5-2016

Design Optimization of Tank Track Pad Meta-Material Using the Cell Synthesis Method

Neehar Milind Kulkarni
Clemson University, neehar.kulkarni@gmail.com

DESIGN OPTIMIZATION OF TANK TRACK PAD META-MATERIAL USING THE UNIT CELL SYNTHESIS METHOD

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Neehar Milind Kulkarni
May 2016

Accepted by:
Dr. Georges Fadel, Committee Chair
Dr. Nicole Coutris
Dr. Gang Li
Dr. Matthew Castanier (US Army – TARDEC)

ABSTRACT

The elastomeric backer pad on the M1 Abrams tank track experiences highly cyclic and dynamic loads during normal operating conditions. As a result, extensive heat is generated within the pad due to its viscoelastic hysteretic nature which leads to its early failure. Research has been carried out in the past at Clemson University to design a meta-material that will mimic the deformation behavior of the elastomeric backer pad but will be made out of a linearly elastic constitutive material to eliminate hysteresis. A meta-material in this context is an artificial material in the form of a periodic structure that exhibits effective properties that differ from its constitutive material. Previous attempts to design a feasible meta-material as an effective replacement to the existing elastomeric backer pad have been unsuccessful. The work carried out in this research therefore, is focused on developing a meta-material that satisfies all the application specific requirements. The meta-material is designed based on the steps prescribed by the Unit Cell Synthesis Method which was developed in previous research. Using this method, a unit cell based periodic meta-material can be designed that exhibits nonlinear deformation behavior by implementing various combinations of different elemental geometries that show geometric nonlinearity under deformation. The idea is to attain a targeted nonlinear deformation response of the meta-material structure by tuning the geometric nonlinearities of one or multiple entities in order to replace the material nonlinearity of the target material. A modification is proposed to the original method to make it more efficient by introducing a multi-objective optimization step that considers all the relevant feasibility criteria

concerning the meta-material design. Two unit cell based meta-material concepts are evaluated and a best meta-material design is chosen based on the results obtained from the multi-objective optimization problem. The optimized meta-material is then subjected to dynamic tank wheel roll-over conditions to compare its deformation response with that of the original pad. Finally, conclusions are drawn and scope for future work is discussed.

DEDICATION

To my family and friends.

ACKNOWLEDGMENTS

I would like to thank my family and friends for their unconditional support. This research would not have been possible without their strong backing. I would like to thank Dr. Fadel for believing in me and providing me the opportunity to be a part of this research project. His guidance, attention to detail and enthusiasm towards the project has helped me during all the stages of the research. I would also like to thank my committee members, Dr. Coutris and Dr. Li for guiding me throughout the course of this research. Their valuable insights in structural mechanics and finite element analysis have helped me make progress with the research. Thanks to Zach Satterfield for his collaboration during the early part of this research.

I would like to thank Matthew Castanier, Bill Bradford, David Ostberg and the US Army TARDEC and Automotive Research Center at University of Michigan for their financial support of this research. I would also like to thank Christopher Cardine of General Dynamics and Joshua Goossens of Tenneco for providing an industrial perspective on this research project.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (CONTINUED)

Page

TABLE OF CONTENTS (CONTINUED)

Page

TABLE OF CONTENTS (CONTINUED)

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF FIGURES (CONTINUED)

Page

LIST OF FIGURES (CONTINUED)

Page

CHAPTER 1. INTRODUCTION

1.1. Overview of the Abrams Tank Track Pad System

The M1 Abrams tanks produced by the United States weigh around 63 tons and are capable of travelling at up to 45 miles per hour over rough terrain [1]. The track of these tanks are equipped with highly compliant elastomeric pads capable of exhibiting high strains at low stress levels. They help minimize damage to roads and also provide vibration damping, noise reduction and better traction in a wide range of terrain conditions [2]. The track components of the M1 tank are shown in Figure 1.1.

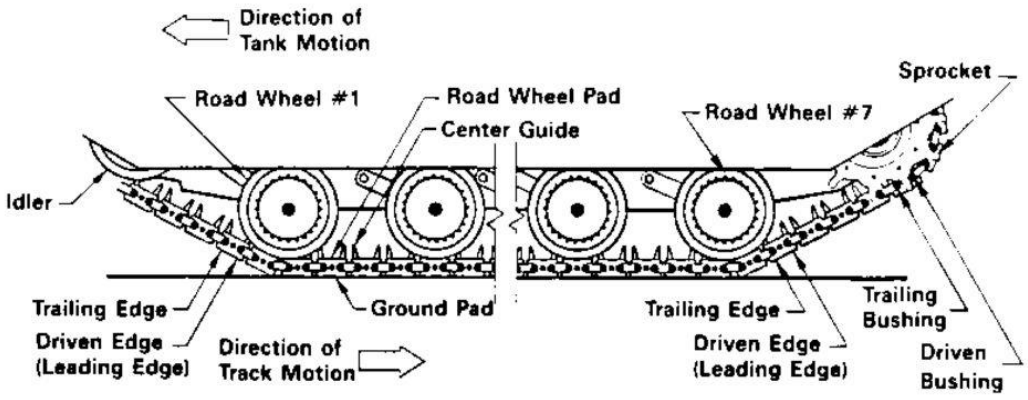

Figure 1.1 M1 Track Components [2]

The tank consists of 7 road wheels on either side of the tank body. These road wheels which are connected by a suspension system, act as idler wheels and support the weight of the tank [2]. The track comprises of steel links and each link consists of three primary components: the ground pad, the steel plate structure, and the backer pad. Figure 1.2 shows a detailed 2D representation of a single wheel-track link assembly. The ground

pad acts between the ground and the steel link while the backer pad (road wheel pad) is in contact with the road wheels. This research primarily focusses on the backer pad.

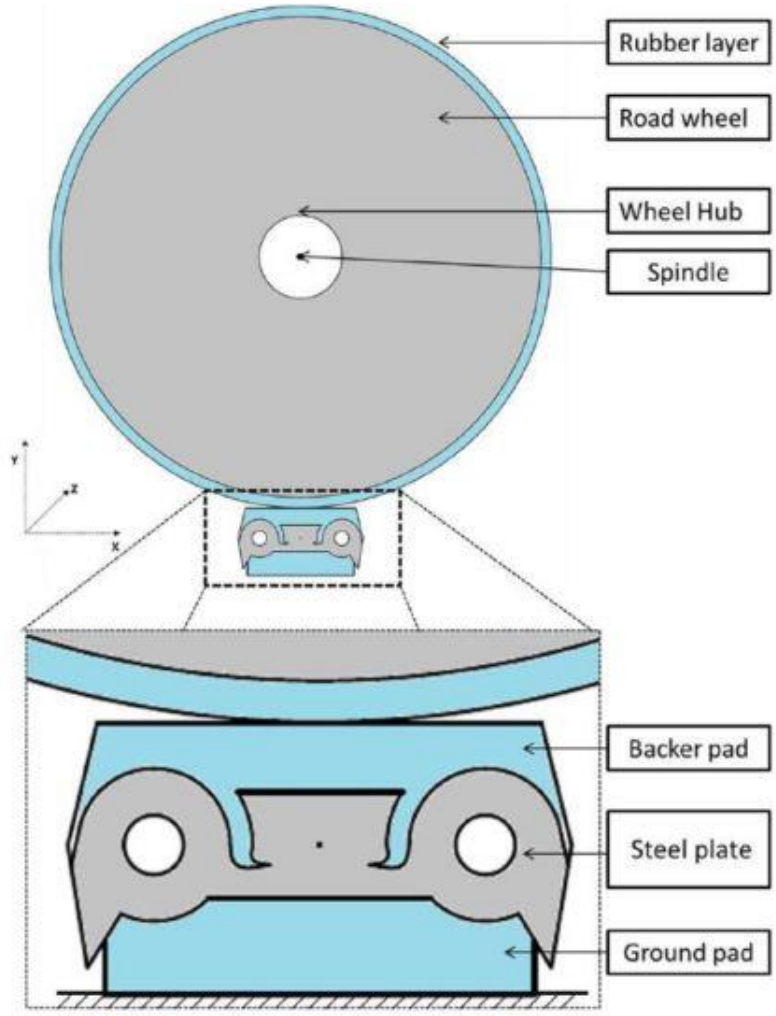

Figure 1.2 2D Representation of Road Wheel and Track Link [3]

As the tank moves forward, the ground pads come in contact with the road, and a succession of seven road wheels pass over the backer pads. As the seventh road wheel passes, the pads pass over the drive sprocket at the rear of the tank and then return to the front as part of an endless belt. During a 500 mile test, a single pad undergoes this cycle approximately 53000 times [2].

1.2. Motivation for Replacing the Current Elastomer Track Pad

The current tank track pads are manufactured from Styrene Butadiene Rubber (SBR) combined with a filler material [1]. The pads show a limited and varying service life depending on the type of terrain they encounter. Due to the stresses developed in tension, compression, and shear, during the normal operation of the tanks, the elastomeric components of the track pad become the life limiting components of the track system [4]. The tank track pad failures are mainly characterized into four types: abrasion, cutting, chunking and blowout. On paved roads, the pad often experiences abrasion and blowouts. Blowouts occur due to excessive internal pad temperatures developed in the rubber due to hysteresis. On gravel or cross-country terrain, cutting and chunking are responsible for pad failure. Road hazards or rigid obstacles produce localized loads on the pads which leads to cutting. Chunking occurs when these cuts are propagated to failure or due to an overloaded impact on the pad [5].

Figure 1.3 Embrittlement and Cracking of Backer Pad and Bushing Assembly [6]

Figure 1.4 Progression of Chunking of a Backer Pad [6]

Figure 1.3 shows an example of embrittlement and cracking in a backer pad and bushing assembly. Figure 1.4 shows the progression of loss of material caused by chunking (highlighted by white arrows) versus component life. Each sample (from left to right) has been collected at an interval increment of 250 miles [6].

Due to premature failure, the track pads on the M1 tank have to be replaced after every 850 miles of use on an average [1]. This has a significant impact on life cycle costs, logistics, field support, and vehicle/war fighter effectiveness [4]. A report by Army Materials and Mechanics Research Center (AMMRC) included this statement from the TACOM Track and Suspension R&D Symposium, 29-30 March 1982: "Current annual repair and replacement costs for track rubber used in tanks and other track vehicles are estimated to be in the range of $ 100,000,000; this estimate is expected to double within the next ten years with the full implementation of the Ml main battle tank into the Army inventory" [1]. Research work has been carried out in the past to analyze the failure modes of the track pads and to improve the reliability of the track components.

1.3. Literature Review

Gogos et al. in [7] have experimentally investigated the effects of material, operational and geometric parameters on the heat generation and subsequent failure in the elastomeric pads. They have also studied the impacts of these parameters on the pad life during its manufacturing using the injection mold curing process. In [8], Lesuer et al. have focused on determining the failure mechanisms that limit the service life of the pads. They have performed finite element analyses simulating structural and thermal response of the pads to tank operation as well as field testing. They have concluded that the normal and effective stresses produced in the pad due to distortion are quite low, relative to stresses that may cause fracture. Further, excessive heat generation and temperature build-up has been recognized as the cause of early failure.

The pads made out of SBR, which is an energy dissipative material, generate significant amount of heat during the compressive loading and unloading cycles due to their hysteretic nature. Due to the low ability of the SBR compound to dissipate heat, the resulting temperature build-up causes premature failure of the pads.

[1] and [9] explore the use of elastomers other than SBR to improve the durability and reliability of the track pads. Lentz et al. in [9], have used a 'tri-blend' rubber-fiber composite for the track pad. It consists of natural rubber, butadiene-styrene rubber, and polybutadiene rubber along with Kevlar 29 which is the aramid fiber in the composite. They have reported improved hot tear resistance of the new composite compound based on the results obtained from lab testing. Also, in order to address the heat build-up in the rubber pads which is attributed to rubber's low thermal conductivity, Katz et al. in [1], have

attempted to use fillers and reinforcements with higher thermal conductivity than rubber to create a mechanism for dissipation of the heat buildup. The heat dissipation is allowed to occur via conduction to the adjacent metal components, so that blowout does not occur. They have also investigated the use of specially formulated polyurethane elastomers as a replacement for SBR. Polyurethanes exhibit higher strength and excellent abrasion resistance compared to rubber. However, they exhibit higher hysteresis and are susceptible to hydrolysis. This limits their use on the track pads.

In [5], Lesuer et al. have shown that apart from compressive loading, the track pads experience tensile stresses of significant magnitude when they encounter obstacles and from the large applied shear stresses produced during turning operations. They have concluded that these conditions are the primary sources of damage and chunking in track pads. They have also reported temperature data where a maximum temperature of 295° F has been recorded in the interior regions of the pads tested on cross-country terrain.

In [10], Lesuer et al. have implemented computer models to get an insight into the field response of tracks. Two models have been developed; first, a mechanical model, has been used to examine the stresses and the irreversible mechanical work done in the rubber portions of the track. Second, a thermal model has been implemented to evaluate the temperatures developed during operation of the tanks. They have shown that the service life of the track pads is a function of temperature, environment, and the number of fatigue cycles. Figure 1.5 shows decaying residual strength (ultimate tensile strength) of the elastomer with increase in temperature and cyclic stress magnitude.

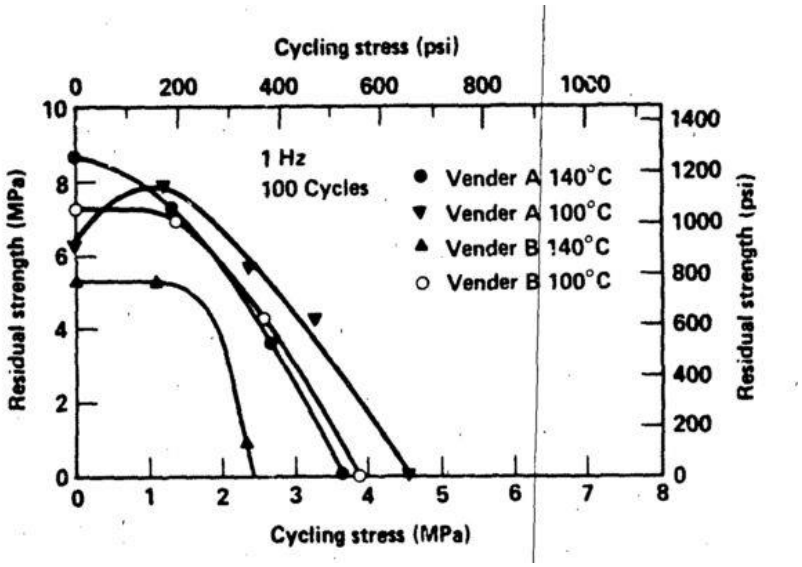

Figure 1.5 Decaying Residual Strength of Elastomer w.r.t Cyclic Stress [10]

Further attempts to analyze the various mechanisms of wear and failure have been carried out in [2] and [11]. Ostberg et al. in [6] have analyzed the loading distribution of the Abrams suspension systems and its impact on track life. Figure 1.6 shows a thermal map of the tank track after undergoing 12 miles of track testing at 40 mph.

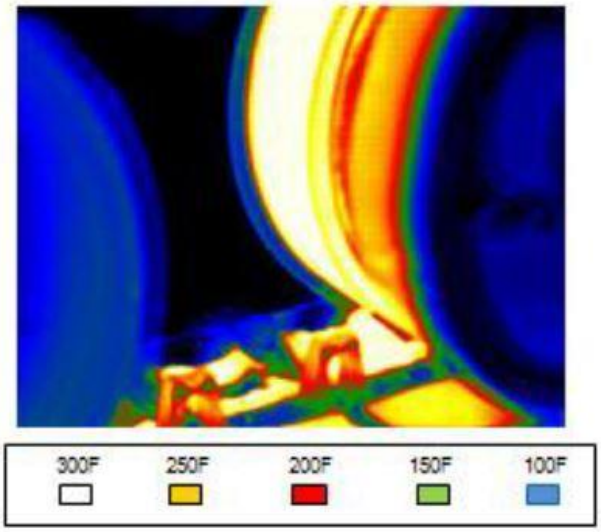

Figure 1.6 Thermal Map of M1 Abrams Track pad and Road Wheel [6]

They have modelled and analyzed a 3D track suspension system of the Abrams tank as shown in Figure 1.7 to calculate the optimum camber angle for uniform load distribution over the track pads. A 2D dynamic analysis of the suspension system has also been carried out to simulate the motion of road wheel on the track assembly to help estimate the change in component life due to change in strain field resulting from the change in loading. They have reported that the Abrams suspension system loads the outer track and road wheels greater than the inner side. They have concluded that optimizing the road arm camber angle will provide a uniform loading distribution and thus, extend the durability and life of current track components by lowering both strain and temperature on the outboard components [6].

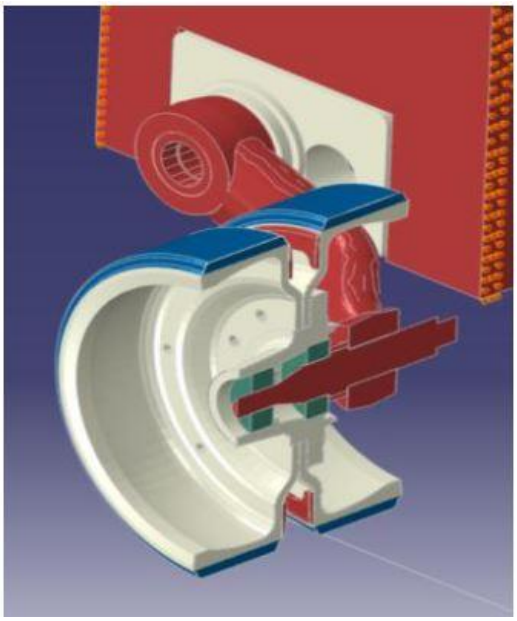

Figure 1.7 1/4th Model of Abrams Suspension Assembly [6]

In [12], Mars et al. have performed a dynamic finite element analysis and a fatigue analysis by simulating wheel roll-over on the track pad assembly. They have modelled the existing elastomer in the track pad using a 2nd Order Ogden Hyperelastic material model.

Based on the results obtained, they have concluded that the backer pad has the least fatigue life among the track pad components. This behavior is in accordance with the results obtained experimentally as shown in Figure 1.8.

Figure 1.8 Evolution of Fatigue Damage in Backer Pads [12]

1.4. Designing a Meta-Material to Replace the Current Track Pads

Throughout the literature, the heat build-up in the track pads, especially the backer pads, has been highlighted as the predominant factor contributing to their premature failure. However, work towards the development of tank track pad has not extended beyond the testing for track pad failure modes and exploration of different elastomeric compounds and filler materials to improve wear resistance and durability [3]. The heat build-up in the pads is primarily attributed to the hysteretic property of the elastomer as shown in Figure 1.9 (Left). Hysteresis is a characteristic of viscoelastic materials where only a portion of the stored strain energy is recovered during unloading and the rest is converted to heat energy [3]. Linearly elastic materials such as steel, aluminum etc. do not show hysteretic

properties. A general stress-strain plot for linearly elastic materials for loading-unloading cycle is shown in Figure 1.9 (Right).

