to evaluate the fitness function. As an example, the programming in APDL in order to write the file "Weight.txt" is shown in Figure 10.

Similarly, by means of programming in APDL a file with the value of stress will be stored in a specific file ("Stress.txt"), as shown in Figure 11.

The Postprocessor allows viewing results by means of plots, as depicted in Figure 12, where von Mises stress is represented.

```

/POST1

*GET, P1, NODE, 2512, RF, FZ
*GET, P2, NODE, 2527, RF, FZ
*GET, P3, NODE, 621, RF, FZ
*GET, P4, NODE, 2612, RF, FZ
*GET, P5, NODE, 2623, RF, FZ
*GET, P6, NODE, 1052, RF, FZ

P=P1+P2+P3+P4+P5+P6

/OUT, Weight, txt
*VWRITE, P
(F15.5, '

```

Figure 10. Example of how to save variables and store them in a file ("Weight.txt")

```

/POST1

ETABLE, SMAXI, NMISC, 1
ETABLE, SMINI, NMISC, 2
ETABLE, SMAXJ, NMISC, 3
ETABLE, SMINJ, NMISC, 4

SMAX, SMAXVAI, SMAXI, SMAXJ, 1, 1
SMAX, SMAXVAJ, SMAXI, SMAXJ, 1, 1
SMAX, SMAXVAEL, SMAXVAI, SMAXVAJ, 1, 1

ESORT, SMAXVAEL
*GET, STRS, SORT, , MAX
PARSAV, ALL

/OUT, Stress, txt
*VWRITE, STRS
(F10.5, '
PARSAV, ALL

```

Figure 11. Example of how to save variables and store them in a file to create file "Stress.txt"

