DEVELOPMENT OF STRUCTURAL OPTIMIZATION AND NEURAL NETWORK DESIGN TOOL FOR BUCKLING BEHAVIOUR OF SKIN-STRINGER STRUCTURES UNDER COMBINED COMPRESSION AND SHEAR LOADING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYDIN OKUL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

JULY 2019

Approval of the thesis:

DEVELOPMENT OF STRUCTURAL OPTIMIZATION AND NEURAL NETWORK DESIGN TOOL FOR BUCKLING BEHAVIOUR OF SKIN-STRINGER STRUCTURES UNDER COMBINED COMPRESSION AND SHEAR LOADING

submitted by **AYDIN OKUL** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmail Hakkı Tuncer
Head of Department, **Aerospace Engineering**

Assoc. Prof. Dr. Ercan Gürses
Supervisor, **Aerospace Engineering, METU**

Examining Committee Members:

Prof. Dr. Altan Kayran
Aerospace Engineering, METU

Assoc. Prof. Dr. Ercan Gürses
Aerospace Engineering, METU

Assoc. Prof. Dr. Demirkan Çöker
Aerospace Engineering, METU

Assoc. Prof. Dr. Melin Şahin
Aerospace Engineering, METU

Prof. Dr. Erdem Acar
Mechanical Engineering, TOBB ETU

Date: 12.07.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Aydın Okul

Signature:

ABSTRACT

DEVELOPMENT OF STRUCTURAL OPTIMIZATION AND NEURAL NETWORK DESIGN TOOL FOR BUCKLING BEHAVIOUR OF SKIN-STRINGER STRUCTURES UNDER COMBINED COMPRESSION AND SHEAR LOADING

Okul, Aydın
Master of Science, Aerospace Engineering
Supervisor: Assoc. Prof. Dr. Ercan Gürses

July 2019, 109 pages

Stiffened panels are commonly used in aircraft structures in order to resist high compression and shear forces with minimum total weight. Minimization of the weight is obtained by combining the optimum design parameters. The skin panel dimensions, the stringer spacing and the stringer dimensions are some of the critical parameters which affect the global buckling behaviour of the stiffened panel. The aim of this thesis is to develop a neural network design tool and to carry out a geometric optimization for panels having a large number of stringers under combined loadings. Before the design tool creation, a simplified panel with minimized number of stringers and the boundary conditions to be substituted for the side stringers are found. Then the effect of some critical design parameters on the buckling behavior is investigated and it is determined which parameter is used in order to increase or decrease the strength effectively. In the Artificial Neural Network (ANN) phase, approximately seven thousand finite element (FE) models are created and analyzed in ABAQUS FE program with the help of a script written in Phyton. The script changes the parametric design variables for the analyses and collect the results. These design variables and analysis results are grouped together in order to create an ANN in MATLAB

NNTOOL toolbox. This process allows faster determination of buckling analysis results than the traditional FE analyses. This tool can predict the critical buckling load and the margin of safety for a given geometry and given loads. In the last phase, a structural optimization study is carried out by using MATLAB OPTIMTOOL toolbox for a specific region with given external dimensions and the applied compression-shear loads by using genetic algorithm method. The only constraint is not to buckle under given loadings. In this way, optimum weight is obtained for optimum design variables, which are the stringer and skin thicknesses, stringer dimensions and stringer placement.

Keywords: Finite element analysis, stiffened panels, artificial neural network, structural optimization, genetic algorithm

ÖZ

BİRLEŞİK BASMA VE KESME YÜKLERİ ALTINDA KABUK-KİRİŞ YAPILARININ BURKULMA DAVRANIŞI İÇİN YAPISAL OPTİMİZASYON VE YAPAY SİNİR AĞ BAZLI TASARIM YÖNTEMİ GELİŞTİRİLMESİ
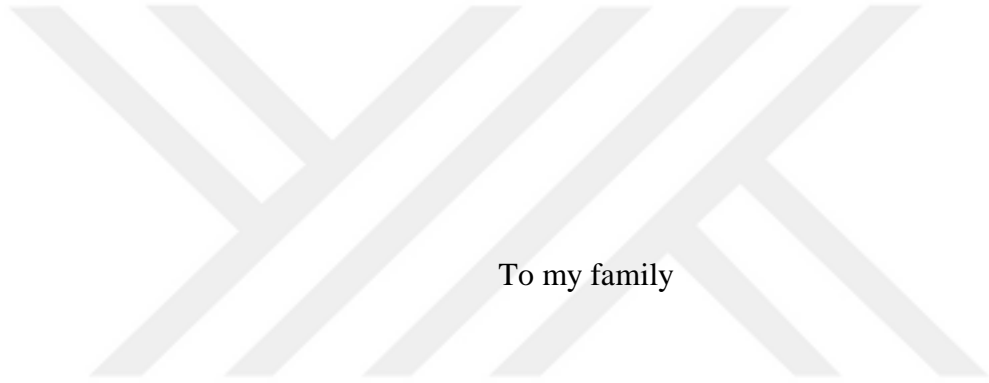
Okul, Aydın
Yüksek Lisans, Havacılık ve Uzay Mühendisliği
Tez Danışmanı: Doç. Dr. Ercan Gürses

Temmuz 2019, 109 sayfa

Kirişlerle güçlendirilmiş kabuk yapıları, hava araçlarındaki yüksek basma ve kesme kuvvetlerine direnç sağlayan ve ağırlığı minimize eden yapılardır. Ağırlığın en aza indirgenmesi, optimum tasarım parametrelerinin birleştirilmesiyle elde edilir. Kabuk ölçüleri, kiriş aralığı ve kiriş kesit ölçüleri kabuk-kiriş yapılarının burkulma davranışını etkileyen kritik parametrelerdendir. Bu tezin amacı, çok sayıda kirişe sahip kabuk yapılarının tasarımına yönelik sinir ağı tabanlı bir tasarım aracı geliştirmek ve verilen yükler altında geometrik bir optimizasyon gerçekleştirmektir. Tasarım aracını oluşturmadan önce, minimum sayıda kiriş içeren basitleştirilmiş bir yapı ve yan kenarlardaki kirişlerin davranışlarını temsil eden sınır koşulları bulunmuştur. Sonrasında da bazı kritik tasarım parametrelerinin burkulma davranışı üzerindeki etkisi incelenmiştir ve hangi parametrenin mukavemeti efektif olarak arttırmak veya azaltmak için kullanılması gerektiği belirlenmiştir. İkinci aşamada, yaklaşık yedi bin sonlu elemanlar (SE) modeli, Phyton dilinde yazılmış bir betik yardımıyla ABAQUS SE programında oluşturulmuş ve analiz edilmiştir. Bu betik, her analiz için farklı bir tasarım parametresini değiştirir ve analiz sonuçlarını toplar. Bu tasarım değişkenleri ve analiz çıktıları, MATLAB NNTOOL araç kutusunda yapay sinir ağı (YSA)

oluşturmak için gruplanır. Bu yöntem, burkulma analiz sonuçlarının geleneksel SE analizlerine göre çok daha hızlı belirlenmesini sağlar. Bu araç, belirli bir geometri için, verilen yükler altında kritik burkulma yükünü ve güvenlik katsayısını tahmin edebilir. Son aşamada ise dış boyutları ve uygulanan basma-kesme yükleri bilinen bir bölge için MATLAB OPTIMTOOL araç kutusu kullanılarak, minimum ağırlığa ulaşmak için bi yapısal optimizasyon çalışması yapılmıştır. Bu çalışmadaki tek kısıt, verilen yükler altında yapının burkulma yaşanmamasıdır. Bu sayede kabuk ve kiriş kalınlıkları, kiriş ölçüleri ve kirişlerin yerleşimi için optimum sonuçlar elde edilir.

Anahtar Kelimeler: Sonlu elemanlar analizi, güçlendirilmiş paneller, yapay sinir ağı, yapısal optimizasyon, genetik algoritma

To my family

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

xiv

LIST OF FIGURES

FIGURES

xvii

LIST OF ABBREVIATIONS

ABBREVIATIONS

ANN	Artificial neural network
BC	Boundary condition
C	Clamped boundary condition
DOF	Degree of Freedom
DSP	Digital Signal Processing
F	Free boundary condition
FE	Finite element
FEA	Finite element analysis
FEM	Finite element method
GA	Genetic algorithm
LB	Lower bounds for the design parameters in optimization
LM	Levenberg-Marquardt
MAE	Mean Absolute Error
mm	Milimeter
MPa	Megapascal
MS	Margin of safety
MSE	Mean Squared Error
N	Newton
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
NN	Neural network
RF	Reserve factor
SS	Simply supported boundary condition
UB	Upper bounds for the design parameters in optimization

LIST OF SYMBOLS

SYMBOLS

a_{skin}	Skin panel length
b_{eff}	Effective width of the panel
b_{skin}	Skin panel width
c_{str}	Flange length of the stringer
d_{str}	Stringer spacing
E	Modulus of elasticity (Young's modulus)
E_c	Compression Young's Modulus of material
F_{cy}	Compression Yield Stress of material
h_{str}	Height of the stringer
i	Buckling mode number
k	Buckling coefficient
k_C	Compression buckling coefficient
k_{eff}	Effective width factor
k_S	Shear buckling coefficient
n_{str}	Number of stringers
R_L	Applied over allowable stress ratio for compressive loading
R_S	Applied over allowable stress ratio for shear loading
t	Thickness of a panel
t_{str}	Thickness of the stringer
ρ	Material density
Ev	Poisson's ratio
$d_{str}h_{str}c_{str}\sigma_{cr}$	Critical buckling stress
$\sigma_{allowable}$	Allowable buckling stress
$\sigma_{applied}$	Applied buckling stress
K^{MN}	Tangent stiffness matrix

v^M	Nontrivial displacement solutions
Q^N	Load increment
K_0^{MN}	Tangent stiffness matrix for base state
P^N	Preload
K_Δ^{MN}	Differential stiffness matrix due to incremental loading Q^N
λ_i	Eigenvalues
v_i^M	Eigenvectors
$[K]$	Stiffness matrix
$\{D\}$	Displacement
$\{R\}_{ref}$	Load on the structure
λ_{cr}	Smallest load at bifurcation point
$F_{cy}E_c b_{eff}$	

CHAPTER 1

INTRODUCTION

Semi-monocoque structures are commonly utilized in many aerospace and marine structures. This form of construction provides a high strength continuous surface for an aircraft, combined with the condition that the structural weight should be as small as possible. The main components of semi-monocoque structures can be classified as follows:

- Skin panels:

This is the most suitable shell for carrying the load on its own surface as membrane stresses. The thin walled skin panel resists all tension, compression and shear loads but reinforcements (lateral and longitudinal supports) are required in order to have thinner skin [1].

- Longitudinal reinforcements:

These are the longerons and stringers for fuselage shells. They can carry the most of the longitudinal tension and the compression loads and secondary small bending loads [1].

- Transverse reinforcements:

These are the rings, frames and bulkheads for fuselage shells. They can carry the most loads in the plane of the member. Therefore, they are usually incapable of carrying much lateral load [1].

These three components together form the external structure of a typical semi-monocoque aircraft such as British Aerospace 146 Structure, which is shown in Figure 1.1.
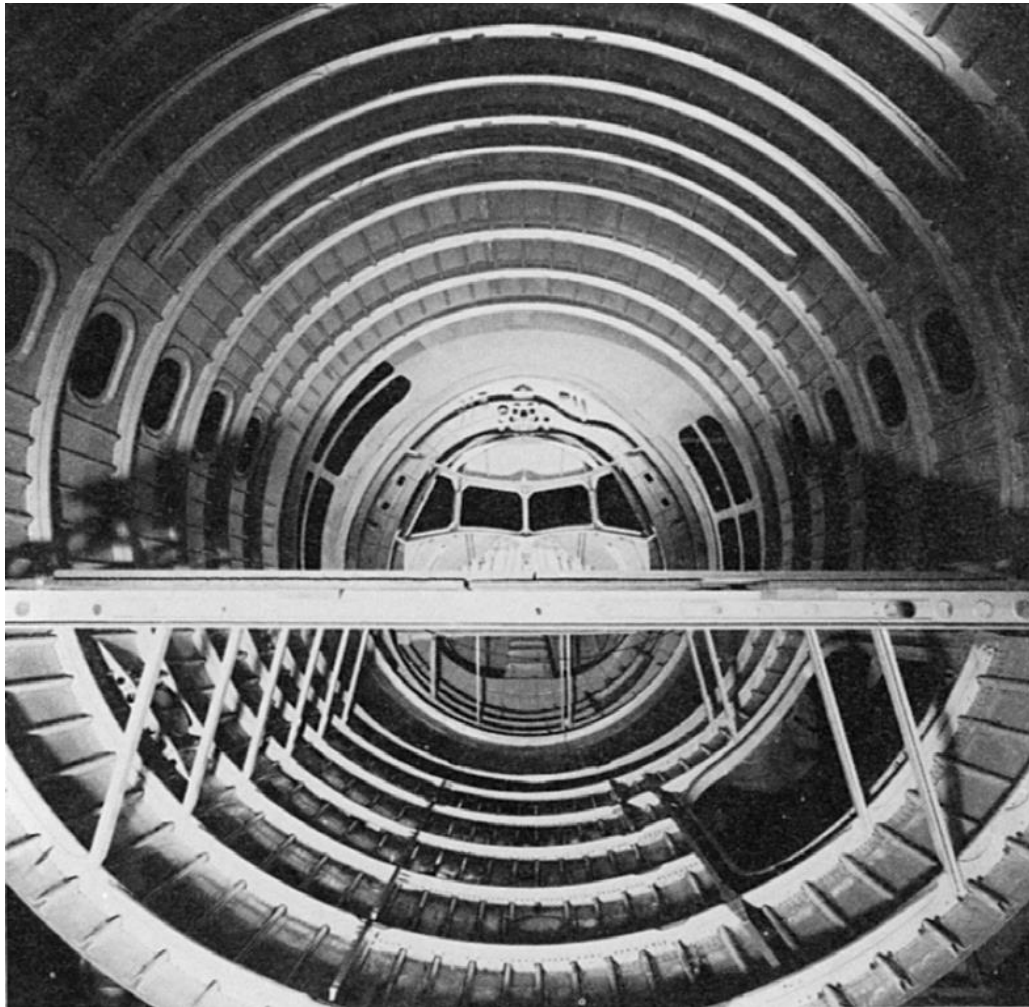
Figure 1.1 British Aerospace 146 Structure [2]

There are many skin-stringer assemblies between each ring frames or bulkheads. As an example, tailcone skin-stringers of T-625 (Turkish Light Utility Helicopter by Turkish Aerospace) are shown in Figure 1.2. Skin-stringers are generally used to resist high compression and shear loads with minimum weight. Since the buckling occurs before the limit load of the skin panels, the stringers are usually utilized to prevent skin panel buckling [3].

In theory, buckling is a loss of balance and is considered as structural instability. This type of failure can take various forms in skin-stringer structures: buckling of the skin

between stiffeners, lateral torsional buckling of the stiffeners, overall buckling of the stiffened panel as a column or local buckling of the stiffener [4]. In the field of aviation, post buckling concept is widely used. In post buckling stage, when the panels between the stiffeners buckle, the skin panel forces are transferred to the stiffeners. As a result of a redistribution of forces occurs between the buckled panel and the stiffeners. However, this concept does not work well for the panels that are subjected to high shear forces. Because for the shear dominant structures, this concept comes up with a little thinner skin panels, but much thicker stiffeners. The structure can converge to an unreasonable final state in terms of weight. As an example, there is very high torsion in the helicopter tail cone, because of the height of the tail rotor. Therefore, the tail cone and its skin-stringer structures between frames are subjected to high shear forces. Generally, the majority of the weight comes from the skin panels in air vehicles. In a typical helicopter structure, skin panels are about three times heavier than the stiffeners. Therefore, keeping the panel thickness low is a primary goal for the design process. For this reason, the first encountered failure mode appears as panel buckling between stiffeners. A typical skin-stringer structure of a helicopter between the ring frames is shown in Figure 1.2.

Figure 1.2: Skin-Stringer Structure of a Helicopter

The critical buckling load is a function of the stringer section, the stringer spacing and the skin thickness for pure compression and pure shear loadings [1]. In the case of a combined loading, it also depends on the compression-shear ratio. In order to find the strength, i.e. the critical buckling load of such a skin-stringer structure, these critical parameters are combined with optimum combinations.

Different techniques are used in literature for the analysis of skin-stringer structures. Testing is probably the best technique but it is a very costly method. After each test, design should be changed based on the results obtained and structure should be manufactured again. Therefore, simulation of these tests with an accurate and robust method is usually preferred.

Numerical methods are commonly used rather than analytical methods in the static analysis of the skin-stringer structures. The most widely used numerical method is the finite element method. In the case of design problems solved by this method, the model of the structure needs to be established firstly. After that, loads and boundary conditions are applied, desired parameters and outputs are arranged. In the last step, computation is performed and the eigenvalues, i.e. the buckling loads, are obtained for a specific geometry. Thereafter, these analyses should be performed repeatedly with new input sets and the results should be updated until the optimum design is attained. All these steps may consume a significant amount of time depending of the complexity of the structure. Details of the structural model, such as geometric parameters, mesh quality, material properties, boundary conditions, load types, can be quite complex [5]. Thus, analyses of large models with FEM may require high computational resources.

1.1. Objective of the Thesis

The main objective of this thesis is to develop a design tool with the help of an artificial neural network (ANN) and to perform a structural optimization for panels with large number of stringers under combined loadings. The design tool will have two major

capabilities. The first one is to obtain critical buckling loads for a specific geometry of a skin-stringer structure. The second is to obtain Margin of Safety result for a specific geometry of a skin-stringer structure under specific applied combined loadings.

The motivation in this work is to contribute to the design of a skin-stringer structures having maximum load carrying capacity with minimum weight. For this purpose, some preliminary studies are performed to understand the buckling behavior of a stiffened panel with a large number of stringers.

1.2. Scope of the Thesis

At the beginning of this study, a short introductory part about the semi-monocoque and skin-stringer structures is presented. After that, the information about the buckling in skin-stringer structures is given.

Finite element based structural analysis of a skin-stringer structure is given in Chapter 2. The aim of the analysis stage is to determine concise baseline parameters in the finite element modelling phase and to come up with a reasonable solution of a specific skin-stringer structure which is the critical buckling curve. In the model creation and FE solver, ABAQUS FE program is used.

Chapter 3 involves several case studies conducted in order to find a representative finite element model with a minimum computational resources. Firstly, a stiffened panel with a minimum number of stringers is found which gives the same buckling behavior with the structures that has higher number of stringers. Secondly, boundary conditions that represents the side edge stringers are investigated with a trade-off study. Lastly, the effects of some critical geometric parameters are examined for a weight gain purposes. Comparing the analysis options is performed in ABAQUS FE program and a trustworthy representative solution for the final finite element model is obtained at the end of this chapter.

In neural network chapter, firstly a database that contains FE analysis inputs and outputs of different sets of parameters is needed for the training of the neural network. For this purpose, scripts are written in Python 2.7 in order to analyze thousands of analyses in ABAQUS and each set of inputs and outputs is stored in several files. Then, an artificial neural network is created by using this database in MATLAB neural network toolbox (NNTOOL).

In Chapter 5, structural optimization is performed for a region with given dimension and given loads by using genetic algorithm method. In the computation, MATLAB optimization toolbox (OPTIMTOOL) is used in order to minimize the weight of the designed structure. In the optimization toolbox, developed ANN tool is used to compute the fitness function.

Lastly, Chapter 6 summarizes the results of all studies performed throughout the thesis and gives recommendations on how these results should be interpreted.

1.3. Literature Review

In the early designs of the aircrafts such as the Wright Brother's aircraft, most aircraft structures were constructed with wire-bracing or cross-bracing methods with wooden rectangle frames between upper and lower wings. Every structural rectangle needs to be braced with diagonal wires since the wings were designed to take the aerodynamic forces only. In this way, shearing, bending and twisting were prevented by the structures between the wings. These biplane methods were used until 1940s. After this point, metal manufacturing techniques were discovered. The thin sheet metal tubes were replaced with wooden spars and struts and internal cross-bracing was found to be sufficient for single or double seated aircraft [6]. However, it did not provide enough strength for multi-passenger structures. To overcome this, thin panels were used with the stiffeners together inside the wing structures which is inspired by the fuselages of early flying boats. Thus, the skin panels and the wing become an active

load carrying member of the structure as well. The stiffeners were generally used to prevent buckling of thin skin panels [3].

In literature, there are many articles for stiffened panels with pure loadings. In a study conducted by Cankur and Gürses [5], pure compression loadings were applied to metallic flat skin-stringer structure for fast determination of buckling and collapse loads of the structure with the generation of the artificial neural network. Besides, Mallela et al. [7] present a parametric study on composite stiffened panels subjected to in-plane shear loads. In the study, the effects of few important parameters were investigated on buckling behavior of the structure by analyzing a total of 450 different models. In addition to pure compression and pure shear loads, combination of these loads were also applied to both flat panels and stiffened panels. Dima [8] obtained a methodology to specify the critical compression and shear forces for a flat isotropic plate and the reserve factor (RF) for given combined applied loads. The RF calculation methodology in Dima's work guides the Margin of Safety (MS) calculation in this study. In 1984, NASA published a technical paper which investigated a few procedures of buckling analysis for stiffened panel under combined loadings [9]. This paper provides a set of accurate benchmarks and insight into buckling characteristics of stiffened panels. However, apart from these studies, there are not many studies in the literature regarding the prediction of the critical buckling loads for stiffened panels under combined compression and shear loads.

The boundary conditions that affect the buckling behavior of the stiffened panels have been examined in several studies. In general, most studies examined the panel independently from the stringers with classical boundary conditions, since most analytical methods suggest a boundary condition for flat panels only, not for stiffened panels. In his book, Bruhn gives buckling coefficient charts for pure loadings which depend on the aspect ratio of the panel and the boundary conditions at the edges [3]. In master's thesis of Muameleci [10], both hinged and clamped edge conditions were investigated and compared with different FE programs for the shear web beams. The study explained the main differences between the boundary conditions and their

effects to the buckling behavior of the plates. Al-Azzawi et al. [11] proposed a modified equation to find the elastic buckling strength of the plate girder web panels. The original equation was found by Lee et al. [12], however several numerical tests were performed in Al-Azzawi's work and by using these test results, an equation taking the stiffener rigidity effect into account was proposed. Boundary conditions in the studies of Dima [8] and Aydın [13] focus on the boundary conditions of a skin-stringer structure under compression loading. Both studies reach the conclusion that neither the simply supported nor the clamped boundary conditions represent the real BC of the unloaded edges, since the actual stringer assures a condition which is in between these two. However, there is no analytical or numerical method in the literature for the real boundary condition provided by the side stringers.

In the preliminary design process of an aircraft, finite element modelling and buckling analysis of each skin-stringer structure are an iterative study and takes long hours until the optimum structure is obtained. In terms of time savings and labor gain, it is very logical to use a smart tool that is reasonably accurate and can produce faster results than the traditional FE analyses. Artificial Neural Network (ANN) is one of these efficient tools. In the literature, there are many studies about the prediction of a quantity in many areas such as future price predictions, image recognition, data estimation etc. As an example use of neural networks, Asilkan and Irmak [14] estimated the future prices of the second-hand automobiles by using neural networks. The input data are gathered from the Europe-based web sites with the prices and features of the vehicles. The results gave acceptable errors at the end of the study. Haas et al. [15] developed a successful neural network that can predict variation in rotating component loads for a helicopter flight test. Similarly, Parlakyıldız [16] identified and classified the fingerprints with ANN by using Digital Signal Processing (DSP).

In addition to these usage areas of the ANN, there are also several studies related with the prediction of the critical loads of a structure by using ANN. Bisagni et al. [17] used ANN for the prediction of load carrying capacity and weight of a skin-stringer in order

to decrease computational time of the analysis. In a similar study, Mallela et al. [18] predicted the buckling load of laminated composite stiffened panels under the in-plane shear load by using ANN. In this study, the database for training and testing was obtained from FE analyses results. With a similar consideration, Cankur [19] developed an artificial neural network which is capable of fast determination of buckling and collapse loads of a stiffened panel under compression loading. For this purpose, 1440 different stiffened panels were modelled and analyzed in ABAQUS FEA program. The input and the output parameters of FE analyses were collected and a neural network was trained with this data. The ANN results were compared with the finite element analysis results and additional sets. A maximum error of 2.5% was obtained from the comparisons and it was concluded that the proposed tool can be used in preliminary design phases.

In literature, there are various studies conducted with the classical optimization methods for the structural design purposes. Only a small portion of studies used an ANN in order to obtain an output, while most of them used FE commercial programs.

In a general discourse, Rao [20] stated that the techniques of finding optimal values are a branch of operational researches and operational research deals with the application of scientific methods to decision making problems and finding appropriate solutions. In his book, he mentioned mathematical programming techniques, especially linear and nonlinear programming, geometric programming, dynamic programming, stochastic programming, and more recently, the genetic algorithm method which is used in the thesis. The author also mentions optimization techniques and applications in engineering.

In his book, Megson [2] describes the aircraft structure analysis with all the details and also included information about elasticity. The importance of the optimization is also mentioned in the preliminary design phase of an aircraft. In their article, Jweeg et al. [21] described their work about creating the optimum design for aircraft static analysis. The research covered both aerodynamic and structural design. In thieir study,

weight gains of different types of materials in different wing structures were determined by an optimization study. A study performed by Rispler and Raju [22] mentioned a parametric optimization of the wing of a mid-sized aircraft. It is aimed to maintain the strength of the structure and to reduce the weight of the structure. By considering structural and production constraints, a parametric optimization was performed and the rib spacings, spar positions and shell thicknesses were optimized in a similar manner with this study. In her article, Falco [23] has optimized a wing structure of beams, ribs and shells considering two cases which are minimum weight and maximum loading conditions. Beams, rib spacing and different structural cross-sectional properties were taken as design variables and improvements in objective function were presented.

For the optimization technique, Charbonneau [24] presented a detailed comparison of genetic algorithm (GA) and other optimization techniques and gave detailed information about the use of GA. In his article, McCall [25] explained the structure of the genetic algorithm with simple examples. Possible applications of GA's were also provided in order to construct the theoretical GA. Lastly, Türe [26] performed a geometric optimization for the helicopter subfloor by using the genetic algorithm method. The main aim was to obtain the structure with maximum energy absorbing capability, i.e. the best crashworthy solution for a structure.

CHAPTER 2

FINITE ELEMENT MODELLING AND BUCKLING ANALYSIS OF SKIN-STRINGER STRUCTURES

Structural analysis of skin-stringer structures with a numerical method requires serious work force and computational power and time in order to achieve reliable results. Finite element method (FEM) is the basis of this thesis. Finite element analyses (FEA) are performed in a parametric manner and the outputs are transferred to the artificial neural network usage.

For the use of artificial neural network (ANN), a large number of finite element models should be created in the first step of this study. To perform these analyses efficiently, the geometry, the mesh and the material properties of the skin-stringer structures would be defined correctly in the modelling process.

The purpose of this chapter is to choose correct baseline parameters for the finite element model and to obtain reliable results for a specific geometry. Thus, a sample skin-stringer structure will be analyzed with the help of baseline parameters. Firstly, the geometric dimensions and the material properties will be defined. After that, selections of load application and boundary condition are explained. In addition, some studies related with mesh properties should be performed for the convergence of the results. Lastly, a sample analysis result will be given.

It is important to set up a reasonable model for a specific geometry in order to obtain correct results for the usage of ANN. For this reason, some sensitivity studies about mesh properties will be performed in the generation the finite element (FE) model.

All finite element models are created and analyzed in ABAQUS FEA software program. ABAQUS can run scripts written in Python, therefore a parametric code written in this language allows all FE analyses to be performed at one-step.

2.1. Buckling Analysis Methodology in Skin-Stringer Structures

In theory, buckling is a loss of balance and is considered as structural instability [3]. This can occur as a sudden deformation in any direction of a skin-stringer structure subjected to both compression and shear loading. For some structures in the aircraft, buckling of panels may not be considered as a failure mode since stiffeners can still resist the loads. This is called post-buckling behavior. However, the stiffeners should be too stiff for this process and if the case is also shear dominant, stiffeners may not resist the loads since the post buckling loads would be higher than the compressive forces. Since the case of interest is combined loading in this study, the panel buckling is chosen as the main failure mode for the skin-stringer structures.