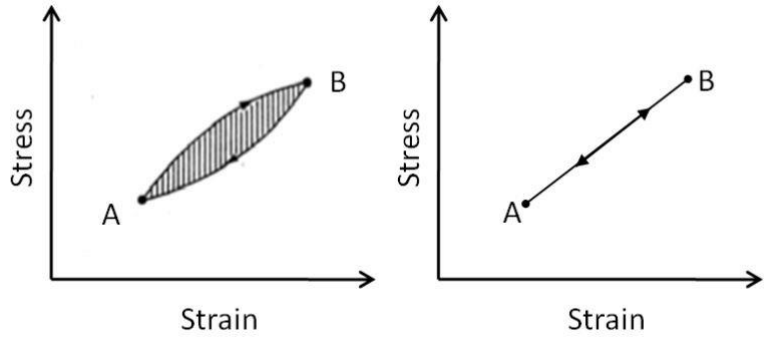

Figure 1.9 Loading – Unloading Stress-Strain Plot for Elastomer (Left) and Linear Elastic Material (Right) [13]

Another material property associated with damping is the loss co-efficient. It measures the degree to which a material dissipates vibrational energy [14]. A high loss coefficient is desirable for damping vibrations whereas a low loss coefficient material is able to transmit energy efficiently. The loss coefficient is also an important factor in determining a material's resistance to fatigue failure. A material with high value of loss coefficient subjected to cyclic loading will dissipate energy into itself leading to fatigue failure [3].

An approach to design a meta-material that takes into consideration the aspects of hysteresis, loss-coefficient and compliance has been introduced in [3]. Rodger Walser coined the term "meta-material" in 1999 and discussed the techniques to design meta-materials for a desired purpose in [11]. A meta-material has been defined in [15] as a macroscopic composite of periodic or non-periodic structures, whose function is due to both cellular architecture and chemical composition.

In [3], Dangeti has identified the properties of a meta-material that would replace the existing elastomeric track pads. It is desired that the proposed meta-material would be made out of linearly elastic materials and have lower loss-coefficient, thereby eliminating hysteresis and at the same time, have compliance similar to the elastomeric pads. The Ashby plot of loss coefficient and Young's modulus in Figure 1.10 shows the desired properties of the proposed meta-material. In [3], the meta-material elastic properties have been defined by determining tangent elasticity tensors for pure-stress states such as uniaxial tension, pure shear and equi-biaxial tension, at different sets of strain levels. These tensors are to be used to define constitutive equations to determine the meta-material unit cell (UC) topology using topology optimization.

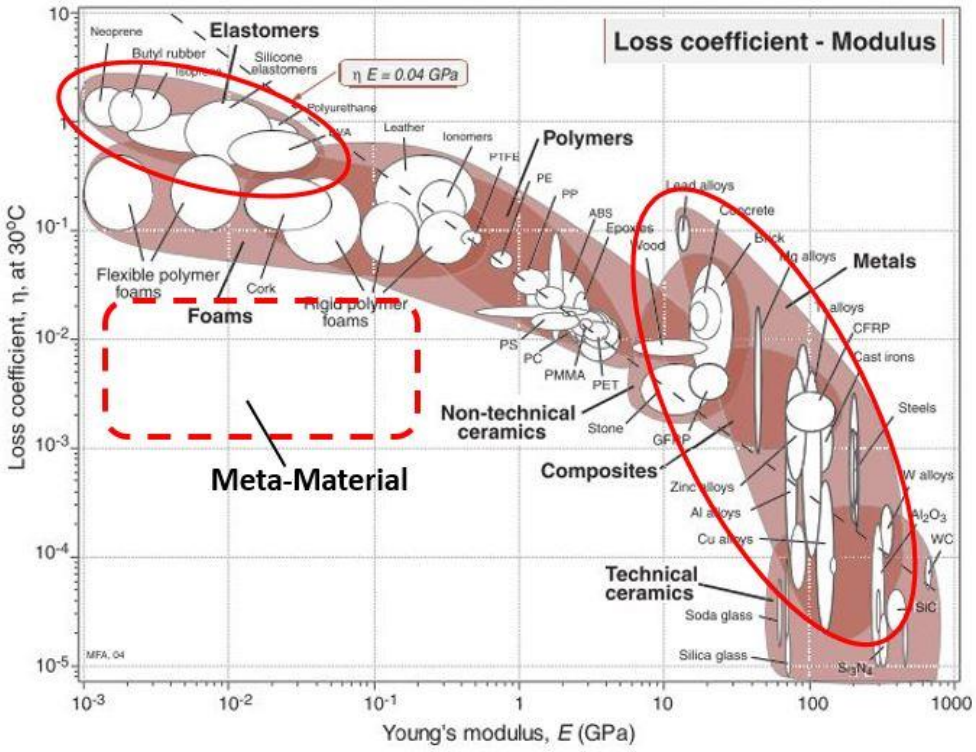

Figure 1.10 Loss-Coefficient - Young's Modulus Plot for Materials [14]

In [16], Satterfield has carried out a literature review of the methods used to design a meta-material and has reported that topology optimization is the predominantly used method. However, upon further investigation, topology optimization has been deemed infeasible to design a meta-material for the tank track pad application with the existing commercial software tools available. Three primary limitations of the current techniques for implementing topology optimization are identified in [16] and they are geometric nonlinearity, periodic boundary conditions and aspect ratio of the UC. Therefore, an alternate strategy to design UC based meta-materials has been adopted. The UC designs have been developed using engineering principles to achieve a desired nonlinear response that mimics the behavior of the elastomer employed in the track pad in uniaxial compression. Figure 1.11 shows a meta-material that consists of a fixed-fixed beam and an oval beam to form a UC which is repeated periodically to form a meta-material structure. Steel has been used as the constitutive material for the meta-material. On carrying out size optimization, it has been reported that the meta-material closely matches the elastomer behavior in compression.

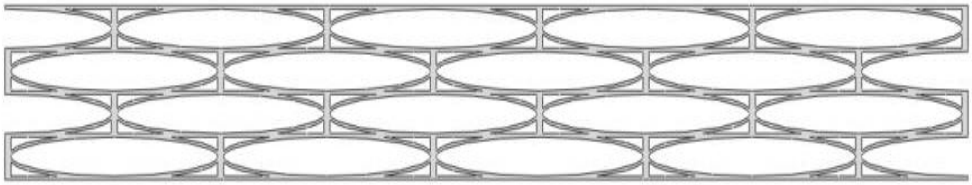

Figure 1.11 BrickOval Metamaterial Design Using Engineering Principles [16]

Based on the steps taken to design the meta-material in [16], a systematic process is abstracted into a design framework to help designers design meta-materials from a UC level to match a targeted nonlinear response. This method is named the Unit Cell Synthesis

Method since it involves combining elemental components with simple geometries that display geometric nonlinearity under deformation. The method is discussed in detail in Chapter 2.

Even though the meta-material structure design described in [16] is able to successfully match the nonlinear compression stress-strain curve of the elastomer, the stresses developed in the structure exceed the yield strength of the constitutive material, Steel, by 400%. Thus, no feasible design is obtained till now that can help validate the aforementioned method. Therefore, the objective of this research is to design a meta-material using the Unit Cell Synthesis Method which can prove feasible for the tank track pad application and in doing so, validate the method as well.

1.5. Research Objectives

The thesis has three primary research objectives:

1) To determine the shortcomings of the current Unit Cell Synthesis Method and, if any, to propose modifications to improve it.

2) To design and develop a feasible meta-material using the Unit Cell Synthesis Method to match the uniaxial nonlinear compression behavior of the current elastomeric backer pads and thereby, validate the method.

3) To validate the performance of the meta-material design obtained by using the Unit Cell Synthesis Method by subjecting it to an application-specific scenario; in this case, the interaction of the backer pad in the track pad assembly with the rolling road-wheel.

1.6. Thesis Organization

This thesis is organized into five chapters. The current chapter has presented the motivation, literature review and research objectives of this work.

Chapter 2 explains the Unit Cell Synthesis Method as described in [16]. A shortcoming is identified in the current method and a modification is proposed to help the designer achieve feasible designs efficiently.

Chapter 3 deals with the steps undertaken to design meta-materials using the modified Unit Cell Synthesis Method to match the compressive behavior of the elastomer which is currently used on the track pad. Two meta-material designs are evaluated against multiple feasibility criteria pertaining to the application.

Chapter 4 presents a dynamic finite element analysis set-up and results of a dynamic wheel roll-over event on the track pad assembly which is performed to validate the performance of the meta-material design obtained in Chapter 3. The objective is to compare dynamic deformation behavior of the meta-material pad acting under a rolling tank road wheel with that of the existing elastomeric pad.

Chapter 5 discusses the conclusions and the scope for future development of the Unit Cell Synthesis Method as well as the meta-material developed for the tank track pad application.

CHAPTER 2. THE UNIT CELL SYNTHESIS METHOD

2.1. Method Introduction

A unit cell (UC) is the smallest repeatable structure and the basic building block of a meta-material [17]. Fazelpour in [18], has reviewed different design methodologies employed in the literature to design meta-materials. He has reported that computational methods including topology optimization, parametric optimization, and synthesis methods are among the most popular methods to design meta-materials. However, there is a gap in systematic design methods for developing new meta-material architectures especially from the UC level as the current methods are limited to either topology optimization or selection and size optimization of predetermined UC geometries.

In view of this research gap and the limitations of topology optimization to design a meta-material for the track pad application as identified in [16], a meta-material design method, called the "Unit Cell Synthesis Method", is developed in [16]. It is aimed at helping the designer develop meta-materials that match targeted nonlinear uniaxial loading curves by combining geometries at the UC level.

The basic principle of this method is to achieve an overall nonlinear deformation behavior of the meta-material by using a combination of "geometric nonlinearities" associated with different elemental geometries. The nonlinear deformation characteristics of these elemental geometries are determined from the nonlinear mechanics of simple geometric entities. The UC is constructed by selecting elemental geometries by comparing their nonlinear deformation characteristics with the target nonlinear deformation response. Finally, size optimization is carried out to obtain the optimized meta-material.

It should be noted that this method is currently restricted to 2D planar UC designs extruded in the out-of-plane direction. The UCs are symmetric about at least one axis. 3D lattice structures are not considered in this work.

2.2. Method Description

The method introduced in [16] is defined in a series of systematic steps as shown in Figure 2.1. Each step is discussed in detail in this section as explained in [16].

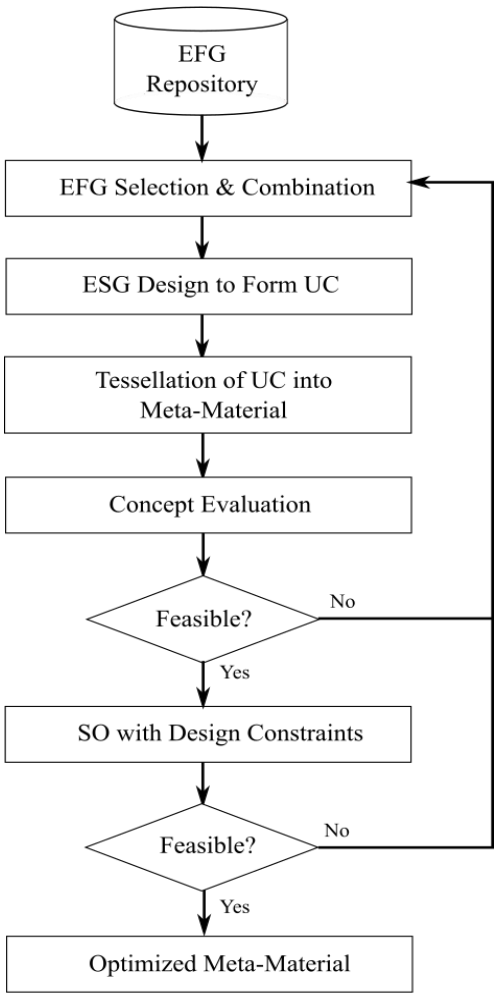

Figure 2.1 Schematic of the Unit Cell Synthesis Method [16]

2.2.1. Step 1: Preparation of EFG Repository

An *Elemental Functional Geometry* (EFG) is an integral part of the UC design and is defined as a geometry whose deformation characteristic is used to match the target nonlinear response [16]. In the Unit Cell Synthesis Method, the first step involves preparation of a repository of geometric elements whose deformation behavior and parameter sensitivities are pre-determined using nonlinear finite element analyses. Entities such as fixed-fixed beams (FFB), cantilever beams (CB) and oval beams (OB) are some examples of EFGs. The general deformation behavior of these EFGs subjected to a concentrated load and undergoing large deformation is shown in Figure 2.2. As shown in the figure, three entities show stiffening behavior with varying rates of stiffening as the applied load increases, whereas the oval EFG subjected to a pushing load has a complementary reciprocal behavior as it softens with the increase in applied load.

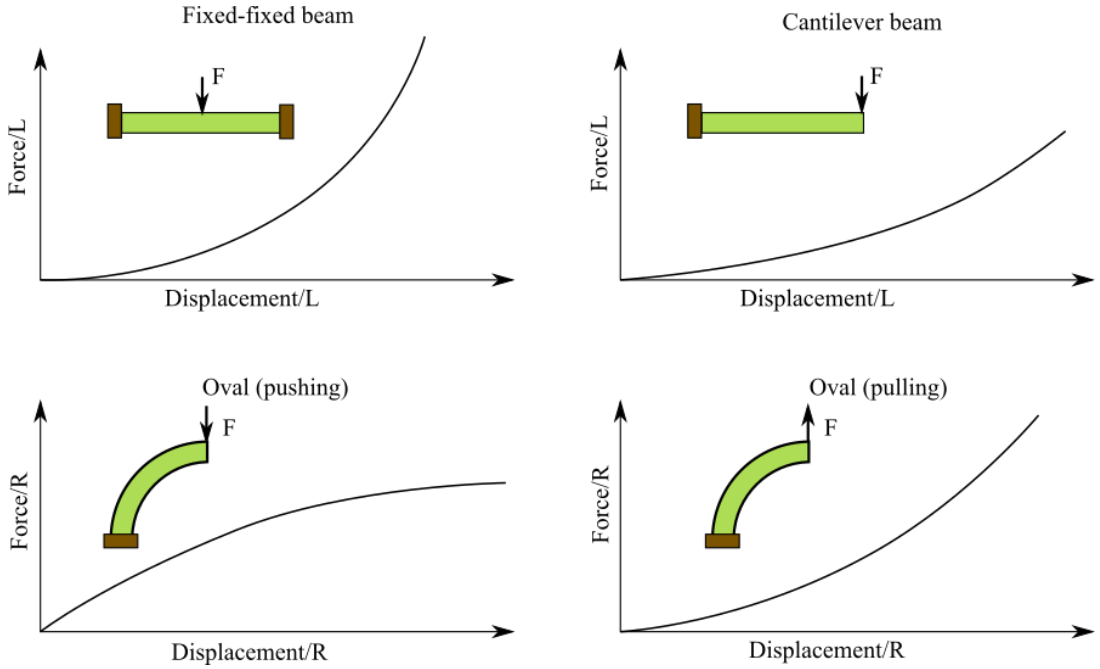

Figure 2.2 EFGs and their general behavior (zeroth order configuration) [16]

A parameter sensitivity study has also been carried out for each EFG to determine the effect of each parameter on the nonlinear deformation characteristics. The results are shown in Figure 2.3. The FFBs and the CBs have two important design parameters (L, h) whereas the OBs have three (R_1, R_2, t). The sensitivity study is carried out by varying the aspect ratio i.e. the ratio of the length to the height in case of the FFB and CB and the ratio of the radii in the case of the OBs and also the thickness in the case of the OBs. As the aspect ratio of the beams increases, the degree of nonlinearity increases with increase in applied loads.

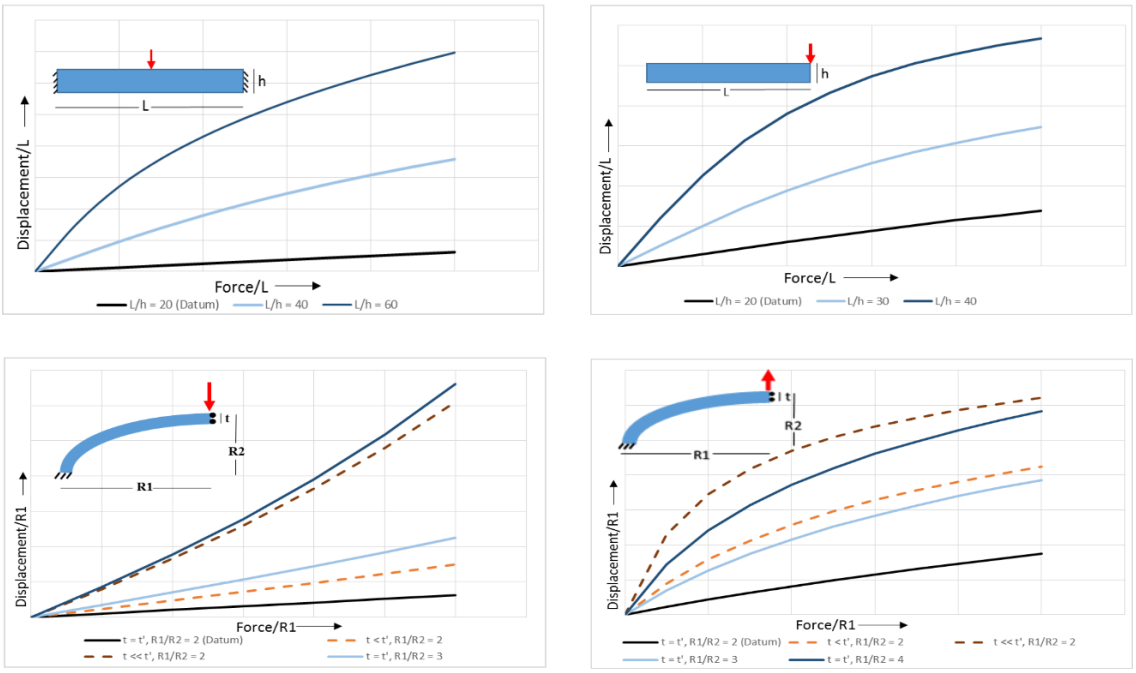

Figure 2.3 Design Sensitivities of EFGs

Thus, a good understanding of these different and complimentary deformation behaviors of the EFGs helps construct a UC based meta-material that can be tuned to match

the target behavior by combining the stiffness of EFGs in series, parallel or combined series-parallel configurations, similar to a spring system.

2.2.2. Step 2: EFG Selection and Combination

The design process is initiated by selecting the most suitable EFG or a combination of EFGs whose associated geometric nonlinearity can match the target response. The EFGs can be combined in several ways to achieve different effective nonlinear behavior of the meta-material. The combinations are categorized into three configurations namely zeroth order, first order and second order as defined in Table 2.1.