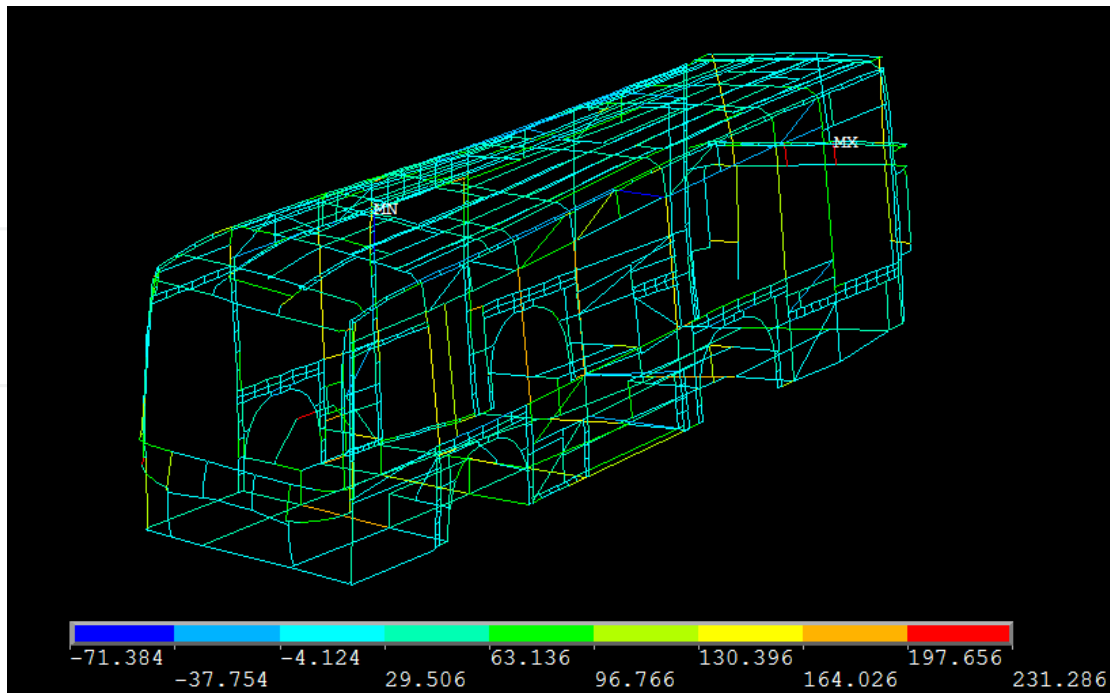


Figure 12. Postprocessing of the bus structure during a torsion rear right load

4. Optimization with genetic algorithms

Genetic algorithms (GA) have been used in this particular case because they are especially suited for multidimensional global search problems where the search space potentially contains the multiple local minimum. In addition, GA do not require having an extensive knowledge of the search space, thus, computing time can be reduced. Last but not least, GA finds better solutions by combining good solutions.

GA has proved to be a suitable tool to solve many optimization problems in the mechanical engineering field, for example, to find the optimum diameter of a rotor shaft as in [9]. In the field of structures it has also been successfully used to optimize truss structures [10, 11]. GA have also been applied for crack detection in shafts [12] as well as to estimate the overall elastic stiffness [13]. In the field of mechanism it has been used for the synthesis of compliant mechanisms [14, 15].

Genetic algorithms consist of three main steps: selection, genetic operation and replacement [16, 17]. The cycle is repeated until a termination criterion is achieved. The criterion can be defined in terms of the maximum number of cycles, the amount of variation of individuals between different generations or of a predefined value of fitness. The best value of the parameters achieved during search will be the final result of the genetic algorithm loop. The function that quantifies the evolutionary mechanism is the fitness function. The fitness function is the function that the user desires to minimize. Therefore, it represents the main link between

the real system (for this particular case, the bus structure) and the genetic algorithm. The two main parameters of a genetic algorithm are the fitness function and the search space.

The search space is the area in which the genetic algorithm will search the solution that best minimizes the fitness function. The genetic algorithm is able to search in the whole search space but this would increase computational effort and time. Thus, a specific search space in which the best solution can be found is provided to the genetic algorithm. For the particular analyzed case, the criteria to specify the genetic algorithm the search space can be based, for example, on minimum allowable thickness to weld two beams or in the minimum standard available beam thickness, etc. For the bus structure, the search space is restricted to values of thickness ranging from 0,25 mm to 9 mm. These values have been selected because they represent standard common beam cross section used in busses.

The fitness function is a function that must be defined by the user. The aim is to minimize this function. Therefore, for the particular case analyzed in this chapter, this function must be proportional to the bus structure weight. In addition, it must be inversely proportional to bus structure torsion stiffness. From equation 1 it can be found that torsion stiffness is proportional to the measured vertical force (F_z). In order to reduce computation time and taking into account that the fitness function is minimized; it must be inversely proportional to the measured vertical force (F_z) at the node at which the vertical displacement is applied in order to maximize torsion stiffness.

A simple genetic algorithm is shown in Figure 13. This genetic algorithm has been implemented in MatLab by means of an integrated toolbox.

One of the most important steps is to define the fitness function. The fitness function will consider the weight, the stress level and the reaction force measured during the torsion stiffness computation for a given applied vertical displacement. This reaction force is an indirect measurement of how stiff a chassis is, as explained above. For the particular example the fitness function defined by the authors is the following one:

$$\begin{aligned}
 & \text{if } \left\{ \left(\frac{\sigma_{\max}}{S_{\text{steel}}} < 1 \right) \& \left(\frac{W}{W_o} < 1 \right) \& \left(\frac{F_{zo}}{F_z} < 1 \right) \right\} \\
 & \quad ff3 = \left(\frac{\sigma_{\max}}{S_{\text{steel}}} \right) + \left(\frac{W}{W_o} \right)^4 + \left(\frac{F_{zo}}{F_z} \right)^2 \\
 & \quad \text{else } ff3 = \text{infinity}
 \end{aligned} \tag{2}$$

where S_{steel} is the steel yield stress (250 MPa), σ_{\max} is the maximum stress measured during the virtual torsion test, W is the weight of the bus structure for each loop, W_o is the initial value of the weight of the bus ($W_o=1580$ kg), F_z is the reaction force at the location at which the vertical displacement is applied in order to generate torsion in the bus structure for each loop and F_{zo} is a fixed value of 5500 N.