Buckling for a skin-stringer panel is mostly independent from the material strength and occurs before the limit load of the panel. For an optimum design of these structures, Hughes et al. suggest an optimum design point with only two design variables, plate thickness and height of the stiffener web, which is shown in Figure 2.1 [27]. The figure shows the critical local plate buckling curve and overall buckling curves, and it is clear that the optimum values of design variables would be at the intersection of these two critical curves. However, the curve is independent from the weight of the structure. As an illustration in the figure, although the point A is not the smallest values of each design variable, it can give the lightest configuration for the design.
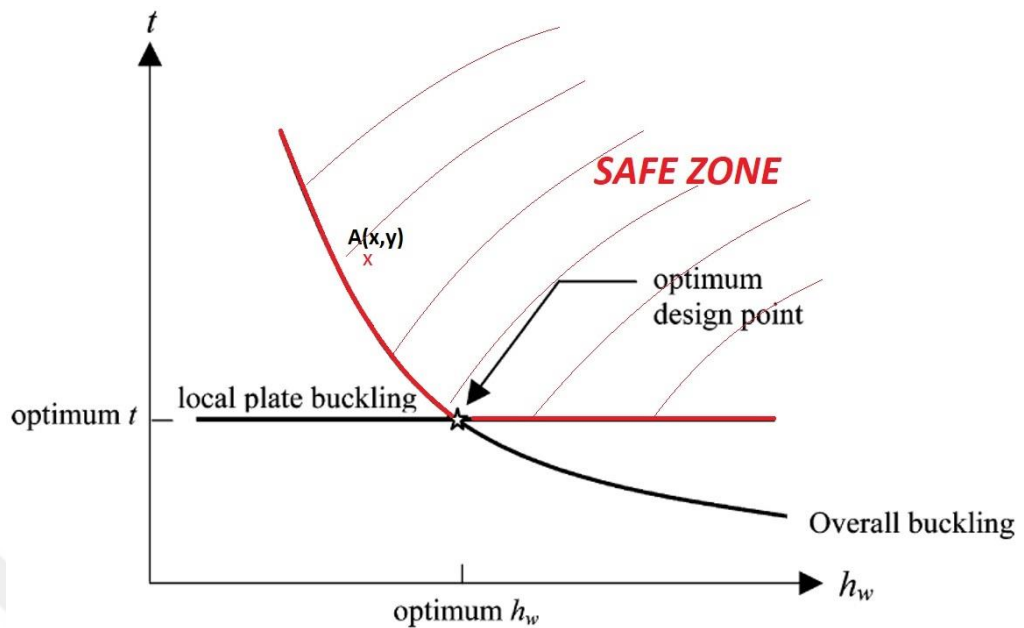
Figure 2.1 Simplified design space for optimum stiffened panel design [27]

The purpose of the analyses is to determine the critical buckling load for specific analysis inputs. In addition to the geometric inputs, compression-shear load ratio should be taken into account for combined loadings. By taking this parameter into account, the inputs are not only the geometric parameters but also compression-shear load ratio. Figure 2.2 shows the inputs and outputs of one FE analysis.
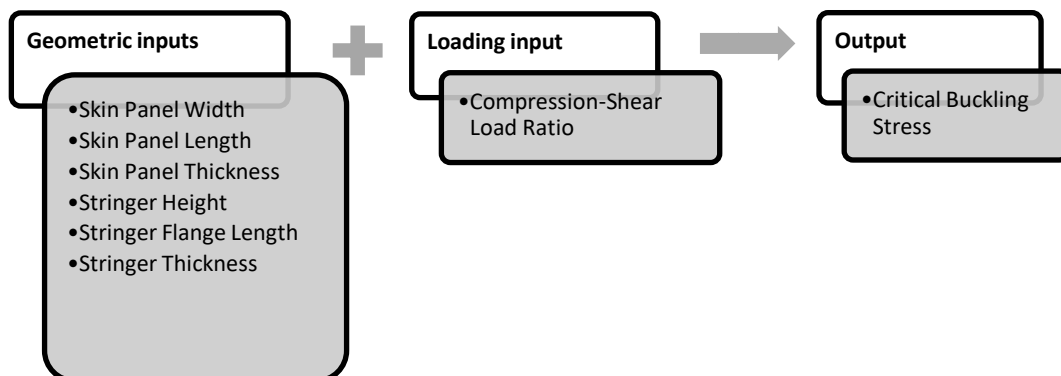
Figure 2.2: Inputs and outputs of one FE analysis

In this study, margin of safety (M.S.) is calculated in the creation of artificial neural network tool. M.S. should be greater than zero for a safe design. Figure 2.3 shows how the M.S. values can be calculated with the help of the allowable/applied stress ratios for pure loadings as (pure compression or pure shear):

$$M.S. \quad = \quad \frac{\sigma_{allowable}}{\sigma_{applied}} - 1 \quad = \quad \frac{b}{a} - 1$$
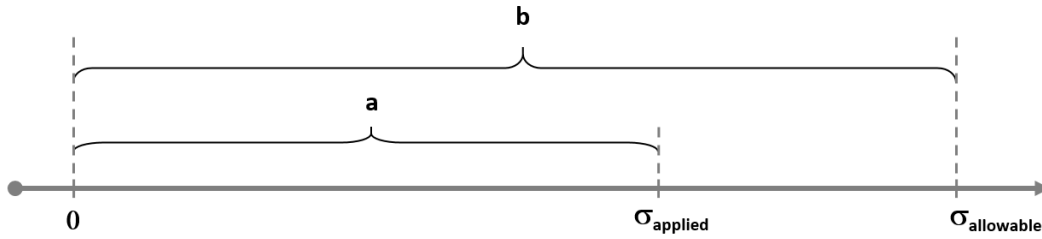
Figure 2.3: Margin of safety calculation for pure loadings

For the preliminary design phase, tool user should be able to obtain margin of safety output for the combination of applied compression and shear loads. For this reason, there should be a critical buckling curve for a specific geometry and one can obtain M.S. output with the graphical method. In order to obtain critical buckling curve, FE analyses should be performed for constant geometric parameters and different load ratios. This process yields a critical buckling curve for a specific geometry in compression-shear graph. A sample curve is shown in Figure 6.

Since the margin of safety calculation is related with the allowable to applied stress ratio from Figure 5, it should be calculated with the same logic for combined loadings at given compression-shear ratio, in Figure 2.4. A sample critical buckling curve is generated with the help of several FE analyses and then the M.S. is obtained with the same concept for a fixed compression-shear ratio:

$$M.S. \quad = \quad \frac{\sigma_{allowable,combined}}{\sigma_{applied,combined}} - 1 \quad = \quad \frac{b}{a} - 1$$

14
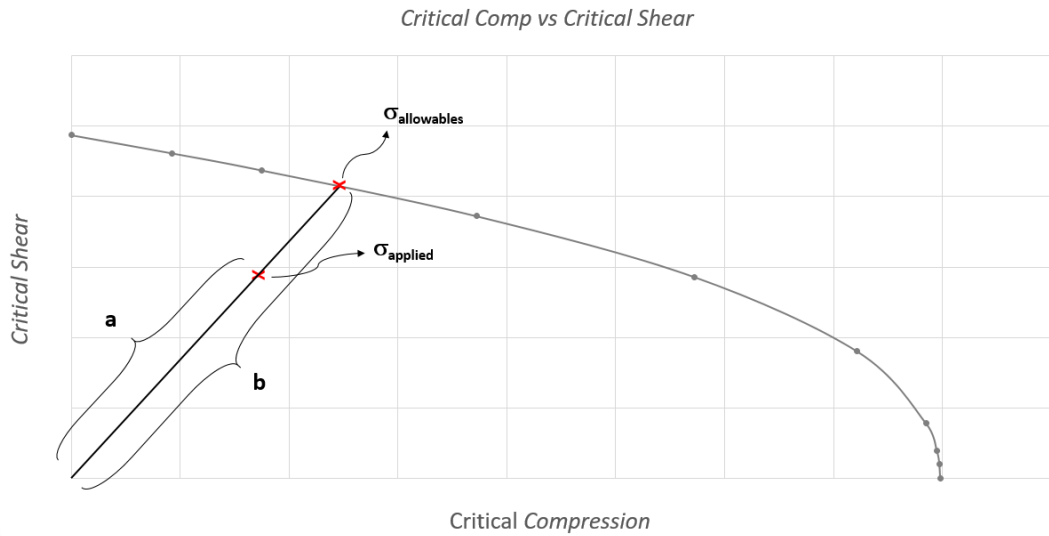
Figure 2.4: Margin of safety calculation for combined loadings

2.1.1. Analytical Methods

Critical buckling formula for skin panels between stiffeners can be obtained for shear and compression loading [3]. Although the critical stress formula is the same, some variables differ in each loading condition. The critical buckling stress is calculated as

$$\sigma_{cr} = \frac{k \ \pi^2 \ E}{12 \ (1-v^2)} \ \left(\frac{t}{b}\right)^2 \tag{2.1}$$

The explanations of the parameters in the critical buckling stress formula are given in Table 2.1.

Buckling coefficient k depends on the panel dimensions, boundary conditions and the loading cases (compression or shear). Compressive and shear buckling coefficient curves are obtained by performing tests and given in literature as graphs [28].

Table 2.1: Definitions of the parameters in the critical buckling stress formula

Buckling coefficient k	$\mathbf{k_c}$ for compression case
	$\mathbf{k_s}$ for shear case
b	Loaded edge length of the panel for compression case
	Short dimension of the panel for shear case
E	Modulus of elasticity
v	Poisson's ratio
t	Skin panel thickness

The compressive curves are given for loaded edge and unloaded edge boundary conditions separately. Three edge conditions for unloaded edges: hinged (simply supported), clamped (fixed) and free edge conditions, and two edge conditions for loaded edges: simply supported (continuous line) and clamped edge (dashed line) conditions are distinguished. Similar to this graph, there are additional graphs for flexural rigidity parameter on unloaded edges. The buckling coefficient for compressive loading is given in Figure 2.5.

Similarly, the buckling coefficient for shear loading is given for simply supported edge and clamped edge conditions. These conditions represent for all loaded and unloaded edges. In figure, symmetric mode means that the deformation due to shear load is symmetric about the load axis. The buckling coefficient for shear loading is given in Figure 2.6.
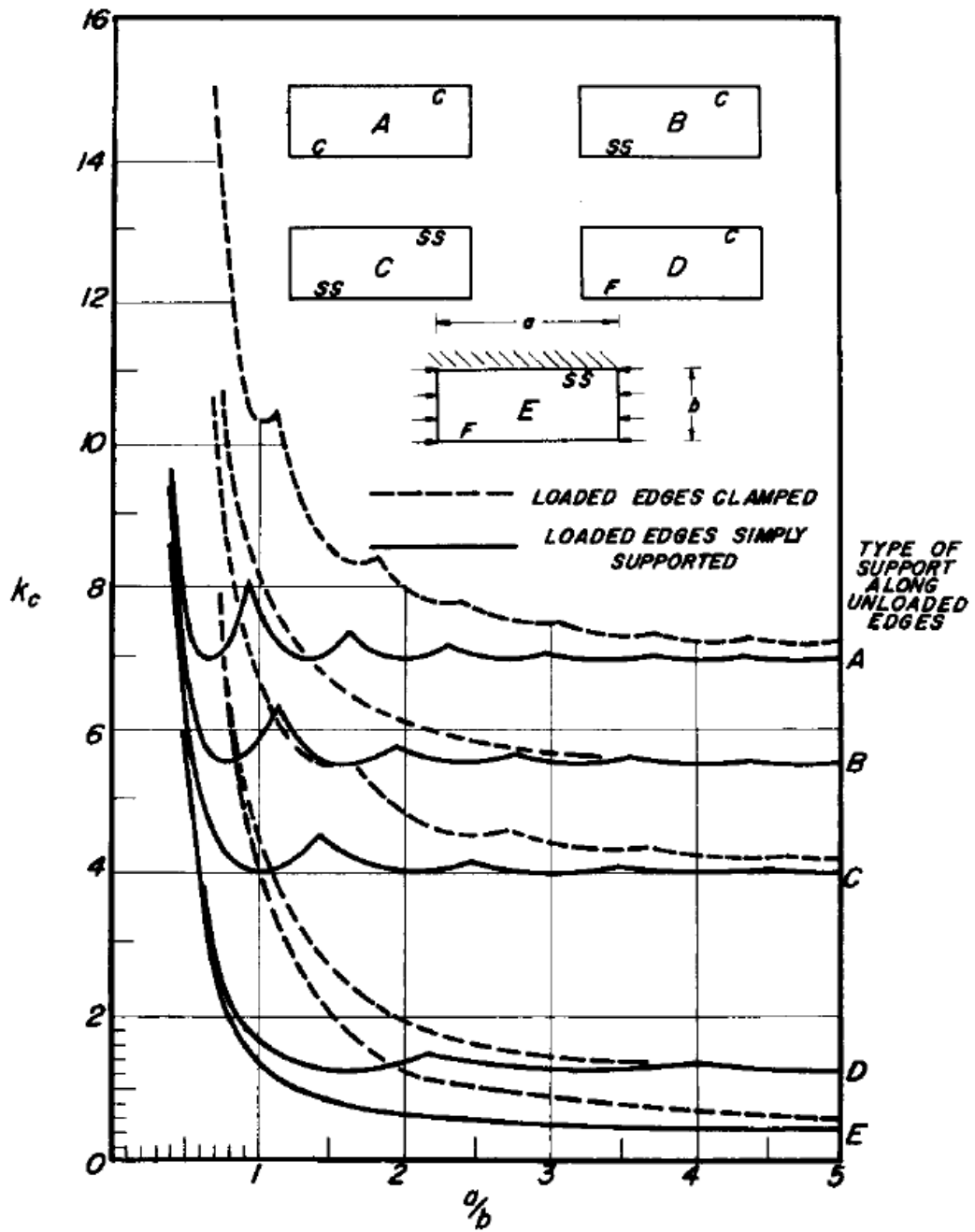
Figure 2.5: Buckling coefficient for compressive loading with respect to panel aspect ratio (a/b) and edge conditions (C:Clamped, SS:Simply supported, F:Free) [3]

17
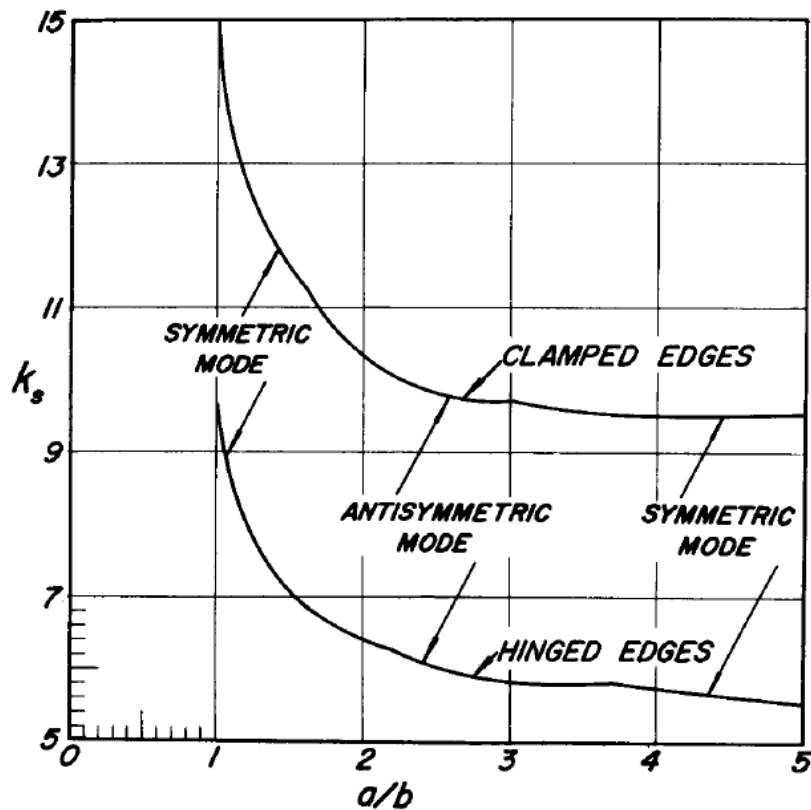
Figure 2.6: Buckling coefficient for shear loading with respect to panel aspect ratio (a/b) and edge conditions (C:Clamped, SS:Simply supported, F:Free) [3]

For the combined loading of compression and shear, an interaction equation is defined as,

$$R_L + R_S^2 = 1.0 \qquad [3] \qquad (2.2)$$

In equation (3.2), R_L and R_S represent [applied stress]/[allowable stress] ratio for longitudinal (compression or tension) and shear loads respectively, which can be calculated by using Equation (3.1). The margin of safety for such a combined loading can be found for R_L and R_S by using Figure 2.7.
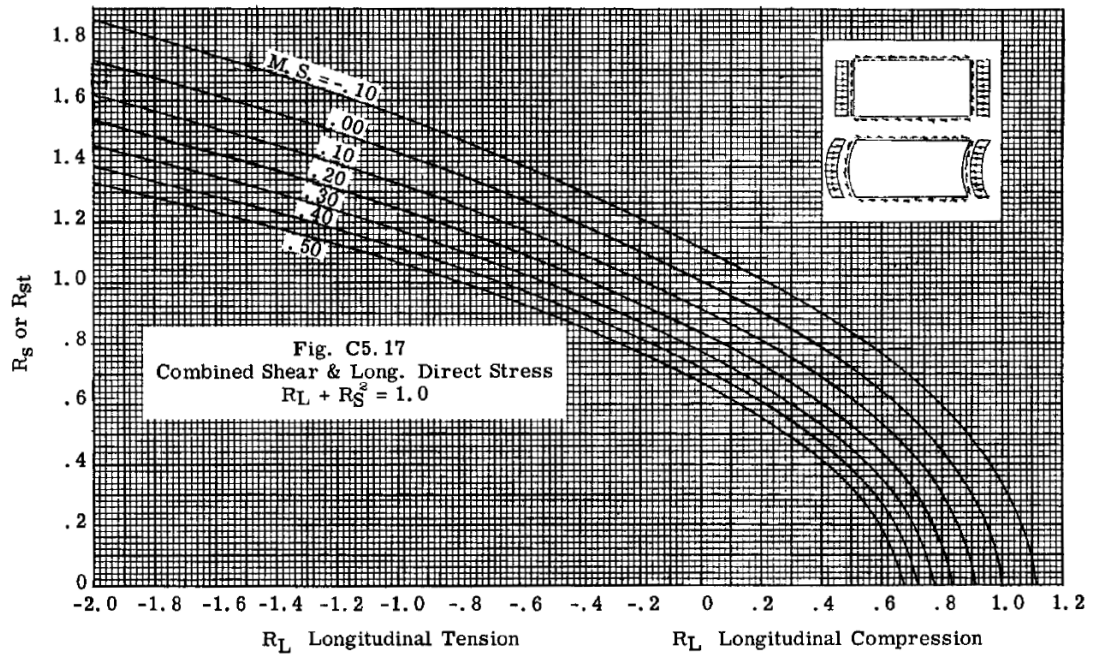
Figure 2.7: Combined compression and shear stress interaction curves [3]

2.1.2. Numerical Methods

In addition to these analytical methods, critical buckling stresses can also be found by using numerical methods. The most widely used numerical method is the finite element method (FEM). Typically, a linear perturbation procedure is used in ABAQUS FE program [29]. This procedure is often used to estimate the critical buckling loads of stiff structures. These structures carry design loads in axial direction and deform in-plane rather than bending. The behavior against applied loads usually involves very little deformation before the buckling. The structure behaves like an Euler beam when it suddenly bends. In the Euler beam theory, the structure has a rigid behavior until a critical load is reached. However, a general eigenvalue buckling analysis approach can provide estimates for buckling mode shapes, even if the response of a structure before the collapse is not linear.

19

In eigenvalue buckling problems, the critical buckling load of a structure where the model stiffness matrix becomes singular has to be founded. Therefore, following problem must have nontrivial solutions,

$$K^{MN}v^M = 0 \qquad (2.3)$$

In equation (3.3), K^{MN} represents the tangent stiffness matrix and v^M represents nontrivial displacement solutions when the loads are applied. The load applied here can consist of pressure loads, discrete loads or thermal loads.

An incremental step for a load increment Q^N is considered. It is embedded to the general eigenvalue problem by scaling it with the load multipliers, λ_i,

$$\left(K_0^{MN} + \lambda_i K_\Delta^{MN}\right) v_i^M = 0 \qquad (2.4)$$

In equation (3.4),

K_0^{MN} is the base state (just before the buckle step) stiffness matrix (including the effects of preloads P^N),

K_Δ^{MN} is the differential stiffness matrix due to the incremental loading, Q^N,

λ_i are the eigenvalues,

v_i^M are the buckling mode shapes (eigenvectors),

i is buckling mode number.

Now the critical buckling load becomes $P^N + \lambda_i Q^N$ which can be found with the help of eigenvalue outputs and initial loadings. Moreover, actual deformation magnitudes could not be found since the eigenvectors are normalized vectors. The maximum displacement component is normalized to 1.0. Moreover, ABAQUS can find the eigenvalues and the eigenvectors for symmetric stiffness matrices only, thus K_0^{MN} and K_Δ^{MN} are symmetric matrices.

For a better understanding of the information in the ABAQUS documentation [29], the book of Cook et al. [30] and thesis of Wallin [31] will also be discussed.

20

The basis of a general buckling problem is a static linear analysis and the equilibrium can be expressed as Equation 3.5. In the equation, $[K]$ represents the stiffness matrix, $\{D\}$ represents displacement and $\{R\}_{ref}$ represents the load on the structure. $[K_\sigma]_{ref}$ and $\{R\}_{ref}$ are proportional to each other and the ratio can be defined by λ_{cr}.

$$[K]\{D\} = \{R\}_{ref} \tag{2.5}$$

$$\left([K] + \lambda_{cr}[K_\sigma]_{ref}\right)\{D\}_{ref} = \lambda_{cr}\{R\}_{ref} \tag{2.6}$$

$$\left([K] + \lambda_{cr}[K_\sigma]_{ref}\right)\left(\{D\}_{ref} + \{\delta D\}\right) = \lambda_{cr}\{R\}_{ref} \tag{2.7}$$

Difference between equation 3.6 and 3.7 gives the eigenvalue problem in equation 3.8. The smallest root here, λ_{cr}, defines the smallest load at bifurcation point. Equation 3.8 can be written with the separation of the variables. Therefore, the determinant of the left hand side would be equal to 0.

$$\left([K] + \lambda_{cr}[K_\sigma]_{ref}\right)\{\delta D\} = 0 \tag{2.8}$$

$$\det\left([K] + \lambda_{cr}[K_\sigma]_{ref}\right) = 0 \tag{2.9}$$

2.2. Finite Element Modelling and Analysis of a Skin-Stringer Structure

In the beginning of a modelling process, geometric models of stringers and skin should be generated and then meshing process takes place. After that, material properties, loads and boundary conditions are defined.

Analysis is performed with thin panels, which are stabilized in out of plane direction by Z section stiffeners. Z section type is used because of their widespread use in aerospace industry [32]. The Z type has inertia and assembling advantages. These stringers are connected to the skin by equally spaced "Rivet" type connections. All parts are modeled with 2-dimensional shell elements with material properties of aluminum 2024. Both compression and shear loads are applied with same magnitude

21

from the skin edges to the structure. This loading case corresponds to compression to shear ratio of 1.

A 3-stringer stiffened panel is shown in Figure 2.8 and the cross section of the stringer with skin panel is shown in Figure 2.9. All design parameters needed for a generation of the FE model are illustrated in these figures. In all the finite element analysis, there is a relation between the skin width (b_{skin}), the stringer spacing (d_{str}) and the number of stringers (n_{str}) which is given in Equation (2.10).

$$b_{skin} = d_{str} \times (n_{str} + 1) \qquad (2.10)$$

It should be noticed that the distances between the stringers are kept constant and the reason why the 3-stringer combination is used will be explained in the first case study.
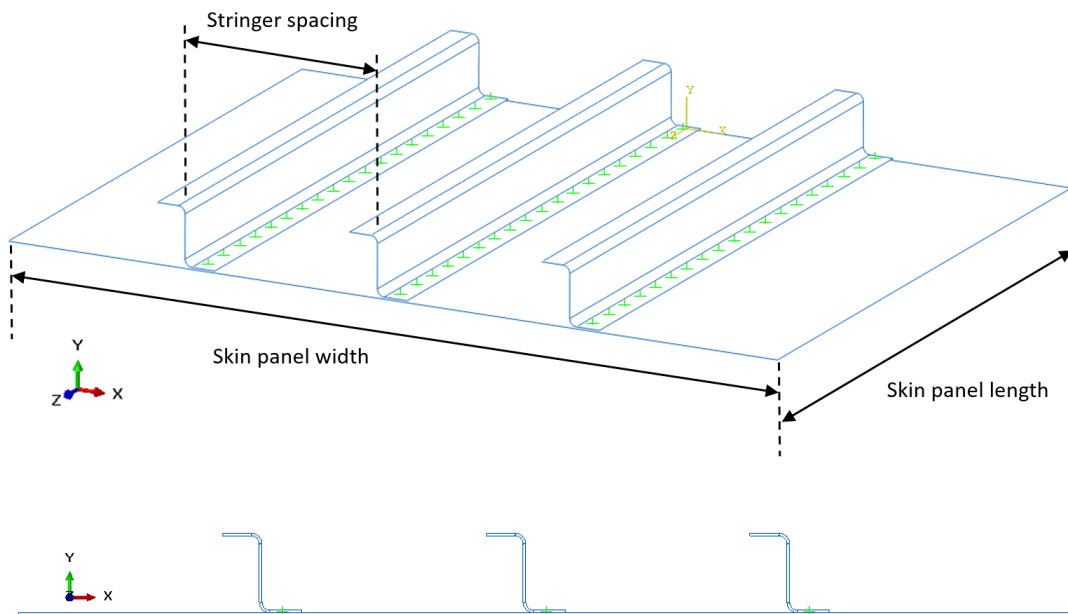
Figure 2.8: 3-stringer stiffened panel isometric and section view
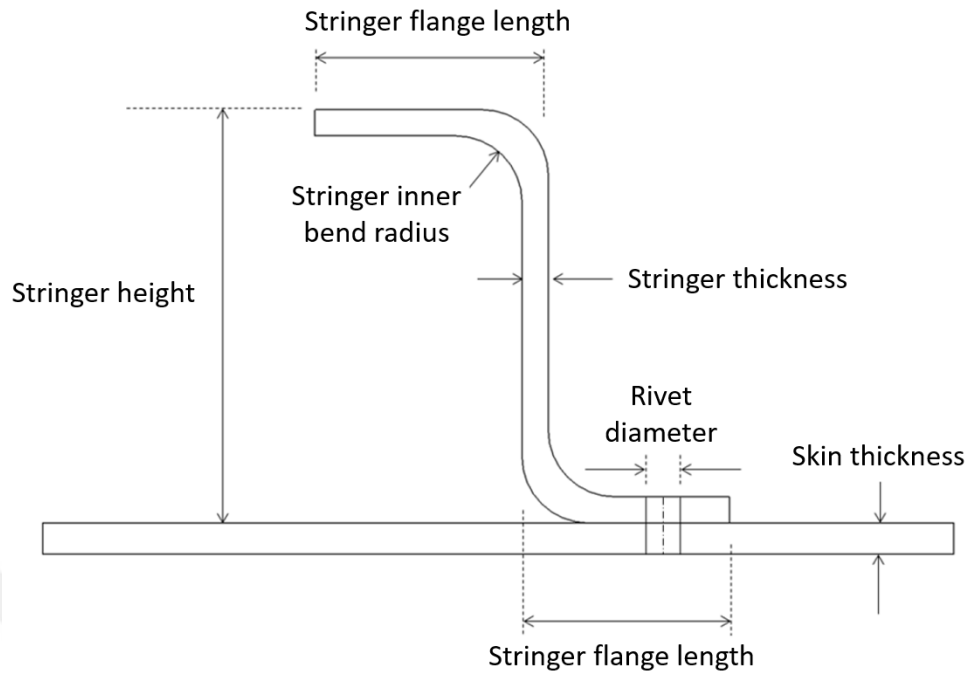
Figure 2.9: The Z type stringer cross section

2.2.1. Analysis Geometry and Meshing

The skin panel is the sheet metal plate with 300x400x1.016 mm dimensions. Besides, the stringer is also sheet metal plate and has 30 mm height, 15 mm flange length and 1.016 mm thickness. The skin panel dimensions and cross-section of the skin-stringer assembly are shown in Figure 2.10 and tabulated in Table 2.2.