Table 2.1 Connection Configurations in a UC [16]

Connection Configuration	Description
Zeroth order	Single EFG (Figure 2.2)
First order	Combination of two Zeroth order configurations (Parallel or Series) (Figure 2.4)
Second order	Combination of two First order or a Zeroth order and First order configuration (Parallel or Series) (Figure 2.5)

Denoting the nonlinear stiffness functions of two zeroth order EFGs as $k_1(u)$ and $k_2(u)$ where u is the displacement, the effective stiffness for first order parallel and series connection combination of the EFGs can be determined in theory as shown below [16]:

$$k_{eff,p} = k_1(u) + k_2(u) \quad \textbf{Eq. 2.1}$$

$$k_{eff,s} = \frac{1}{\frac{1}{k_1(u)} + \frac{1}{k_2(u)}} \quad \textbf{Eq. 2.2}$$

where $k_{eff,p}$ and $k_{eff,s}$ are the effective nonlinear stiffness functions in parallel and series configurations respectively for a first order system.

Similarly, connection configurations of higher orders can be constructed by following the pattern explained in Table 2.1. However, it is observed from the illustrations shown in Figure 2.4 and Figure 2.5 that the first few orders of connection configurations are able to generate a wide range of nonlinear deformation behavior of the resultant meta-material. Hence, configurations only up to the second order are considered in this work.

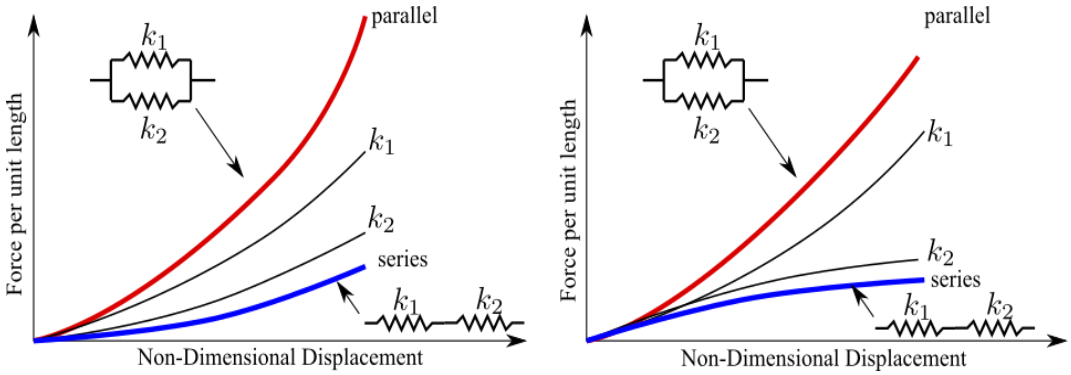

Figure 2.4 First Order Connection Configuration [16]

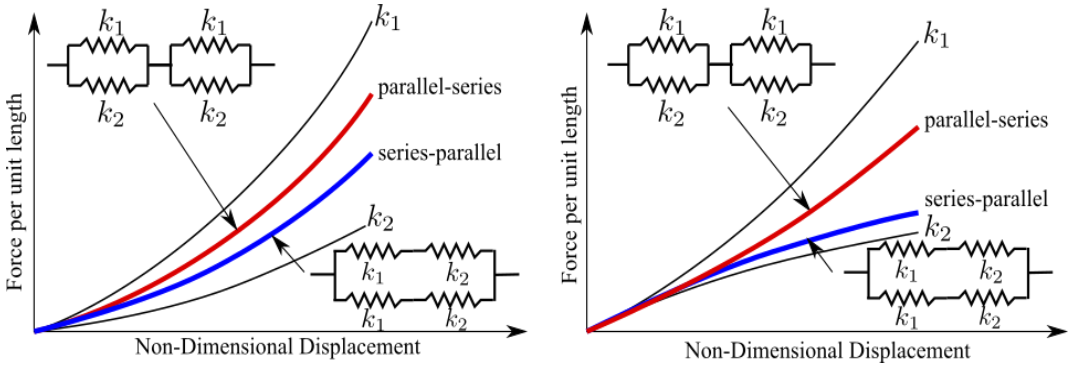

Figure 2.5 Second Order Connection Configuration [16]

2.2.3. Step 3: ESG Design to Form UC

Along with the EFGs, the other integral member needed to construct a UC is the *Elemental Structural Geometry* (ESG). The ESGs are the structural components in a UC that serve as the rigid connection to the EFGs and the adjacent UCs [16]. They are designed to have a much higher stiffness so that they do not interfere with the deformation of EFGs.

They may also be designed such that they help in completing the load path and transmit the loads from one UC to another. The next step after EFG selection, therefore, involves designing suitable ESGs to complete the preliminary UC geometry. The ESGs should adhere to the following requirements [16]:

1) *They must exhibit high stiffness and low deformation compared to the EFGs*
2) *They should complete the topology of the UC by connecting the EFGs between the UC.*

The first requirement helps to isolate the required deformation characteristics of the EFGs. The second requirement serves to complete the UC in order to allow tessellation into a meta-material while maintaining symmetry of the UC. The ESGs are intended to serve purely as structural entities that help shape the UC topology. Thus, it is not necessary to determine their deformation characteristics beforehand. However, they might have some associated design variables which may have to be considered in Step 6.

2.2.4. Step 4: Tessellation of UC into a Meta-Material

Once the preliminary UC geometry has been designed comprising of the EFGs and the ESGs, the meta-material structure is formed by tessellating (i.e. repeating periodically) the UC multiple times in the x- and y- directions. Since, the 2D meta-material geometry is to be extruded in the out-of-plane direction, no tessellation is carried out in the z- direction. Also, the tessellation may be carried out in a manner such that each UC undergoes similar deformation in the meta-material. One way of achieving this is to offset each alternate layer of UCs in the direction in which load is applied such that the load gets transmitted to all the layers efficiently. To reduce computational cost while carrying out finite element

analysis and optimization, a representative volume element (RVE) of the meta-material is constructed through tessellation. Two cases arise when considering the size of tessellation to form the RVE [16]:

1) Size of Meta-material >> Size of UC

In this scenario, a tessellation with a relatively large number of UCs is necessary for the performance analysis and optimization, especially when the boundary conditions are not exactly known and are only approximated. Figure 2.6 shows the selection of RVE from a meta-material and then its further decomposition into a UC. However, in the context of this method, the meta-material is constructed from a UC level with the RVE being the intermediate step. In order to ensure that the deformation behavior of the RVE accurately represents the behavior of the bulk meta-material, a convergence study is required.

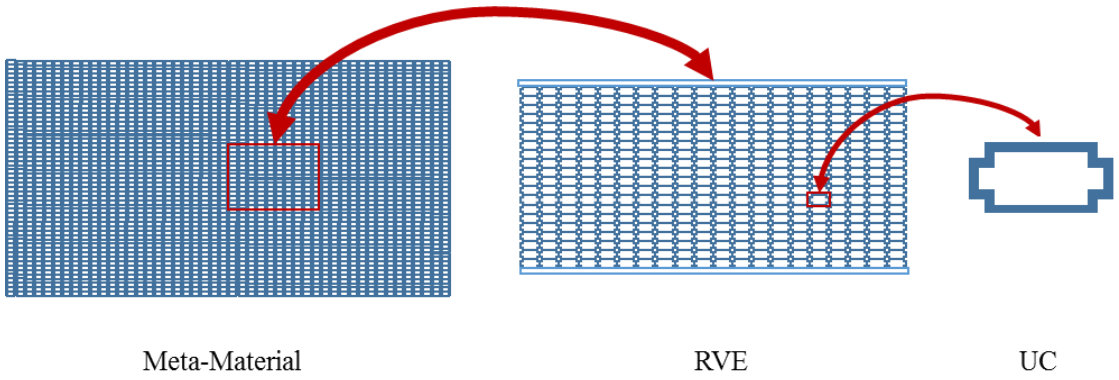

Figure 2.6 Decomposition of a Meta-Material into RVE and Tessellation of UC [16]

2) Size of Meta-material > Size of UC

This scenario usually arises for applications with a restrictive design space. In such cases, the dimension of the RVE can be determined by the size of the target structure, and this overall size becomes the driving factor which may limit the number of UCs in the tessellation. Thus, the tessellation size might be restricted in view of the manufacturing and

the overall size constraints. It should be noted that if few UCs exist in each direction, the effect of boundary conditions becomes more prominent. Hence, exact application specific boundary conditions should be applied for the resulting meta-material structure.

2.2.5. Step 5: Concept Evaluation

The deformation behavior of a nonlinear material can be defined completely by a material tensor which is comprised of multiple nonlinear components [19]. This nonlinear material tensor can be determined by the material's stress-strain responses under prescribed loading conditions and deformation modes. However, it is often observed that one or two deformation modes dominate the deformation of the target material in any given application. Therefore, it is assumed that in most cases, it is sufficient to consider only the dominant deformation mode(s) of the target material and its associated stress-strain response and develop a meta-material design that matches that response [16].

However, since the meta-material is intended to have a target deformation behavior which is different from that of its linearly elastic constitutive material, a means of evaluating the effective mechanical properties of the meta-material is to be determined. For a meta-material RVE comprising of a large number of UCs, the meta-material performance is evaluated based on the RVE's deformation response. Taking into the consideration the stress-strain curves associated with the target deformation behavior, finite element analyses are performed on the RVE to obtain its force-displacement behavior as shown in Figure 2.7. A "meta-strain" can then be defined as the percentage of the "bulk" uniaxial deformation (i.e. averaged displacement) of the RVE (meta-material) as [16]:

$$\text{meta-strain}=\%(\text{"bulk" deformation}) = \frac{\delta}{H}(100) \qquad \textbf{Eq. 2.3}$$

where δ is the displacement and H is the original height of the meta-material.

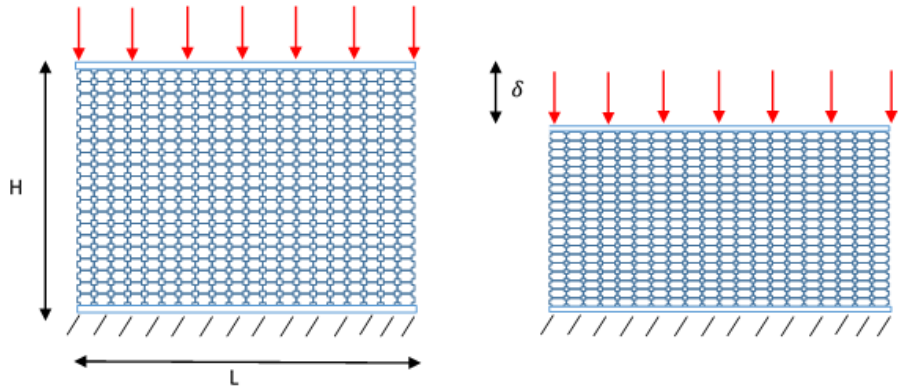

Figure 2.7 Meta-material with Uniaxial Loading (Left) and After Deformation (Right) [16]

Now, the meta-material is subjected to a series of static load cases corresponding to the range of the target stress-strain response curve(s). The meta-strain is calculated for each load case to determine the RVE deformation response which is compared to the target curve in order to perform the feasibility evaluation. For a meta-material with restrictive design space, the definition of the meta-strain remains the same. However, it should be noted that the meta-material RVE containing a relatively small number of UCs exhibits deformation behavior which is influenced by the effects of the material boundary. Thus, appropriate boundary conditions should be applied based on the actual boundary conditions of the target structure.

It is necessary to determine if the UC parameters can be tuned to match the nonlinearity of the target stress-strain response before moving on to the next step. This feasibility can be determined by carrying out a design of experiments or a sensitivity study. This is a necessary intermediate step between the development of the concept UC topology

and optimizing UC parameters to achieve the target deformation behavior of the meta-material in order to save computational time and resources.

If the concept UC based meta-material (RVE) shows the ability to match the nonlinearity of the desired material response during the concept evaluation stage, the concept UC is regarded as a "feasible" concept design. If not, a different EFG in the same configuration or a higher order connection configuration is selected and Steps 2-5 are repeated until the UC concept feasibility is obtained. It should be noted that a higher order EFG configuration usually leads to an increase in the design parameters associated with the UC topology which may impart more tuning ability to match the target behavior by augmenting the design space [16]. As shown in Figure 2.4 and Figure 2.5, there are multiple ways of combining the EFGs to achieve the desired deformation behavior. However, it is logical for the designer to start with the lowest possible connection configuration for the simplicity of the UC design and to save computational time and costs [16].

2.2.6. Step 6: Size Optimization (SO) with Design Constraints

A size optimization of the design parameters associated with the EFGs and ESGs in the UC is conducted once the UC design concept is deemed feasible. Size optimization is employed in [20] [21] [22] [22] for different applications and objectives. For this method, the objective of the optimization procedure is to achieve a meta-material design which has deformation behavior that matches the target material response. The optimization setup can be mathematically written as [16]:

$$\min_{f} : f = \sum_{i=1}^{N} \left(\varepsilon_i^t - \varepsilon_i^c \right)^2 \qquad \textbf{Eq. 2.4}$$

where ε_i^t and ε_i^c are the target strain and the meta-strain (i.e. the % deformation) of the meta-material RVE, respectively, at the i^{th} load case for a total of N load cases. The optimization algorithm should be implemented taking into consideration its convergence properties and the ability to explore the design space efficiently. The selection of gradient based or evolutionary algorithms can be made based on the need to explore local or global design space respectively. Once the optimization is complete, the deformation response of the resulting meta-material (RVE) should have converged with that of the target response. The acceptance of the results obtained from the optimization step depend on their evaluation against the application-specific design criteria and constraints.

After the optimization run is complete, the optimal design points are evaluated against the application specific design constraints/criteria to rule out an application-specific infeasible meta-material design. Some examples of design constraints include manufacturing limitations and maximum stress allowance, and the requirement of non-contact of the structure within the targeted deformation limits [16]. If an optimal meta-material design is obtained that meets all the feasibility criteria, a feasible solution exists for the given application. If the meta-material design is found to be infeasible, the starting design points for the optimization are changed and another SO is carried out with the new starting conditions until the target deformation behavior is achieved and the feasibility criteria are met. However, if the SO iterations are not able to generate a feasible optimal design, then the design process goes back to Step 2 and a different EFG configuration of the same or a higher order is selected and Steps 2-6 are repeated for the new UC concept [16].

2.3. Drawbacks of the Current Method

The current Unit Cell Synthesis Method is based on the steps that were undertaken to design a meta-material that mimics the nonlinear compressive behavior of the M1 Abrams tank track pad [16]. Figure 2.8 shows the progression of the BrickOval UC based meta-material which was developed in [16].

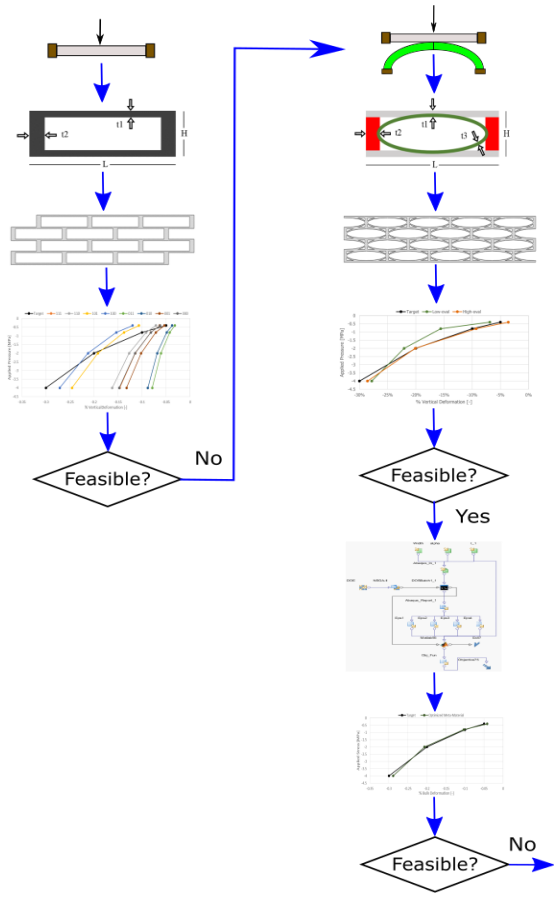

Figure 2.8 Progression of BrickOval UC Meta-material from [16]

The initial UC design consisting of just the Fixed-Fixed beam (zeroth order) was proven infeasible in the concept evaluation step (Step 5). Therefore, an Oval beam was added in parallel with the Fixed-Fixed beam (first order configuration) in accordance with

the method, which was found to be a feasible UC concept. On carrying out size optimization, it was observed that the resulting meta-material showed nonlinear deformation behavior that closely matched the target response. However, the maximum stresses developed in the meta-material structure exceeded the yield strength of its constitutive material, Steel, by 400%. Thus, physical implementation of the meta-material as a replacement for the track pad was not possible.

An infeasible final design may have been obtained due to two reasons. First, it is probable that that the first order combination of FFB and OB as considered in the UC design may not be sufficient to match the target curve without exceeding the yield limit of the constitutive material. This issue can only be addressed by looking at higher order configurations or considering different EFG combinations altogether.

Second, the concept of performing a single objective size optimization to match the target response curve and subsequently checking feasibility of the design for stress and manufacturing criteria might not be the most efficient way of exploring the design space. With the task of minimizing (or maximizing) just one objective, the mathematical optimizer does not get any feedback about other criteria which may be equally important. Also, it involves the designer's intervention to determine if an overall feasible design solution has been obtained. Hence, it is necessary to formulate the optimization problem such that all the necessary feasibility criteria are accounted for while carrying out the optimization. The determination of final feasibility of the meta-material design would then become a more manageable task by examining the design points and using engineering judgement to decide which design is to be selected.

2.4. Proposed New Method

In view of the shortcomings of the current method, a modification is proposed to the existing method. Figure 2.9 shows the proposed new method. Step 6 now consists of a multi-objective optimization set-up. All the application specific requirements such as strain deviation from the target response, maximum internal stress, weight etc. can be formulated as individual objective functions.

It is possible to lump all the different objective functions into a single objective function with predetermined weights for each function. However, such a single objective optimization problem will only be useful to help designers gain insight into the nature of the problem. It may not provide a set of alternative solutions that trade different objectives against each other [23]. Similarly, if one objective function is chosen and the other objectives are incorporated as constraints, the designer risks limiting the design choices available [24]. Also, the entire design space may not be explored efficiently.

However, in a multi-objective optimization with conflicting objectives, there is no single optimal solution. Due to the interaction between different objective functions, a multi-objective optimization set-up leads to a set of compromised solutions, commonly known as the trade-off or Pareto-optimal solutions [23].

Further, constraints may be incorporated for each objective function in order to help the optimization algorithm explore the feasible design space efficiently. They also serve as a feasibility check such that if the optimizer is able to generate sufficient design points that are easily satisfying the said constraints and further improving the objective function values as desired, the meta-material design may be deemed feasible for the specific application.

Also, once the optimization process has been completed and sufficient feasible points have been obtained, the designer can choose a design solution from among the Pareto points that best suits the target application.