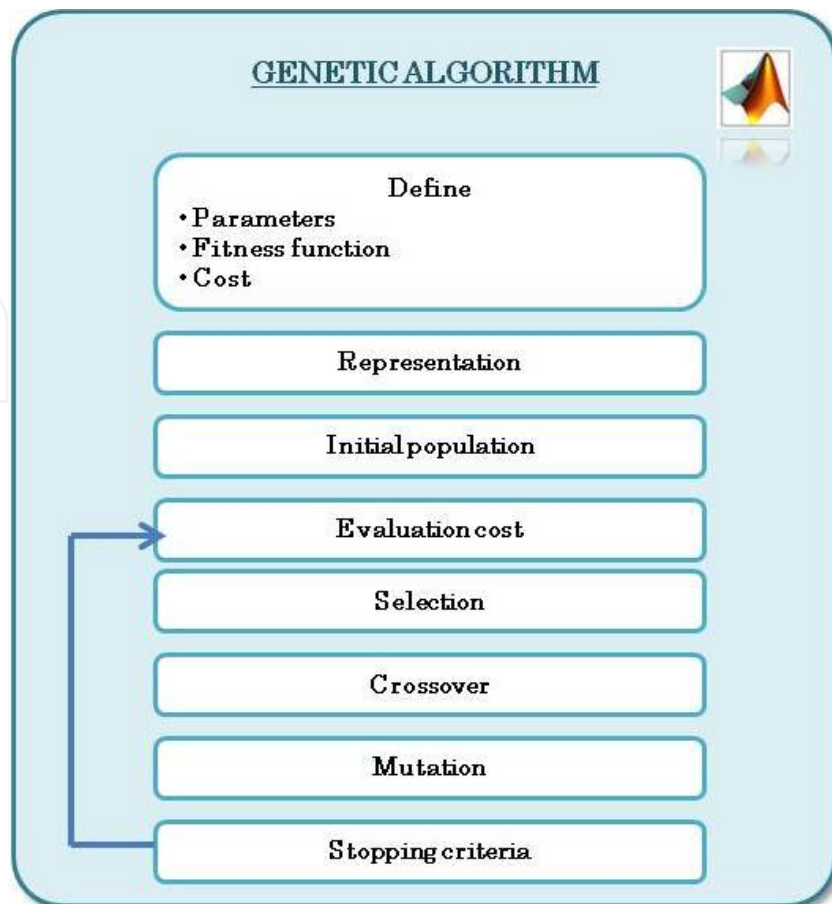


Figure 13. Genetic algorithm done in MatLab

The fitness function is defined in a MatLab file which is the one that will call ANSYS so that it runs in batch mode, that is, so that it runs under the operating system without intervention of the user. The file "fitnessfunction.m" stores the optimized parameters in the file "parameters.inp", executes ANSYS in batch mode and evaluates the fitness function.

The first part of "fitnessfunction.m" is depicted in Figure 14.

Programming of the fitness function, defined by equation (2), is shown in Figure 15.

During the first part the developed program stores the provided values of beam thickness from the GA ("OptiBus.m") to individual variables. Next, these individual variables are stored in the file "parameters.inp" which will be used to create the bus geometry. Afterwards, ANSYS is executed from "fitnessfunction.m" by means of the command specified in Figure 16.

If another type of finite element software is employed "C:\Program Files (x86)\Ansys Inc\.....\ANSYS.exe" must be substituted by the appropriate executable route file. In addition, it must be checked on the help tool how to run the finite element software in batch mode. Batch mode allows the software to be run under system without specifically opening it by providing an input text file in which the finite element model is programmed. For ANSYS, batch mode is run with the specific commands described in Figure 16. The third part of the code provided in

the figure specifies the file route that points to the input finite element model text file in which the finite element model is programmed. When using different finite element software the programming written in the input file will vary as it has to follow the appropriate commands for the specific employed finite element software.