Bend radius of the stringer is 3 mm and the rivet diameter is chosen as 3.2 mm for all skin-stringer assemblies, since the diameter of the most common and smallest rivet (for weight gain purposes) is 3.2 mm. Additionally, the distance between fasteners is suggested to be between 4 times and 6 times of the rivet diameter in literature [33]. In this study, it is chosen to be 5d. These three values are constant through the whole study. Finite element model of this sample skin-stringer assembly is shown in Figure 2.11.
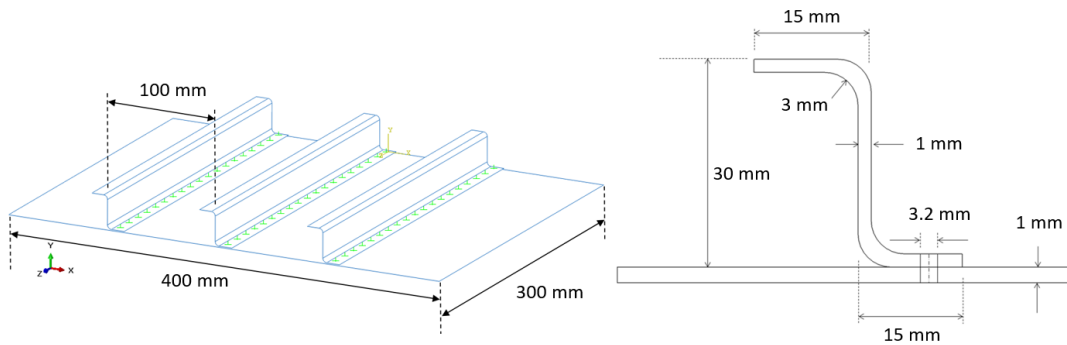
Figure 2.10: All input dimensions for the analysis

Table 2.2: Dimensions of the Skin-Stringer Geometry

Skin Panel Width [mm]	400
Skin Panel Length [mm]	300
Skin Panel Thickness [mm]	1.0
Stringer Height [mm]	30
Stringer Flange Length [mm]	15
Stringer Thickness [mm]	1.0
Stringer Inner Bend Radius [mm]	3 (constant)
Rivet Diameter [mm]	3.2 (constant)
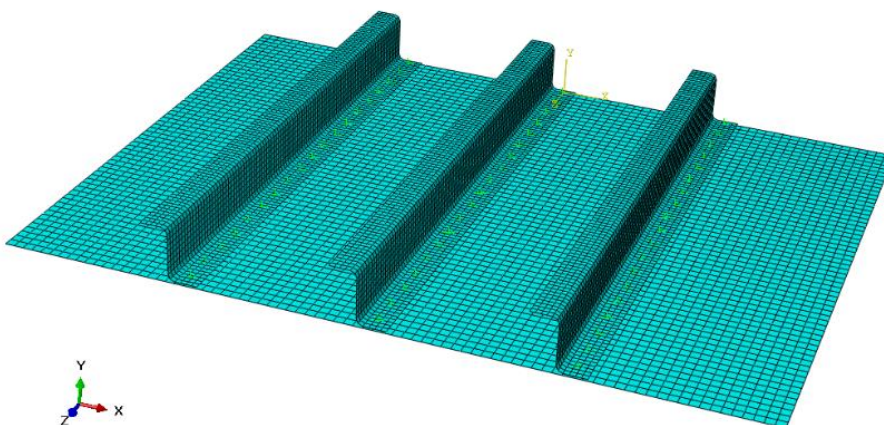
Figure 2.11: Finite element model of the structure reinforced with three stringers

2.2.2. Material Properties

In previous parts, the material used in FE analysis is stated as aluminum. There are many options for its series and temper selection. In this study, 2024-T3 clad sheet aluminum is selected throughout the entire FE analyses.

2000 series alloys are commonly used in aircraft industry for sheet metal applications. For this series, copper is the primary alloying element (1.5% magnesium, 0.5% manganese, 4.5% copper). 2024-T3 alloy is solution heat treated, cold worked, and naturally aged to a substantially stable condition. Elastic material properties used in the modelling are shown in Table 2.3.

Table 2.3: Material Properties of Aluminum 2024-T3 Clad Sheet

Aluminum 2024-T3 Clad	
Young's Modulus [MPa], E	72395
Poisson's Ratio, υ	0.33
Density [kg/m3], ρ	2768
Compression Yield Stress [MPa], F_{cy}	296

2.2.3. Loads and Boundary Conditions

The assembly is subjected to both compression and shear loading. In this chapter, compression and shear load ratio is fixed to 1 and the simply supported edges are used as boundary conditions.

The loads applied to the assembly are shown in Figure 14. Unit compression load is applied from one skin panel edge (edge AB) through the stringer direction (z-direction in global axis). Therefore, the translation of the opposite edge is restricted in the same direction. Unit shear loads are applied from all edges of the skin panel as shown in Figure 2.12. ABAQUS "shell edge load" module is used for all steps of the loading.
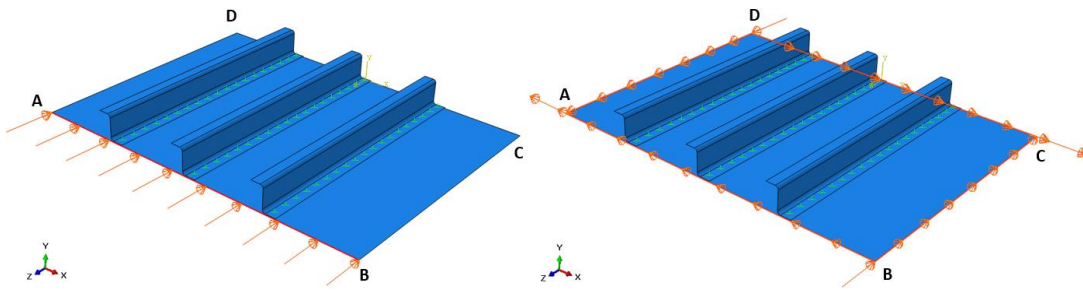
Figure 2.12: Compressive and shear loading respectively

Simply supported boundary condition is chosen for this sample analysis which is shown in Figure 2.13. This condition allows rotation for the edges but no translation occurs in out of plane direction. For edges AD and BC, boundary conditions should simulate the skin-stringer connections. For this purpose, a trade-off study will be performed in following parts. A detailed look for boundary conditions is given in Table 2.4, which shows the restricted translations and rotations.
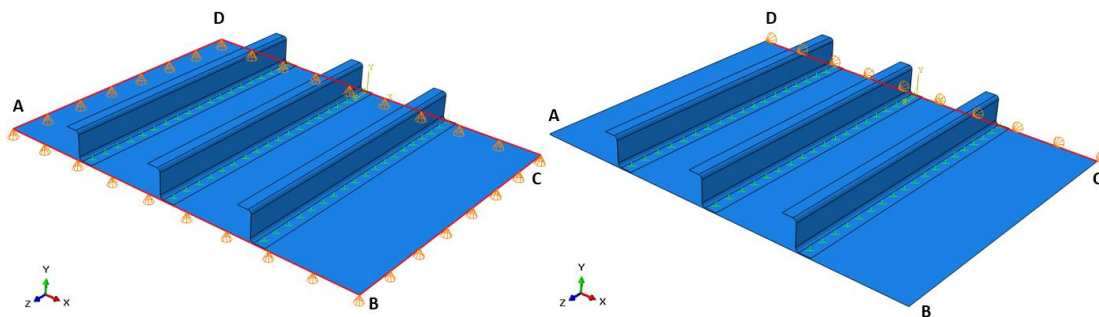
Figure 2.13: Boundary conditions for simply supported condition, as y-constraint (left figure) and z constraint (right figure)

Table 2.4: All constraints for simply supported condition where U1, U2, U3 are the translational constraints and R1, R2, R3 are the rotational constraints

Constraints	U1	U2	U3	R1	R2	R3
Point A	X	X	X			
Point B						
Point C		X				
Point D						
Edge AB		X				
Edge BC		X				
Edge CD		X	X			
Edge DA		X				

2.2.4. ABAQUS FEA Properties

ABAQUS finite element program is used to perform the eigenvalue buckling analyses. 2-D shell elements are preferred because of a faster solution and a faster modelling of both skin and stingers.

Firstly, geometry forming unit is used and section sketch is made. In the second step, material and property assignment are set. Then, assembly is created and fastener between the skin and the stringers are modeled. After that, all geometry is meshed with Quad elements.

For both compression and shear loading of the skin panel, "loading category" and "type for selected step" are chosen as "mechanical" and "shell edge load" respectively. This type of loading represents the amount of load per unit length. Therefore, N/mm is used for its unit.

Lastly, "Step Module" is used for the selection of the type of analyses. For the Procedure type, "Linear Perturbation – Buckle" step is chosen. There are two options for eigensolvers: Supspace and Lanczos. Although the Lanczos provides a faster solution, it has limitations and not suitable for complex problems. In order to converge the solutions, Subspace solver is used and the maximum iteration number is selected as "150". Because, some diverged results are obtained with lower iteration numbers.

2.2.5. Sensitivity Studies

There are some factors that need to be taken into consideration for the modelling the geometry. These are to obtain realistic results and to obtain these results in a reasonable time.

In this part, some trade-off studies are performed in order to verify the finite element model characteristics of the skin-stringer assembly. Accordingly, the results of different analyses with different parameters are compared for the baseline characteristics and they are used later in the analyses section. Sensitivity studies are performed for the finite element size and finite element type. The values of the element size are chosen as 3 mm for the stringer and 5 mm for the skin panel, and the element type is chosen as S4R shell element, which is 4-node doubly curved thin or thick shell, reduced integration, hourglass control, finite membrane strains.

For the element size, the objective is to find the number of elements at stringers and skin panel where the critical buckling stress value converges. With this intention, skin panel element size and the stringer element size are studied separately by making various analyses with different element sizes. The number of elements versus the first eigenvalue curves for the skin panel and the stringers are generated in Figure 2.14 and Figure 2.15, respectively. As can be seen in the figure 16, the first eigenvalue converges after the number of elements exceeded 6400 in the skin panel section. This number corresponds to 5 mm element size. Similarly, the curve converges after the number of elements exceeded 2500 for the stringer and it corresponds to 3 mm element size. In Figure 2.16, the enlarged view of meshed skin panel and a stringer is shown.

28
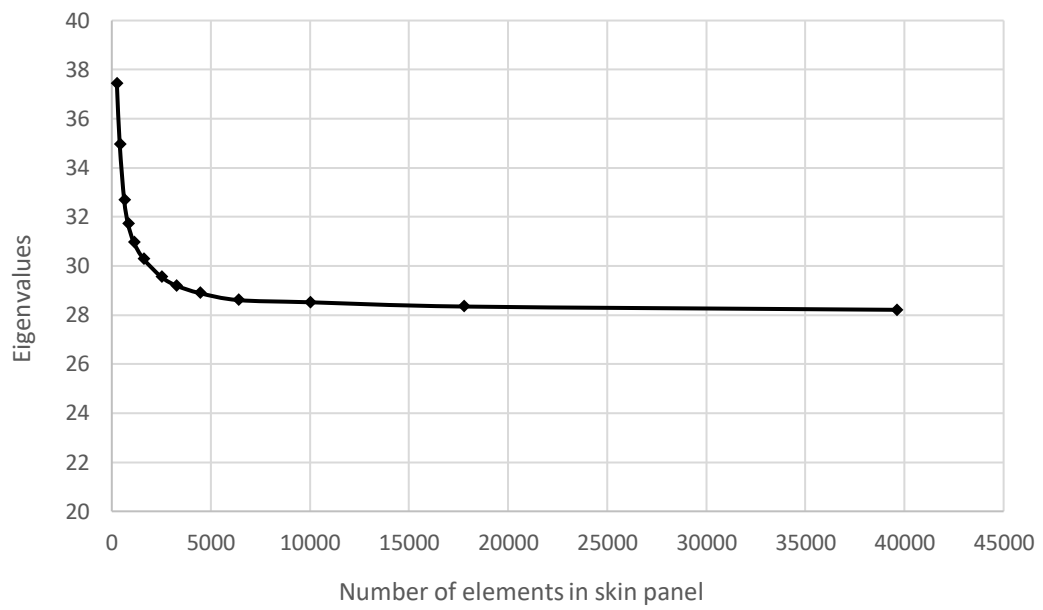
Figure 2.14: Eigenvalues for different number of elements at skin panel section
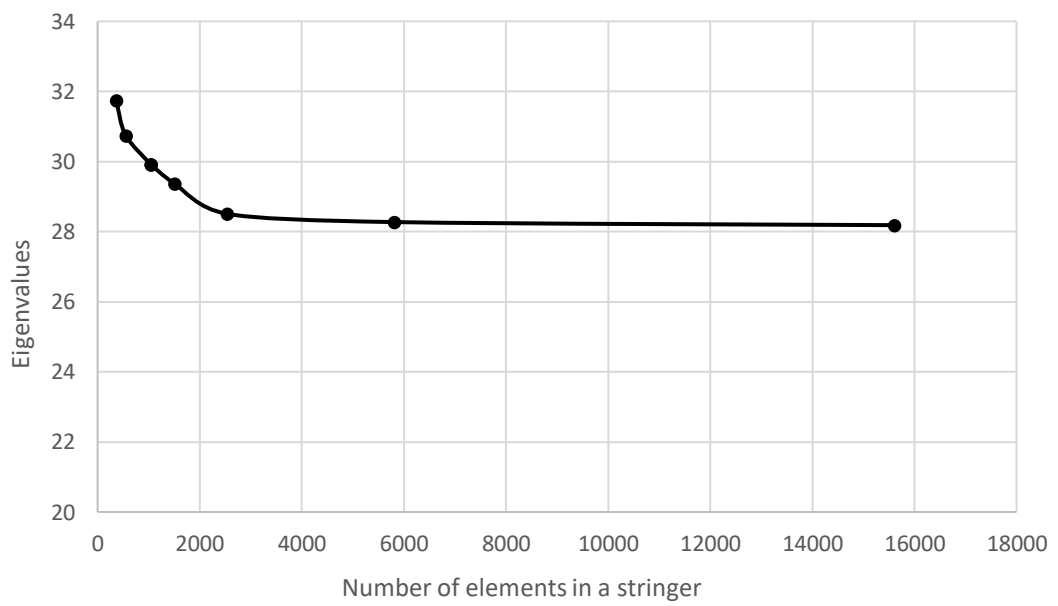
Figure 2.15: Eigenvalues for different number of elements at stringer section
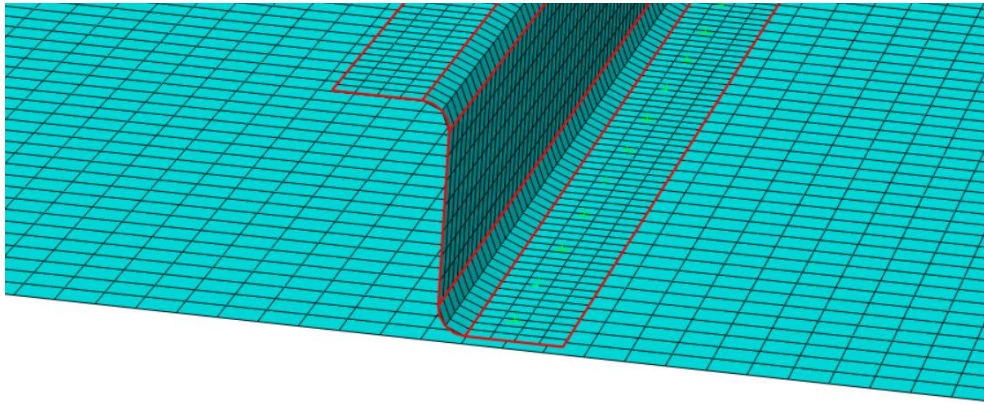
Figure 2.16: Enlarged view of meshed skin and a stringer

In addition to element size, a trade-off study is performed for the element type selection. It is found that the S4R (a 4-node element with reduced integration, having hourglass control and finite membrane strain) element type has the most suitable features compared to the other element types.

In this study, following element types are considered and compared with their solution times and eigenvalue outputs as given in Table 2.5.

- S4: 4-node doubly curved general-purpose shell, finite membrane strains.
- S4R: 4-node doubly curved thin or thick shell, reduced integration, hourglass control, finite membrane strains.
- S8R: 8-node doubly curved thick shell, reduced integration (small strains).
- S8R5: 8-node doubly curved thin shell, reduced integration, using 5 DOFs per node (2 in-surface rotations).

Table 2.5: Sensitivity study results for different element types

Element type	Time [sec.]	First Eigenvalue
S4	45	28.607
S4R	43	28. 730
S8R	108	27.538
S8R5	138	27.206

30

Although all of above element types can be used for general-purpose shell, the small strain element types are not suitable for eigenvalue problems [29]. Besides that S8R and S8R5 element types are not good in terms of computational resources. The advantage of using S4 and S4R is that they take into account the change in the shell thickness. S4R is reduced integration (lower-order) and has one integration point per element, while S4 is fully integrated and has four integration points per element, which makes the computation more expensive compared to S4R [34]. A slight difference between their computation times can be seen in   Additionally, S4R has more conservative approach than S4. Therefore, S4R element type is used throughout this thesis.

2.2.6. Finite Element Results of Skin-Stringer Buckling Problem

As stated in the previous sections, the aim of the FE analyses is to obtain critical buckling curves for a specific geometry and to determine the margin of safety for given combined compression and shear loadings.

As a sample result, Figure 2.17 shows a critical buckling curve for the input geometric parameters given in Section 2.2.1. The area below this curve represents the safe zone, while the area above represents the failure zone. Safe zone means the margin of safety is greater than zero, while the failure zone means that the M.S. is lower than zero.

Three different analyses for three different compression-shear load ratio are chosen in order to show the deformation of each analyses. All remaining parameters are the same for all analyses. Figure 2.18 shows the deformation of the pure shear loading point "A", while the Figure 2.19 shows the deformation of the pure compression loading point "C". Deformation of the combined loading (with compression/shear load ratio 1) is shown in Figure 2.20. All buckling analysis results are obtained for the first eigenvalue.

In addition to the graphical approach, critical buckling curve can be drawn by the analytical approach in Figure 2.17. The equation for combined loading for critical buckling was given in Equation (2.2):

$$R_L + R_S^2 = 1.0 \qquad [3] \qquad\qquad (2.2)$$
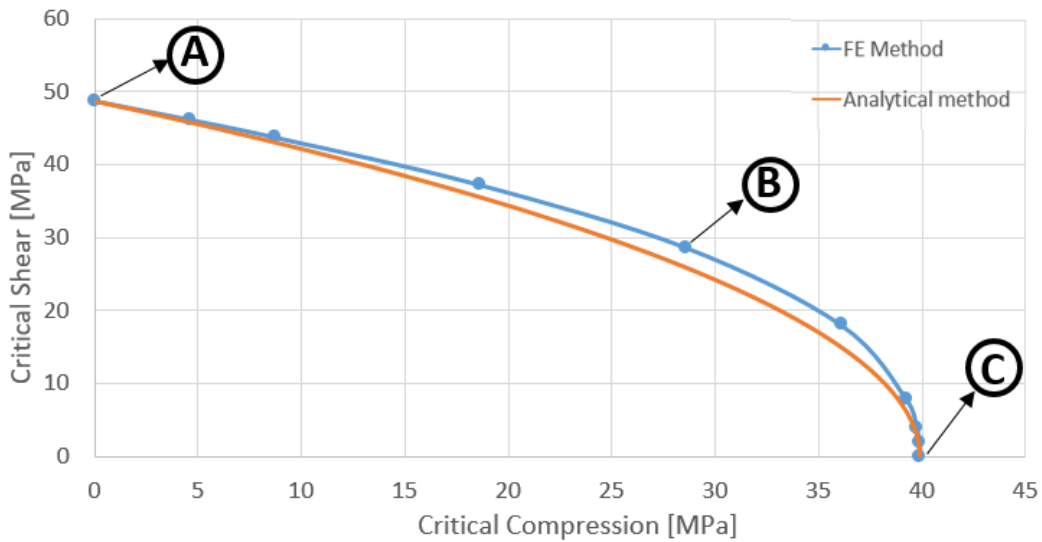
Figure 2.17: Critical Buckling Curve for a specific geometry
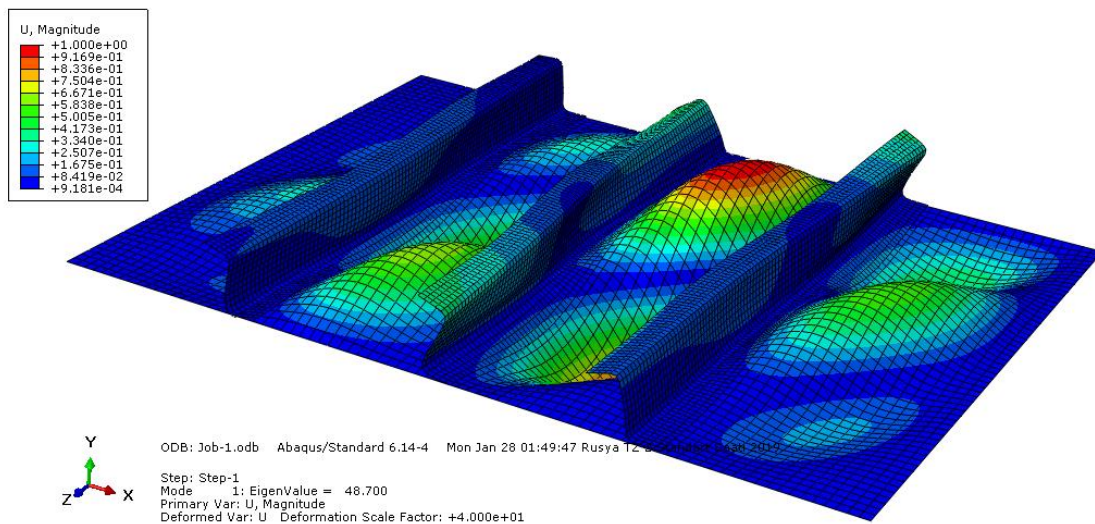
Figure 2.18: Buckling deformation of pure shear loading (Point A)
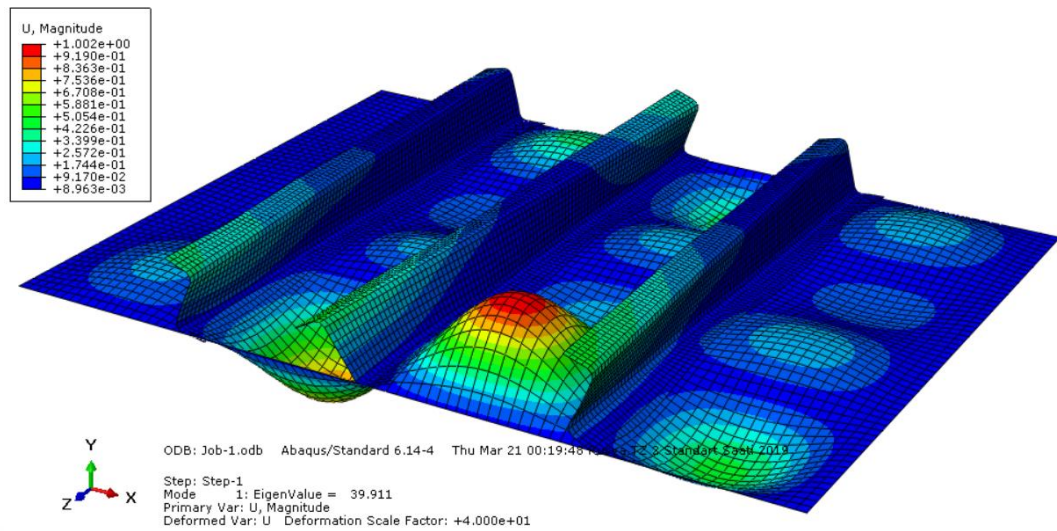
Figure 2.19: Buckling deformation result of pure compression loading (Point C)
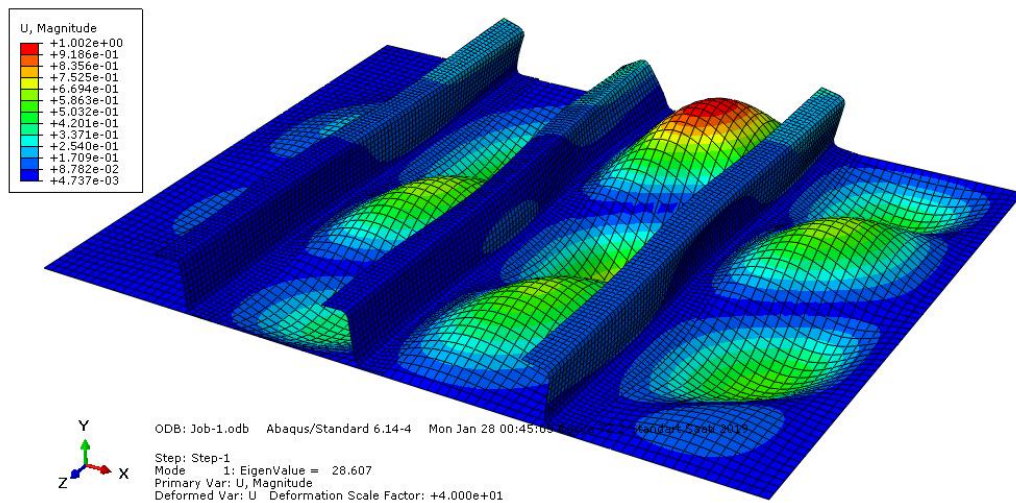
Figure 2.20: Buckling deformation result of combined loading (Point B)

For a better understanding the margin of safety calculations in the script, sample analyses need to be performed with the help of graphical method. The geometry is the same for both cases. In Figure 2.21, one of the applied combined loads is in the safe

region, while the other is in the failure region. The allowable loads, having the same compression-shear ratio with the combined applied loads, can be computed by linear interpolation between two neighbor allowable points. Their M.S. values are shown in Table 2.6. In addition to graphical approach, the MS values can also be calculated analytically by using Equation (2.2). MS calculation steps are given in Equation (2.11) and (2.12). For a better comparison, the results of the analytical approach is given in Table 2.6. The difference between the MS's may be due to the difference between the critical buckling curves in Figure 2.17. Other than that, given methods have different perspective of computation.

$$1 = \frac{1.0}{R_L + R_S^2} \tag{2.2}$$

$$MS = \frac{1.0}{R_L + R_S^2} - 1 \tag{2.2}$$
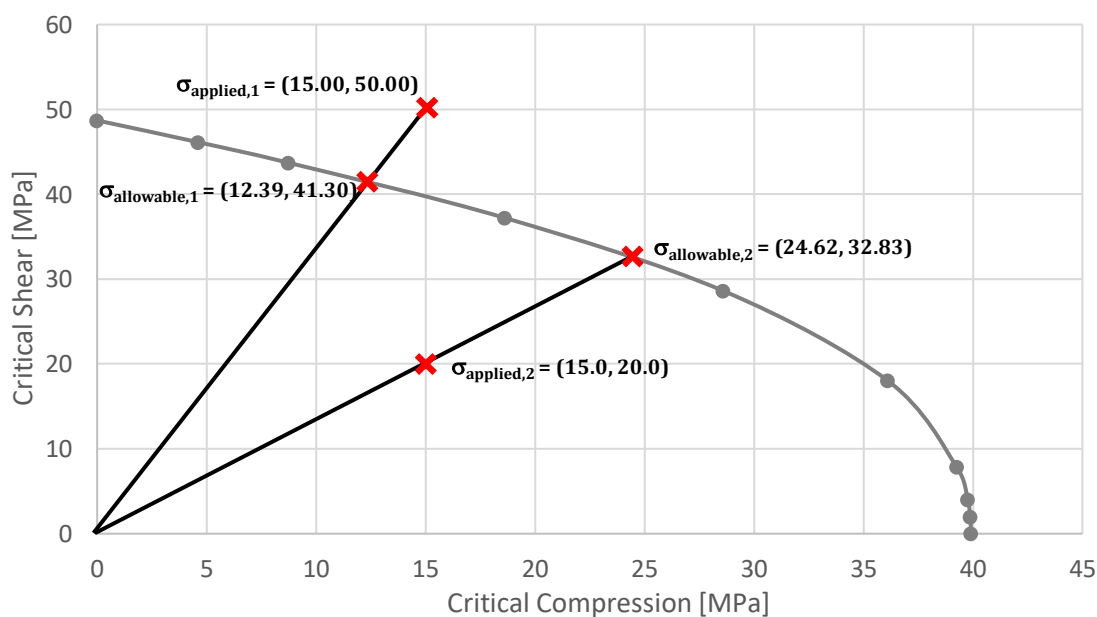
Figure 2.21: Critical buckling curve and two different loading for a given geometry

Table 2.6: Margin of safety and failure status for given designs

	Analysis - 1	Analysis - 2
Applied Loads	(15.0, 50.0)	(15.0, 20.0)
Allowable Loads	(12.39, 41.30)	(24.62, 32.83)
MS (graphical)	-0.17	+0.64
MS (analytical)	-0.30	+0.84
Status	FAILURE	SAFE

CHAPTER 3

CASE STUDIES

In this part, some case studies are performed before proceeding with the artificial neural network and optimization process. These studies help building a representative finite element model. In order to compare the analysis options and to prove that the final model bears trustworthy force reaction and stress results, a few critical settings are studied in this section. These studies are listed below:

- Finding a stiffened panel with a minimum number of stringers that has the same buckling behavior with a larger model that contains more stringers (determination of representative FE model size)

- Finding boundary conditions representing side edge stringers

- The effect of some critical design parameters on buckling behavior (Trade-off studies)

In this section, only one compression/shear ratio is considered for all analyses (compression/shear = 1.0). Therefore, there is only one compression load and one shear load for each geometry, not a curve.