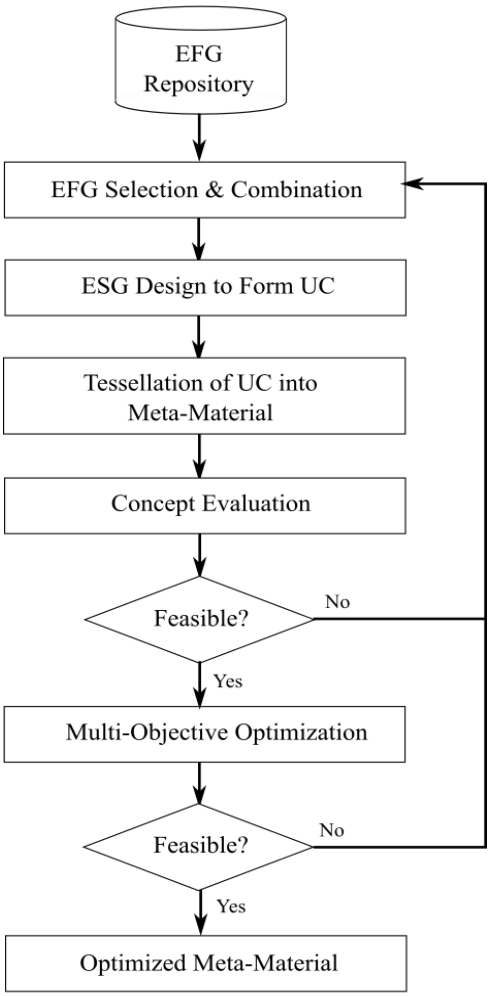

Figure 2.9 Schematic of the New Unit Cell Synthesis Method

The proposed multi-objective optimization set-up is one step towards automating the entire meta-material design process. In the next chapter, the modified Unit Cell Synthesis Method is used to design a new meta-material to replace the existing elastomeric track pads.

CHAPTER 3. META-MATERIAL DEVELOPMENT USING THE UNIT CELL SYNTHESIS METHOD

3.1. Meta-material Design Requirements

In order to design a meta-material using the Unit Cell Synthesis Method to replace the existing elastomeric backer pads in the tank tracks, it is essential to determine the design requirements. Two important requirements are identified for the meta-material. Firstly, the overall dimensions of the meta-material structure should be known beforehand. Secondly, the target response behavior of the meta-material should be determined which will be used to mimic the behavior of elastomer in the current backer pads.

Figure 3.1 shows the overall dimensions of a single track pad link assembly. The dimensions of the backer pad can be extracted from the figure. The length of the meta-material has to lie between 0.130-0.152 m while the height can vary between 0.020-0.025 m. The meta-material geometry is extruded in the out-of-plane direction and the depth of the pad is 0.170 m [3].

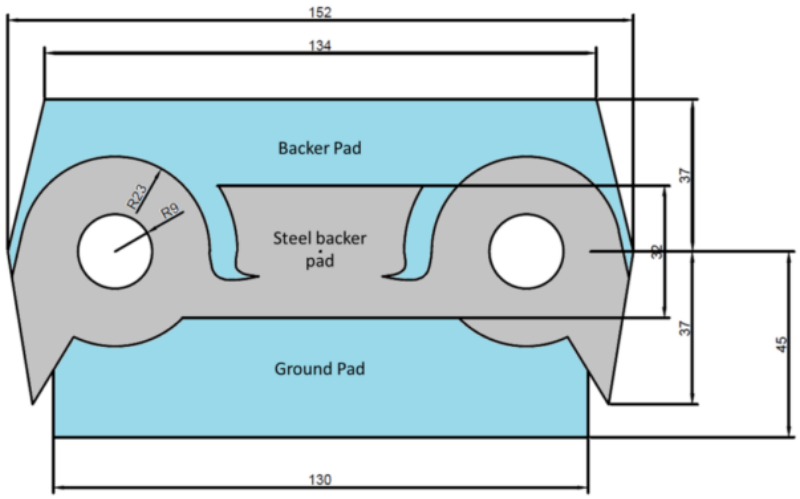

Figure 3.1 Track Link Dimensions [3]

Since the predominant mode of deformation of the backer pad is compression, the design objective of the meta-material is to achieve a nonlinear behavior under compression similar to the elastomer in use, as shown in Figure 3.2.

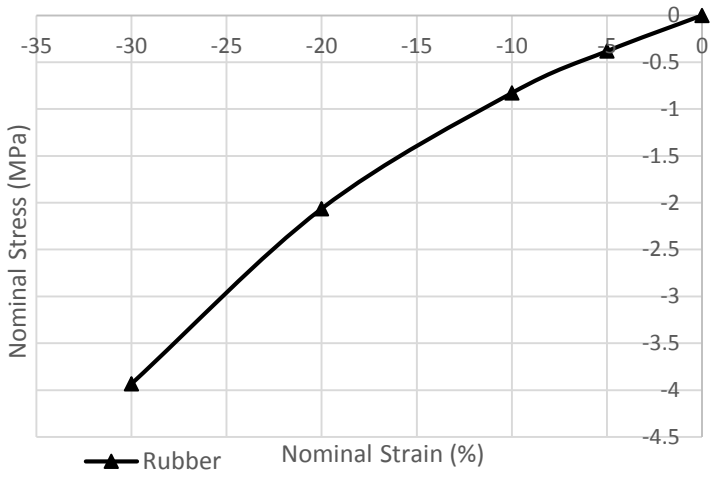

Figure 3.2 Target Compression Response for the Meta-material [16]

The compression plot of the elastomer on the current backer pads has been obtained by fitting a 2nd order Ogden Hyper-elastic polynomial to the experimental data which has been presented in [12]. It has been determined in [3] that the backer pad experiences a maximum compressive strain of 20% as the tank wheels pass over it. Table 3.1 [16] shows the target strain values for the meta-material when subjected to compression which are obtained from Figure 3.2. Each pressure value corresponding to the four strain levels will act as a separate load-case for analyzing the meta-material deformation response. Note that even though the meta-material is to be designed to have a maximum compressive strain of 20%, the inclusion of the fourth load-case corresponding to a strain of 30% ensures that the meta-material behavior will closely resemble the overall behavior of the elastomer even beyond the targeted range.

Table 3.1 Meta-Material Target Property Values

Applied Pressure [MPa]	% Meta-Strain
-0.3817	05.0
-0.8284	10.0
-2.0632	20.0
-3.9327	30.0

After understanding these requirements, the meta-material design can now be initiated by following the steps prescribed in the Unit Cell Synthesis Method.

3.2. Canti-Duo UC Based Meta-material Design

The first step in the method, which involves the preparation of the EFG repository is not discussed again since the repository is already created as explained in section 2.2.1.

3.2.1. Step 2: EFG Selection & Combination

Since the elastomer in the backer pad shows a stiffening effect with increase in load, an EFG is sought which shows a similar deformation behavior. A fixed-fixed beam (FFB) has been used as a base geometry to construct two UC concepts in [16]. However, both designs were found to be infeasible for the meta-material application as they either could not match the target response or exceeded the yield strength of the constitutive material while deforming. Therefore, a cantilever beam (CB) is chosen for the advantages it offers over the fixed-fixed beam. It is known from Figure 2.2 and Figure 2.3 that a CB shows increased stiffening behavior as its aspect ratio increases. Further, it allows for a relaxed boundary condition at its tip (free end) which helps in achieving a larger deformation with a smaller elastic strain in the beam structure.

At this point, a material selection study is carried out to determine the optimal elastic constitutive material for the meta-material. It was explained in [16] that a material with the lowest ratio of Young's Modulus to Yield Strength ($E:\sigma_y$) is the most suitable material for the tank track pad application. The Young's Modulus, which is a measure of the stiffness of the material determines the amount of deformation a material will undergo when subjected to a force. The Yield Strength of a material is the amount of stress that can be developed in the material before it experiences plastic deformation. Thus, a low ratio of ($E:\sigma_y$) indicates a higher ability of the material to undergo large deformation before yielding.

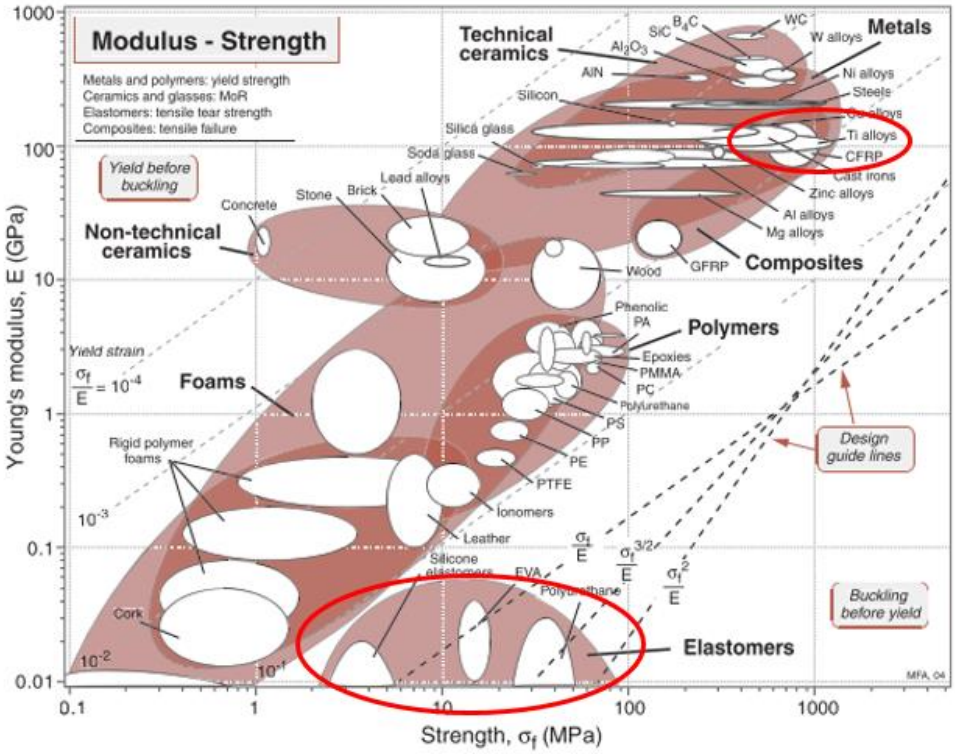

Figure 3.3 Ashby Chart of Young's Modulus v/s Strength for Materials [14]

In the Ashby chart shown in Figure 3.3 comparing these two properties for different materials, it can be seen that elastomers have a very low ratio in the range of 1:1 – 10:1 [16]. However, the meta-material has to be made out of a linearly elastic material and titanium alloys have the least value of the ratio among metals.

Hence, a grade of titanium alloy ($E: \sigma_y = 92.5: 1$) with material properties as shown in Table 3.2 is identified in [16] as an ideal constitutive elastic material for the track pad meta-material.

Table 3.2 Properties of Meta-material's Constitutive Material

Titanium Alloy	Ti 3Al-8V-6Cr-4Mo-4Zr-0.05Pd
Young's Modulus (E)	102 GPa
Yield Strength (σ_y)	1103 MPa
Poisson Ratio (ν)	0.32

3.2.2. Step 3: ESG Design to Form UC

Once the EFG has been selected, the next step involves designing ESGs to complete the UC topology. Figure 3.4 shows the completed conceptual UC topology along with the associated design variables. The ESGs have been designed by adhering to the requirements explained in section 2.2.3. The UC consists of 5 independent design variables – *H, W,* t_2, t_3, and g. The variable 't_1' is dependent on the design variables 'g' and 't_3'. The UC is symmetric about the central axis and its overall dimension is *H* x *2W*. The gap between the beam tips introduced by the variable 'g' gives a relaxed boundary condition to the half-beams, allowing them to deflect like cantilever beams. This UC concept has been termed Canti-Duo as it contains two CBs.

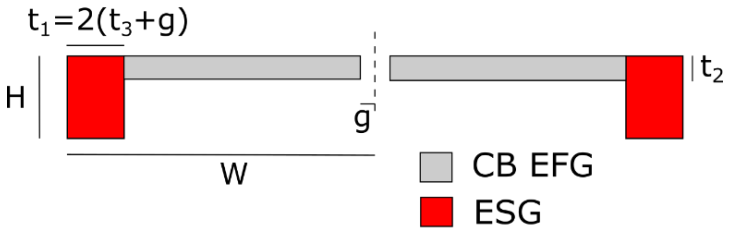

Figure 3.4 Canti-Duo UC Design with Design Variables

Note that due to symmetry requirements and the need to allow for relaxed boundary conditions for the CBs, the current UC configuration acts as first order connection system since both the half beams act in parallel. Thus, even though the CB is a zeroth order EFG, its inclusion in a UC leads to a first order configuration. Figure 3.5 shows the first order spring system that is equivalent to the current UC configuration. The nonlinear stiffness function of each CB has been denoted by the symbol 'K'.

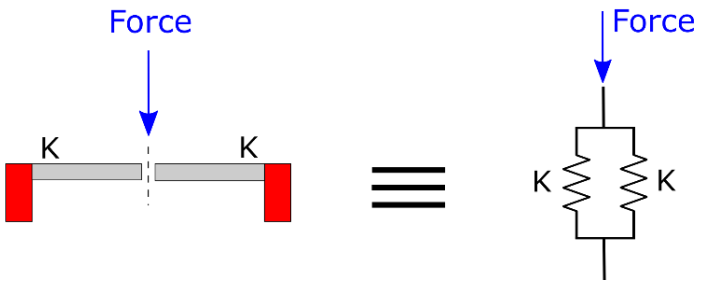

Figure 3.5 Equivalent First Order Spring System of Canti-Duo UC

The variable 't_3' denotes the part of the ESG that is placed directly on top of the CB tips during tessellation and it can be easily visualized once the UCs are assembled to form the meta-material structure.

3.2.3. Step 4: Tessellation of UC into a Meta-material

Since each UC in the meta-material structure is required to undergo similar deformation when acting under a compressive load, UC tessellation is carried out in a

manner shown in Figure 3.6. The UC in the upper layer is offset by half the UC width such that the ESG of the top UC imparts the necessary boundary conditions to its EFGs and at the same time transmits force down to the EFGs of the UCs beneath. As mentioned before, the overlap of the top ESGs on the CB tips underneath them has been represented by the variable 't_3'. Thus, complete tessellation of the UCs to form the meta-material structure is carried out in this manner such that each alternate UC layer is offset from the one beneath it.

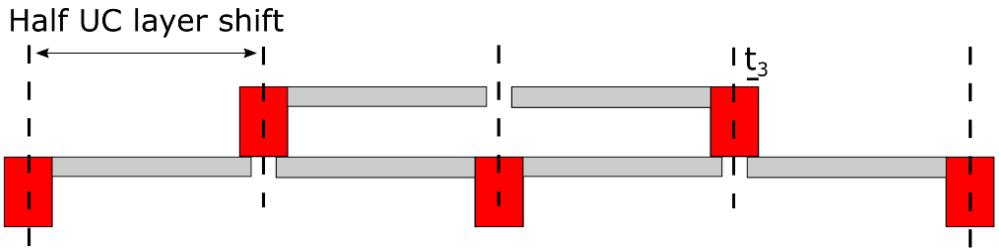

Figure 3.6 Tessellation of UC to Form Meta-material

As explained in section 2.2.4, the number of tessellations required to construct a RVE for the meta-material may be restricted by the over-all size constraints imposed by the application. In this case, the total length of the meta-material has to lie between 0.130-0.152 m whereas the total height has to be between 0.020-0.025 m. The UC dimensions can be made small enough to accommodate several UCs in the over-all design space. However, manufacturing constraints and feasibility need to be taken into consideration while determining the minimum allowable design variable dimensions that may be accurately manufactured. In view of these factors, it has been decided to restrict the number of UCs in the x- direction (n_{xdir}) to three and that in the y- direction (n_{ydir}) to six. The resulting RVE and the meta-material structure become one and the same in this case, as is shown in Figure 3.7.

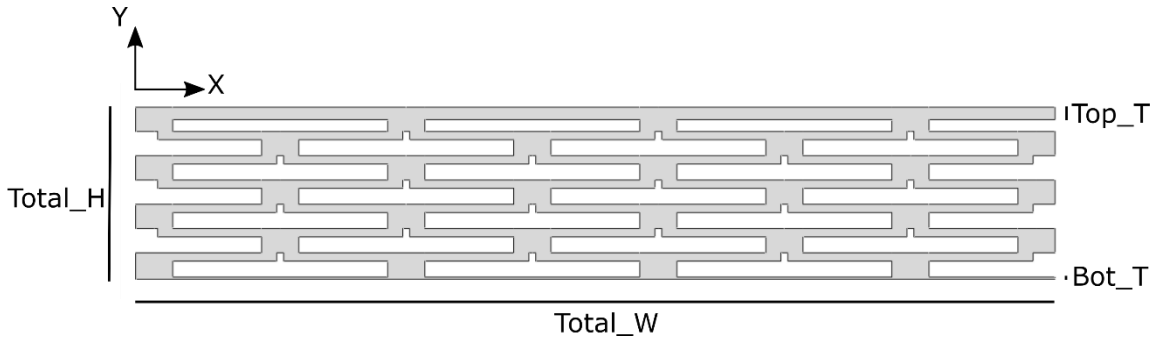

Figure 3.7 Tessellated Canti-Duo Meta-material with Over-all Dimensions

Once the tessellation has been carried out, top and bottom face sheets with thicknesses '*Top_T*' and '*Bot_T*' respectively, are added to the meta-material structure (Figure 3.7) to account for uniform application of loads and boundary conditions. The top face sheet has to be thick enough to transmit the loads efficiently to the UC layers below it without undergoing significant self-deformation. The bottom face sheet, however, can have negligible thickness as it neither contributes to the deformation behavior of the meta-material nor undergoes any self-deformation. Its only function is to enclose the bottom boundary of the meta-material structure to allow a uniform base.

At this point, the total height and width of the meta-material structure denoted by *Total_H* and *Total_W* respectively, can be calculated as shown below:

$$Total\_H = 2 \times H \times n_{ydir} + H + Bot\_T \quad \textbf{Eq. 3.1}$$

$$Total\_W = 2 \times W \times n_{xdir} - \left(n_{xdir} - \frac{1}{2}\right) \times t_1 + W \quad \textbf{Eq. 3.2}$$

Note that, the out-of-plane extruded depth of the meta-material is fixed at 0.170 m as mentioned before.

3.2.4. Step 5: Concept Evaluation

Since the preliminary meta-material concept design is now ready, the next step is to evaluate the design to determine if it is feasible in terms of matching the target nonlinear response curve. In order to determine feasibility, a full factorial study is carried out by performing static nonlinear finite element analysis on the meta-material structure. Since, the meta-material is thicker in the z- direction as compared to the x- and y- directions, a 2D plane strain formulation is adopted for ease of computation [25]. Figure 3.8 shows the load and boundary conditions applied to the meta-material. Pressure is applied on the top face of the structure with values corresponding to those shown in Table 3.1. The left and right edges of the meta-material are only allowed to translate vertically whereas the bottom face is fixed with all the degrees of freedom constrained.