Next, to evaluate the fitness function, the files "ReactionForce.txt", "Displacement.txt", "Stress.txt" and "Weight.txt" generated by "BusModel.txt" must be read. Thus, reading commands must be specified in "fitnessfunction.m", as shown in Figure 17.

In addition, another MatLab file will call the GA tool to which several parameters will be given. Other parameters regarding initial values of optimized parameters, its minimum and maximum values, etc. must be defined in order to correctly define the genetic algorithm. An example is shown in Figure 18.

```
function salud=fitnessfunction(T)

T1=abs(T(1));
T2=abs(T(2));
T3=abs(T(3));
T4=abs(T(4));
T5=abs(T(5));
T6=abs(T(6));
T7=abs(T(7));
T8=abs(T(8));

fid=fopen('parameters.inp','w+');
fprintf(fid,'T1=%f\n',T1);
fprintf(fid,'T2=%f\n',T2);
fprintf(fid,'T3=%f\n',T3);
fprintf(fid,'T4=%f\n',T4);
fprintf(fid,'T5=%f\n',T5);
fprintf(fid,'T6=%f\n',T6);
fprintf(fid,'T7=%f\n',T7);
fprintf(fid,'T8=%f\n',T8);
fclose(fid);
```

Figure 14. First part of file "fitnessfunction.m"

```

fid=fopen('ReactionForce.txt','r');
FR=fscanf(fid,'%f',[1,1]);
fclose(fid);

fid=fopen('Weight.txt','r');
P=fscanf(fid,'%f',[1,1]);
fclose(fid);

fid=fopen('Stress.txt','r');
STRS=fscanf(fid,'%f',[1,1]);
fclose(fid);

%Torsion stiffness in Nm/deg
%atan(5,1592)=0.179948 degrees
R=(FR*1.592)/0.179948;

if ((STRS/250)<1) && ((P/15508.826)<1) && ((5500/FR)<1)
    salud=((STRS/250)+((P/15508.826)^4)+((5500/FR)^2));
else
    salud=10000000000;
end

```

Figure 15. Programming of the fitness function in "fitnessfunction.m"

```

!"C:\Program Files (x86)\Ansys Inc\.....\ANSYS.exe" -b -p
ANSYSUL-i "C:\Documents and Settings\.....\ BusModel.txt"

```

Figure 16. Command to run ANSYS in batch mode from "fitnessfunction.m"

Additional values for options must be defined, such as the population size, etc. Please refer to MatLab help for further information. In Figure 18 it can be seen that there are 8 variables of thickness (T) and a minimum value for all of them has been set to 1 mm. The maximum value of each thickness is set to 8 mm. It is worth highlighting that all of the files generated either by the MatLab scripts and by ANSYS (by means of programming in APDL) must be placed on the same folder so as to be recognized.

In Figure 19 depicts an example of the options that the user must specify in order to set the genetic algorithm options.

```

fid=fopen('Reactionforce.txt','r');
FR=fscanf(fid,'%f',[1,1]);
fclose(fid);

fid=fopen('Displacement.txt','r');
DP=fscanf(fid,'%f',[1,1]);
fclose(fid);

fid=fopen('Weight.txt','r');
P=fscanf(fid,'%f',[1,1]);
fclose(fid);

fid=fopen('Stress.txt','r');
STRS=fscanf(fid,'%f',[1,1]);
fclose(fid);

```

Figure 17. Commands written in "fitnessfunction.m" to read the values of the files generated by the FEM "BusModel.txt"

```

function OptiBus

options = gaoptimset('PopInitRange',[1 1 1 1 1 1 1 1;8 8 8 8 8 8 8 8],'Generations',100);
[T,fval,output,population] = ga(@fitnessfunction, 8,options)