3.1. Finding a Stiffened Panel with a Minimum Number of Stringers

In this study, various models that have different number of stringers are compared for buckling behavior in order to make a decision for representative FE model. In all analyses, the stringer spacing between stringers is the same and the total panel width increases with increasing number of stringers. The skin panel width is a function of

the stringer spacing and the number of stringers. The number of stringers are 1, 2, 3, 4, 5, 7, 10 in the FE models and stringer spacing is fixed to 100 mm. Total skin width calculation is given in Equation (3.1). In the equation, b_{skin} is the skin width, d_{str} is the stringer spacing and n_{str} is the number of stringers. All other parameters are fixed to specific values for seven analyses considered and they are shown in Table 3.1.

$$b_{skin} = d_{str} \times (n_{str} + 1) \tag{3.1}$$

Table 3.1: The values of parameters for all models with different number of stringers

Stringer Spacing [mm]	100.0
Skin Panel Length [mm]	400.0
Skin Panel Thickness [mm]	1.0
Stringer Height [mm]	30.0
Stringer Flange Length [mm]	15.0
Stringer Thickness [mm]	1.0

In Figure 3.1, the displacements and the first eigenvalues for all analyses are shown. The mode shape results show that the first eigenvalues are observed in similar locations of analyses. From the outputs, it appears that the first eigenvalues converge to a specific value for three or more stringers combination. For this reason, it can be concluded that a minimum number of three-stringer structure would be sufficient in order to obtain the first eigenvalues correctly. Figure 3.2 shows the change of first eigenvalues with respect to the number of stringers. In Figure 3.1 and Figure 3.2, all analyses are performed for a combined loading with a compression to shear load ratio of "1".

A study conducted by Örün [35], the effect of aspect ratio of the web plate is investigated for the thin-walled beams under pure compression, pure bending and pure shear loadings. It can be concluded that the more slender the web plate, the critical buckling stress converges to a certain value in all type of loadings.
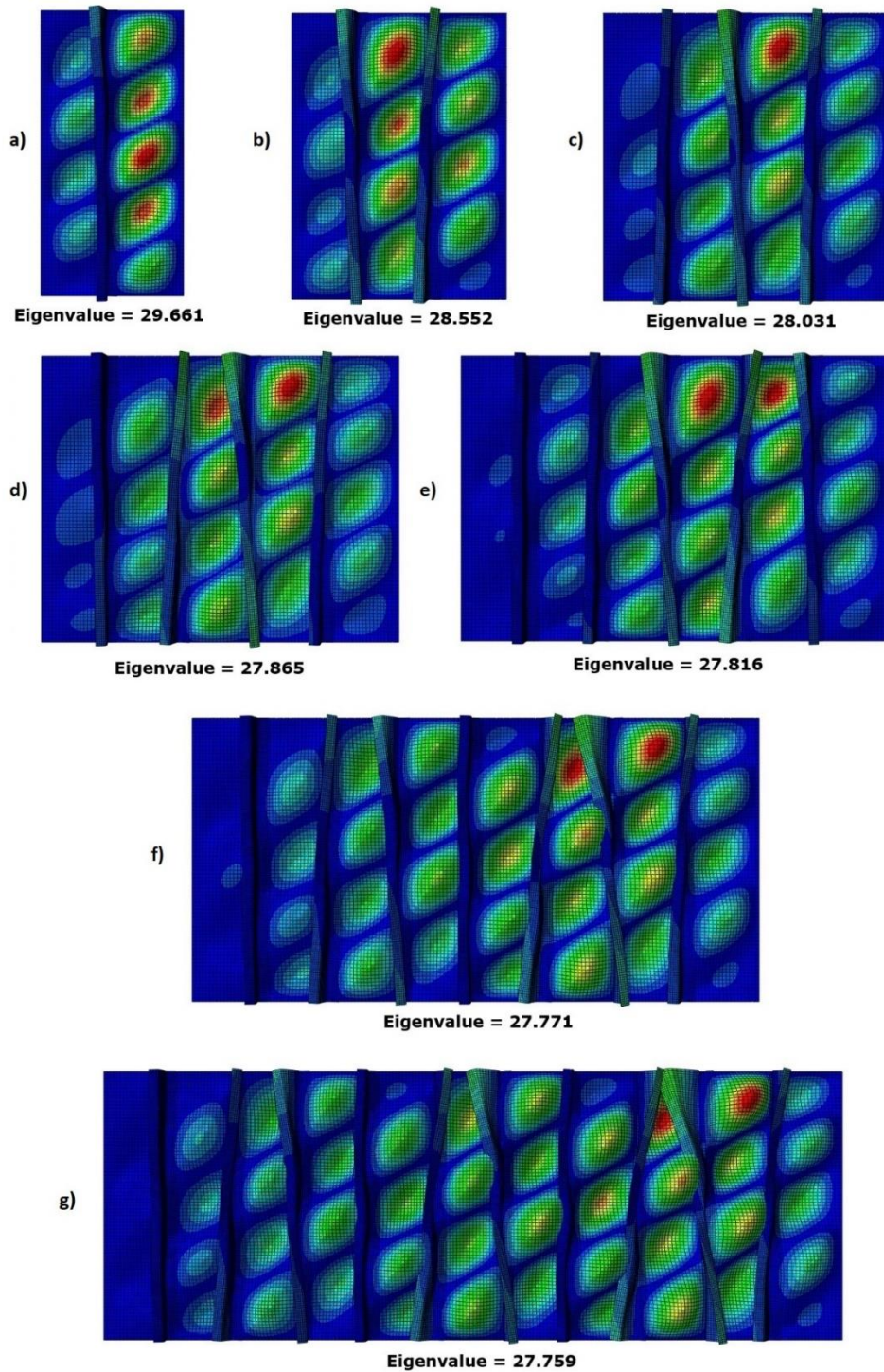
Figure 3.1: Displacement and the mode shapes of the first Eigenvalues of the models
a) with one stringer b) with two stringers c) with three stringers d) with four stringers
e) with five stringers f) with seven stringers f) with ten stringers
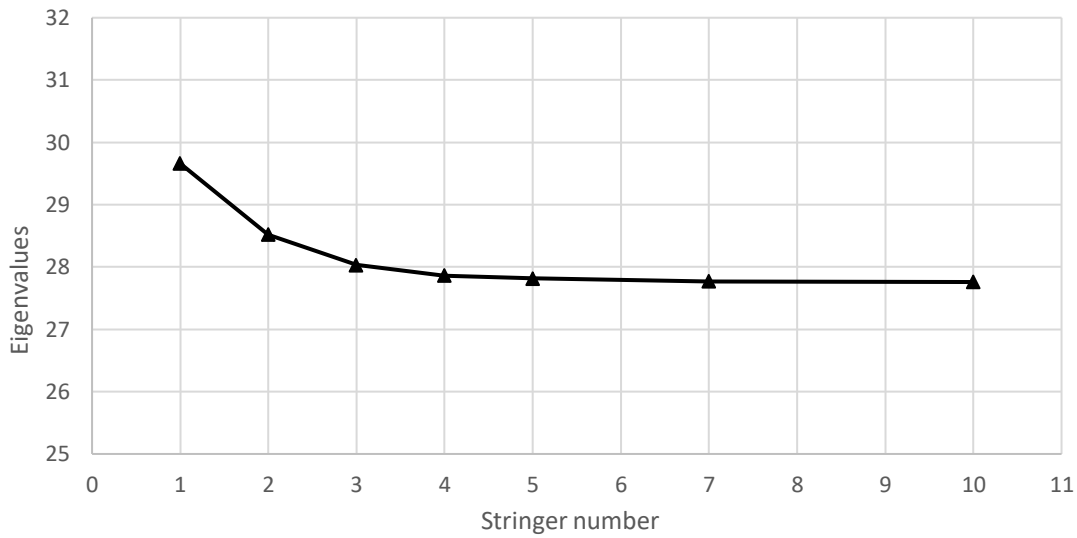
Figure 3.2: Change of the first eigenvalue with respect to the number of stringers for a compression-shear ratio of "1"

It is important to check the pure loadings results for this case study. For this reason, the same steps followed for a combined loading are performed for pure compression and pure shear loadings. Figure 3.3 and Figure 3.4 shows the change of the first eigenvalues with respect to the number of stringers in the finite element model for pure compression and pure shear loading, respectively. According to the results, it can be concluded that 3 stringer model is representative of models with higher number of stringers having the same stringer spacings. It also has advantages about computational cost and computational time. Therefore, this configuration will be used in all ANN trainings.
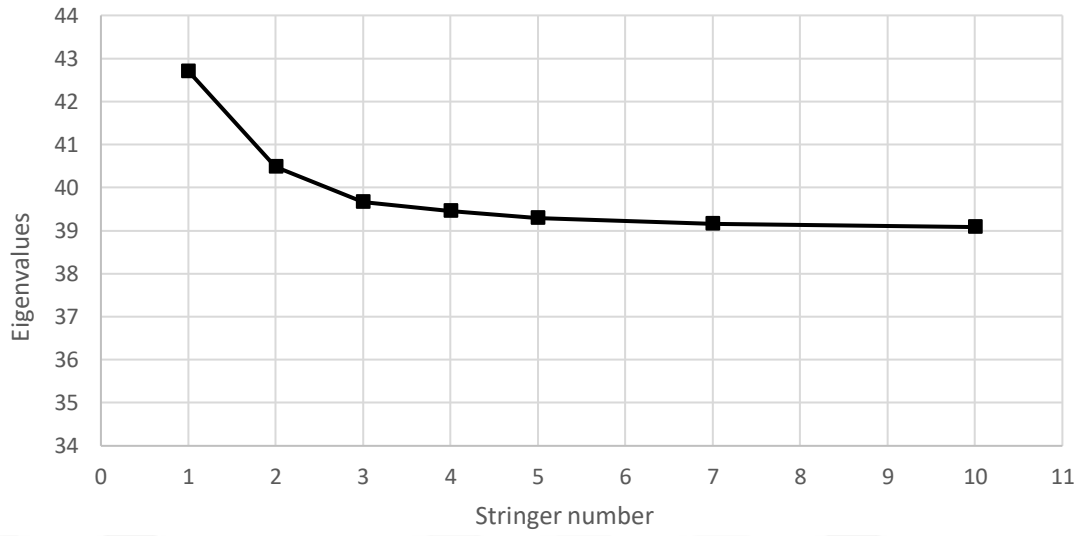
Figure 3.3: Change of the first eigenvalue with respect to the number of stringers for pure compression loading
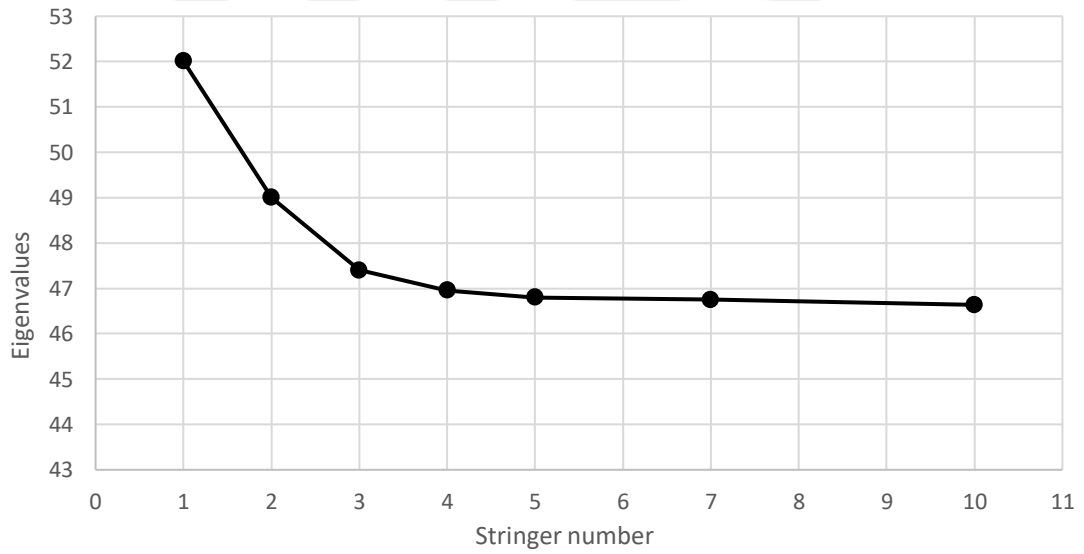
Figure 3.4: Change of the first eigenvalue with respect to the number of stringers for pure shear loading

3.2. Finding Boundary Conditions representing Side Edge Stringers

In this study, appropriate boundary conditions that represent side edge stringers are investigated. For this purpose, two different skin-stringer structures are analyzed with and without side edge stringers. In both analyses, the stringer spacing between stringers and panel dimensions are fixed to specific values. The only difference between the two models is whether there are stringers on the side edges or not. The skin panel and the stringer dimensions for the FE models are shown in Table 3.2. All analyses are performed for a combined loading, which has compression to shear load ratio of "1".

Table 3.2: The values of parameters for both models

Skin Panel Width [mm]	400.0
Skin Panel Length [mm]	400.0
Stringer Spacing [mm]	100.0
Skin Panel Thickness [mm]	1.0
Stringer Height [mm]	30.0
Stringer Flange Length [mm]	15.0
Stringer Thickness [mm]	1.0

For the first analysis, a five-stringer structure is modeled by placing two stringers on the side edges and no boundary condition is applied to these edges. The first eigenvalue is found to be 28.866. The first mode shape of the model is given in Figure 3.5.
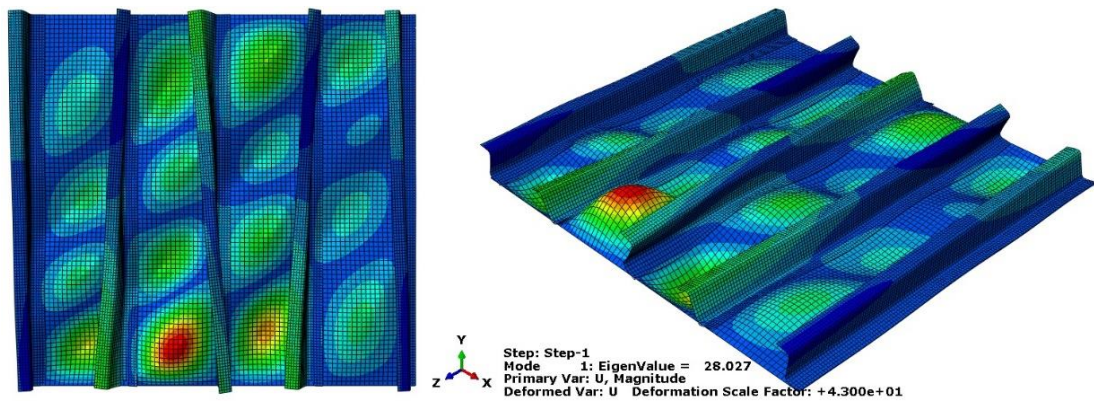
Figure 3.5: The first mode shape of the model with side stringers

In the second step, a three-stringer structure is modeled without side edge stringers. Both simply supported and clamped conditions on side edges are tested instead of placing stringers. For the simply supported case, the first eigenvalue is found to be 24.268 and for the clamped case, the first eigenvalue is found to be 29.461. The first mode shapes for these configurations are shown in Figure 3.6 and Figure 3.7, respectively.
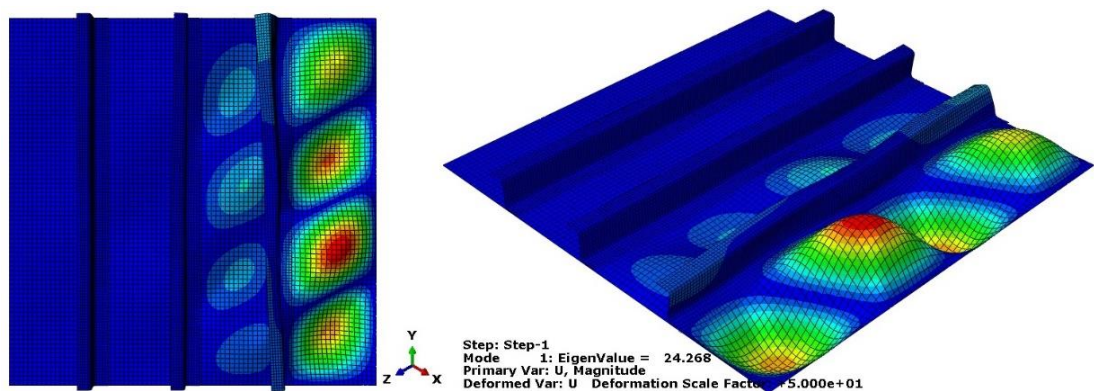
Figure 3.6: The first mode shape of the model without side stringers where side edges are simply supported
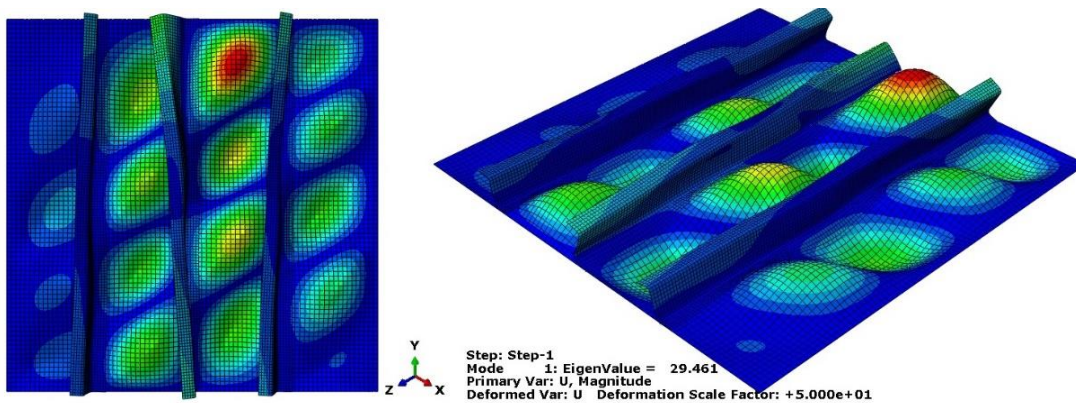
Figure 3.7: The first mode shape of the model without side stringers where side edges are clamped

From these results, it is found that neither the clamped nor the simply supported conditions represents the actual boundary condition of the side edge stringers. While the first eigenvalue comes up smaller for the simply supported case, it becomes larger for the clamped case. Therefore, it can be stated that the real boundary condition representing the stringers, must be in between these two conditions. Nevertheless, the first eigenvalue of the clamped case is closer to the model with side stringers. A research by Aydın et al [36] examines the boundary conditions instead of a stringer in a stiffened panel by comparing the finite element method and the analytical method. The study shows that the finite element method gives closer but slightly higher eigenvalue results when the clamped condition is applied to the side edges. At this point, effective width phenomenon from the literature helps to determine the appropriate width of the stiffened panel model.

In the design of stiffened panels, if post-buckling process is taken into account, the effective width concept is frequently used. When a panel buckles between two stiffeners, a portion of the load of the panel shifts to the stringers and to an effective section of the skin panel [4]. That means, the buckling of the plate takes place outside this effective width region. Stringers and effective skin panel section carry the excess

44

load until collapse. The distribution of the loads and the idealization for the computation at the moment of buckling are illustrated in Figure 3.8 [3].
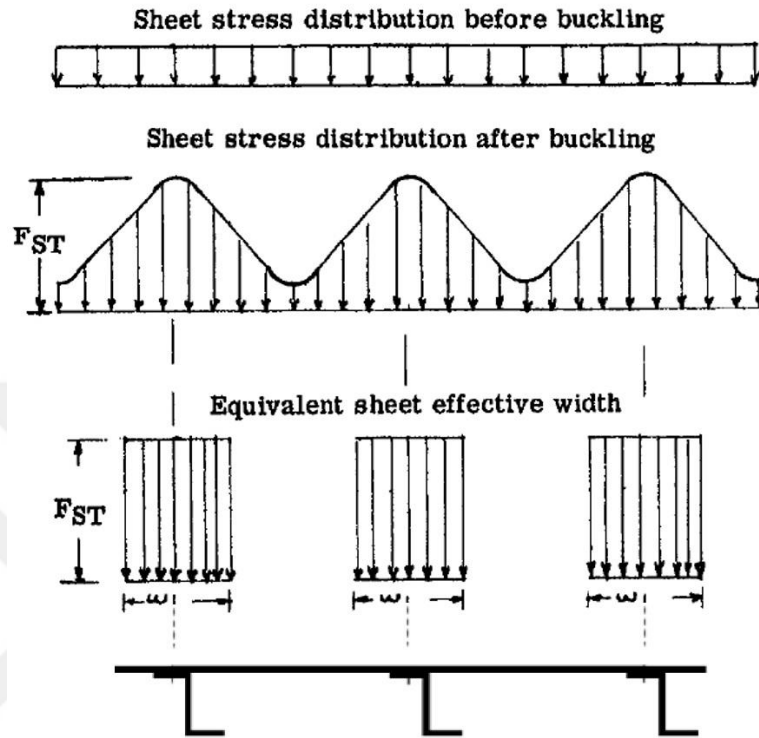
Figure 3.8: Stress distribution and idealization of a skin-stringer structure before and after the buckling [3]

In Fairchild Dornier Calculation Method for airframe static stresss strength and stability analysis, calculation of the effective skin width is stated as [37]:

$$b_{eff} = \left(\frac{E_{c,skin}}{E_{c,stiffener}}\right) \times k_{eff} \times t_{skin} \times \sqrt{\frac{E_{c,skin}}{F_{cy,stiffener}}} \tag{3.2}$$

where E_c is the compression Young's Modulus, k_{eff} is the effective width factor of the material and F_{cy} is the compression yield allowable.

- If stringer is machined → $k_{eff} = 1.9$
- If stringer is a sheet metal → $k_{eff} = 1.7$

In this study, materials of the skin panel and the stringer are always sheet metal and the same. Therefore, the first term of the equation $\left(\dfrac{E_{c,skin}}{E_{c,stiffener}}\right)$ is always "1" and the equation (3.2) can be rewritten as,

$$b_{eff} = 1.7 \times t_{skin} \times \sqrt{\frac{E_{c,skin}}{F_{cy,stiffener}}} \tag{3.3}$$

$$= 1.7 \times 1.0 \ \times \sqrt{\frac{72395}{296}} \ = \ 26.59 \text{ mm}$$

From this knowledge, the panel in three-stringer model is extended symmetrically by the effective width amount and reanalyzed for the clamped condition. This reduces the first eigenvalue of the classic clamped condition and a much closer first eigenvalue to the five-stringer model is reached. A closer look at the geometries that have been analyzed throughout this case study is given in Figure 3.9. Note that effective skin width is independent from the applied loads and the number of stringers by its definition.
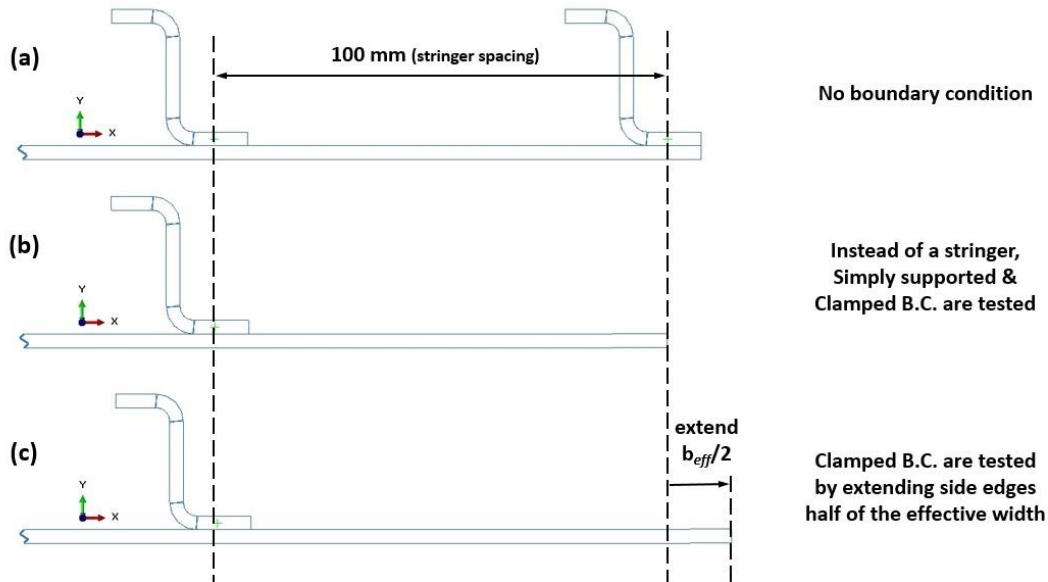
Figure 3.9: The cross-section of tested geometries (a) Five-stringer model without a displacement/rotation boundary condition at side edges (b) Three-stringer model with simply supported and clamped boundary conditions at the side edges (c) Three-

stringer model with symmetrically extended panel by the effective width amount with clamped boundary conditions at the side edges

For the geometry created with this approach, the first eigenvalue is found to be 27.981 and the mode shape obtained is very similar to the five-stringer model. The mode shape of the final configuration is shown in Figure 2.29. The results of this study are summarized in Table 3.3.
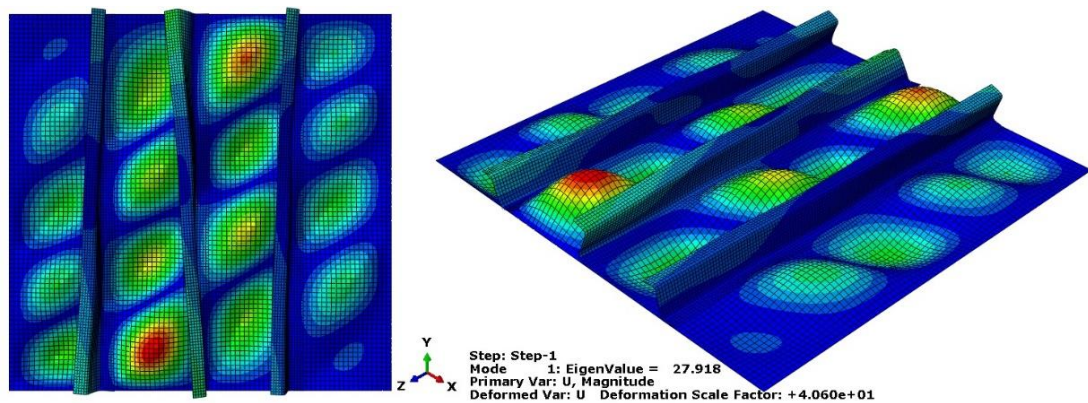
Figure 3.10: The first mode shape of the three-stringer model with symmetrically extended panel by the effective width amount with clamped boundary conditions at the side edges

Table 3.3: Summary of 5-stringer and 3-stringer models with different boundary conditions

Model Description	Boundary Condition (for side edges)	First Eigenvalue	Error compared to base model
5-stringer model (BASE)	No B.C. (stringers at sides)	28.027	-
3-stringer model	Simply supported	24.268	13.41%
3-stringer model	Clamped	29.461	5.12%
3-stringer model (by extending side edges 13.3 mm.)	Clamped	27.918	0.39%

For this case study, a total of five different sets of geometry is created and each set is analyzed for three different compression-shear load ratios. The results of only one of them is represented in previous tables and figures, which is Set-1 with compression-shear ratio of "1". All geometric parameters for each analysis are given in Table 3.4. Each analysis is repeated for both the five-stringer model and the three-stringer model. According to the results, the first eigenvalues for the three-stringer models show an average 1.13% difference in comparison to the five-stringer models. The maximum error from all alternatives is found as 5.11%.