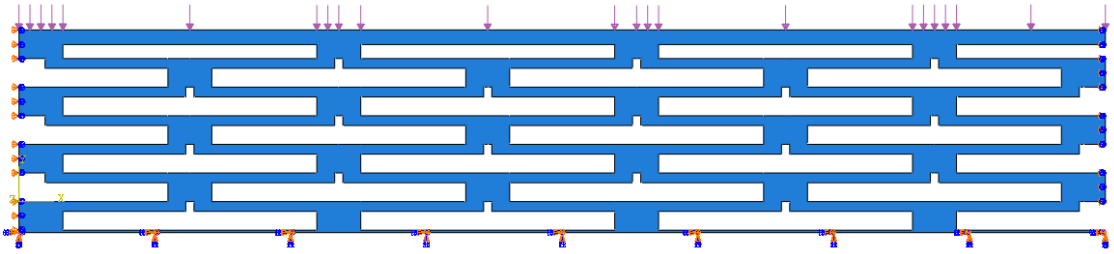

Figure 3.8 Load and Boundary Condition Application on the Canti-Duo Meta-material

The vertical displacement (δ_y) at the top center of the meta-material is calculated for each load-case and the resulting vertical deformation is expressed in terms of meta-strain as shown below:

$$meta-strain\ (\%\ Vertical\ Deformation) = \frac{\delta_y}{Total\_H} \times 100 \qquad \textbf{Eq. 3.3}$$

In order to carry out a full factorial study, the design variables that directly affect the deformation behavior of the meta-material are identified. It is known from the

sensitivity study shown in Figure 2.3 that the degree of nonlinear behavior obtained from a CB undergoing large deformations can be controlled by varying the aspect ratio of the beam. It can be seen from Figure 3.4 that variables '*W*' and 't_2' are directly related to the aspect ratio of the CBs. Also, the variable '*H*' is directly related to the total height of the meta-material which plays a crucial role in calculating the meta-strain as shown in Eq. 3.1 and Eq. 3.3. Hence, only these three variables are considered to carry out the sensitivity study.

For a full factorial study experiment, if the combinations of k factors are investigated at two levels, a factorial design will consist of 2^k experiments [26]. In this case, the study consists of $2^3 = 8$ study experiments. Low and high level values for the three design variables considered for the study are shown in Table 3.3. The values are chosen such that the overall dimensions of the meta-material are within the design space requirements. Other variables that do not contribute directly to the deformation response of the concept meta-material are held constant and are shown in Table 3.4.

Table 3.3 Full Factorial Study Parameters for Canti-Duo UC Concept

Design Variable	Low Level (0) (m)	High Level (1) (m)
W	0.0200	0.0220
H	0.0030	0.0036
t_2	0.0010	0.0020

Table 3.4 Constant Design Variables Values for Canti-Duo Concept Evaluation

Design Variable	Value (m)
g	0.0005
t_3	0.0015
Top_T	0.0017
Bot_T	0.0003

The results of the full factorial study experiment are shown in Figure 3.9. The values indicated in the legend of the plot denote the levels for *W*, *H* and t_2 respectively. It can be inferred by looking at the plots that the concept meta-material shows the ability to match the nonlinearity of the target response curve at higher values of aspect ratio of the CBs. For instance, the curve which denotes the configuration '100' indicates that the variable '*W*' is at high level and the variable 't_2' is at a low level. This directly culminates in the CBs within the UC that have a higher aspect ratio value.

The full-factorial study, of course, does not directly yield a design solution that closely matches the target curve. However, the results obtained from this step deem the concept Canti-Duo UC feasible. The next step involving multi-objective optimization can now be carried out to obtain an optimized meta-material design.

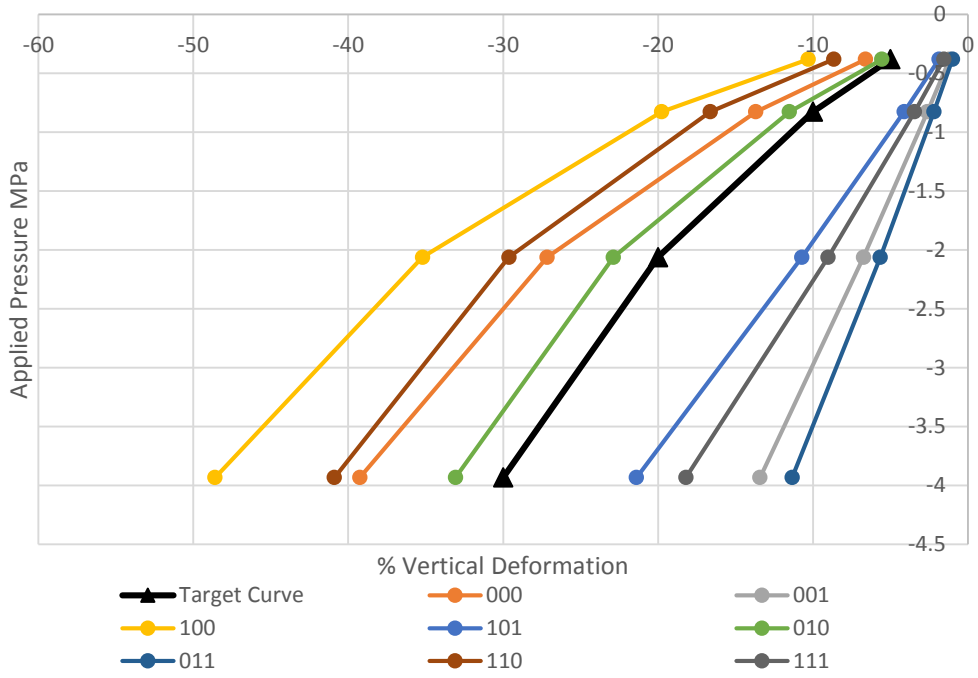

Figure 3.9 Full Factorial Study for Concept Evaluation of Canti-Duo Meta-material

3.2.5. Step 6: Multi-objective Optimization

Two important factors have been identified for the tank track pad application that will lead to the two objective functions to be employed in the optimization set-up. Firstly, the meta-material should closely match the target nonlinear response. Secondly, the maximum stress developed within the meta-material structure when deforming under compression should be well below the yield stress of its constitutive material. Figure 3.10 shows the Von-Mises stress contour plot of a generic Canti-Duo meta-material which is deformed under compressive loading.

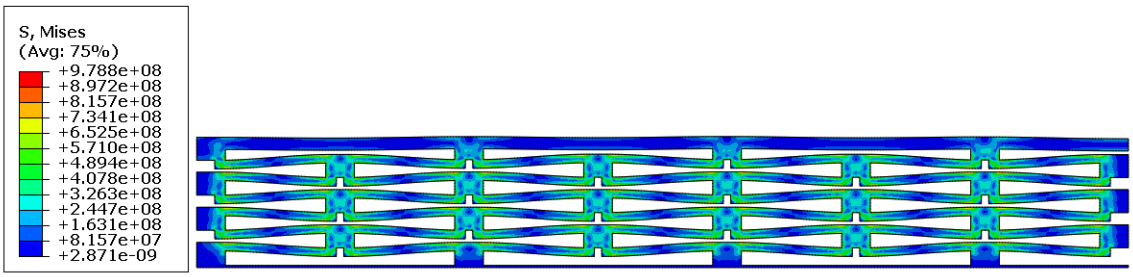

Figure 3.10 Stress Contour Plot of Canti-Duo Meta-material Under Deformation

The optimization procedure is carried out in a commercial optimizer modeFRONTIER 4.4.2. Figure 3.11 shows the work-flow set-up for the optimization process. The steps carried out in formulating the optimization problem are explained below in detail.

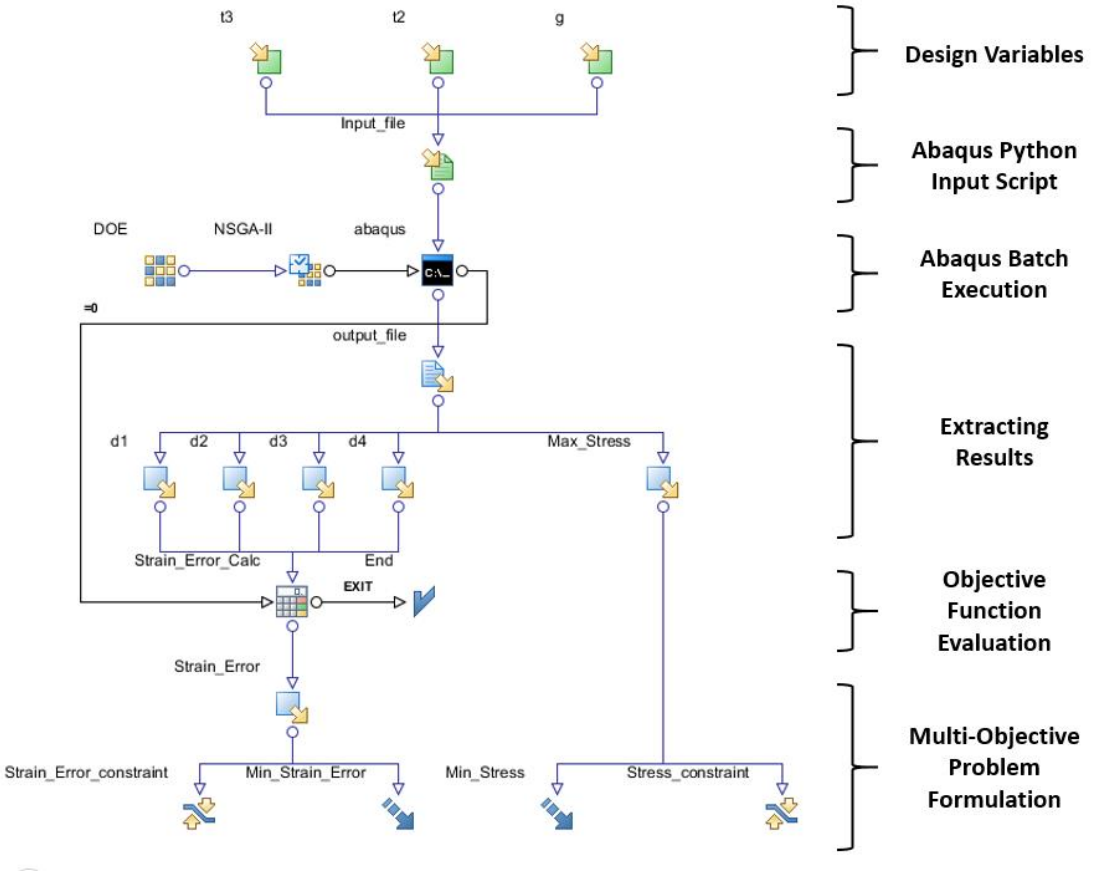

Figure 3.11 Optimization Set-up for Canti-Duo Meta-material

In order to carry out size optimization, the design variables that have to be included in the optimization are determined. Since, the overall dimensions of the meta-material are already known, the variables '*H*' and '*W*' can be held constant since they directly determine the total height and the total width of the meta-material as shown in Eq. 3.1 and Eq. 3.2. Keeping the variable '*H*' constant also imparts some control in making sure that there is no contact between the UC layers when the meta-material undergoes 20% vertical deformation. This is possible as only the CB EFG thickness which is defined by the variable 't_2' is varied during the optimization process. Table 3.5 shows the design variable values which are kept constant for the optimization run.

Table 3.5 Constant Design Variable Values for Canti-Duo Optimization

Constant Design Variable	Value (m)
H	0.0032
W	0.0205
Top_T	0.0017
Bot_T	0.0003

Thus, the total height (*Total_H*) of the meta-material is calculated to be 0.0227 m. The total width (*Total_W*) of the meta-material, however, also depends on the variables 't_3' and '*g*' as shown in Eq. 3.2. Since the variable '*W'* has been kept constant, once the upper and lower limits for these variables are fixed, the limits of the total width of the meta-material can be determined.

Hence, only the variables t_2, t_3 and *g* are considered for the size optimization process. With the variable '*W*' as a constant, a change in the value of the variable 't_2' changes the aspect ratio of the CBs in the UC. Also, it is observed from preliminary analyses that variables 't_3' and '*g*' which form the ESGs are directly related to the maximum stress developed in the meta-material structure. The lower and upper bound values considered for these three variables are shown in Table 3.6.

Table 3.6 Limits of Design Variables for Canti-Duo Optimization

Design Variable	Lower Bound (m)	Upper Bound (m)
g	0.0001	0.0006
t_2	0.0010	0.0018
t_3	0.0010	0.0030

While the lower bound value of the CB EFG thickness 't_2' is determined keeping in mind the manufacturing constraints, its upper limit is determined after considering the requirement of non-contact during deformation at 20% meta-strain. This requirement can

be satisfied if the total available vertical gap between all the UCs along the y- direction in the meta-material is greater than the allowable deformation of the meta-material experienced during 20% vertical deformation. It is expressed mathematically in Eq. 3.4 as shown below:

$$n_{ydir} \times (H - t_2) \geq 20\% \times Total\_H + n_{ydir} \times \delta \qquad \textbf{Eq. 3.4}$$

where δ is the clearance between each UC once the EFG in it is deformed to account for 20% overall vertical deformation of the meta-material. Since the values for n_{ydir}, H and *Total_H* are known beforehand, and taking δ = 0.0006 m, the upper limit for 't_2' can be found to be 0.0018 m.

As shown in Figure 3.11, the design variables are linked to the Abaqus python input script which is attached in Appendix A. The Abaqus python script has been written such that it constructs the UC geometry based on the design variable values, carries out the pre-processing, runs the analyses for four load-cases and extracts the required results in a report file. It is executed in batch mode and the design variables are changed by the optimization algorithm for every run.

NSGA-II or the Non-dominated Sorting Genetic Algorithm II is chosen as the optimization algorithm for this particular problem. A Genetic Algorithm (GA) is an evolutionary algorithm that is based on a biological systems' improved fitness through evolution. It can efficiently search the global design space as compared to a gradient based algorithm. Usually for a GA, a large population size and a large number of generations increase the likelihood of obtaining a global optimum solution, but substantially increases

processing time [27]. Hence, it is important to decide on the optimum number of initial population size and generations.

The initial design of experiments (DOE) are generated using Uniform Latin Hypercube samplings. This ensures that for each variable, the points are randomly, uniformly distributed [28]. Proper selection of initial DOE points is of utmost importance as these points serve as starting points for the GA. Table 3.7 shows the parameters considered for the optimization algorithm.

Table 3.7 Optimization Algorithm Parameters for Canti-Duo Optimization

Optimization Parameters	Value
Number of initial DOE	25
Number of NSGA-II generations	40
Cross-over Probability	0.9
Mutation Probability	1.0
Total number of design points	1000

For each Abaqus simulation, the y- displacement values at the top center of the meta-material for all the four load-cases and the maximum Von-Mises stress developed in the meta-material structure at the third load-case corresponding to 20% strain are extracted from an output report file. Two objective functions are then formulated as shown below:

$$min\, Strain\_Error : f_1 = \sum_{i=1}^{4} (\varepsilon_i^t - \varepsilon_i{}^c)^2 \quad \textbf{Eq. 3.5}$$

$$min\, Max\_Stress : f_2 = max\ \sigma_{Von-Mises\ @\ 3^{rd}\ load-case} \quad \textbf{Eq. 3.6}$$

Note that ε_i^t and $\varepsilon_i{}^c$ are the target strain and the meta-strain at the i^{th} load-case.

Further, as discussed in section 2.4, constraints are imposed on the objective functions as shown in Eq. 3.7 and Eq. 3.8. While it is expected that a good meta-material

design would attain strain error and stress values much lower than the constraint values, these prescribed values are expected to give a good indication of the feasibility of the meta-material based on the results obtained from the optimization.

$$Strain\_Error \leq 2.5E-04 \quad \textbf{Eq. 3.7}$$

$$Max\_Stress \leq 950\ MPa \quad \textbf{Eq. 3.8}$$

The summary of the design points generated by the optimization run is shown in Figure 3.12. It can be seen that only 1% (10) of the total number of design points (1000) are found to be feasible i.e. satisfying both the constraints.

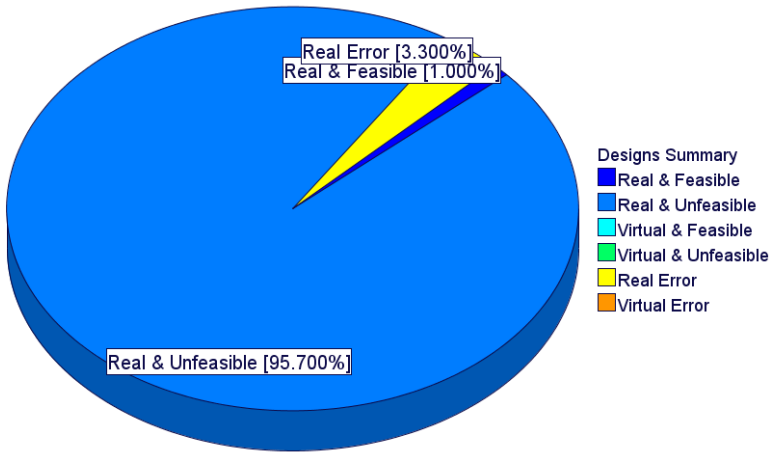

Figure 3.12 Design Summary for Canti-Duo Optimization

Table 3.8 shows the optimization results obtained for one of the best feasible designs. Note that the maximum stress value in the deforming meta-material is about 85% of the yield strength of the Titanium alloy. Figure 3.13 shows the deformation response of this design as compared to the target curve.

Table 3.8 Best Canti-Duo Design Parameters

Parameter	Value
g	0.00024855 m
t_2	0.0011166 m
t_3	0.0020549 m
Strain_Error	2.38E-04
Max_Stress	933 MPa

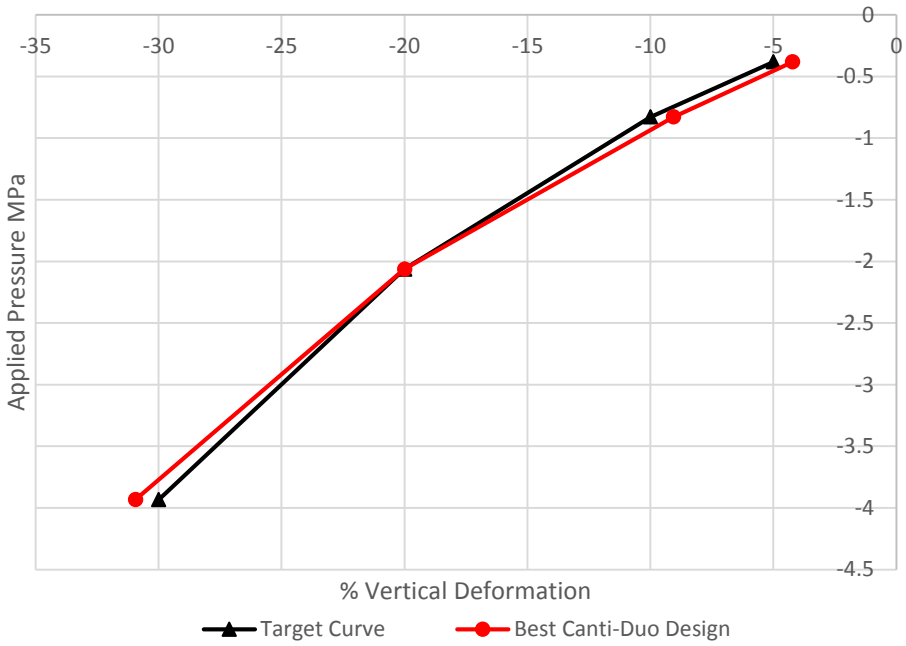

Figure 3.13 Optimized Canti-Duo – Target Properties Comparison

3.2.6. Discussion

The reasons for obtaining such a low percentage of feasible design points are investigated. One of the classical traits of multi-objective optimization with conflicting objectives is that improvement in one objective function leads to degradation of other objective function values [29]. With just three optimization variables, the optimization algorithm may not have a large enough design space to minimize both the conflicting objective functions simultaneously. Also, the stresses developed in the optimized Canti-

Duo design are not sufficiently low to account for protection against fatigue failure during normal tank operation. Hence, even though the Canti-Duo design leads to certain constraint-satisfying feasible design points, it is desired to further improve the meta-material design by deeming the current design as undesirable and going back to Step 2 to modify the UC geometry.