```

Figure 18. Defining the genetic algorithm parameters for optimization ("OptiBus.m")

```

options = gaoptimset('PopInitRange',[1 1 1 5 1 1 1 1;2 2 2 10 8 8 8 8],
'Generations';100;'PopulationSize';100;'EliteCount';2;'MigrationFraction';0.2;
'MigrationInterval';50;'PopulationType';'doubleVector';'StallGenLimit';Inf,
'SelectionFcn'; @selectionroulette;'PlotFcns';{@gplotbestf});
[T,fval,output,population] = ga(@fitnessfunction, 8,options);

```

Figure 19. Defining the options of the genetic algorithm ("OptiBus.m")

5. Results

In order to show the effectiveness of the proposed methodology this section shows some of the obtained results. Figure 20 shows the evolution of the fitness function until a minimum value of the fitness function was achieved. In the figure it can be seen that the initial values of the fitness function are very big. The reason is that the initial thickness of the bus beam cross sections are not optimized in weight, as expected. The genetic algorithm searches the search space and for the different feasible thickness values inside the defined search space the fitness function is evaluated. For the thickness values that provide lower values of the fitness function are combined between them in order to obtain an even better solution. Finally, an optimum is found, for the defined fitness function. It is worth noting that the user must define different fitness functions in order to see which one better suits the optimization which is being performed. A better minimum might be reached with a different defined fitness function. In addition, it must also be checked that the genetic algorithm has stopped due to a minimum solution and not due to have reached a maximum number of iterations. If the genetic algorithm has stopped for having reached the maximum number of iterations, either the defined maximum number of iterations is too small or the defined fitness function is not the most appropriate to find the minimum.

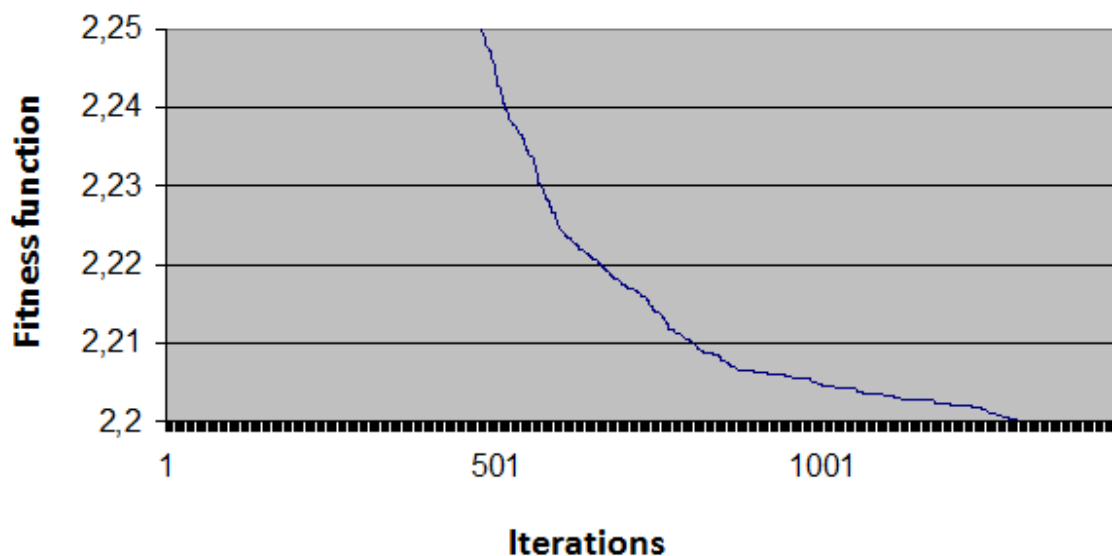


Figure 20. Evolution of the fitness function value

In addition, as an example of some of the obtained results, Figure 21 shows the evolution of the thickness value of one of the beam cross sections.

From Figure 21 it can be observed that although very big values of the beam thickness were obtained, once smaller values minimize the fitness function the genetic algorithm converges to a solution.