Table 3.4: All design parameters for 15 different analyses

Parameters	Set-1	Set-2	Set-3	Set-4	Set-5
Skin Panel Width [mm]	400.0	300.0	400.0	400.0	400.0
Skin Panel Length [mm]	400.0	400.0	400.0	400.0	400.0
Stringer Spacing [mm]	100.0	100.0	80.0	100.0	100.0
Skin Panel Thickness [mm]	1.0	1.0	1.0	0.81	1.0
Stringer Height [mm]	30.0	30.0	30.0	30.0	30.0
Stringer Flange Length [mm]	15.0	15.0	15.0	15.0	15.0
Stringer Thickness [mm]	1.0	1.0	1.0	1.0	0.81
Compression/Shear Ratio	0.25	0.25	0.25	0.25	0.25
	1.0	1.0	1.0	1.0	1.0
	4.0	4.0	4.0	4.0	4.0

3.3. Effects of Design Parameters on Buckling Behavior

In this part, a design problem is identified and a solution is searched. For this purpose, the critical buckling values with respect to some geometric parameters, obtained from the buckling FEA, will be presented. After investigating their effects on buckling, a comparison will be made in order to see which action is needed to solve the problem. The effects of three design parameters are examined and they are listed as

- Skin Thickness
- Stringer Spacing
- Stringer Thickness

In the beginning of the design process of an aircraft, an initial structure is provided by design groups for the use of analysis groups. Then, it is modelled with a finite element program and analyzed with given aerodynamics and inertial loads. If a region fails under given loads, sizing is performed to increase the allowable loads. On the other hand, if a region does not fail, this time sizing for lightening is performed to decrease the allowable loads. At the end of this study, for a skin-stringer structure that buckles under a combined shear and compression loading, it needs to be decided which parameter should be changed for an optimized structure.

The parameters of the examined structure are based on the sample analysis in Chapter 2.2. Throughout the whole trade-off studies, all parameters are selected from the inputs of this sample analysis except the parameter that is being examined. Additionally, the shear-compression load ratio is set to 1. In Table 3.5, all parameters of the skin-stringer structure are given.

Table 3.5: Base parameters of the skin-stringer structures in the trade-off studies

Skin Panel Width [mm]	400.0
Skin Panel Length [mm]	400.0
Stringer Spacing [mm]	100.0
Skin Panel Thickness [mm]	1.0
Stringer Height [mm]	30.0
Stringer Flange Length [mm]	15.0
Stringer Thickness [mm]	1.0
Shear-Compression Ratio	1.0

In order to see the effects of the examined parameter properly, the results are investigated in terms of critical stress per unit mass. At the end of this section, in addition to each trade-off study, a comparison will be made showing which parameter has more effect on the buckling behavior.

3.3.1. The Effect of the Skin Thickness

In this part, a trade-off study is carried out for skin panels with different thicknesses. While all other parameters are constant, only the skin panel thickness is changed and the analyses are performed for six standard sheet metal thickness: 0.635 mm, 0.813 mm, 1.016 mm, 1.27 mm, 1.42 mm and 1.62 mm. The values of the other parameters for the analyses are given in Table 3.5.

Table 3.6 shows the critical buckling load and mass results from FEA for different skin thicknesses. In order to see the effects of the skin thickness change on buckling, critical load per unit mass is also computed. For a better comparison, both the critical load and the critical load per unit mass are drawn in Figure 3.11. In the figure, as the skin thickness is increased, the critical load is increased as expected. This phenomenon is also represented analytically in chapter 2.2.1. In addition to this rise, one can see that the critical load per unit mass is increasing also. If this parameter is increasing, which means the slope is positive in the graph; the load carried by unit mass is also increasing.

Table 3.6: FEA results of skin thickness trade-off study

Skin Thickness [mm]	Critical Load [MPa]	Critical Load per unit mass [MPa/kg]	Mass [g]
0.635	8.62	17.15	502.67
0.813	16.64	28.33	587.49
1.016	28.03	41.49	675.58
1.270	51.22	63.61	805.24
1.420	67.94	77.49	876.71
1.620	95.76	98.04	976.77
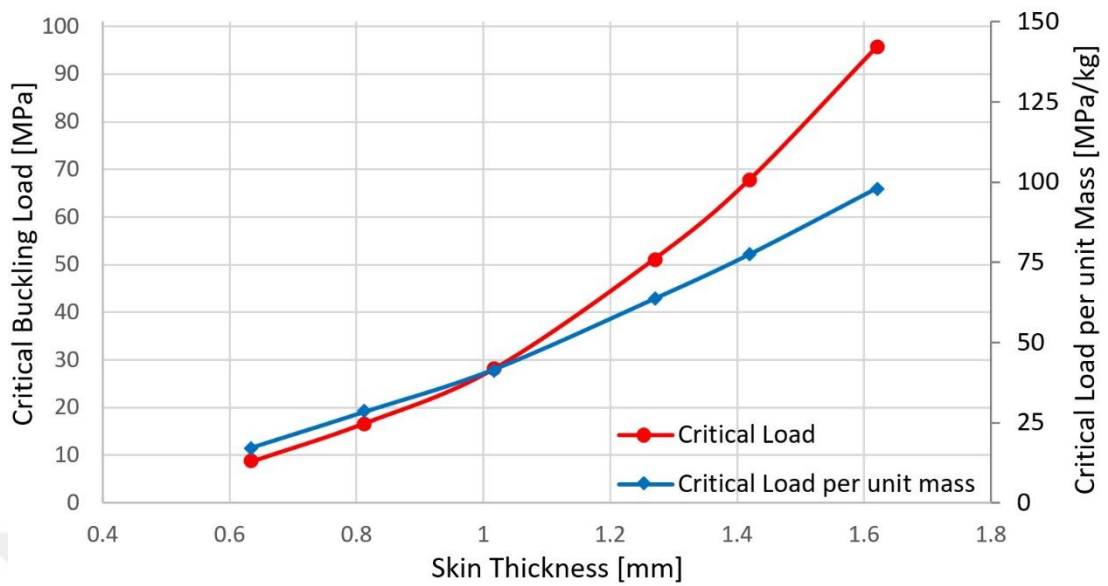
Figure 3.11: The effect of skin thickness on buckling load capacity

3.3.2. The Effect of the Stringer Spacing

In this part, a trade-off study is carried out for the stringer placements spacing. While all other parameters are kept constant, only the stringer spacing is changed and thus the number of stringer is changed for a stiffened panel of 400 mm width. The analyses are performed for six different stringer spacings: 133.3 mm, 100 mm, 80 mm, 66.67 mm, 57.14 mm, and 50 mm. The values of the other parameters for the analyses are given in Table 3.5.

Table 3.7 shows the critical buckling load and mass results for different stringer spacing values. In order to see the effects of the stringer spacing on buckling, *critical load per unit mass* is also given. For a better comparison, both the *critical load* and the *critical load per unit mass* are drawn in Figure 3.12. In the figure, as the stringer spacing is decreased, the critical load is increased as expected. Moreover, one can see that *the critical load per unit mass* is also increasing which means that the decrease in the stringer spacing is an efficient way to increase the critical buckling load.

51

Table 3.7: FEA results of stringer spacing trade-off study

Stringer Spacing [mm]	Critical Load [MPa]	Critical Load for unit mass [MPa/kg]	Mass [g]
133.3	16.13	26.44	609.87
100.0	28.03	41.49	675.58
80.0	44.16	59.41	743.29
66.7	62.26	76.86	809.99
57.1	78.43	89.46	876.69
50.0	91.68	97.18	943.39
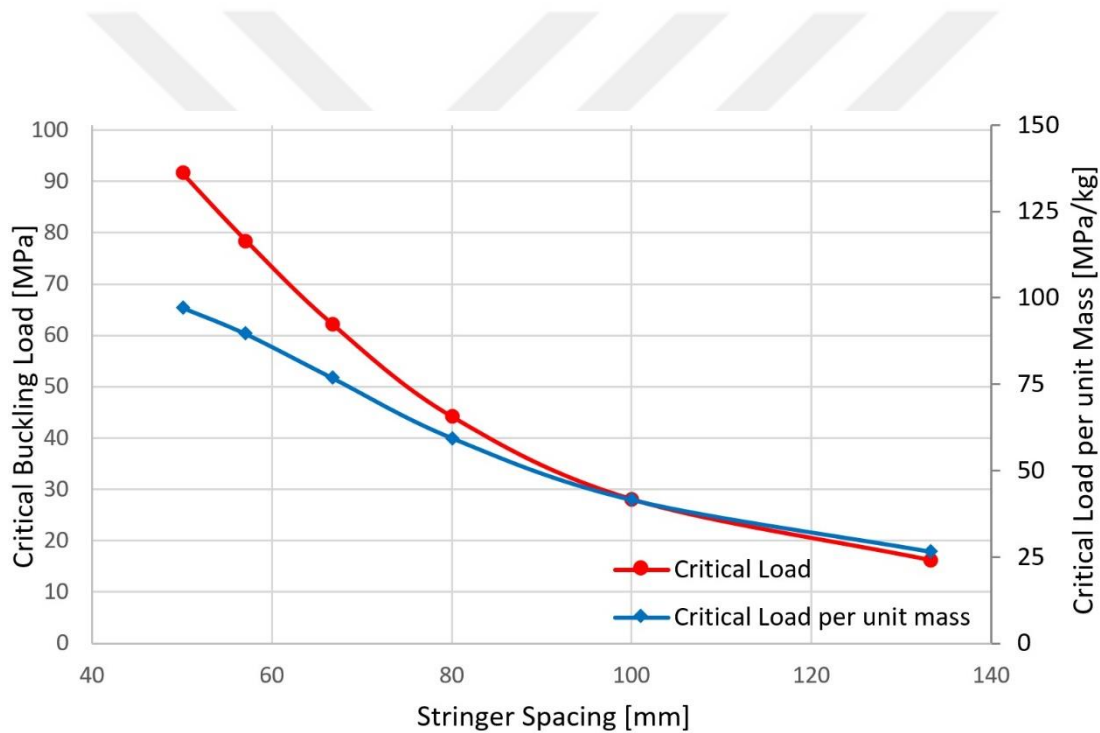
Figure 3.12: The effect of stringer spacing on buckling load capacity

3.3.3. The Effect of the Stringer Thickness

In this part, a trade-off study is carried out for the stringers with different thicknesses. While all other parameters are kept constant, only the stringer thickness is changed

and the analyses are performed for eight standard sheet metal thicknesses: 0.635 mm, 0.813 mm, 1.016 mm, 1.27 mm, 1.42 mm, 1.62 mm, 2.032 mm and 2.54 mm. The values of the other parameters for the analyses are given in Table 3.5.

Table 3.8 shows the critical buckling load and mass results of FEA for different stringer thicknesses. In order to see the effects of the stringer thickness on buckling, the *critical load per unit mass* is also given. For a better comparison, both the *critical load* and the *critical load per unit mass* are drawn in Figure 3.13. In the figure, as the stringer thickness is increased, the critical load per unit mass increases until the 1.42 mm stringer thickness value. After that thickness, the slope becomes negative which means the load carried by unit mass is not increasing anymore. It is not efficient to increase the stringer thickness in order to increase the critical buckling load.

Table 3.8: FEA results of stringer thickness trade-off study

Stringer Thickness [mm]	Critical Load [MPa]	Critical Load for unit mass [MPa/kg]	Mass [g]
0.635	23.14	38.27	604.73
0.813	25.58	39.97	639.94
1.016	28.03	41.43	676.59
1.270	31.26	42.89	728.86
1.420	32.60	43.03	757.58
1.620	33.79	42.37	797.41
2.032	35.09	40.22	872.38
2.540	36.04	37.35	964.77
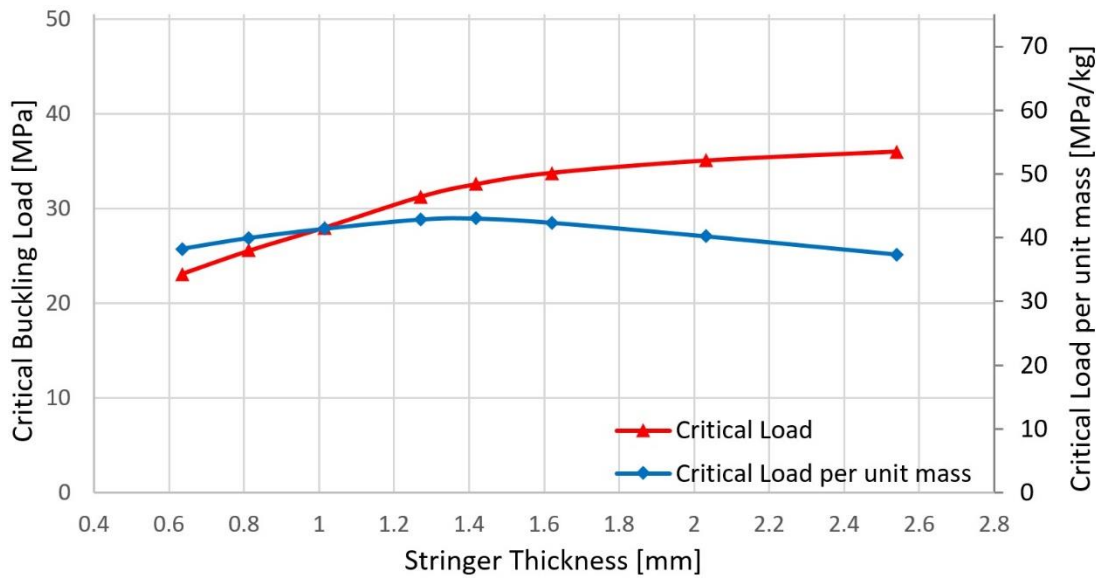
Figure 3.13: The effect of stringer thickness on buckling load capacity

3.3.4. Results of the Trade-off Studies

In the investigation of the effects of the design parameters, the critical buckling load and the critical buckling load per unit mass values are found from FE analysis. Increasing the skin thickness and decreasing the stringer spacing are efficient methods because they increase not only the critical load but also the critical load per unit mass. However, increasing the stringer thickness is efficient only up to 1.42 mm stringer thickness for the shear-compression load ratio of "1". Beyond that point, the load carried by unit mass is not increasing anymore.

In order to compare the effects of design parameters and to decide which parameter is the most efficient for the geometry defined in Table 3.5, critical load values must be compared in terms of their masses. For each parameter change through the previous studies, mass values are known.

Figure 3.14 shows the critical buckling loads with respect to the masses of models of each parameter study. According to the figure, most rational method to strengthen the

skin-stringer structure for the considered load case is to decrease the stringer spacing. However, if the structure is not buckled under combined loads, it is reasonable to decrease thickness in order to lighten the structure. As an example, if the critical buckling load of the base design is to be increased from 28.03 MPa to 50 MPa, one should change the stringer spacing. From three design parameters, it is not possible to reach 50 MPa by changing the stringer thickness. On the other hand, by changing the stringer spacing, 50 MPa critical value can be achieved with a lighter structure.
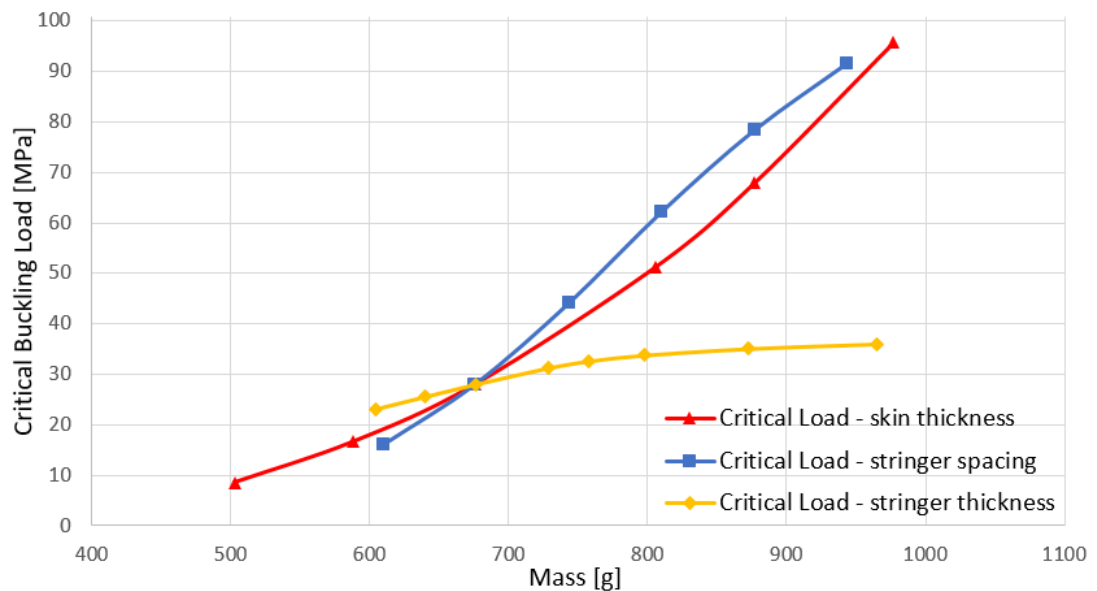
Figure 3.14: Critical buckling loads for each trade-off study for each trial

CHAPTER 4

ARTIFICIAL NEURAL NETWORK

Artificial neural network (ANN) is a smart tool that originated in order to model the human brain and understand how it works. In the most general sense, an ANN can be defined as a complex system that is formed by the connection of simple processors with different levels of effect, similar to the relationship between neurons in the human brain. In accordance with the processing method of human brain, ANN is designed to solve problems in parameter prediction, pattern identification and optimization. In today's world, ANNs are widely used extensively in many disciplines such as medicine, physics, mathematics, robotics, statistics, signal processing and nonlinear control areas [38]. It has high computing abilities and this makes it suitable for complex nonlinear problems that are inconvenient to solve with traditional mathematical methods [39].

The purpose of this chapter is to develop an artificial neural network design tool. For this purpose, a large number of finite element analyses are performed with the scripts written in Python 2.7. In the first part, a general information about the artificial neural network is given. The next part describes how the scripts are created and how they work. Then, artificial neural networks are generated with the help of these scripts and lastly, the results of the ANN are presented.

4.1. The Theory of Artificial Neural Network

Artificial neural networks (ANNs) try to imitate the structure of human brain and aim to mimic biological neural networks in a simpler scale [40]. The first studies on biological neural networks were about the structure of a nerve cell and communication

with each other. The basic elements of a nerve cell are cell body, dendrite and axon. Dendrites collect the signals from other nerve cells. The opposite end of the nerve cell is the axon, which transmits the signals to other nerve cells with the help of synapses. Nerve impulses are received by the dendrites, travel down the branches of the dendrites to the nerve cell body, and are carried along the axon. The structure of a biological nerve cell is given in Figure 4.1.

One human brain cortex is estimated to have about 10 billion neurons and approximately 60 trillion synapses or connections. Therefore, the human brain can be defined as an extremely complex, non-linear and parallel-scattered information processing system, thanks to its ability to learn, combine, adapt and generalize.
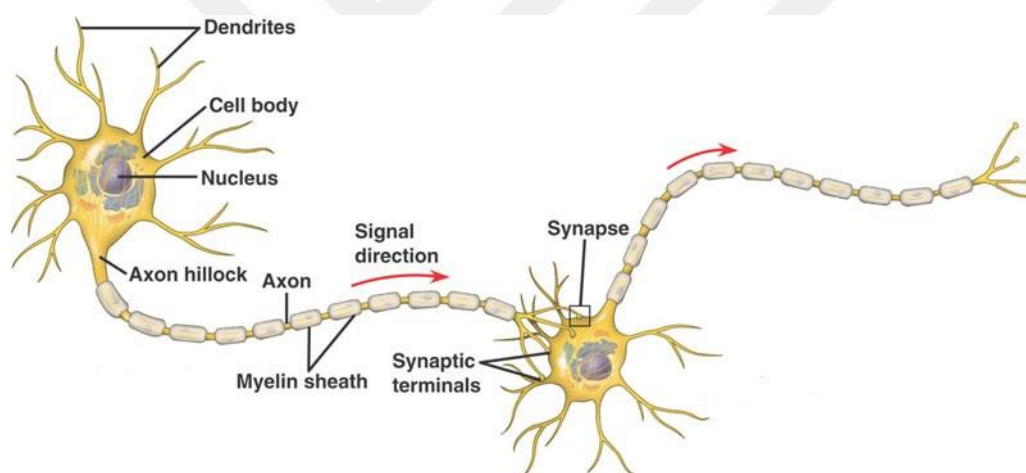
Figure 4.1: Structure of a biological nerve cell [41]

Similar to the biological neural networks, an artificial neural network consists of multiple artificial neurons that join each other with parallel connections in various layers. Neurons of each layer are connected to the neurons of the next layer [19]. The ANN is composed of three layers and these layers are respectively:

- Input Layer: The neurons in this layer transfer the data from the outside world to intermediate layers. There is always one input layer and the number of neurons is equal to the number of inputs.
- Intermediate (Hidden) Layer: The data from the input layer is processed in this layer and sent to the output layer. There may be more than one hidden layer in a network.
- Output Layer: The neurons in this layer take the data from the hidden layer to produce the output to the outside world. There is always one output layer and the number of neurons is equal to the number of outputs.

Since the input and output layers can only transmit the data, the computation capability of the ANN is determined with the number of hidden layers and the number of neurons in each hidden layer. These numbers can be determined with a trial and error method [42]. Figure 4.2 shows the structure of a sample ANN with three hidden layers for n input and m output.
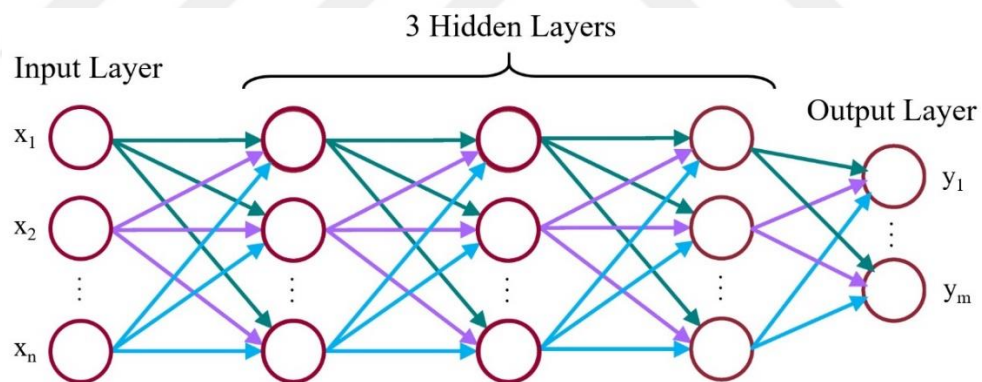
Figure 4.2: Sample ANN structure for n input and m output

In this study, *function fitting* tool is used in order to compute the margin of safety and the critical buckling load. It can construct a function that has the best fit to a series of data points. Among all back-propagation neural network methods, Levenberg-

Marquardt (LM) algorithm is extensively used one, which provides a faster Gauss-Newton optimization method and the stability of the steepest descent method in minimizing the sum-squared errors of the output results [43]. In Figure 4.3, the mathematical operations of a simple artificial neuron can be seen. Each input is multiplied with the weight vector of the network "w" and summed up with the bias "b". Then, it enters the activation function to give the neuron output. Among the options, generally, the sigmoid function is used for the activation function whose outputs are in between "-1" and "+1", instead of step functions and linear functions [44].
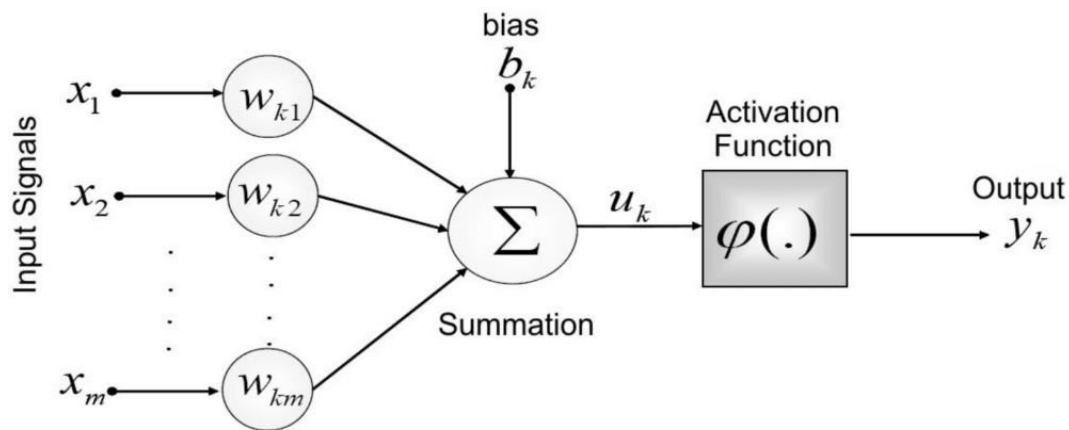
Figure 4.3: Mathematical operation steps of a simple artificial neuron [45]

4.2. Python Script for the Finite Element Analyses Run

As mentioned in the previous sections, a data set is required to train the ANN. The size of the data set is a crucial parameter in the training process since it should be large enough to contain enough history about a fitting function. For this purpose, a total of 6720 different skin-stringer structures were created with a script written in Python 2.7, and these models were analyzed in ABAQUS FE program in accordance with the methods described in Chapter 2. These scripts speed up the process of model creation

and running the analysis. ABAQUS allows easy analysis of the models with different element sizes, different boundary conditions or different materials.

All of analyses are performed for seven different compression-shear ratios. For each load ratio, there are six variable inputs. Except for these inputs, some input parameters are fixed throughout the study. The variable inputs and the constant inputs are given in Table 4.1 and Table 4.2, respectively. In the load ratio parameter, "zero" and "infinity" values correspond to pure shear loading and pure compression loading. Apart from these parameters, there is one output from the scripts, which is the first eigenvalue. Since the mass can be computed by hand calculation, it is not defined an output for the FE analyses. The first eigenvalue will enter the post-process calculation after the ANN training and generate a margin of safety value for given applied loads.

Table 4.1: Variable inputs for all FE analyses in script

Skin Panel Thickness [mm]	[0.635, 0.813, 1.0, 1.27]
Skin Panel Length [mm]	[250, 350, 450, 550]
Stringer Spacing [mm]	[50, 100, 150, 200, 250]
Stringer Height [mm]	[20, 30]
Stringer Flange Length [mm]	[15, 20]
Stringer Thickness [mm]	[0.635, 0.813, 1.0]
Compression/Shear Load Ratio	[0, 0.1, 0.25, 1.0, 4.0, 10.0, ∞]

Table 4.2: Constant inputs for all FE analyses in script

Stringer Inner Bend Radius [mm]	3 (constant)
Rivet Diameter [mm]	3.2 (constant)
Skin Panel Material	Aluminum 2024-T3 Clad
Stringer Material	Aluminum 2024-T3 Clad

Before the analyses, the script creates a text file by writing the input parameters for each skin-stringer structure. Then, it creates the FE model and analyze it. After that, it extracts the first eigenvalue and it is written to the same text file. All recorded input and output parameters are used to create an artificial neural network in the MATLAB NNTOOL toolbox. The script for the FE analyses and their storage can be found in Appendix A.

4.3. The Generation of Artificial Neural Network

In the previous section, it is mentioned that a text file is created and all input and output parameters are saved in this file. In this section, it is explained how to generate the ANN by using the values of all parameters in the text file.

First of all, the input and the output parameters of the FE analyses script together form the inputs of the ANN. At this stage, normalization for the inputs takes place, since the magnitude of the values of the parameters are very different from each other. While there are 3-digit values for the panel lengths and widths, there are values around 1 mm for skin and stringer thicknesses and it gave a high mean squared error in the first attempt of the training with the initial values. Normalization is made by dividing all parameter values by the highest value of that parameter except the load ratio. Normalized values of the parameters are shown in Table 4.3.

Table 4.3: Normalized values of the inputs

Input Parameters	Original Values	Normalized Values
Skin Thickness [mm]	[0.635, 0.813, 1.0, 1.27]	[0.5, 0.64, 0.787, 1.0]
Skin Panel Length [mm]	[250, 350, 450, 550]	[0.455, 0.636, 0.818, 1.0]
Stringer Spacing [mm]	[50, 100, 150, 200, 250]	[0.2, 0.4, 0.6, 0.8, 1.0]
Str. Height [mm]	[20, 30]	[0.667, 1.0]
Str. Flange Length [mm]	[15, 20]	[0.75, 1.0]
Str. Thickness [mm]	[0.635, 0.81, 1.0]	[0.635, 0.81, 1.0]

On the other hand, the compression-shear load ratio is a critical parameter in the training. Since its values change from zero to infinity, any normalization method do not give proper results. Therefore, instead of taking the compression-shear ratio as input parameter, seven different networks are trained for each load ratio and a MATLAB script is created in order to combine these networks. This script is also used to calculate the mass of each structure and to export the FE results' text file to MATLAB. Inputs and outputs are exported as two separate matrices and they are used for the training of the ANN. This script can be found in Appendix B.