3.3. Canti-Oval UC Based Meta-material Design

Even though the Canti-Duo design was deemed undesirable, it was able to generate few feasible design points that satisfied the constraints. Hence, according to the Unit Cell Synthesis Method, going back to Step 2, attempts are made to design a higher order UC configuration and to combine existing CB EFG along with another EFG rather than replacing the original EFG by another entity altogether.

3.3.1. Step 2: EFG Selection and Combination

Based on the results obtained from the Canti-Duo design, it was observed that the design variables associated with the CB EFG alone are not sufficient to satisfactorily match the target curve and simultaneously reduce the stresses. Hence, looking at the EFG repository, an EFG is sought that can be easily combined with the CB EFG and also impart more control in achieving the aforementioned targets.

The Oval beam (OB) in pushing configuration as shown in Figure 2.2 shows a stiffening behavior similar to the CB. Combining the OB and CB EFGs together will form a second order configuration system as shown in Figure 2.5. Furthermore, as illustrated in Eq. 2.1 and Eq. 2.2, if these two EFGs are added in parallel, the net effective stiffness

would naturally be more than the stiffness of either EFG. If added in series, the effective stiffness would be smaller than the smallest stiffness value among the two EFGs. Hence, in order to compensate for the drop in the effective stiffness value, the series combination provides an opportunity to have thicker EFG entities which would better comply with the manufacturing constraints. Hence, it is decided to combine the two EFGs in series to form a second order connection configuration.

3.3.2. Step 3: ESG Design to Form UC

The conceptual Canti-Oval UC design is shown in Figure 3.14. It can be seen from the figure that the CB and the OB are placed in series on either side of the central axis. The ESG design is similar to the Canti-Duo design except for additional ESGs that connect the OB to the outer boundary of the UC. These ESGs are important to provide connectivity while carrying out tessellation and in transmitting the loads from one UC layer to another beneath it.

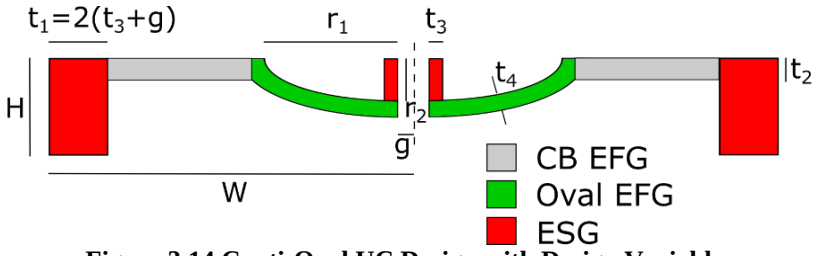

Figure 3.14 Canti-Oval UC Design with Design Variables

The UC consists of three additional design variables – r_1, r_2 and t_4. This leads to a total of eight independent design variables. Apart from representing the major radius of the OB, the variable 'r_1' also determines the percentage of OB and CB included in the UC design. Thus, as the values of 'r_1' increases, the contribution of the CB in the UC reduces.

Figure 3.15 shows an equivalent second order spring system for the Canti-Oval UC configuration. The nonlinear stiffness functions of the CB and the OB half beams are represented by K_1 and K_2 respectively.

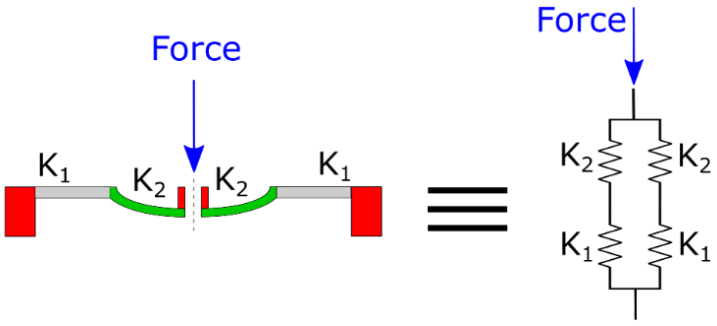

Figure 3.15 Equivalent Second order Spring System of Canti-Oval UC

3.3.3. Step 4: Tessellation of UC into a Meta-material

The tessellation of the Canti-Oval UC is carried out in a manner similar to the Canti-Duo design as explained in section 3.2.3. Each alternate UC layer in the y- direction is given a half UC width shift for efficient transmission of loads. Top and bottom face sheets are added once the tessellation is complete. Similar to the Canti-Duo design the number of cells in the x- and y-direction are kept constant at (n_{xdir} =) 3 and (n_{ydir} =) 6 respectively. Figure 3.16 shows the preliminary meta-material structure once the tessellation has been completed. The design concept can now be tested to evaluate its feasibility.

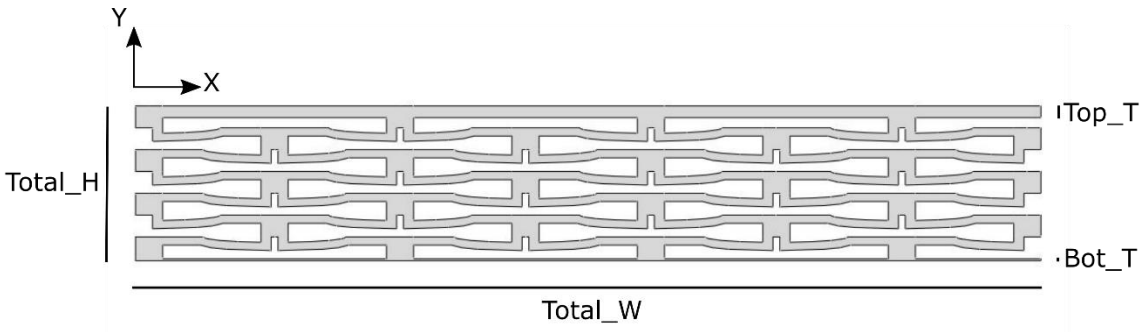

Figure 3.16 Tessellated Canti-Oval Meta-material with Over-all Dimensions

3.3.4. Step 5: Concept Evaluation

Loads and boundary conditions are applied on the concept meta-material in a manner shown in Figure 3.8. Variables H, W, t_2, t_4, r_1, r_2 are directly related to the EFGs and are responsible in controlling the deformation behavior of the meta-material. However, the values of variables 'H' and 'W' can be held constant based on the values obtained from the Canti-Duo Design. Thus, in order to carry out a full factorial sensitivity study for four variables with 2 levels each, a total of $2^4 = 16$ design experiments will have to be carried out. However, since the CB based UC was already deemed feasible, the combination of CB and OB, both of which exhibit stiffening behavior, is expected to show a resultant stiffening behavior as well. Hence, to save computational cost, a reduced factorial study is carried out. Table 3.9 and Table 3.10 show the design parameters considered for the sensitivity study.

Table 3.9 Reduced Factorial Parameters for Canti-Oval UC Concept

Design Variable	Low Level (0) (m)	High Level (1) (m)
r_1	0.0008	0.0150
r_2	0.0004	0.0010
t_2	0.0010	0.0012
t_4	0.0008	0.0012

Table 3.10 Constant Design Variable Values for Canti-Oval Concept Evaluation

Design Variable	Value (m)
H	0.0032
W	0.0205
g	0.0005
t_3	0.0015
Top_T	0.0017
Bot_T	0.0003

The results of the reduced factorial sensitivity study can be seen in Figure 3.17. The values indicated in the legend of the plot denote the levels for r_1, r_2, t_2 and t_4 respectively. As expected, the Canti-Oval UC design exhibits a nonlinear deformation behavior and the degree of nonlinearity closely resembles that of the target response. In fact, some of the curves obtained from the study such as '0110' and '0011' are found to be very close to the target curve. Hence, the Canti-Oval concept UC is deemed feasible and the multi-objective optimization step can be carried out to find the optimized meta-material geometry.

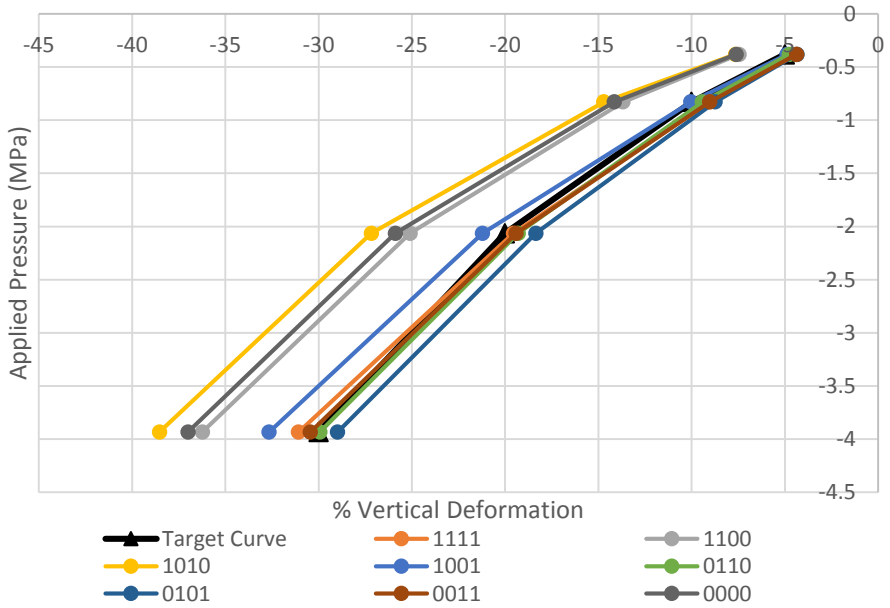

Figure 3.17 Reduced Factorial Study for Concept Evaluation of Canti-Oval Meta-material

3.3.5. Step 6: Multi-Objective Optimization

Similar to the optimization set-up for the Canti-Duo design, the optimization variables have to be determined for the Canti-Oval UC geometry. Out of the eight independent design variables, '*H*' and '*W*' are held constant as before to fix the UC boundaries. Also, the face-sheet thickness values '*Top_T*' and '*Bot_T*' remain unchanged. Thus, six variables are considered for the optimization process. Figure 3.18 shows the

optimization set-up in modeFRONTIER 4.4.2. It is similar to the set-up for the Canti-Duo optimization except for three additional design constraints. These constraints are essential for constructing the UC geometry in Abaqus and to ensure that there is no contact within the meta-material structure during deformation for the third load-case. The formulation of these constraints is explained below in detail.

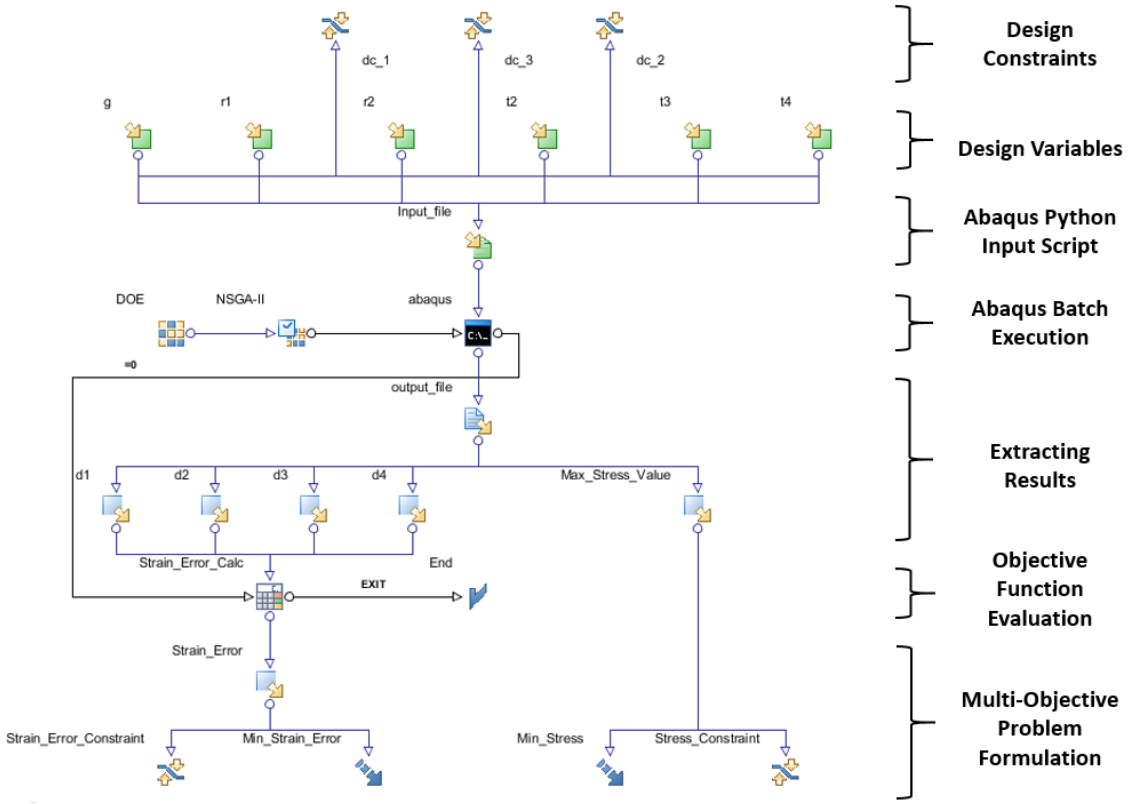

Figure 3.18 Optimization Set-up for Canti-Oval Meta-material

The first design constraint ensures that there is no contact within the meta-material structure as it undergoes 20% vertical deformation. Eq. 3.4 has been modified to account for the Canti-Oval design parameters to develop Eq. 3.9. Plugging the known values for n_{ydir}, H and $Total\_H$ and taking δ = 0.0006 m, the first design constraint obtained is shown in Eq. 3.10.

$$n_{ydir} \times (H - r_2 - t_4) \geq 20\% \times Total\_H + n_{ydir} \times \delta \quad \textbf{Eq. 3.9}$$

$$r_2 + t_4 - 0.0018 \leq 0 \quad \textbf{Eq. 3.10}$$

The second design constraint shown in Eq. 3.11 ensures that the CB thickness is less than the sum of the minor radius and the thickness of the OB. This ensures that the UC geometry is error-less when it is constructed in Abaqus.

$$t_2 - r_2 - t_4 + 0.00001 \leq 0 \quad \textbf{Eq. 3.11}$$

The third design constraint too, as shown in Eq. 3.12, ensures that the UC geometry is built without any errors by constraining the major radius of the OB such that it not does not intersect the ESGs on the sides.

$$r_1 + 3g + 2t_3 - W \leq 0 \quad \textbf{Eq. 3.12}$$

Table 3.11 shows the limits of the design variables considered for the optimization. The python script for the Abaqus input file for the Canti-Oval design is attached in Appendix B. The optimization algorithm parameters employed are the same as the Canti-Oval design and shown in Table 3.7. The initial DOE is generated using Latin Hypercube samplings and NSGA-II is used as the optimization algorithm.

Table 3.11 Limits of Design Variables for Canti-Oval Optimization

Design Variable	Lower Bound (m)	Upper Bound (m)
g	0.0001	0.0006
r_1	0.0100	0.0160
r_2	0.0002	0.0006
t_2	0.0010	0.0018
t_3	0.0010	0.0030
t_4	0.0010	0.0018

The constraints and objective function formulations for the Canti-Duo design as shown from Eq. 3.5 to Eq. 3.8, remain the same for the Canti-Oval design. Once the optimization process is carried out, the designs are evaluated to find the optimal meta-material design.

Figure 3.19 shows the design summary of the design points that are obtained from the optimization process. Out of 1000 design points, 65.6 % feasible designs are obtained that satisfy the imposed constraints. This is a good improvement over the Canti-Duo design which managed to generate only 1% feasible designs.

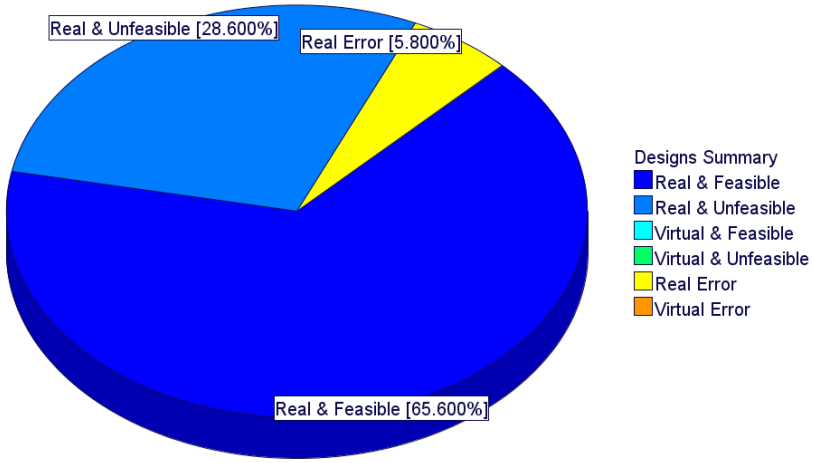

Figure 3.19 Design Summary for Canti-Oval Optimization

Figure 3.20 shows the Scatter Plot for all the feasible design points plotted against the two objective functions. The points marked in green are the Pareto optimal points. As explained before, a multi-objective optimization problem does not have a single optimum solution but multiple alternate solutions called Pareto points. Therefore, it becomes a matter of selecting a design point that best suits the meta-material application. It can be observed from the plot that the Canti-Oval design has managed to lower the stresses

considerably as compared to the Canti-Duo design. Also, the strain error values are lower which suggests that the new design closely matches the target nonlinear response.

The final design is selected from among the Pareto points that has moderate values of both objective functions and not from among the extreme points since both the objective functions are equally important for the track pad application. The optimized design variable values for the design chosen are shown in Table 3.12.

To take into account the manufacturing tolerances, the design variable values are rounded off. This has an effect on the strain error and the maximum stresses developed in the meta-material as shown in Table 3.12. While the maximum stress value reduces, the strain error increases by about 15%. However, it is an acceptable change in view of the overall results obtained.