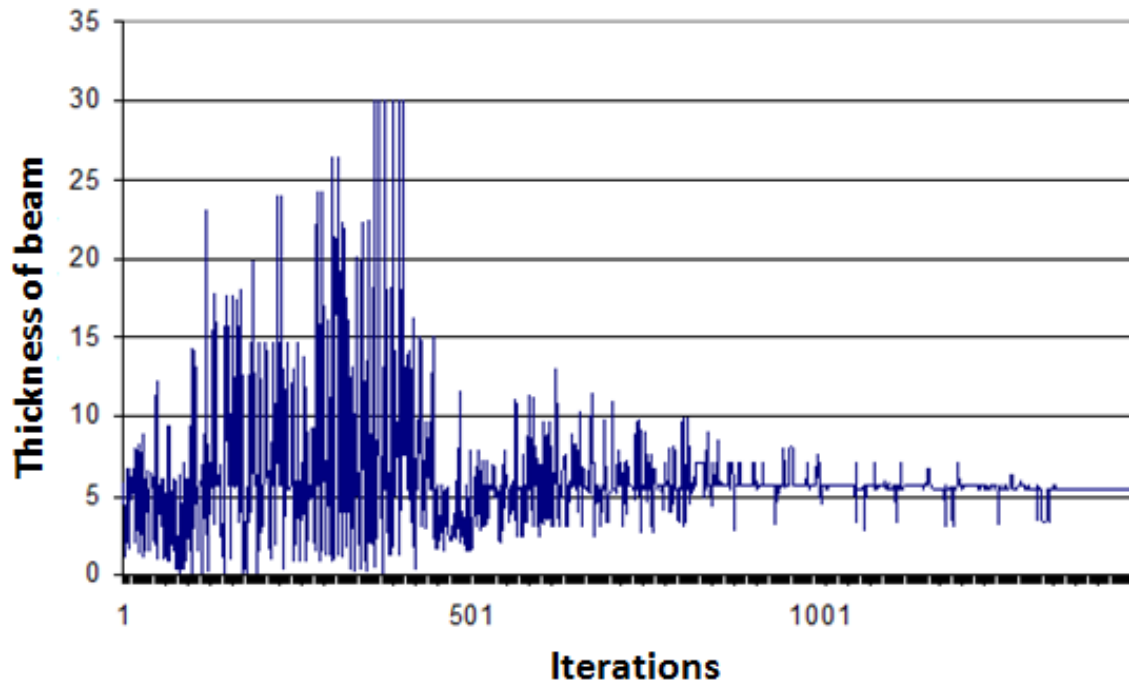


Figure 21. Example of the evolution of the value of beam thickness

The optimized vehicle structure achieved the optimized reductions shown in Table 2.

Weight	Torsion stiffness
Reduction of 4%	Increased 0,23%

Table 2. Optimization results

A reduction of 4% of the weight was achieved while improving the torsion stiffness in 0,23%. Prior to this optimization a sensitivity analysis was carried out in order to apply the optimization loop on certain beams more sensitive to variations in weight and torsion stiffness. It is worth highlighting that the optimization that has been performed keeps the original bus structure geometry. Thus, further weight reduction and torsion stiffness improved results may arise by changing the beam placement in the bus structure.

6. Conclusions

In the current chapter the authors have shown, by means of an optimization of a real bus structure, how to perform an advance analysis in the automotive field based on the usage of MatLab and ANSYS. The methodology to couple both programs has been presented, as well as all of the files necessary to optimize the beam thickness of a real bus structure in weight and torsion stiffness. On one hand, MatLab provides an advanced software environment in which

the user may, either program complex scripts or use available toolboxes. For the optimization loop analyzed in this chapter the genetic algorithm toolbox has been employed, having shown to be a very useful tool. On the other hand, due to the fact that the optimization problem to be solved required of vehicle structure calculations, finite element software (ANSYS) was employed. In addition, for each iteration of the optimization loop, both programs require of calculations, thus, having to be coupled together and perform such computations without user intervention. The present chapter has shown how to couple both programs as well as how to manage the different files created during the simulations. In addition, in order to perform the bus structure calculations and due to the fact that the thickness value changes during each loop, the bus finite element model must be defined in terms of parametric variables. It has been shown how to use the ANSYS Parametric Design Language (APDL) in order to parametrize the bus structure finite element model. The methodology presented has shown to be a very successful tool for advance analysis in the automotive field.