The ANN should be configured to provide an efficient performance by arranging some parameters. Especially, the number of hidden layers and the number of neurons in the hidden layer are significant parameters. The use of one hidden layer is found to be sufficient in terms of mean squared errors and duration of the training process [46]. Besides, a trial and error method is performed for the number of neurons. Although it is known that the accuracy of the fitting function is increasing as the number of neurons is increased, inaccurate results may be obtained for the points between the training parameter values if there are too many neurons. This is called over-fitting the function [19]. Therefore, it is found that 50 artificial neurons in one hidden layer give the best performance. The final configuration of the ANN setup becomes 6:50:1 and it is shown in Figure 4.4.
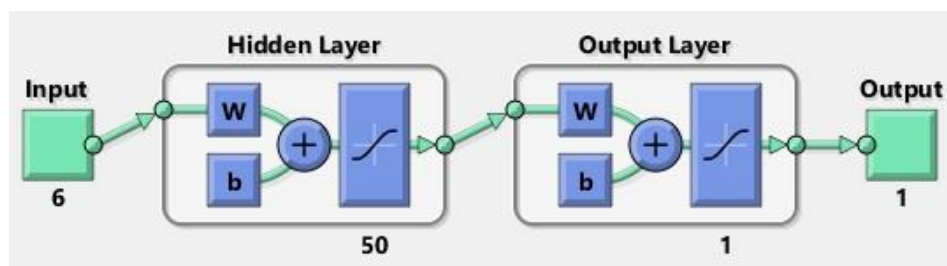
Figure 4.4: The final configuration of the neural network structure

In MATLAB neural network tool, all data of inputs and outputs are separated to three categories. These are the training set, the validation set and the test set. In MATLAB default settings, 70% of the analyses are used for the training, 15% for the validation and 15% for the testing of the ANN [44]. The training set is used in order to specify the weight and bias of each neuron. The validation set is used in order to check the performance of the network after each iteration. The test set is used again in order to check the performance of the network when the training is finished. It should be noticed that the validation and test sets are not connected with the training process.

There are a few termination criteria during training. Epoch limit is chosen as (1×10^3), which is the maximum number of iteration during training. Maximum momentum coefficient is set to (1×10^{10}), which slows the speed of the descent so that the search value does not fly back and forth across the minimum without stopping sufficiently near it [13]. If this value is small, it is difficult to get rid of local minimum, while being too large may cause problems in reaching a single solution [47].

4.4. The Results of Artificial Neural Network

MATLAB gives the mean squared error (MSE) to show the performance of the tool. The MSE implies the mean square of the deviations between the trained ANN output values and the target output values. In addition to MSE values, mean absolute error (MAE) values of each compression-shear load ratio are calculated from the absolute difference between the outputs of the each network and the FE analyses.

The training of all networks take approximately 8-9 minutes. Since each network is trained for a single compression-shear load ratio, there are seven different MSE values and seven ME values. The error performances are shown together in Table 4.4. The average MSE and ME values of seven networks are found as 7.0912×10^{-4} and 1.26% respectively. In addition to being very small average ME value, it can be understood that MSE value is also very small when the normalized values are

considered. In addition, maximum absolute error for each network is shown in the table. 17.87% maximum absolute error is obtained among all the networks. Although all the error values are slightly increasing with the increasing shear dominance, it is still pretty satisfactory. On the other hand, if the designs with maximum absolute errors are examined, it is seen that stringer section properties appear to have an impact. Maybe it can be concluded that training with more than 2 design variables for stringer flange length and stringer height would have been more successful.

Table 4.4: Error values for compression-shear load ratio of each network

Compr-Shear Load Ratio of each network	Mean Squared Error (MSE)	Mean Absolute Error (MAE)	Maximum Absolute Error
0	8.454×10^{-6}	0.88%	5.42%
0.1	2.534×10^{-6}	0.93%	6.44%
0.25	2.415×10^{-6}	0.61%	6.38%
1	4.491×10^{-6}	0.94%	15.75%
4	3.236×10^{-4}	1.60%	15.80%
10	5.064×10^{-5}	1.90%	15.45%
∞	1.043×10^{-4}	2.00%	17.87%
Average	7.091×10^{-5}	1.26%	

For an illustration, MSE values with respect to the number of iterations are shown in Figure 4.5. In the first iterations, performance of the mean square error values of all sets are very high, but the performance is quickly increasing after about 100 iterations. After the performances are slowly increasing, the best validation performance comes up at 432nd iteration. It is noticed that after this point, training set performance is still improving. Figure 4.6 shows the termination criteria of the training. Among three criteria, which are gradient, momentum coefficient and maximum validation failure, the last one becomes the termination criteria. The first two criteria do not reach the limits before the maximum validation number. In the figure, vertical axis is the number

of consecutive validation failure and the horizontal axis is the iteration number. From iteration number 432 to 732, 300 consecutive validation checks are made and none of them pass the criteria.
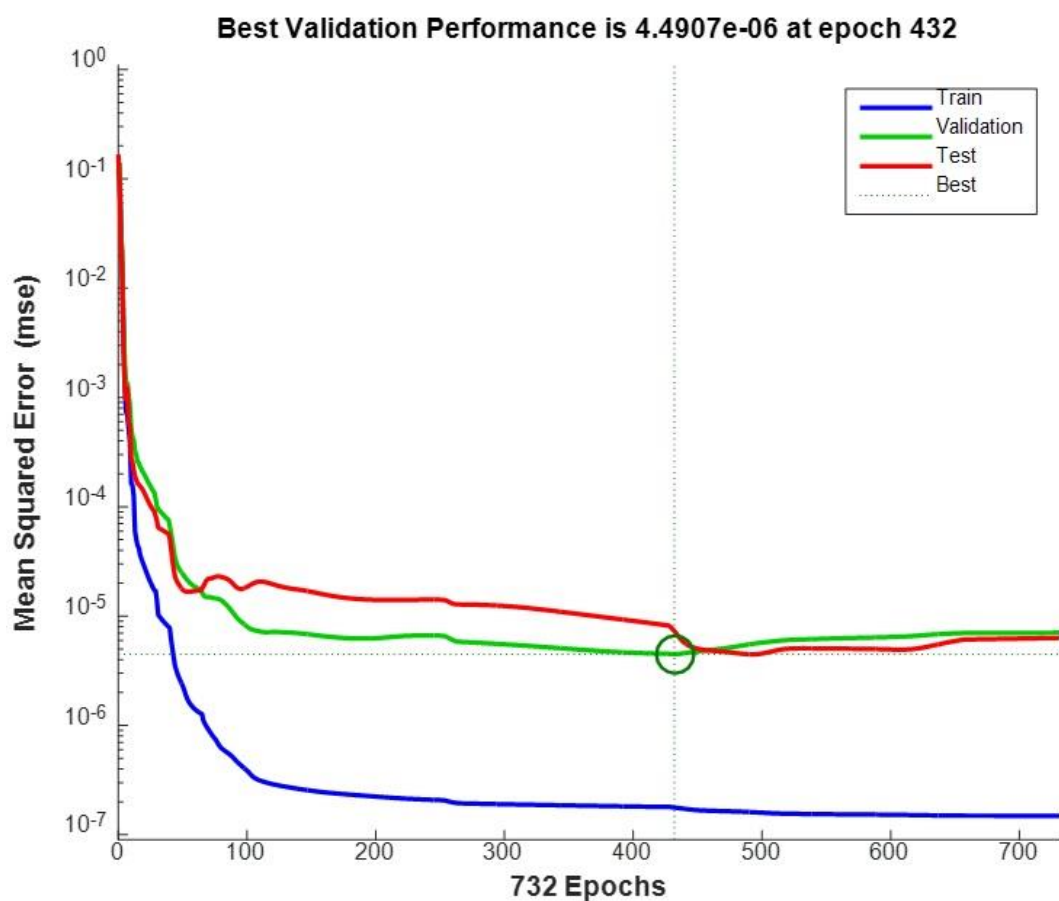
Figure 4.5: The performance plot of ANN for compression-shear ratio of "1"
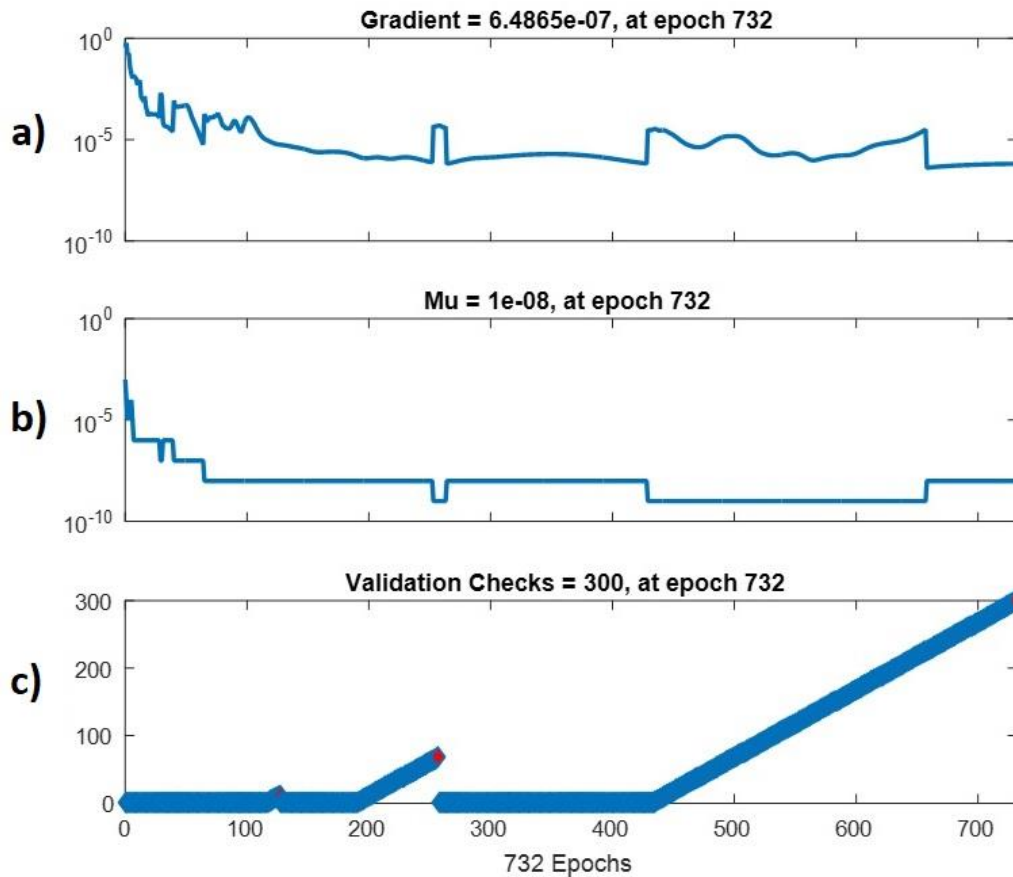
Figure 4.6: Termination criteria for ANN training of compression-shear ratio of "1"
a) Gradient b) Momentum coefficient c) Maximum Validation Failure

The best results that the tool can give is the FE analysis solution since the tool is created based on FE result database [48]. Thus, another criterion for the error evaluation is the regression plots. Figure 4.7 shows the regression plots, which compare the target values with the ANN output values for training, validation, test set and all together. The dashed lines (x=y) show the perfect fitting and the colorful lines show the fitted lines by the ANN. It is observed that these two lines almost coincide, which means that the training of ANN is very good. In the figures, R value implies how much the selected input parameters affect the output in total. Since the R values are nearly 1, the selection of the parameters and input data size are successful for the training.
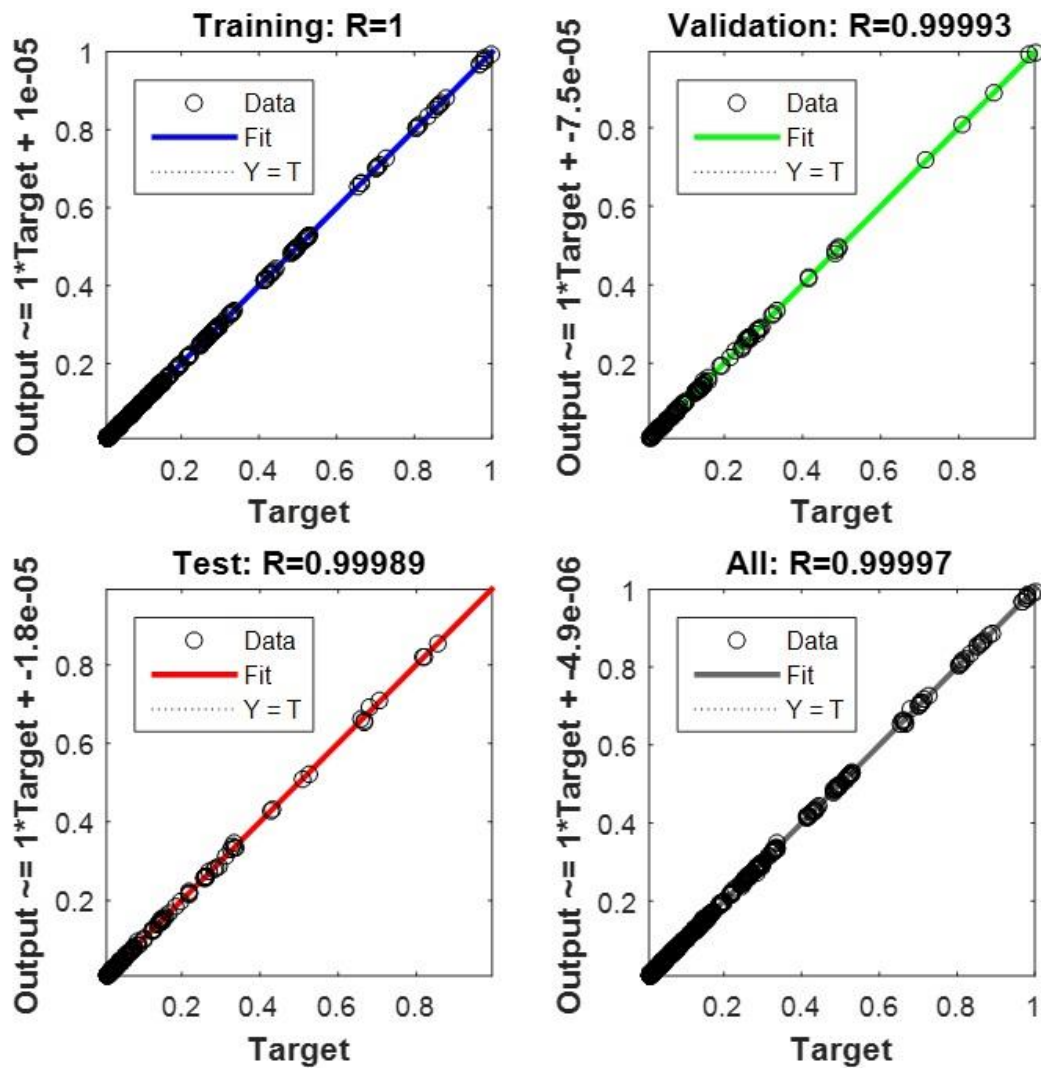
Figure 4.7: Regression plots of the ANN for training, validation and test sets of compression-shear ratio of "1"

The trained networks are tested with the stated 15% of all given sets at the end of training process and the results are shown in previous figures. When all data sets (6720 analyses results) are considered for testing the tool, the average percentage error is found as 1.26%, which was calculated from seven different error values for seven different load ratios in Table 4.4. Among 6720 analyses results, 6391 analyses (95.1%) give 5% or less difference than the target values. 259 (3.9%) analyses comes up with

the errors between 5% and 10% while only 70 (1.0%) analyses give more than 10% difference. For all load ratios, the number of analyses and percentages are tabulated in Table 4.5 for different error intervals.

Table 4.5: Number of analyses and percentages for different error intervals

Compression-Shear Load Ratio	Mean Error <5%	Mean Error 5% - 10%	Mean Error >10%
0	930	20	10
	96.9%	2.1%	1.0%
0.1	940	17	3
	97.9%	1.8%	0.3%
0.25	946	14	0
	98.5%	1.5%	0.0%
1	938	19	3
	97.7%	2.0%	0.3%
4	897	48	15
	93.4%	5.0%	1.6%
10	879	60	21
	91.6%	6.3%	2.2%
∞	861	81	18
	89.7%	8.4%	1.9%
Total	**6391**	**259**	**70**
	95.1%	**3.9%**	**1.0%**

Although the results of training points itself give low percentage errors, it is also important that the design tool can predict the results for which the input parameters between the training data points. Such sets truly show the performance of the approximation. Therefore, 16 additional FE analyses are performed. Inputs of 10 additional analysis are chosen between the training data points while the one of the inputs of 6 additional analysis are chosen from outside the training points. After the FE analyses, the input parameters of these analyses are sent to ANN design tool and the results from both FEM and ANN are compared with each other. Table 4.6 shows the input parameter values of these sixteen random sets. Note that, red marked values

are interpolation points while the blue marked values are extrapolation points between the training input sets.

Table 4.6: Input parameters of additional analyses sets

	Skin thickness [mm]	Stringer spacing [mm]	Skin length [mm]	Stringer flange length [mm]	Stringer height [mm]	Stringer thickness [mm]	Comp-Shear Ratio
Set-1	1.016	210	300	15	20	0.635	4
Set-2	0.813	90	400	20	30	1.016	1
Set-3	1.016	125	500	15	30	0.813	3
Set-4	1.270	160	420	20	30	0.635	0.3
Set-5	0.635	70	480	15	20	0.813	0.5
Set-6	1.016	120	280	16	25	0.635	1
Set-7	0.635	60	350	17	22	1.016	0.75
Set-8	1.270	240	450	18	24	0.813	10
Set-9	0.813	200	310	19	21	0.635	0.45
Set-10	0.635	250	540	17	22	0.813	20
Set-11	0.813	110	410	23	25	1.016	0.1
Set-12	1.270	150	450	18	33	0.813	2.3
Set-13	1.420	100	515	19	21	1.016	0.86
Set-14	1.016	85	250	20	26	1.270	0.05
Set-15	0.813	140	600	20	29	1.016	1.35
Set-16	0.635	45	260	15	23	0.635	6

It is should be noticed that the MATLAB script makes interpolations between the seven input load ratio values. Therefore, it is important to see the error values for those sets. MATLAB script also calculates the allowable shear and compression load with the help of the eigenvalue output of the ANN design tool. In Table 4.7, the critical shear and compression load outputs for additional analyses and the comparison with the FE results are shown. According to the results, Set 11, 12 and 14 comes up with an error higher that 6%. The reason for this outcome can be that all these sets have an

extrapolation variable related with the stringer section property. In the training, stringer section parameters are trained with 2 or 3 different variables. For this reason, these variables can be more sensitive than others.

After the results, it can be concluded that the percent error levels are acceptable to verify the skin-stringer design tool for buckling behavior in a preliminary design study in case the input parameters are in the interpolation points.

Table 4.7: The results of additional analyses sets

Parameters	FEM		ANN		Error
	Shear Load	Compression Load	Shear Load	Compression Load	
Set-1	1.89	7.56	1.88	7.52	0.64%
Set-2	9.85	9.85	10.16	10.16	3.12%
Set-3	5.66	17.00	5.67	17.02	0.23%
Set-4	16.16	4.85	16.11	4.83	0.30%
Set-5	8.87	4.44	9.14	4.57	3.01%
Set-6	12.27	12.27	11.69	11.69	4.71%
Set-7	11.72	8.79	11.69	8.77	0.21%
Set-8	1.12	11.24	1.06	10.60	5.72%
Set-9	4.29	1.93	4.19	1.89	2.25%
Set-10	0.08	1.69	0.09	1.71	0.96%
Set-11	8.68	0.87	7.76	0.78	10.63%
Set-12	9.58	22.03	8.82	20.28	7.96%
Set-13	39.30	33.80	38.24	32.89	2.69%
Set-14	31.82	1.59	37.47	1.87	17.78%
Set-15	3.69	4.98	3.62	4.88	1.98%
Set-16	4.49	26.94	4.32	25.90	3.86%

After the training of the ANN, Margin of Safety and weight calculations are performed with two different MATLAB scripts. They will be utilized for the optimization process of a skin-stringer structure with given width and length of the skin panel and given applied loads.

CHAPTER 5

OPTIMIZATION

Optimization is the process of obtaining the best result under the specified conditions. It can also be defined as the process of finding a minimum or maximum of a specific function. Techniques of finding optimal values are a branch of operational researches and operational research deals with the application of scientific methods to decision making problems and finding appropriate solutions [20]. Reduction of structural weight with optimization techniques is used extensively in the aerospace industry. Weight of aircraft structure effects the aircraft performance, the fuel consumption and the cost negatively, that is why the structural optimization is a crucial work area.

In the literature, there are various studies conducted with the classical optimization methods for the structural design purposes. Each method has its own advantages and disadvantages. The method chosen to solve a problem may not work well for another problem. Depending on the nature of the problem, the problems of divergence or finding local minimum are frequently observed [49], [50]. Genetic algorithm techniques provide generally more dynamic and appropriate solutions than the other techniques.

The objective of this chapter is to perform structural optimization of a rectangular skin-stringer structure, which can idealize a specific region of an aircraft by using the genetic algorithm method. The external dimensions of the rectangular area and the applied loads from the boundaries are known at the beginning of the optimization process.

In the computation, MATLAB OptimTool is used in order to optimize the design variables. For the objective and constraint functions, there are two MATLAB scripts. The first script is for objective function, which is the weight of the skin-stringer

structure and the second script is for constraint function, which is the Margin of Safety of the skin-stringer structure. Since the weight can be found by hand calculation, this script does not use the trained ANN. However, for the MS values, the script is written by using the ANN trained in the previous chapter.

5.1. Theory of Genetic Algorithm

Genetic algorithm (GA) is a stochastic search method. It is used as an optimization technique based on natural selection and genetic science. The GA is modelling the development process of biological systems based on Darwin's best-lives principle [51].

The GA consists of five basic phases, which are the initial population, the fitness (penalty) function, the selection, the crossing and the mutation. In GA, each solution is coded as an individual. At the beginning of the process, the initial population is generated randomly. Each individual in the population can be encoded in binary form. These individuals are decrypted during the evaluation phase and they are evaluated according to their performance in certain objective functions. Individuals are more likely to be selected according to their fitness values (roulette-wheel selection). Selected individuals are eligible to enter the crossover pool, while the others eliminate. Selection operators provide the best survivors, just as in the real nature. Each survival individual is then randomly matched with another individual in the crossover population. The main aim is to provide individuals (parents) with higher fitness values by crossing the good characteristics. In this manner, their genes passes to the next generations after the crossing. For each pair of parents, a crossover point is chosen randomly within the genes. If no crossing occurs, the parents are transferred to the next population. If the GA consisted of only the selection and the crossover operators, they could reach a local optimum very quickly. In order to maintain diversity in the population, to regain the good features that can be lost after the crossing and to achieve the best optimum, the genes in the individuals are subjected to a mutation with a low

random probability. This process continues until a predetermined termination criterion is achieved. The flowchart of the GA optimization process steps is given in Figure 5.1.
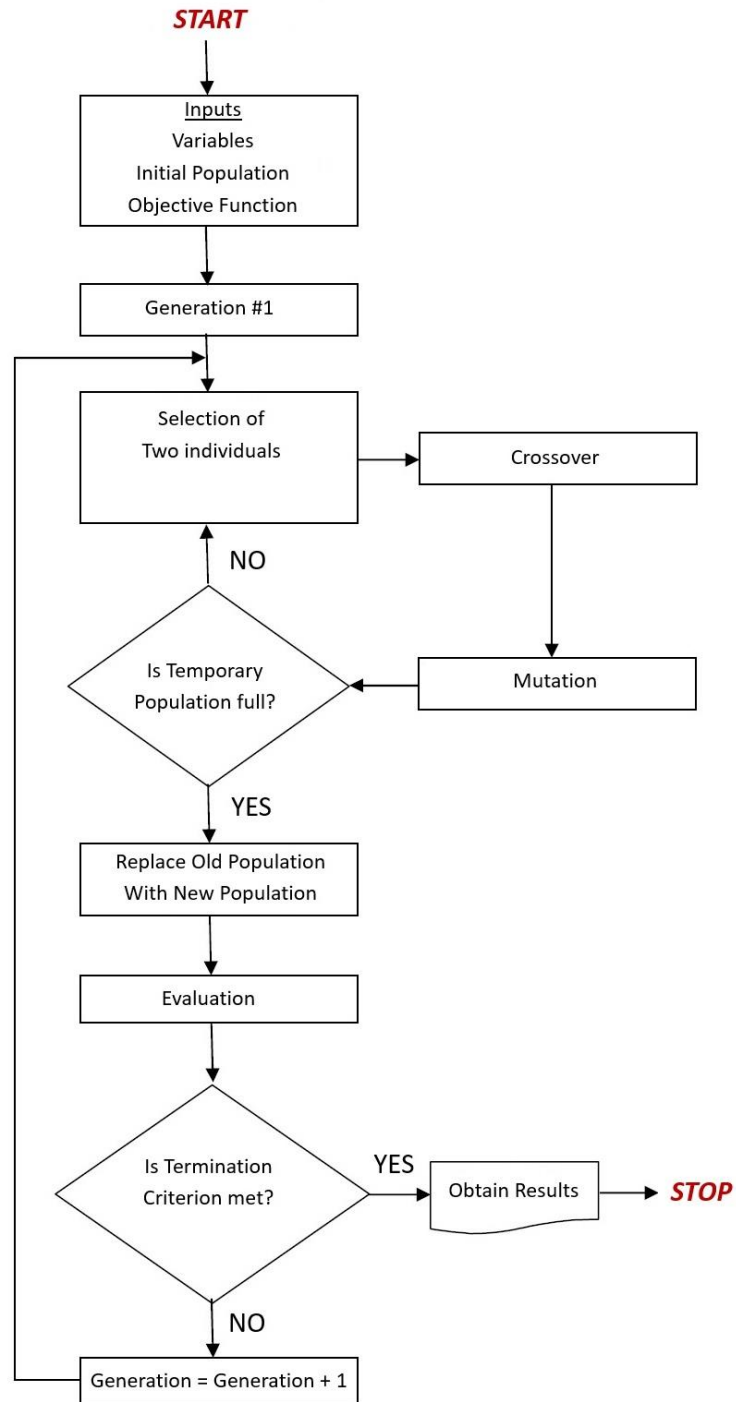
Figure 5.1: Optimization flowchart of the genetic algorithm process

5.2. Definition of the Optimization Problem

An optimization problem can be expressed as basically [52];

Find the design variables, $[x] = [\, x_1, \, x_2, \dots, x_n]^T$

which satisfy m inequality constraints and p equality constraints,

$$g_i(x) > 0, \qquad i = 1,2, \dots, m$$

$$h_j(x) = 0, \qquad j = 1,2, \dots, p$$

which minimize the objective function,

$$f(x).$$

In an optimization problem, design variables can be discrete, continuous and/or integer variables. According to the nature of the problem, they may also be preassigned values.

In this study, an array of nine design variables is defined to obtain the optimum result. Four of them are preassigned, two of them are discrete, two of them are continuous and one of them is integer parameters. That means, at the end of the optimization, five optimum values for five design variables will be obtained. These variables and their variable types are given in Table 5.1 with the given order.

In this study, a skin-stringer structure should be placed in a specific rectangle region which has 500x500 mm dimensions. A compressive stress of 20 MPa and a shear stress of 10 MPa are applied from the boundaries of this region. The pre-assigned parameters are presented graphically in Figure 5.2.