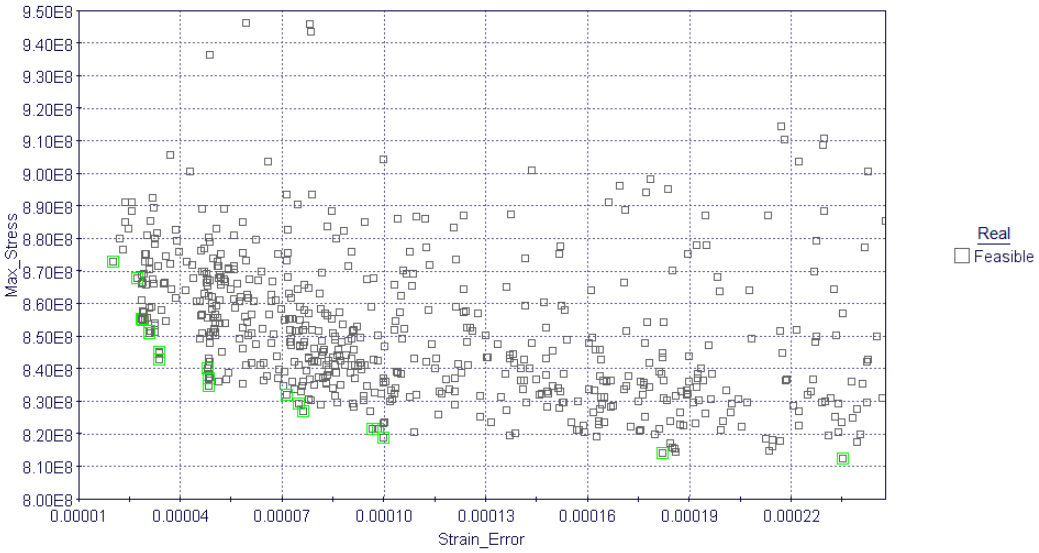

Figure 3.20 Pareto Front for Canti-Oval Optimization

Figure 3.21 shows the deformation plot of the final optimized meta-material design. With a strain error of 5.09E-5, it can be seen that the curve closely matches the nonlinear

target curve of the elastomer. Also, the maximum stress developed in the structure at 20% vertical deformation is about 75% of the yield strength of the Titanium Alloy. Figure 3.22 shows the meta-material structure constructed with the optimized design variable values and Table 3.13 gives a summary of all the parameters associated with the design.

Table 3.12 Optimized Canti-Oval Design Parameters

Parameter	Optimized Value	Rounded-off Value
g	0.00020455 m	0.00020 m
r_1	0.01458600 m	0.01460 m
r_2	0.00040549 m	0.00040 m
t_2	0.00116280 m	0.00117 m
t_3	0.00184250 m	0.00184 m
t_4	0.00110420 m	0.00111 m
Strain_Error	4.45E-05	5.09E-05
Max_Stress	834 MPa	833 MPa

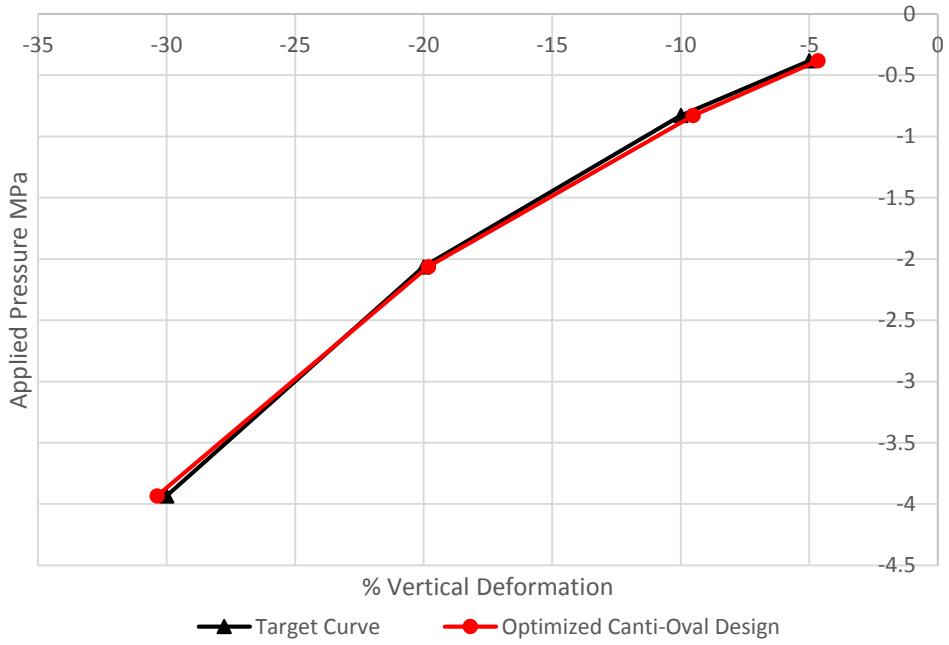

Figure 3.21 Optimized Canti-Oval - Target Properties Comparison

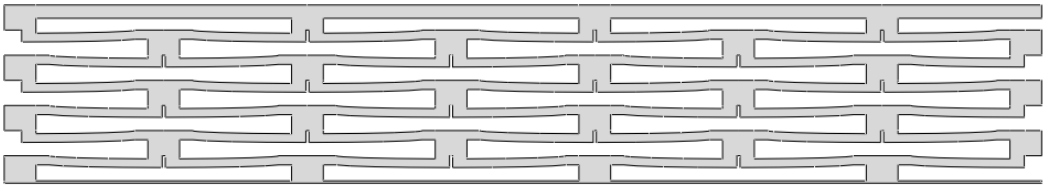

Figure 3.22 Optimized Canti-Oval Meta-material

Table 3.13 Summary of Final Canti-Oval Design

Parameter	Value
H	0.00320 m
W	0.02050 m
g	0.00020 m
r_1	0.01460 m
r_2	0.00040 m
t_2	0.00117 m
t_3	0.00184 m
t_4	0.00111 m
t_1	0.00408 m
n_{xdir}	3
n_{ydir}	6
Top_T	0.00170 m
Bot_T	0.00030 m
Total_H	0.02270 m
Total_W	0.13330 m
Total Depth	0.17000 m
Total Volume	0.00023725 m^3
Strain Error	5.09E-05
Max. Stress	833 MPa

3.3.6. Discussion

Hence, it can be concluded from the Canti-Oval design that higher order connection configurations increase the tuning ability of the meta-material deformation behavior by increasing the number of design variables and thus, augmenting the design space. With

eight independent design variables, the Canti-Oval design offers better solutions than its base design based on CB EFGs. However, combining multiple EFGs leads to complications arising due to the inclusion of additional design constraints. The Canti-Oval design consists of three design constraints that are essential for constructing an error-free UC geometry in Abaqus. These design constraints are dependent on the type of EFG and the order of connection configuration and need a good understanding of the UC geometry for their formulation.

3.4. Design Considerations for Improved Fatigue Life

The original Canti-Oval UC consists of sharp corners and abrupt change of cross-sections. This directly affect the stress concentration factors at those locations. It usually leads to a higher stress in these areas than the rest of the part. Since the meta-material is subjected to cyclic loading, faituge failures will usally initiate in these regions [30]. Therefore, it is intended to minimize the amount of stress concentration by providing fillets at the critical locations in the UC.

Figure 3.23 shows the positions of four fillets that are introduced in the optimized Canti-Oval UC design obtained in section 3.3.5. Table 3.14 shows the fillet radii values considered for the preliminary analysis. Fillets are not introduced on the end boundaries of the UC to prevent geometry errors while carrying out tesselallation.

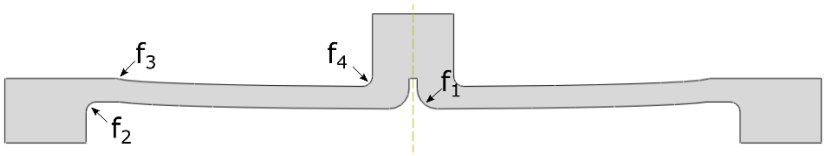

Figure 3.23 Fillet Positions on the Original Optimized Canti-Oval UC

Table 3.14 Preliminary Fillet Radii Values for Canti-Oval UC

Fillet	Radius (m)
f_1	0.0010
f_2	0.0005
f_3	0.0005
f_4	0.0005

Table 3.15 shows the deformation response of the Canti-Oval meta-material when the fillets are introduced in the UC as compared to the original optimized meta-material performance. It is obvious the meta-material gets stiffer with the introduction of fillets since the fillets are directly affecting the performance of the EFGs. The strain error, which is one the objective functions, calculated for the resulting meta-material response is found to be about 26% greater than the constraint value specified in Eq. 3.7. Hence, modifications have to be made to the UC design if fillets are to be taken into consideration.

Table 3.15 Comparison of Meta-material Performance With and Without Fillets

Stress (MPa)	Target % Strain	Original Meta-material % Strain Response	Modified Meta-material % Strain (with Fillets)	% Error between Original and Modified design
-0.3817	-5	-04.66	-04.31	-7.51
-0.8284	-10	-09.52	-08.87	-6.82
-2.0632	-20	-19.82	-18.89	-4.69
-3.9327	-30	-30.37	-29.35	-3.39
Strain_Error		5.09E-05	3.36E-04	-5.6 (Average)
Max_Stress		833 MPa	820 MPa	

Therefore, attempts were made to re-optimize the entire Canti-Oval UC based meta-material geometry by taking into consideration the fillets in the UC. However, it was observed that as the optimizer changed the design variable values during the optimization run, errors were introduced in the UC geometry that prevented Abaqus from constructing the UC geometry and perform the subsequent analyses. Hence, an alternative approach is adopted to determine the optimal UC geometry that consists of fillets as well.

It can be seem from Table 3.15 that the average strain (vertical deformation) experienced by the meta-material is about 5.6% lower when the fillets are introduced. Hence, to take into account the stiffening effect of fillets, a two-level optimization is proposed as shown in Figure 3.24. The idea is to optimize the original meta-material design without taking the fillets into consideration with augmented target strain values. The target strain values are augmented by 5.5% as shown in Table 3.16. The optimization set-up is the same as discussed before in section 3.3.5. Once the optimization is complete, the best Pareto optimal design is chosen and the design variable values are rounded off to take into consideration the manufacturing tolerances similar in a manner shown in Table 3.12. The strain error and the maximum stress developed in the structure are also dependent on the fillet radii. Hence, a shape optimization is carried out with the fillet radii as the design variables to match the original target response curve and simultaneously reduce the stresses [31]. Note that the fillet 'f_1' does not affect the performance of the meta-material and is excluded from the optimization run. Also, the design variables values obtained in the first optimization step are held constant during this process. The fillet radii are optimized with discrete values to take into account the manufacturing feasibility.

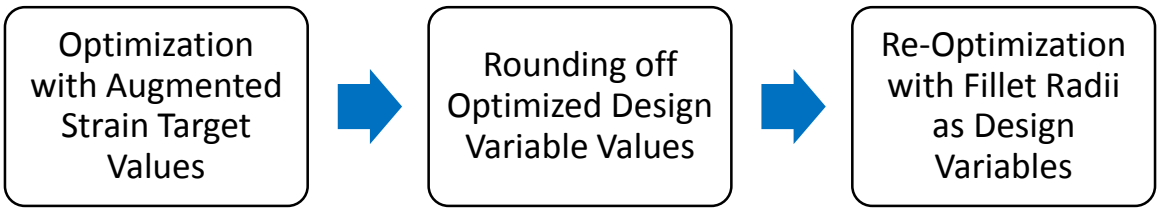

Figure 3.24 Two-Level Optimization for Fillet Consideration in Canti-Oval Meta-material

Table 3.16 Augmented Target Strain Values

Stress (MPa)	Target Augmented % Strain
-0.3817	5.275
-0.8284	10.55
-2.0632	21.1
-3.9327	31.65

The values of the final optimized design variables along with the fillet radii and objective functions are summarized in Table 3.17. Figure 3.25 shows the deformation responses of the designs obtained from the two optimization runs. Once, the final design is obtained, the geometry is exported to Solidworks software, end fillets are added to the edges and extrusion is carried out to produce the complete meta-material structure. Figure 3.26 shows the final meta-material design which is rendered in Solidworks.

Table 3.17 New Optimized Design Parameters with Fillet Radii

Design Variable	Value (mm)	Design Variable	Value (mm)	Design Variable	Value (mm)
W	20.5	r_1	11.6	f_1	1
H	3.2	r_2	0.45	f_2	0.75
g	0.39	t_3	1.58	f_3	0.3
t_2	1.15	t_4	1.04	f_4	0.55
Strain Error	7.25E-5		**Max Stress**	833 MPa	

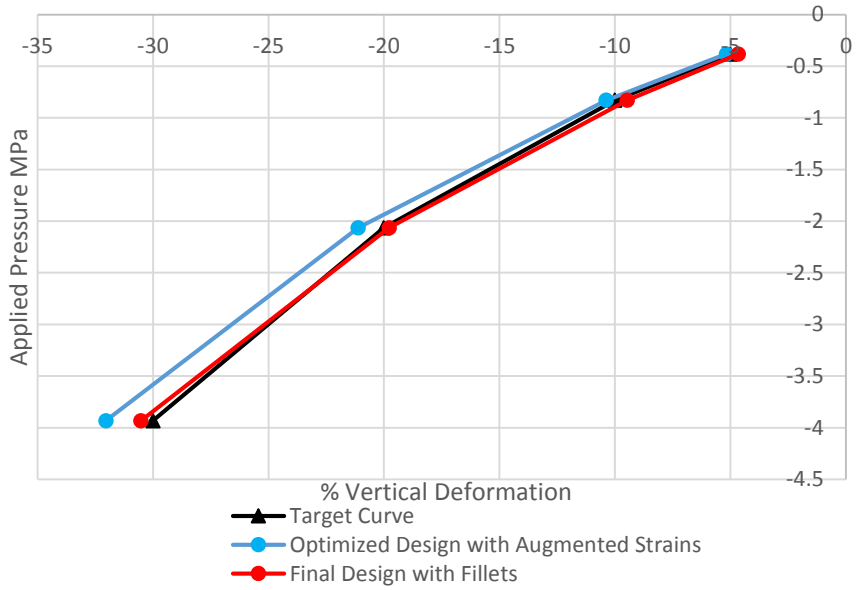

Figure 3.25 Target Properties Comparison for First and Second Level Optimization

Figure 3.26 Final Rendered Canti-Oval Design with Fillets

3.5. Conclusion

To summarize, using the Unit Cell Synthesis Method, the meta-material design process was initiated with a UC concept consisting of CB EFGs. The concept UC design was deemed feasible in the concept evaluation step for its ability to match the nonlinearity of the target response. Then multi-objective optimization was carried out to find a meta-material design that matches the target compression curve and, at the same time, has maximum stress developed in the deforming structure that is well within the elastic yield

limit of its constitutive material. However, the results obtained from the optimization were not satisfactory and the CB EFG based design was deemed undesirable.

Then, according to the method, a higher order connection configuration design was conceptualized which combined the CB and the OB EFGs in series. Once again, the steps involved in the Unit Cell Synthesis Method were followed. After carrying out concept evaluation and subsequent multi-objective optimization, a design was selected that best suits the requirements of the meta-material application. Finally, design changes for improving fatigue life were taken into consideration.

Since the meta-material is designed based on the results obtained by subjecting it to static loading conditions, it is paramount to compare its behavior with the elastomer pad in dynamic conditions as it deforms under the rolling tank wheels. This comparison is carried out in the next chapter.

CHAPTER 4. VALIDATION WITH DYNAMIC FINITE ELEMENT ANALYSIS

4.1. Motivation for Dynamic Analysis

Using the Unit Cell Synthesis Method, a meta-material design is obtained that matches the elastomer compression curve. However, the meta-material was designed by optimizing its deformation response under static load conditions. The backer pad on the tank track pad undergoes a combination of compression and shear forces as it passes under the road wheels. Also, the loads acting on the backer pad through the road wheel are not uniformly applied on the entire backer pad top surface. If the existing elastomeric pad is to be replaced by the meta-material ultimately, it is essential to determine the meta-material dynamic behavior as it deforms under the road wheel and compare it to the elastomeric pad. In order to carry out this comparison, dynamic finite element analyses are carried out to simulate a wheel roll-over event on the track link assemblies consisting of the original elastomeric backer pad as well has the meta-material. The set-up for these analyses and the results are discussed in this chapter.

4.2. Dynamic Analysis Set-up

The dynamic finite element analysis simulations are carried out in Abaqus 6.14. It was decided to use the Dynamic Explicit Scheme offered by Abaqus to carry out the analysis. An explicit dynamic analysis is computationally efficient for the analysis of large models with relatively short dynamic response times as compared to an implicit analysis [25]. Figure 4.1 shows the set-up adopted for the analysis. Three track link assemblies are

modeled along with a road wheel. The track links are placed on a flat ground which is modeled as a rigid surface. To save computational cost, all the entities are modeled with 2D plane strain formulation with out-of-plane thickness of 0.170 m. Each track link assembly consists of a backer pad, ground pad and a middle steel plate. Two simulations are carried out – one with all elastomeric backer pads as shown in Figure 4.1 and another with the central backer pad replaced by the meta-material. The set-up for both the simulations is similar.

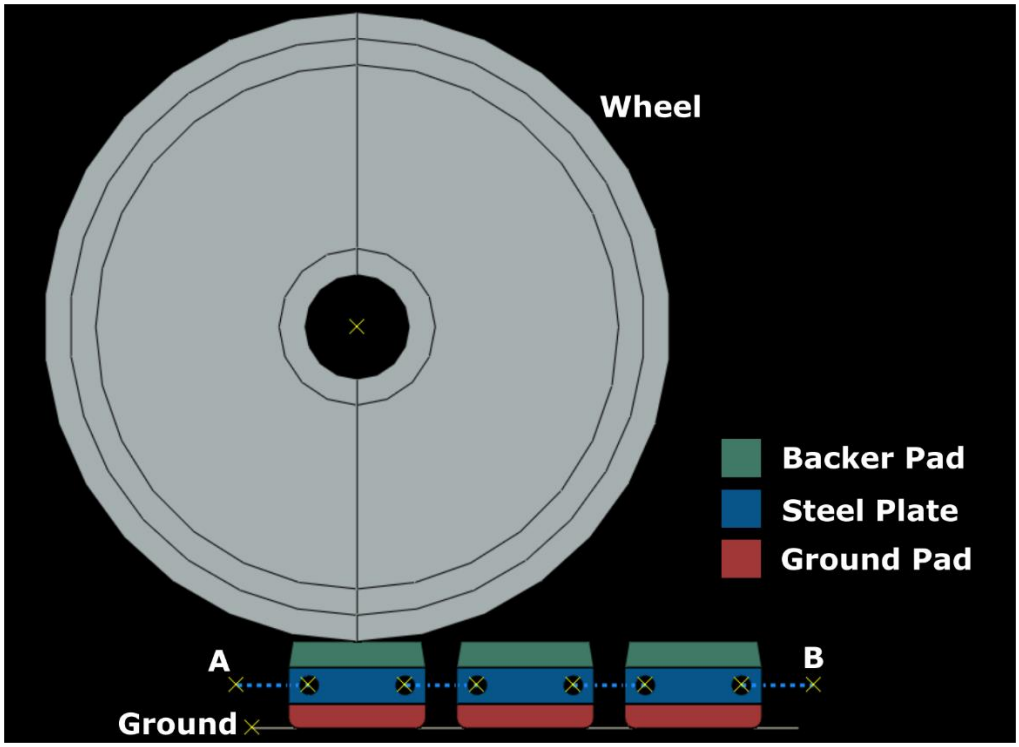

Figure 4.1 Dynamic Analysis Set-up

The dimensions for the road wheel and the components of the track pad assembly are obtained from [3] and are shown in Figure 4.2. The track pad assembly design is simplified for ease of geometry construction in Abaqus and computational analysis. The road wheel is made out of Steel and has a rubber layer with a thickness of 0.0254 m around

its periphery. The steel part of the wheel and track pad assembly are modeled with standard material properties for Steel (E= 210 GPa, υ = 0.3, ρ = 7850 kg/m^3). The elastomer for the backer pad and the ground pad are modeled using a 2nd Order Ogden Hyper-elastic material model. The model parameters are shown in Table 4.1 [12]. Note that the material properties for the rubber layer around the road wheel are not known. Hence, the same Ogden material model is used to model it.