Author details

A. Gauchía*, B.L. Boada, M.J.L. Boada and V. Díaz

*Address all correspondence to: agauchia@ing.uc3m.es

Mechanical Engineering Department, Research Institute of Vehicle Safety (ISVA), Carlos III University, Madrid, Spain

References

- [1] Gauchía, A., Boada, M.J.L., Boada, B.L., Díaz, V. Simplified dynamic torsional model of an urban bus. *Int. J. Heavy Vehicle Systems* 2009; 16(3) 341-353.
- [2] Gauchía, A., Díaz, V., Boada, M.J.L., Olatunbosun, O.A., Boada, B.L. Bus structure behaviour under driving manoeuvring and evaluation of the effect of an active roll system. *Int. J. Vehicle Structures & Systems* 2010; 2(1) 14-19.
- [3] Gauchía, A., Olmeda, E., Aparicio, F., Díaz, V. Bus mathematical model of acceleration threshold limit estimation in lateral rollover test. *Vehicle System Dynamics* 2011; 49(10) 1695-1707.
- [4] Cazzola, G.J., Alcalá, E., Aparicio Izquierdo, F. Study of the bending response of metal foam-filled beams applied to enhance the rollover behavior of coach structure. *International Journal of Chashworthiness* 2013. DOI: 10.1080/13588265.2013.831516.
- [5] Lan, F., Chen, J., Lin, J. Comparative analysis for bus side structures and lightweight optimization. *Proc. IMechE Part D: J. Automobile Engineering* 2004; 218 1067-1075.

- [6] MATLAB 2011, The MathWorks Inc. 2011.
- [7] ANSYS. Structural Analysis Guide. Ansys Inc 2001.
- [8] Gauchía, A., Díaz, V., Boada, M.J.L., Boada, B.L. Torsional stiffness and weight optimization of a real bus structure. *International Journal of Automotive Technology* 2010; 11(1) 41-47.
- [9] Choi, B., Tang, B. Optimum shape design of rotor shafts using genetic algorithm. *Journal of Vibration and Control* 2000; 6 207-222.
- [10] Gil, L., Andreu, A.. Shape and cross section optimisation of a truss structure. *Computers and Structures* 2001; 79 681-689.
- [11] Adeli, H., Cheng, N.-T. Concurrent genetic algorithm for optimization of large structures. *Journal of Aerospace Engineering* 1994; 7(3) 276-296.
- [12] Xiang, J., Zhong, Y., Chen, X., He, Z.. Crack detection in a shaft by combination of wavelet-based elements and genetic algorithm. *International Journal of Solids and Structures* 2008; 45 (17) 4782-4795.
- [13] Li, S., 2000. The micromechanics theory of classical plates: a congruous estimate of overall elastic stiffness. *International Journal of Solids and Structures* 37 (40), 5599-628.
- [14] Joo, J., Kota, S., Noboru, K. Nonlinear synthesis of compliant mechanisms: Topology design. In: DETC2000/MECH-14141 (Ed.), *Proceedings of 2000 ASME Biannual Mechanism Design Conference*.
- [15] Saxena, A. Topology optimization of large displacement compliant mechanisms with multiple materials and multiple ports. *Structural and Multidisciplinary Optimization* 2005; 30(6) 477-490.
- [16] Mitchell, M. *An Introduction to Genetic Algorithms*. The MIT Press. Cambridge. Massachusetts. London. England 1996.
- [17] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc 1989.