Table 5.1: All design variables and variable types

Design Variables		Variable Type
t_{skin}	Skin Panel Thickness [mm]	Discrete
a_{skin}	Skin Panel Length [mm]	Pre-assigned
b_{skin}	Skin Panel Width [mm]	Pre-assigned
$c_{stringer}$	Stringer Flange Length	Continuous
$h_{stringer}$	Stringer Height [mm]	Continuous
$t_{stringer}$	Stringer Thickness [mm]	Discrete
σ_{comp}	Applied Compression Load [MPa]	Pre-assigned
σ_{shear}	Applied Shear Load [MPa]	Pre-assigned
$n_{stringer}$	Number of stringer	Integer
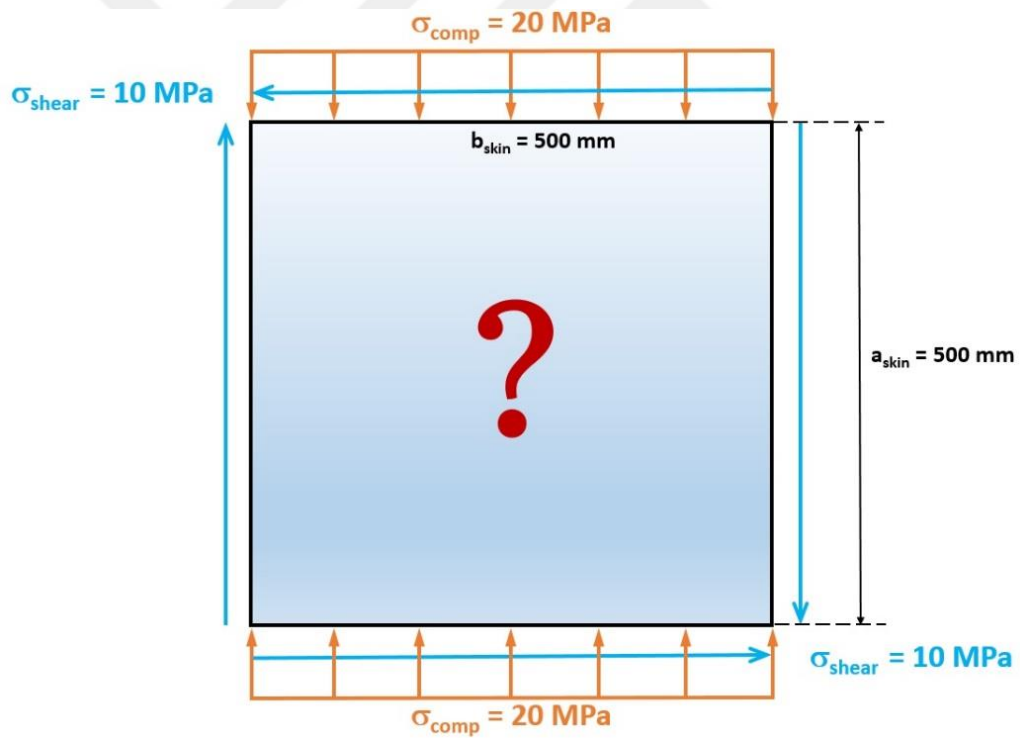
Figure 5.2: Pre-assigned parameters of the optimization problem

Based on the given infomation, the mathematical expression of the problem can be given as;

Find the design variables,

$$[x] = [t_{skin}, c_{stringer}, h_{stringer}, t_{stringer}, n_{stringer}]$$

which satisfy the margin of safety inequality constraint,

$$MS > 0$$

Which satisfy the following equality constraints for four design variables,

$$a_{skin} = 500\ mm \qquad b_{skin} = 500\ mm$$

$$\sigma_{comp} = 20\ MPa \qquad \sigma_{shear} = 10\ MPa$$

which minimize the weight function of the designed structure,

$$f(x) = weight(t_{skin}, c_{stringer}, h_{stringer}, t_{stringer}, n_{stringer})$$

It should be noticed that the number of stringers variable will be an integer during the iteration process. In addition to that, since they are sheet materials, discrete values are implemented for the thickness values of the skin and the stringers. In Table 5.2, nine standard thickness values are chosen for both the skin and the stringers.

Table 5.2: The values of discrete design variables

Discrete Design Variables	
$t_{skin}(i)$	$[0.406, 0.508, 0.635, 0.813, 1.016, 1.270, 1.420, 1.630, 2.032]$
$t_{str}(j)$	$[0.406, 0.508, 0.635, 0.813, 1.016, 1.270, 1.420, 1.630, 2.032]$

In the MATLAB Optimization Toolbox, lower and upper bounds of the design variables need to be specified. For the pre-assigned design variables, both lower and upper bound values are the same. However, for discrete variables, it is not possible to identify all points by stating just their bounds. Therefore, an array of discrete values should be entered in the MATLAB scripts. The bounds of these variables need to be identified as an integer in the Optimization Toolbox. From an aerospace engineering point of view, all lower and upper bounds of the design variables are specified in Table 5.3.

Table 5.3: Lower and upper bounds for the design variables

Design Variables	$[t_{skin},\ c_{str},\ h_{str},\ t_{str}, n_{str}]$				
Lower Bounds	[1,	15,	15,	1,	3]
Upper Bounds	[9,	35,	35,	9,	8]

For the algorithm parameters, default values of the toolbox are used mostly. Only the population size is found as 50 with a trial-error method. For bigger values of population size, the error does not decrease significantly, while the solution time significantly increases. On the other hand, since the smaller values lead to more error, 50 individual for each population is found to be sufficient. In addition, crossover fraction is used as 0.8, migration fraction is used as 0.2 and initial penalty and penalty factor are used as 10 and 100, respectively.

5.3. Results of the Optimization Problem

After 197 iterations, optimization is terminated, since the average change in the penalty fitness value and the constraint funciton is less than the stopping criteria. With the initial population, a total of 198 generations have been created and 9900 analyses have been conducted. At the end, the optimum value of the objective function is obtained as 0.7134 kg, which is the weight of the optimized skin-stringer structure. For the constraint function, MS value is expected to be slightly bigger than zero and as expected it is found as "3.85×10^{-5}".

Table 5.4: Optimum values of the design variables

	Design Variables	Optimum Value
t_{skin}	Skin Panel Thickness	0.813 mm
c_{str}	Stringer Flange Length	15 mm
h_{str}	Stringer Height	20.926 mm
t_{str}	Stringer Thickness	0.406 mm
n_{str}	Number of stringer	6

It is also important that the optimum design of skin-stringer structure are re-analyzed with finite element method. The inputs of optimum structure are defined in ABAQUS and analyzed in linear buckling. According to the solution, MS value can be found by substact "1" from the eigenvalue output. *Figure 5.3* shows the mode shape of the FE solution of the optimum solution. From its eigenvalue output, MS can be found as "0.018" and this value is very close to the ANN output for the optimum structure.
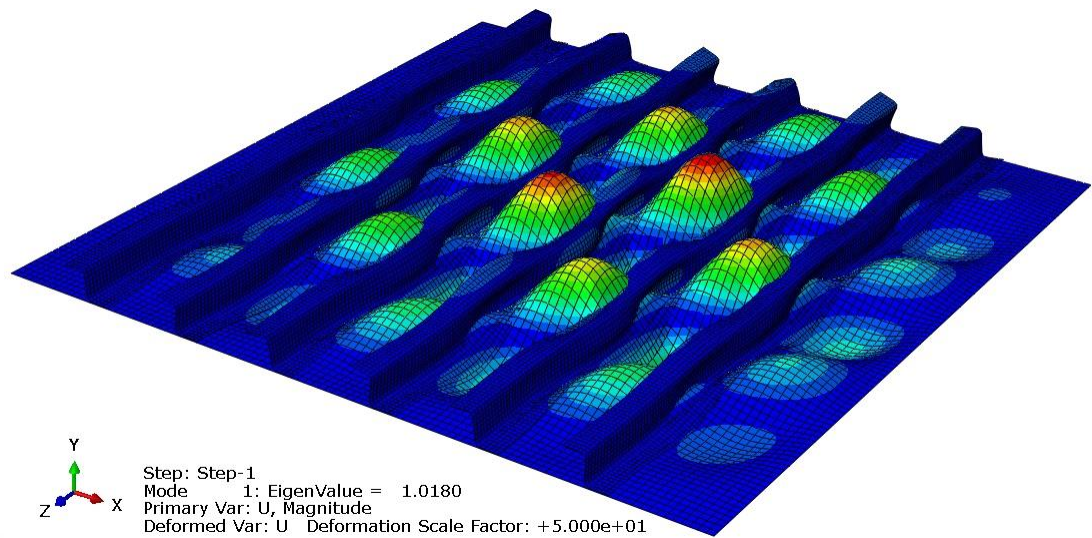
Figure 5.3: Finite element solution of the optimum structure

After all iterations, the MATLAB Optimization Toolbox shows the optimum design variables at the bottom of the window, as seen in Figure 5.4. The integer values for thickness outputs should be converted to their actual values in their arrays. Optimum values of varible design parameters are shown in

Table 5.4. It should be noticed that the optimum stringer thickness is close to the lower bound, the optimum skin panel thickness comes up an average value and the optimum number of stringers is close to the upper bound. The reason can be explained by the results of trade-off studies in Part 3.3.4. In Figure 3.14, the most reasonable method to strengthen the skin-stringer structure with minimum weight increase is chosen as decreasing the stringer spacing. MATLAB toolbox has also the same approach with the trade-off studies.
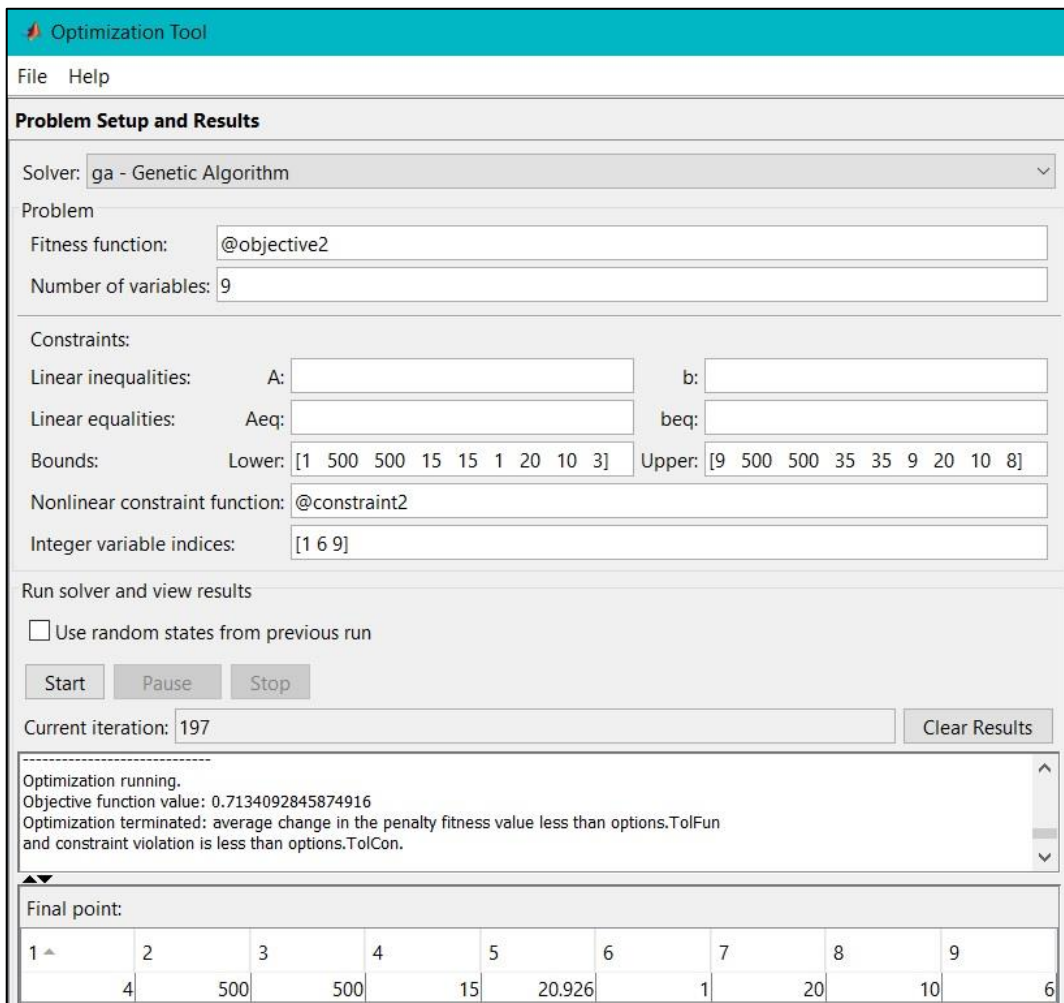
Figure 5.4: Optimum result of Optimization Toolbox window

After each generation, 50 different individuals have been created and updated. After 197th generation, optimization have converged to a certain value. For a sample illustration, the 1st, 50th, 100th, 150th and 197th generation outputs are shown for each individual in Figure 5.5. It can be seen that, the outputs are getting closer to the optimum value with the increasing number of generations.
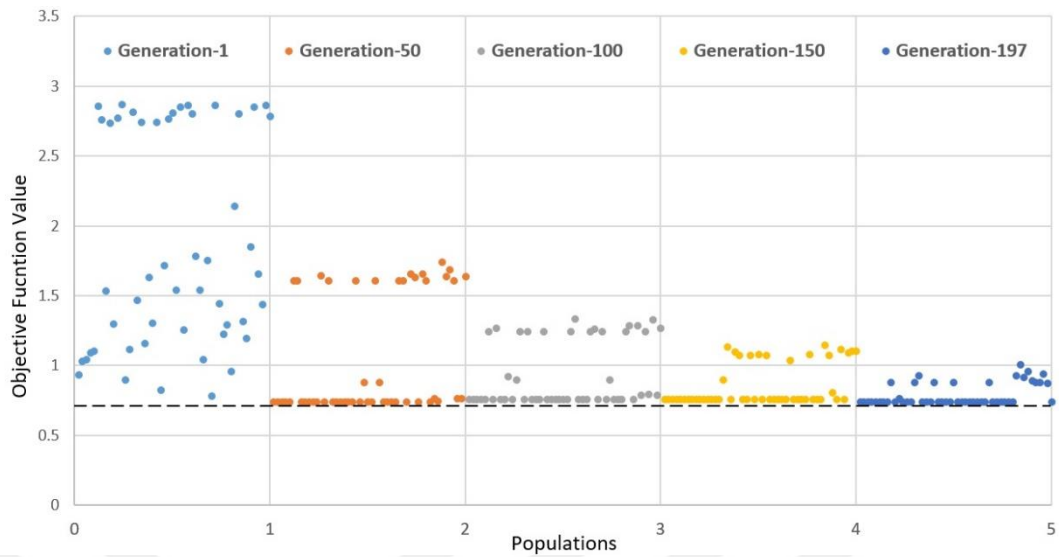
Figure 5.5: Sample outputs for the 1st, 50th, 100th, 150th and 197th generations

The average and the best fitness (penalty) values of each generation are compared in Figure 5.6. A significant reduction is observed not only the best fitness value but also the average fitness value with the increasing number of generations. The best fitness value converges to a certain value at the 60th generation. After this generation, altough the fitness value converges to a solution, iterations do not stop. The reason is that the average distance between the individuals does not converge to a solution at those generations. The average distance between the individuals for each generation is given in Figure 5.7. At the last generations, it can be seen that the average distance rapidly decreases.
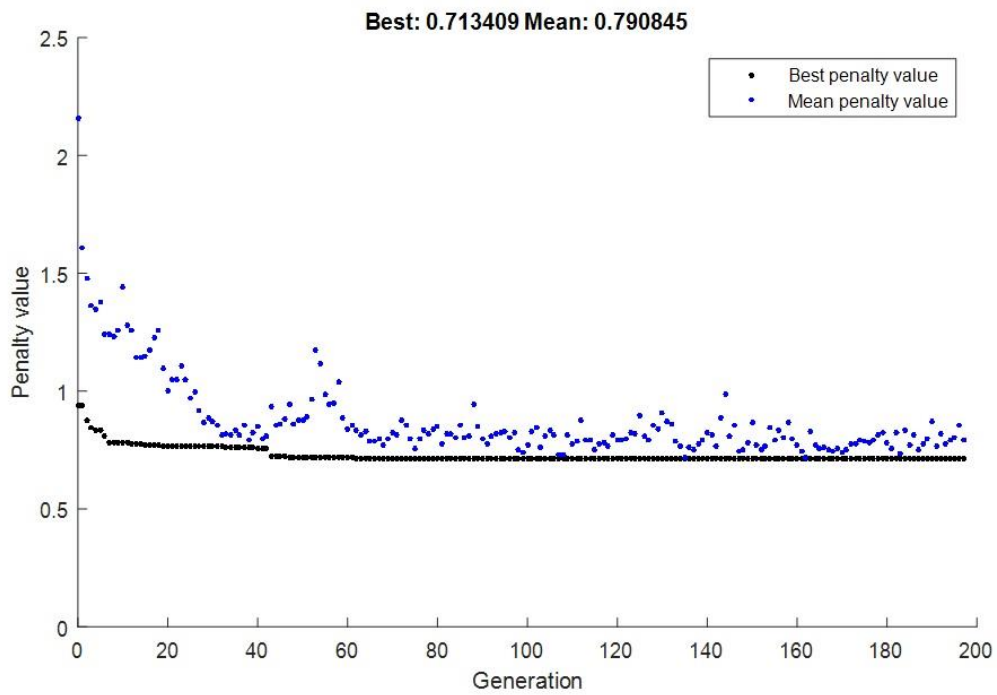
Figure 5.6: The average and the best values for each generation
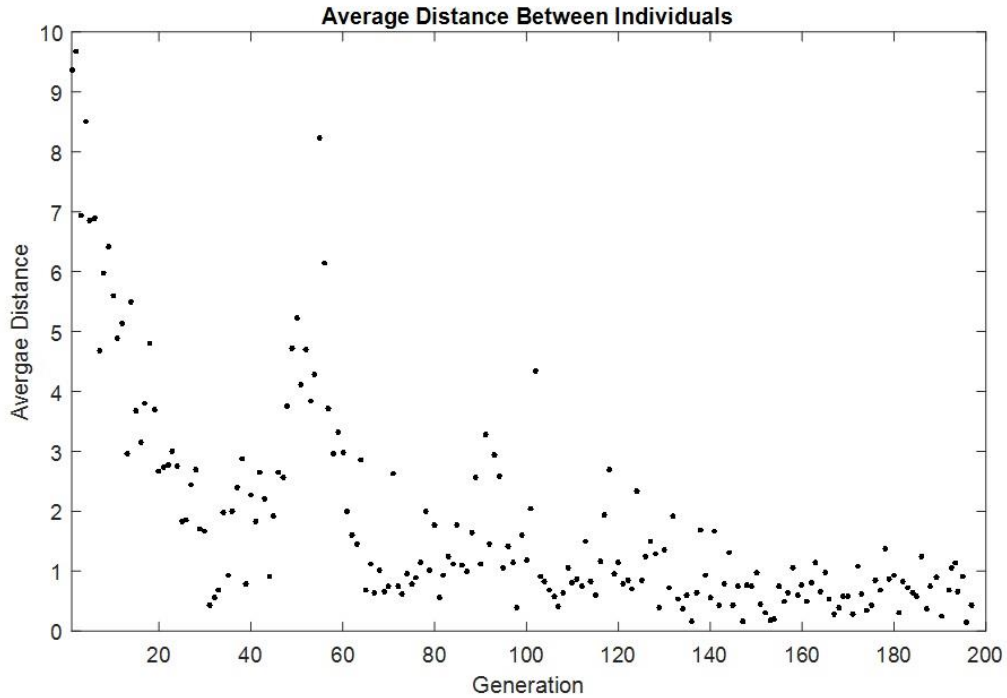
Figure 5.7: The average distance between individuals for each generation

84

CHAPTER 6

CONCLUSION

The main purpose of the structural design of an aircraft is to achieve the maximum strength with a minimum weight. Since the minimization of the weight means an increased range and more payload capability, all costs can be reduced with one optimization process. The main purpose of this thesis is to develop a design tool based on artificial neural network for buckling prediction and to carry out a geometric optimization for skin-stringer structures having a large number of stringers. Using an ANN for the buckling load and Margin of Safety prediction decreases the analysis time and labor work. In addition, a safe skin-stringer structure with minimum weight can be obtained with an optimization process, which uses the genetic algorithm technique.

For the use of artificial neural network (ANN), a large number of finite element models have been created and analyzed. In the studies in Chapter 2 and 3, baseline parameters for the finite element analyses were determined in order to perform these analyses efficiently and a sample analysis was performed. It is seen that a critical buckling curve can be generated for the combined buckling of a skin-stringer structure. This curve also gives the Margin of Safety value for a specific geometry and given applied compression and shear loads.

In Chapter 2, some sensitivity studies were performed in the generation the finite element (FE) model for mesh size and element types. After that, some case studies were performed in Chapter 3, in order to help building a representative finite element model. From the outputs of the first case study, it appears that the first eigenvalues converge to a specific value for three or more stringers combination with equal stringer spacing. It can be concluded that a minimum number of three-stringer structure provides sufficient results in order to obtain the first eigenvalues correctly. In the

second case study, appropriate boundary conditions that represent side edge stringers were investigated. At the end of the study, three-stringer model was extended symmetrically by the amount of effective width and reanalyzed for the clamped condition. It was found that this configuration gave the nearest results to the five-stinger model. In the third case study, the effects of some design parameters on buckling behavior were investigated. The trade-off studies of the skin thickness, the stringer spacing and the stringer thickness were performed and it converges to a conclusion that, in order to increase the strength of a structure with minimum weight increase, the most effective method was found to reduce the stringer spacing, then to increase the skin thickness and lastly to increase the stringer thickness. This outcome was later confirmed in the optimization chapter.

In Chapter 4, an artificial neural network was developed as a design tool. Initially, a large number of finite element analysis were performed in ABAQUS with the scripts written in Python language. By using these analyses inputs and outputs, 6720 sets were separately trained for seven different compression/load ratios and the performance of the ANN was tested with two different data sets. Firstly, it was tested with all data points in each training set at the end of the training process and 6391 of 6720 data points (95.1%) gave 5% or less difference. In addition, it is also important that the design tool can predict the results for which the input parameters between the training data points. For this reason, total of 16 analyses are performed and it can be concluded that the percent error levels are acceptable especially for interpolation points to verify the skin-stringer design tool for buckling behavior in a preliminary design study.

After development of the ANN, a structural optimization was performed for the specific region of an aircraft by using the developed neural network in previous chapter. Genetic algorithm was used for the solver method. The optimization problem is to find geometric parameters of a skin-stringer structure with minimum weight which is placed in a region with given dimensions and applied loads. The structure was constrained with the requirement that margin of safety must be higher than zero.

After 197 generations created and 9900 total analyses conducted, an optimum solution was reached with a 0.7134 kg minimum weight and "3.85×10^{-5}" margin of safety. It was found that for design of optimum skin-stringer structures, the stringer spacing is the most important variable in order to withstand the applied compression and shear loads.

The studies performed in this thesis not only demonstrate the successful use of ANN as a skin-stringer design tool, but also show a consistency about usability of this tool for a structural optimization. Furthermore, these studies will shed light to similar structural problems in a way for the applicability of ANN and genetic algorithm structural optimizations.

6.1. Future Work

This thesis intends to develop a new approach to the combined compression and shear buckling phenomena in skin-stringer structures. This methodology can be improved by integrating the bending load condition into the design tool. During this study, considered failure mode is always the buckling failure of the skin panel between stiffeners. Moreover, considered structure is always the skin-stringers. By adding some failure modes into the tool and increasing the number of geometrical parameters like frames, curve panels etc, this tool can be converted to a complete design tool for a larger components of an aircraft. Other than these improvements, for a better usage the tool, a graphical user interface (GUI) can be generated, so that an optimized structure can be obtained in a very short time.

REFERENCES

[1]   M. C. Niu, "Airframe Structural Design - Practical Design Information and Data on Aircraft Structures (2nd Edition)," *AD Adaso/Adastra Engineering LLC*, 1999. .

[2]   T. H. G. Megson, *Introduction to Aircraft Structural Analysis*, Second Edi. 2010.

[3]   E. F. Bruhn, *Analysis and Design of Flight Vehicle Structures*. Jacobs Publishing, 1973.

[4]   G. Y. Grondin, A. E. Elwi, and J. J. R. Cheng, "Buckling of stiffened steel plates - A parametric study," *J. Constr. Steel Res.*, 1999.

[5]   A. Cankur and E. Gurses, "An Artificial Neural Network Based Analysis Method for Skin-Stringer Structures," *Int. Ankara Conf. Aerosp.*, 2017.

[6]   N. J. Hoff, "Innovation in Aircraft Structures - Fifty years ago and today," *25th Struct. Struct. Dyn. Mater. Conf.*, 1984.

[7]   U. K. Mallela and A. Upadhyay, "Buckling of laminated composite stiffened panels subjected to in-plane shear : A parametric study," vol. 44, pp. 354–361, 2006.

[8]   I. Dima, "Buckling of Flat Thin Plates under Combined Loading," vol. 7, no. 1, pp. 83–96, 2015.

[9]   W. J. Stroud, W. H. Greene, and M. S. Anderson, "NASA Technical Paper, Buckling Loads of Stiffened Panels Subjected to Combined Longitudinal Compression and Shear: Results Obtained With PASCO, EAL, and STAGS Computer Programs," 1984.

[10]   M. Muameleci, "Linear and Nonlinear Buckling Analyses of Plates using the Finite Element Method," Master's Thesis, Linkoping University.

[11]   Z. Al-azzawi, T. Stratford, J. M. Rotter, and L. A. Bisby, "Effect of Flange and Stiffener Rigidity on the Boundary Conditionss and Shear Buckling Stress of Plate Girders," *16th Eur. Bridg. Eng. Conf.*, no. January, 2015.

[12]   S. C. Lee, J. S. Davidson, and C. H. Yoo, "Buckling Coefficients of Plate Girder Web Panels," *Comput. Struct. J.*, vol. 59, no. 5, pp. 789–795, 1996.

[13]   E. Aydın, "Post-Buckling Behaviour of Metallic Skin-Stringer Assemblies and Buckling of Composite Flat Panels," Master's Thesis, Middle East Technical University, 2018.

[14] Ö. Asilkan and S. Irmak, "İkinci El Otomobillerin Gelecekteki Fiyatlarinin Yapay Sinir Aglari ile Tahmin Edilmesi," *J. Fac. Econ. Adm. Sci.*, vol. 14, no. 2, pp. 375–391, 2009.

[15] D. J. Haas, J. Milano, and L. Flitter, "Prediction of Helicopter Component Loads Using Neural Networks," *J. Am. Helicopter Soc.*, vol. 40, no. 1, pp. 72–82, 2009.

[16] Ş. Parlakyıldız, "Yapay Sinir Ağlari Kullanılarak Parmak İzi Tanıma ve Sınıflandırma," Master's Thesis, Gazi University, 2014.

[17] C. Bisagni and L. Lanzi, "Post-buckling optimisation of composite stiffened panels using neural networks," *Compos. Struct.*, vol. 58, no. 2, pp. 237–247, 2002.

[18] U. K. Mallela and A. Upadhyay, "Buckling load prediction of laminated composite stiffened panels subjected to in-plane shear using artificial neural networks," *Thin-Walled Struct.*, vol. 102, pp. 158–164, 2016.

[19] A. Cankur, "Development of an Artificial Neural Network Based Analysis Method for Skin-Stringer Structures," Master's Thesis, Middle East Technical University, 2017.

[20] S. S. Rao, *Engineering Optimization: Theory and Practice*, 4th ed. John Wiley & Sons, 2009.

[21] M. J. Jweeg, S. J. AL-Tornachi, and S. H. Abid-Aun, "Optimization of Light Weight Aircraft Wing Structure," *J. Eng. Dev.*, vol. 12, no. 1, pp. 1–22, 2008.

[22] A. Rispler and J. Raju, "Optimization of an Aircraft Control Surface," *ANSYS Conf. Seattle*, vol. 3, no. 11, 2003.

[23] S. A. Falco and A. R. de Faria, "Optimization of a simple aircraft wing," *Struct. Multidiscip. Optim. J.*, vol. 21, no. 4, pp. 2–13, 2002.

[24] P. Charbonneau, "An Introduction to Genetic Algorithms for Numerical Optimization," *Artif. Life*, vol. 3, no. 1, pp. 63–65, 2014.

[25] J. McCall, "Genetic algorithms for modelling and optimisation," *J. Comput. Appl. Math.*, 2005.

[26] M. O. Türe, "Shape Optimization of Helicopter Subfloor Intersection Element Under Crash Loading," Master's Thesis, Middle East Technical University, 2017.

[27] O. F. Hughes, B. Ghosh, and Y. Chen, "Improved prediction of simultaneous local and overall buckling of stiffened panels," *Thin-Walled Struct.*, vol. 42, no. 6, pp. 827–856, 2004.

[28] G. Gerard and H. Becker, "Handbook of Structural Stability Part I - Buckling

of Flat Plates," 1957.

[29] "Abaqus 6.13 Online Documentation," *Dassault Systemes*, 2013. [Online]. Available: http://dsk.ippt.pan.pl/docs/abaqus/v6.13/pdf_books/HELP.pdf. [Accessed: 20-Jun-2019].

[30] R. D. Cook and H. Saunders, "Concepts and Applications of Finite Element Analysis (2nd Edition)," *J. Press. Vessel Technol.*, 2009.

[31] M. Wallin, "A Finite Element tool for linear buckling analysis," Master's Thesis, Linkoping University, 2014.

[32] R. Kaur and A. Ambri, "Spars and Stringers- Function and Designing," *Int. J. Aerosp. Mech. Eng.*, vol. 1, no. 1, pp. 58–61, 2014.

[33] F. Aman, S. H. Cheraghi, K. K. Krishnan, and H. Lankarani, "Study of the impact of riveting sequence, rivet pitch, and gap between sheets on the quality of riveted lap joints using finite element method," *Int. J. Adv. Manuf. Technol.*, vol. 67, no. 1–4, pp. 545–562, Jul. 2013.

[34] A. Shahbazian and Y. C. Wang, "Calculating the global buckling resistance of thin-walled steel members with uniform and non-uniform elevated temperatures under axial compression," *Thin-Walled Struct.*, 2011.

[35] A. E. Örün, "Kiriş Yapılarındaki Hafifletme Deliği Etrafındaki Güçlendirmenin, Birleşik Yükler Altında Kritik Burkulma Gerilmesi Üzerindeki Etkisi," Master's Thesis, TOBB University of Economics and Technology, 2017.

[36] E. Aydin and A. Kayran, "Investigation of the Effect of Boundary Conditions on the Buckling Coefficients of Stiffened Flat Panels," *Int. Ankara Conf. Aerosp.*, 2017.

[37] S. Bräutigam, *Fairchild Dornier Calculation Methods for Airframe Static Stress, Strength and Stability Analysis Chapter 4, Issue B*. 2001.

[38] M. O. Efe, E. Abadoglu, and O. Kaynak, "Novel analysis and design of a neural network assisted nonlinear controller for a bioreactor," *Int. J. Robust Nonlinear Control*, vol. 9, no. 11, pp. 799–815, 1999.

[39] S. Gholizadeh, A. Pirmoz, and R. Attarnejad, "Assessment of load carrying capacity of castellated steel beams by neural networks," *J. Constr. Steel Res.*, vol. 67, no. 5, pp. 770–779, 2011.

[40] D. Graupe, *Principles of Artificial Neural Networks*, 3rd ed. Singapore: World Scientific Publishing Company, 2013.

[41] S. S. P. Kode, "Conversion of Neural Network Models to State-Space Models for Model-Based Control Design," Master's Thesis, The University of Alabama, 2015.

[42] H. M. Gomes and A. M. Awruch, "Comparison of response surface and neural network with other methods for structural reliability analysis," *Struct. Saf.*, pp. 2649–2667, 2004.

[43] A. Suratgar, M. Tavakoli, and A. Hoseinabadi, "Modified Levenberg–Marquardt method for neural networks training," *World Acad Sci Eng Technol*, vol. 1, no. 6, pp. 1719–1721, 2005.

[44] H. Demuth and M. Beale, "Neural Network Toolbox For Use with MATLAB," *The Math Works*. 2013.

[45] M. R. Veronez, S. F. de Souza, M. T. Matsuoka, A. Reinhardt, and R. M. da Silva, "Regional mapping of the geoid using GNSS (GPS) measurements and an artificial neural network," *Remote Sens.*, vol. 3, no. 4, pp. 668–683, 2011.

[46] A. Yıldırım, "Development of the Bolted Flange Design Tool Based on Finite Element Analysis and Artificial Neural Network," Master's Thesis, Middle East Technical University, 2015.

[47] M. Sarı, "Yapay sinir ağlari ve bir otomotiv firmasinda satiş talep tahmini uygulamasi," Master's Thesis, Sakarya University, Turkey, 2016.

[48] A. Yıldırım, A. A. Akay, H. Gülaşık, D. Çöker, E. Gurses, and A. Kayran, "Development of Bolted Flange Design Tool Based on Finite Element Analysis and Artificial Neural Network," *ASME Int. Mech. Eng. Congr. Expo. Proc.*, 2015.

[49] E. Tosun, "Frezeleme İşlemlerinde Genetik Algoritma Yaklaşımı İle Kesme Koşullarının Optimizasyonu," Master's Thesis, Selçuk University, Turkey, 2006.

[50] A. Coşkun and N. Arıcı, "Genetik Algoritma Kullanılarak Deriden Geçiş Özelliklerine Göre Kimyasal Maddelerin Molekül Yapılarının Bulunması," *Politek. Derg.*, vol. 9, no. 4, pp. 255–261, 2006.

[51] S. Gözütok and O. N. Özdemir, "Refinement of Hydraulic Calibration for Water Supply Networks with Genetic Algorithms," *Fournal Fac. Eng. Archit. Gazi Univ.*, vol. 19, no. 2, 2004.

[52] C. A. Coello and A. D. Christiansen, "Multiobjective optimization of trusses using genetic algorithms," *Comput. Struct.*, vol. 75, no. 6, pp. 647–660, 2000.

APPENDICES

A. PYTHON SCRIPTS FOR LARGE NUMBER OF FINITE ELEMENT ANALYSIS

```python
from abaqus import *
from abaqusConstants import *
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import math
import odbAccess
import job
from jobMessage import *

# skin dimensions
skin_t_array=[0.635,0.81,1.0,1.27]
str_spc_array=[50.0, 100.0, 150.0, 200.0, 250.0]
#str_spc_array=[50.0,100.0,150.0,200.0,250.0]
skin_a_array=[250.0,350.0,450.0,550.0]

# stringer dimensions(should not be changed)
str_c_array=[15.0,20.0]
str_h_array=[20.0,30.0]
str_t_array=[0.635,0.81,1.0]

load_ratio_array=[0.0001, 0.1, 0.25, 1.0, 4.0, 10.0, 10000.0]

str_n=3
ri=3.0
edge_d=7.4
```

```python
def fonk(skin_t, str_spc, skin_a, str_c, str_h, str_t, load_ratio):
        path = ""
        #----SKIN SKETCH----

        skin_b=str_spc*(str_n+1)

        mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
sheetSize=400.0)
        mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-skin_b/2,
0.0),
                point2=(skin_b/2, skin_a))
        mdb.models['Model-1'].Part(dimensionality=THREE_D,        name='Part-1',
type=
                DEFORMABLE_BODY)
        mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
                mdb.models['Model-1'].sketches['__profile__'])
        del mdb.models['Model-1'].sketches['__profile__']

        #----STRINGER SKETCH----
        iter=0

        mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
sheetSize=400.0)

        while (iter<str_n):

                mdb.models['Model-1'].sketches['__profile__'].Line(point1=
                        (-(str_c-edge_d+str_c)+(-(str_n-1)*str_spc/2+iter*str_spc),
str_h+skin_t/2-str_t/2), point2=
                        (-(str_c-edge_d+str_t/2)+(-(str_n-1)*str_spc/2+iter*str_spc),
str_h+skin_t/2-str_t/2))
                mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
                        addUndoState=False, entity=
                        mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+2])

                mdb.models['Model-1'].sketches['__profile__'].Line(point1=
                        (-(str_c-edge_d+str_t/2)+(-(str_n-1)*str_spc/2+iter*str_spc),
str_h+skin_t/2-str_t/2), point2=
                        (-(str_c-edge_d+str_t/2)+(-(str_n-1)*str_spc/2+iter*str_spc),
skin_t/2+str_t/2))
                mdb.models['Model-
1'].sketches['__profile__'].VerticalConstraint(addUndoState=
```

```
                        False,                              entity=mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+3])
            mdb.models['Model-
1'].sketches['__profile__'].PerpendicularConstraint(
                addUndoState=False, entity1=
                mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+2], entity2=
                mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+3])

            mdb.models['Model-1'].sketches['__profile__'].Line(point1=
                (-(str_c-edge_d+str_t/2)+(-(str_n-1)*str_spc/2+iter*str_spc),
skin_t/2+str_t/2), point2=
                (edge_d+(-(str_n-1)*str_spc/2+iter*str_spc), skin_t/2+str_t/2))
            mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
                addUndoState=False, entity=
                mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+4])
            mdb.models['Model-
1'].sketches['__profile__'].PerpendicularConstraint(
                addUndoState=False, entity1=
                mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+3], entity2=
                mdb.models['Model-
1'].sketches['__profile__'].geometry[3*iter+4])
            iter=iter+1

    mdb.models['Model-1'].Part(dimensionality=THREE_D,        name='Part-2',
type=
        DEFORMABLE_BODY)
    mdb.models['Model-1'].parts['Part-2'].BaseShellExtrude(depth=skin_a,
sketch=
        mdb.models['Model-1'].sketches['__profile__'])
    del mdb.models['Model-1'].sketches['__profile__']

    # Fillet radius yapabilmek icin "nearpoint1" ve "nearpoint2" secimleri
yapiliyor, fakat bu secimler icin formul kabul etmiyor.
    # Dolayisiyla edit cinsinden pointleri tanimlayip ona gore girmek gerekiyor.
    # Bu konuda tek kistas spacing icin 100 mm kabul edilmesi. inputlari spacing
100 olacak sekilde girmek gerek!!!

    mdb.models['Model-1'].ConstrainedSketch(name='__edit__', objectToCopy=
        mdb.models['Model-1'].parts['Part-2'].features['Shell         extrude-
1'].sketch)
```

```
mdb.models['Model-1'].parts['Part-2'].projectReferencesOntoSketch(filter=
        COPLANAR_EDGES,                    sketch=mdb.models['Model-
1'].sketches['__edit__'],
        upToFeature=
        mdb.models['Model-1'].parts['Part-2'].features['Shell extrude-1'])


it1=0

aa1x = -14.758232116699-(str_n-1)*str_spc/2
# -114.758232116699
#                -(str_c-edge_d+str_t/2)+(-(str_n-1)*str_spc/2+iter*str_spc)-
6.658232116699
aa1y = 30.4793510437012
# 30.4793510437012
# str_h+skin_t/2-str_t/2+0.4793510437012

aa2x = -7.666877746582-(str_n-1)*str_spc/2
# -107.666877746582
#                                      -(str_c-edge_d+str_t/2)+(-(str_n-
1)*str_spc/2+iter*str_spc)+0.433122253418
aa2y = 23.1867713928223
# 23.1867713928223
# str_h+skin_t/2-str_t/2-6.8132286071777

aa3x = -7.46989440918-(str_n-1)*str_spc/2
# -107.46989440918
#                -(str_c-edge_d+str_t/2)+(-(str_n-1)*str_spc/2+iter*str_spc)-
6.658232116699
aa3y = 7.41904067993164
# 7.41904067993164
# str_h+skin_t/2-str_t/2+0.4793510437012

aa4x = -2.151374816895-(str_n-1)*str_spc/2
# -102.151374816895
#                                      -(str_c-edge_d+str_t/2)+(-(str_n-
1)*str_spc/2+iter*str_spc)+0.433122253418
aa4y = 1.11194610595703
# 1.11194610595703
# str_h+skin_t/2-str_t/2-6.8132286071777


while (it1<str_n):
```

```
            mdb.models['Model-1'].sketches['__edit__'].FilletByRadius(curve1=
                    mdb.models['Model-
1'].sketches['__edit__'].geometry[3*it1+2], curve2=
                    mdb.models['Model-
1'].sketches['__edit__'].geometry[3*it1+3], nearPoint1=(
                    aa1x+str_spc*it1, aa1y), nearPoint2=(
                    aa2x+str_spc*it1, aa2y), radius=ri+str_t/2)
            mdb.models['Model-1'].sketches['__edit__'].FilletByRadius(curve1=
                    mdb.models['Model-
1'].sketches['__edit__'].geometry[3*it1+3], curve2=
                    mdb.models['Model-
1'].sketches['__edit__'].geometry[3*it1+4], nearPoint1=(
                    aa3x+str_spc*it1, aa3y), nearPoint2=(
                    aa4x+str_spc*it1, aa4y), radius=ri+str_t/2)
        it1=it1+1

    mdb.models['Model-1'].parts['Part-2'].features['Shell extrude-1'].setValues(
            sketch=mdb.models['Model-1'].sketches['__edit__'])
    del mdb.models['Model-1'].sketches['__edit__']
    mdb.models['Model-1'].parts['Part-2'].regenerate()



    # CREATE MATERIAL - ALUMINUM

    mdb.models['Model-1'].Material(name='aluminum')
    mdb.models['Model-1'].materials['aluminum'].Density(table=((2780.0e-12, ),
))    #tonne/mm3
    mdb.models['Model-1'].materials['aluminum'].Elastic(table=((70000.0, 0.33),
))    #MPa

    # CREATE SECTION

    mdb.models['Model-
1'].HomogeneousShellSection(idealization=NO_IDEALIZATION,
            material='aluminum',                        name='skin_section',
poissonDefinition=DEFAULT,
            preIntegrate=ON,        thickness=skin_t,        thicknessField='',
thicknessModulus=None,
            thicknessType=UNIFORM, useDensity=OFF)
    mdb.models['Model-
1'].HomogeneousShellSection(idealization=NO_IDEALIZATION,
            material='aluminum',                        name='str_section',
poissonDefinition=DEFAULT,
```

```
            preIntegrate=ON,           thickness=str_t,           thicknessField='',
thicknessModulus=None,
            thicknessType=UNIFORM, useDensity=OFF)

    # SECTION ASSIGNMENT - SKIN

    mdb.models['Model-1'].parts['Part-1'].Set(faces=
            mdb.models['Model-1'].parts['Part-1'].faces, name='Set-1')
    mdb.models['Model-1'].parts['Part-1'].SectionAssignment(offset=0.0,
            offsetField='', offsetType=MIDDLE_SURFACE, region=
            mdb.models['Model-1'].parts['Part-1'].sets['Set-1'], sectionName=
            'skin_section', thicknessAssignment=FROM_SECTION)

    # SECTION ASSIGNMENT

    fc_n=5*str_n

    mdb.models['Model-1'].parts['Part-2'].Set(faces=
            mdb.models['Model-1'].parts['Part-2'].faces[0:fc_n], name='Set-1')
    mdb.models['Model-1'].parts['Part-2'].SectionAssignment(offset=0.0,
            offsetField='', offsetType=MIDDLE_SURFACE, region=
            mdb.models['Model-1'].parts['Part-2'].sets['Set-1'], sectionName=
            'str_section', thicknessAssignment=FROM_SECTION)

    # CREATE ASSEMBLY

    mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
    mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,  name='Part-
1-1',
            part=mdb.models['Model-1'].parts['Part-1'])
    mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,  name='Part-
2-1',
            part=mdb.models['Model-1'].parts['Part-2'])

    # SKIN ROTATION

    mdb.models['Model-1'].rootAssembly.rotate(angle=90.0,
axisDirection=(skin_b,
            0.0, 0.0), axisPoint=(-skin_b/2, 0.0, 0.0), instanceList=('Part-1-1', ))

    # CREATE ATTACHMENT POINTS

    it2=0
```

```
        while (it2<str_n):

                mdb.models['Model-
1'].rootAssembly.AttachmentPointsOffsetFromEdges(edges=
                        mdb.models['Model-1'].rootAssembly.instances['Part-2-
1'].edges[(16*it2+3):(16*it2+4)],
                        name='Attachment Points-'+str(it2+1),
                        numberOfRows=1,                    offsetFromEdges=edge_d,
offsetFromStartPoint=8.0,
                        patterningMethod=PATTERN_ORTHOGONALLY,
pointCreationMethod=BY_SPACING,
                        referenceFace=
                        mdb.models['Model-1'].rootAssembly.instances['Part-2-
1'].faces[(5*it2)], setName=
                        'Attachment                    Points-'+str(it2+1)+'-Set-1',
spacingBetweenPoints=16.0, startPoint=
                        mdb.models['Model-1'].rootAssembly.instances['Part-2-
1'].vertices[(12*it2)])
                it2=it2+1


        # CREATE FASTENERS

        mdb.models['Model-1'].ConnectorSection(assembledType=BEAM,
name='ConnSect-1')
        mdb.models['Model-1'].rootAssembly.Surface(name='skin_face',
side1Faces=
                mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].faces[0:1])

        it3=0
        while (it3<str_n):

                mdb.models['Model-1'].rootAssembly.Surface(name='att_face-
'+str(it3+1), side1Faces=
                        mdb.models['Model-1'].rootAssembly.instances['Part-2-
1'].faces[(5*it3):(5*it3+1)])
                it3=it3+1

        it4=0
        while (it4<str_n):

                mdb.models['Model-
1'].rootAssembly.engineeringFeatures.PointFastener(
                        additionalMass=0.0001,        name='Fasteners-'+str(it4+1),
```

```
physicalRadius=1.6, region=
                    mdb.models['Model-1'].rootAssembly.sets['Attachment Points-
'+str(it4+1)+'-Set-1'],
                    sectionName='ConnSect-1', targetSurfaces=(
                    mdb.models['Model-1'].rootAssembly.surfaces['skin_face'],
                    mdb.models['Model-1'].rootAssembly.surfaces['att_face-
'+str(it4+1)]), unsorted=OFF)
            it4=it4+1

    # CREATE STEP

    mdb.models['Model-1'].BuckleStep(name='Step-1',                numEigen=10,
previous='Initial'
            , vectors=18)
    mdb.models['Model-1'].steps['Step-1'].setValues(maxIterations=100)

    # CREATE MESH

    mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1,
            minSizeFactor=0.1, size=5.0)
    mdb.models['Model-1'].parts['Part-1'].generateMesh()
    mdb.models['Model-1'].parts['Part-2'].seedPart(deviationFactor=0.1,
            minSizeFactor=0.1, size=3.0)
    mdb.models['Model-1'].parts['Part-2'].generateMesh()

    # BOUNDARY CONDITION - SKIN EDGES

    # y-direction translational
    mdb.models['Model-1'].rootAssembly.Set(edges=
            mdb.models['Model-1'].rootAssembly.instances['Part-1-
1'].edges[0:1]+\
            mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].edges[2:3],
name='Set-6')
    mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Initial',
            distributionType=UNIFORM,    fieldName='',    localCsys=None,
name='BC-1',
            region=mdb.models['Model-1'].rootAssembly.sets['Set-6'],
u1=UNSET, u2=SET,
            u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=UNSET)

    mdb.models['Model-1'].rootAssembly.Set(edges=
            mdb.models['Model-1'].rootAssembly.instances['Part-1-
1'].edges[1:2]+\
```

```
        mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].edges[3:4],
name='Set-7')
        mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Initial',
            distributionType=UNIFORM,     fieldName='',     localCsys=None,
name='BC-2',
            region=mdb.models['Model-1'].rootAssembly.sets['Set-7'],
u1=UNSET, u2=SET,
            u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=UNSET)


    # x-direction translational
    mdb.models['Model-1'].rootAssembly.Set(edges=
        mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].edges[3:4],
name='Set-8')
        mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Initial',
            distributionType=UNIFORM,     fieldName='',     localCsys=None,
name='BC-3',
            region=mdb.models['Model-1'].rootAssembly.sets['Set-8'],   u1=SET,
u2=UNSET,
            u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=UNSET)


    # z-direction translational
    mdb.models['Model-1'].rootAssembly.Set(edges=
        mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].edges[2:3],
name='Set-9')
        mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Initial',
            distributionType=UNIFORM,     fieldName='',     localCsys=None,
name='BC-4',
            region=mdb.models['Model-1'].rootAssembly.sets['Set-9'],
u1=UNSET, u2=UNSET,
            u3=SET, ur1=UNSET, ur2=UNSET, ur3=UNSET)


    # Z-direction ROTational
    mdb.models['Model-1'].rootAssembly.Set(edges=
      mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].edges[1:2]+\
            mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].edges[3:4],
name='Set-10')
        mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Initial',
        distributionType=UNIFORM, fieldName='', localCsys=None, name='BC-
5',
        region=mdb.models['Model-1'].rootAssembly.sets['Set-10'],    u1=UNSET,
```

```
u2=UNSET,
        u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=SET)

    # CREATE COMPRESSION & SHEAR EDGE LOADS

    mdb.models['Model-1'].rootAssembly.Surface(name='Surf-0', side1Edges=
        mdb.models['Model-1'].rootAssembly.instances['Part-1-
1'].edges[0:1])
    mdb.models['Model-1'].rootAssembly.Surface(name='Surf-1', side1Edges=
        mdb.models['Model-1'].rootAssembly.instances['Part-1-
1'].edges[1:2])
    mdb.models['Model-1'].rootAssembly.Surface(name='Surf-2', side1Edges=
        mdb.models['Model-1'].rootAssembly.instances['Part-1-
1'].edges[2:3])
    mdb.models['Model-1'].rootAssembly.Surface(name='Surf-3', side1Edges=
        mdb.models['Model-1'].rootAssembly.instances['Part-1-
1'].edges[3:4])

    if load_ratio == 0.0001:
        compr = 0.0001
        shear = 1.0
    elif load_ratio == 0.1:
        compr = 0.1
        shear = 1.0
#   elif load_ratio == 0.25:
#       compr = 0.25
#       shear = 1.0
    elif load_ratio == 1.0:
        compr = 1.0
        shear = 1.0
#   elif load_ratio == 4.0:
#       compr = 1.0
#       shear = 0.25
    elif load_ratio == 10.0:
        compr = 1.0
        shear = 0.1
    else:
        compr = 1.0
        shear = 0.0001

    mdb.models['Model-1'].ShellEdgeLoad(createStepName='Step-1',
distributionType=
        UNIFORM,    field='',    localCsys=None,    magnitude=compr,
name='compression',
```

```
            region=mdb.models['Model-1'].rootAssembly.surfaces['Surf-0'],
resultant=ON)

        mdb.models['Model-1'].ShellEdgeLoad(createStepName='Step-1',
distributionType=
            UNIFORM,    field='',    localCsys=None,    magnitude=shear,
name='shear0', region=
            mdb.models['Model-1'].rootAssembly.surfaces['Surf-0'],
resultant=ON,
            traction=SHEAR)


        mdb.models['Model-1'].ShellEdgeLoad(createStepName='Step-1',
distributionType=
            UNIFORM,    field='',    localCsys=None,    magnitude=-shear,
name='shear1', region=
            mdb.models['Model-1'].rootAssembly.surfaces['Surf-1'],
resultant=ON,
            traction=SHEAR)


        mdb.models['Model-1'].ShellEdgeLoad(createStepName='Step-1',
distributionType=
            UNIFORM,    field='',    localCsys=None,    magnitude=shear,
name='shear2', region=
            mdb.models['Model-1'].rootAssembly.surfaces['Surf-2'],
resultant=ON,
            traction=SHEAR)


        mdb.models['Model-1'].ShellEdgeLoad(createStepName='Step-1',
distributionType=
            UNIFORM,    field='',    localCsys=None,    magnitude=-shear,
name='shear3', region=
            mdb.models['Model-1'].rootAssembly.surfaces['Surf-3'],
resultant=ON,
            traction=SHEAR)


    # CREATE JOB

    mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
            explicitPrecision=SINGLE,    getMemoryFromAnalysis=True,
historyPrint=OFF,
            memory=90,    memoryUnits=PERCENTAGE,    model='Model-1',
modelPrint=OFF,
            multiprocessingMode=DEFAULT,    name=jname,
nodalOutputPrecision=SINGLE,
```

```
                    numCpus=1,   numGPUs=0,   queue=None,   resultsFormat=ODB,
scratch='', type=
                    ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)


        mdb.jobs[jname].submit(consistencyChecking=OFF)
        mdb.jobs[jname].waitForCompletion()


        dat_file = open(jname+".dat","r")
        # her satir "lines" array'inin bir elemani oldu.
        lines = dat_file.readlines()
        for line in lines:
                if " MODE NO      EIGENVALUE" in line:
                        eigenval = lines[lines.index(line)+3].split('      ')[2]
                        floatEigenVal = float(eigenval)
                        floatEigenVal = abs(floatEigenVal)

        myOdb = odbAccess.openOdb(path=path+jname+".odb", readOnly=True)
        mass = myOdb.steps["Step-1"].mass


        return [floatEigenVal,mass,compr,shear]

st1 = 1
out_file = open("out_file.out", "wb")
out_file.write("          comp_app  skin_t  str_spc  skin_a    str_c    str_h    str_t
mass_total    shear_app  \n")
for skin_t in skin_t_array:
      for str_spc in str_spc_array:
            for skin_a in skin_a_array:
                  for str_c in str_c_array:
                        for str_h in str_h_array:
                              for str_t in str_t_array:
                                    for load_ratio in load_ratio_array:
                                          jname = "Set-"+str(st1)

        #+str(skin_t)+"v2_"+str(str_spc)+"v3_"+str(skin_a)+"v4_"+str(str_c)+"v5_"
+str(str_h)+"v6_"+str(str_t)

                                          fonk_out=fonk(skin_t,    str_spc,
skin_a, str_c, str_h, str_t, load_ratio)


                                          #eigenvalue = fonk_out[0]
                                          eigenvalue=fonk_out[0]
```

104

```
                                          mass_total=fonk_out[1]

        comp_app=eigenvalue*fonk_out[2]

        shear_app=eigenvalue*fonk_out[3]
                                          out_str = "%9s %10s %9s %9s
%9s %9s %9s %9s  %9s  %14s \n" %(str(jname), comp_app, skin_t, str_spc, skin_a,
str_c, str_h, str_t, mass_total, shear_app)
                                          #asagiya yazilmasi icin mass
eklenecek
                                          out_file.write(out_str)

                                          st1=st1+1

out_file.close()
```

B. MATLAB MARGIN OF SAFETY SCRIPT

```matlab
function [c,c_eq] = constraint2(x)


allX1_6 = [0.406, 0.508, 0.635, 0.813, 1.016, 1.27, 1.42, 1.63, 2.032];


x(1) = allX1_6(x(1));

x(6) = allX1_6(x(6));


% set-99
% constraint2 ([0.813,  500,   500,   15,   20,   1,   10.1,   10,   5   ])
% constraint2  [skin_t, skin_b, skin_a, str_c, str_h, str_t, comp_app, shear_app, str_n]
%  for optimtool -   [0.635 500 500 15 20 1 10.1 10 3]    [1.27 500 500 15 20 1 10.1 10 8]

load network1.mat
load network2.mat
load network3.mat
load network4.mat
load network5.mat
load network6.mat
load network7.mat

%Compression/Shear Ratio Define
load_ratio=x(7)/x(8);

%Geometric Parameters not normalized
skin_t=x(1)/1.27;

str_spc=(x(2)/(x(9)+1))/250;

skin_a=x(3)/550;
str_c=x(4)/20;
str_h=x(5)/30;
str_t=x(6)/1;
str_n=x(9);

%Comp/Shear=0
shear1=network1([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*145.44;
comp1=0;
%Comp/Shear=0.1
shear2=network2([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*140.71;
comp2=shear2*0.1;
%Comp/Shear=0.25
```

```matlab
shear3=network3([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*133.96;
comp3=shear3*0.25;
%Comp/Shear=1.0
shear4=network4([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*105.8;
comp4=shear4*1.0;
%Comp/Shear=4
shear5=network5([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*45.865;
comp5=shear5*4;
%Comp/Shear=10
shear6=network6([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*19.59;
comp6=shear6*10;
%Comp/Shear=10000
x7=network7([skin_t,str_spc,skin_a,str_c,str_h,str_t]')*0.019838;
comp7=x7*10000;
shear7=0;

if (load_ratio>0.0001) && (load_ratio<0.1)
    x1=comp1;
    y1=shear1;
    x2=comp2;
    y2=shear2;

elseif (load_ratio>0.1) && (load_ratio<0.25)
    x1=comp2;
    y1=shear2;
    x2=comp3;
    y2=shear3;

elseif (load_ratio>0.25) && (load_ratio<1)
    x1=comp3;
    y1=shear3;
    x2=comp4;
    y2=shear4;

elseif (load_ratio>1) && (load_ratio<4)
    x1=comp4;
    y1=shear4;
    x2=comp5;
    y2=shear5;

elseif (load_ratio>4) && (load_ratio<10)
    x1=comp5;
    y1=shear5;
    x2=comp6;
    y2=shear6;

elseif (load_ratio>10) && (load_ratio<10000)
    x1=comp6;
    y1=shear6;
```

```matlab
    x2=comp7;
    y2=shear7;

end

shear_all=(x1*(y2-y1)-y1*(x2-x1))/(load_ratio*(y2-y1)-(x2-x1));
comp_all=load_ratio*shear_all;

c= (-comp_all)/x(7)+1;
MS_shear=shear_all/x(8)-1;

c_eq=[];

%shear1,comp1
%shear2,comp2
%shear3,comp3
%shear4,comp4
%shear5,comp5
%shear6,comp6
%shear7,comp7
%shear_all,comp_all
%MS_comp,MS_shear

%MS_comp

end
```

C. MATLAB WEIGHT SCRIPT

```matlab
function weight = objective2(x)

 allX1_6 = [0.406, 0.508, 0.635, 0.813, 1.016, 1.27, 1.42, 1.63, 2.032];

 x(1) = allX1_6(x(1));

 x(6) = allX1_6(x(6));

 % set-99
% objective2 ([0.813,  500,   500,   15,   20,   1,    10.1,   10,    5  ])
% objective2  [skin_t, skin_b, skin_a, str_c, str_h, str_t, comp_app, shear_app, str_n]
% for optimtool -  [0.635 500 500 15 20 1 10.1 10 3]    [1.27 500 500 15 20 1 10.1 10 8]

skin_t=x(1);
skin_b=x(2);
skin_a=x(3);
str_c=x(4);
str_h=x(5);
str_t=x(6);

ri=3.0;

str_spc=skin_b/(x(9)+1);

%Skin Section Area Calculation
skin_area=skin_t*skin_b;

%Stringer Section Area Calculation
str_c1=str_c-(ri+str_t);
str_h1=str_h-2*(ri+str_t);
ro=ri+str_t;

str_area1=str_t*(2*str_c1+str_h1);
str_area2=0.5*((pi*ro*ro)-(pi*ri*ri));

str_area=x(9)*(str_area1+str_area2);

%Total volume (mm3) & weight (kg)
volume=skin_a*(skin_area+str_area);

weight=(volume/(10^9))*(2734);

end
```