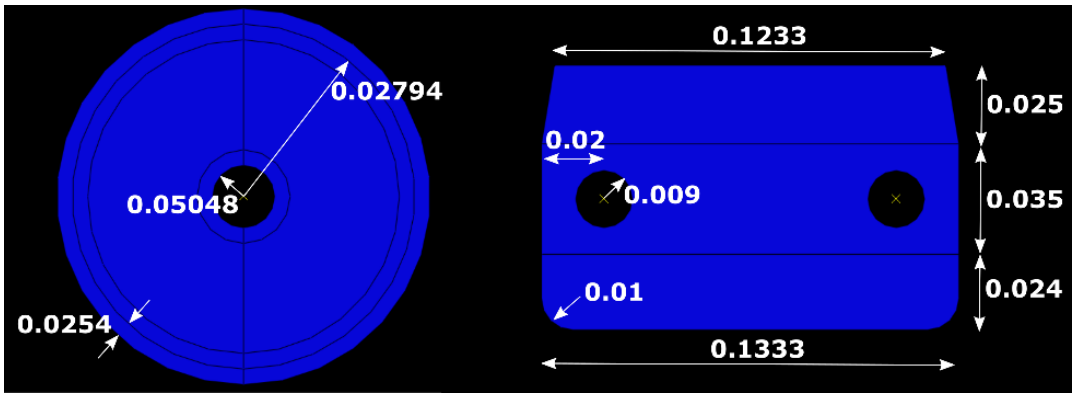

Figure 4.2 Dimensions for Road Wheel and Track Pad Assembly (m)

Table 4.1 Elastomer Material Properties [12]

Parameter	Value
mu_1	2275319 Pa
mu_2	54452 Pa
$alpha_1$	-1.00837
$alpha_2$	7.863497
D_1	1.42E-08
Density	950 kg/m^3

In order to simulate track-tension, three track link assemblies are connected using beam connector elements as shown in Figure 4.3. A beam connector element provides a rigid beam connection between two nodes [25]. These reference node points are the center points of each circular slot in the steel plate. Kinematic coupling constraints are modeled between the reference node points that connect the beam connector elements and the

periphery of the circular holes in the steel plate for each track link assembly. A kinematic coupling constrains the motion of the coupling nodes to the rigid body motion of the reference node [25]. A force of magnitude 22500 N is applied on the left most side (Point A) to simulate track-tension. This load simulates tension in the entire track link assembly due to the kinematic coupling. All degrees of freedom of the right most part of the link (Point B) are constrained. The ground which is modeled as a rigid surface is also fixed with all its degrees of freedom constrained.

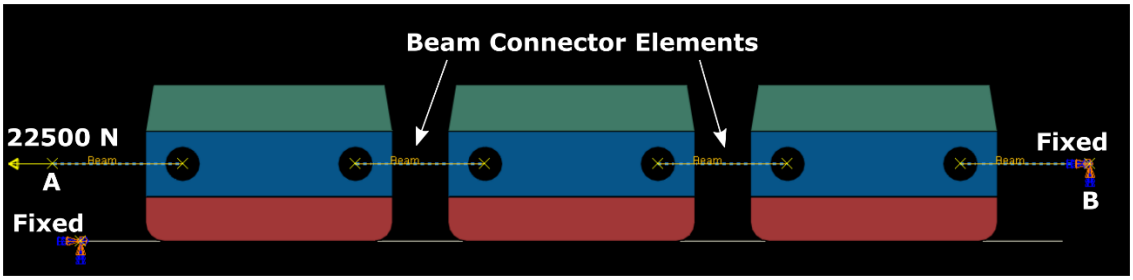

Figure 4.3 Boundary Conditions on Track Link Assemblies

Further surface to surface based tie constraints are modeled at the interface between the backer pad and the steel plate and between the steel plate and the ground pad. In Abaqus, a tie constraint ties two separate surfaces together so that there is no relative motion between them even though the meshes created on the surfaces of the regions are dissimilar [25].

For the road wheel, a rigid body constraint is employed to model the entire wheel as a rigid body. In Abaqus, a rigid body constraint is used to constrain the motion of a body to the motion of a reference point [25]. This reference point is defined at the center of the road wheel. There are two important reasons for considering the road wheel as a rigid entity. Firstly, the road wheel is stiff as compared to the deformable track pads. So,

considering it as a rigid entity helps to save computational costs as it is not necessary to determine the deformation and the stress field developed in the wheel. Secondly, as mentioned before, the material properties of the rubber layer used around the wheel periphery are not known and can create numerical errors in the model if they are not accurately modeled.

Figure 4.4 shows the boundary conditions applied at the reference point on the center of the road wheel. The weight of the tank acting on each track pad is calculated. The tank consists of 14 road wheels and each track link consists of two pads in the transverse direction. Assuming the tank weighs 63 tons, the load acting on each pad is calculated to be 22500 N. Also, linear and angular velocities corresponding to a tank speed of 40 mph are applied to the wheel. The idea is to let the wheel roll under its weight over the backer pads and to capture the deformation response in the pads during the course of the roll-over event.

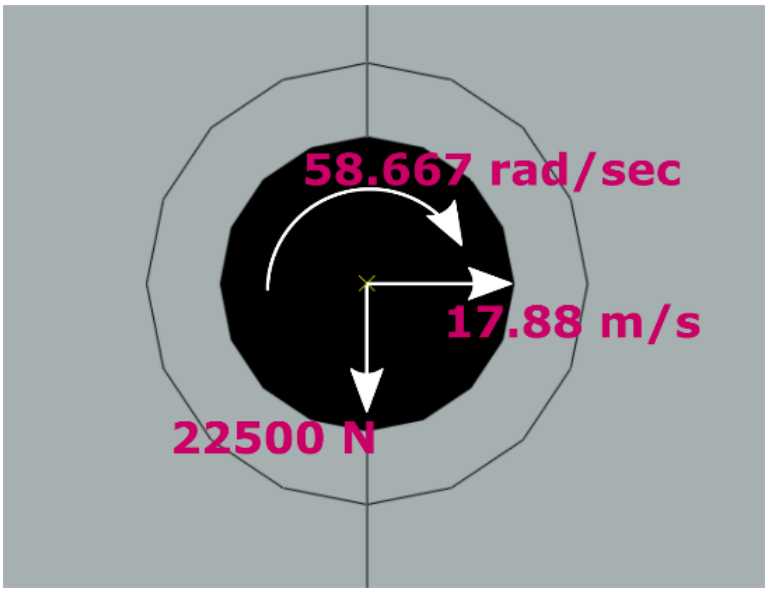

Figure 4.4 Boundary Conditions on the Road Wheel

Another important aspect that has to be modeled in the set-up is that of the interaction between the wheel and the backer pads and between the ground and the ground pads. Without these interactions, the loads cannot be transmitted effectively from the wheel to the track pads. For this purpose, surface to surface interactions are modeled with normal and tangential properties defined for the interaction behavior. A penalty formulation with friction co-efficient of 0.4 is chosen to define the tangential behavior.

Once the entire set-up is completed, the nonlinear dynamic explicit finite element analysis is run for a time period of 0.026 seconds. The explicit scheme integrates through time by using many small time increments [25]. The time step is a function of the smallest element dimension in the mesh. Hence, a trade-off has to be carried out in determining the optimum mesh size. A fine mesh will give more accurate results but will take a large amount of time to run the simulation. As mentioned before, the first simulation consists of all three backer pads which are modeled with the elastomer material properties. For the second analysis, the central backer pad is replaced by the Canti-Oval meta-material as shown in Figure 4.5. The meta-material geometry considered for the analysis is the one summarized in Table 3.13 and not the one consisting of fillets in the UC geometry. Note that the central backer pad has been chosen as the point of interest since it accurately represents the real conditions experienced by the pads.

The meta-material is made out of Titanium Alloy with properties shown in Table 3.2. To protect the pad from abrasion, it is proposed to use a 0.0023 m thick Steel plate, which is bonded on to the top face of the meta-material. This plate is intended to act as an abrasion plate which comes in the contact with the road wheel. The total height of the meta-

material and the steel plate combined is 0.025 m. The rest of the set-up remains the same. The deformation responses of the elastomeric backer pad and the meta-material are compared once both the simulations are completed.

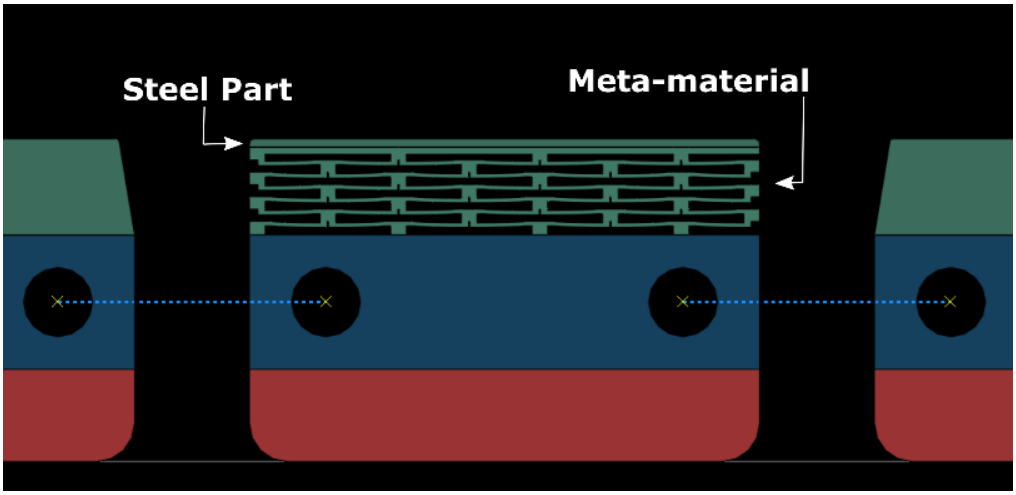

Figure 4.5 Meta-Material Placement in Dynamic Analysis

4.3. Results of Dynamic Analysis

The deformation and stress fields developed in the central backer pad are recorded at different time steps as the wheel rolls over it. Figure 4.6 shows the Von-Mises stress contour plot of the deformed meta-material when the wheel has just passed over its center.

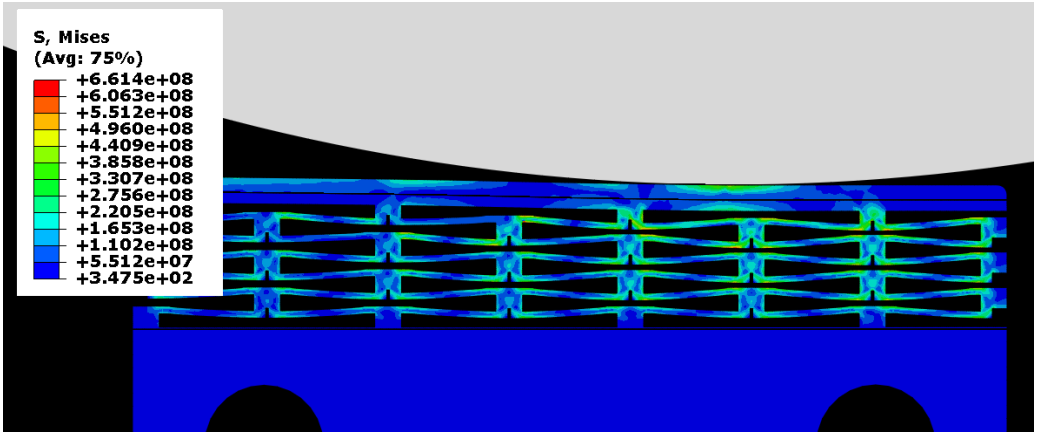

Figure 4.6 Stress Contour Plot of Meta-material Deforming under the Wheel

To compare the deformation response between the elastomer and the backer pad, the y- displacement history at three different locations on the backer pads is recorded for each case. Figure 4.7 shows the comparison of the deformation response at three different locations as shown in each plot. A good indication of meta-material performance is to compare the meta-material deformation value with that of the elastomer at the time step where the elastomeric pad undergoes maximum deformation for each of the three locations. The elastomer will typically undergo maximum deformation at a particular location when the wheel is directly above it. Table 4.2 shows the deformation response error calculation. For each of the locations the meta-material shows lower deformation as compared to the elastomer and the deformation error value increases from left to right on the pad surface. However, the average deformation error is less than 7%.

Table 4.2 Deformation Response Error between Elastomer and Meta-material

Location	Elastomer (m)	Meta-material (m)	Error (%)
Left	-0.00144682	-0.00141406	2.264
Center	-0.00278316	-0.00262151	5.808
Right	-0.00440671	-0.00385738	12.465
Average Error			6.846

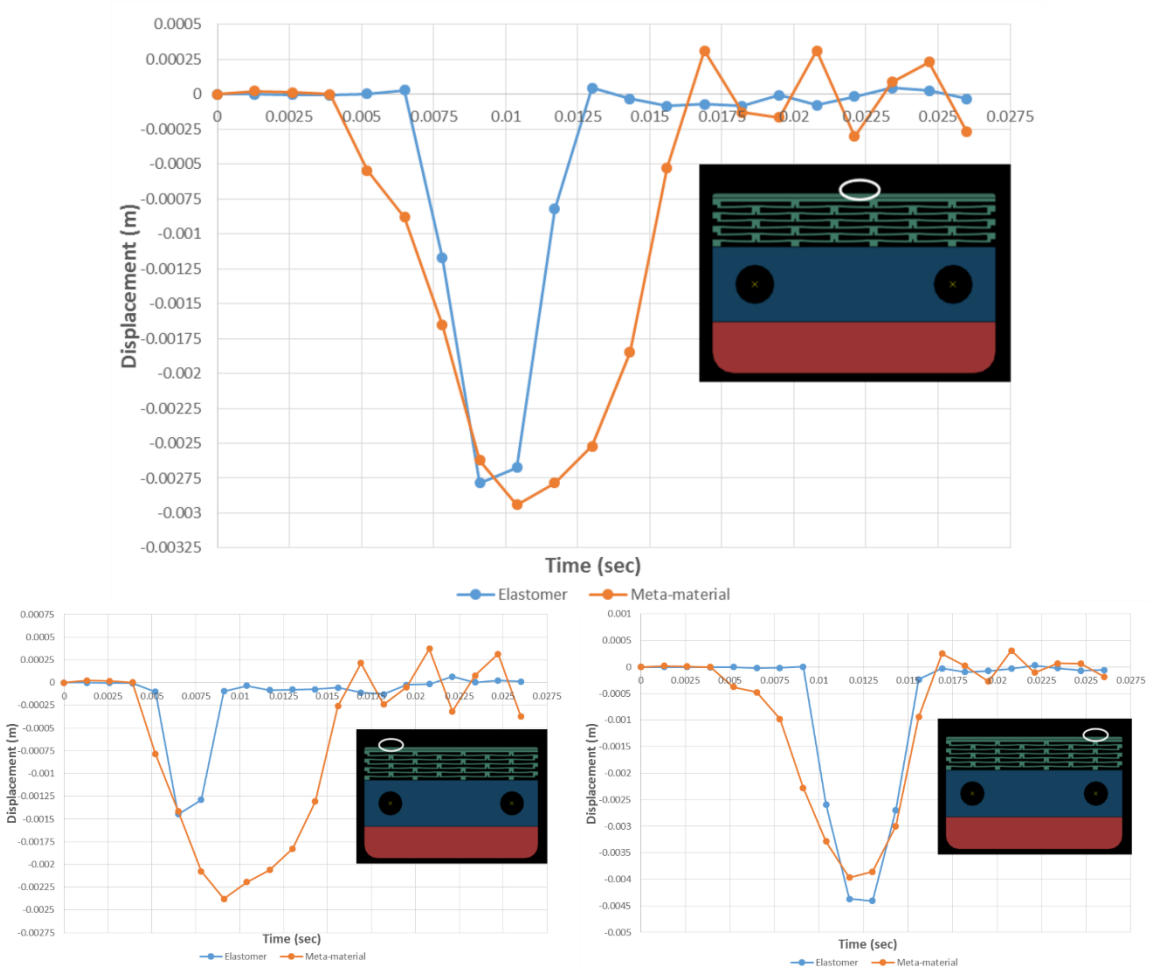

Figure 4.7 Dynamic Deformation Response Comparison between Elastomer and Meta-material

4.4. Discussion

Since the average error in the meta-material and the elastomer deformation response is less than 7%, it is reasonable to say that the designed meta-material will serve as an effective replacement for the existing elastomeric track pads. One glaring aspect of the deformation plots shown in Figure 4.7 is that every deformed region of the elastomeric pad seems to be quickly regaining its original configuration as soon as the wheel passes over it. This is not possible in the case of the meta-material as its entire flat top surface

remains deformed as long as the wheel stays in its contact at any point. However, if the wheel and the rubber layer around it are modeled as a deformable body, the rubber layer on its periphery will deform as it gets in contact with the backer pad surface. It is expected that this deformation of the rubber layer will transmit the loads over a larger region of the backer pad surface than the current scenario. This phenomenon will have two effects. Firstly, any point on the elastomeric backer pad will stay deformed for a longer duration than it currently is. This will help close the gap between the deformation behavior of the elastomer and the meta-material in terms of their tendency to regain their un-deformed state. Secondly, the meta-material deformation response itself will be improved as the deformed wheel rubber layer would simulate uniform load application conditions much like the loading conditions under which the meta-material has been designed.

Another factor that can be attributed to the difference between the deformation responses is that the meta-material has been designed for the dominant mode of deformation which is compression. However, as the wheel rolls over it, the backer pad experiences a combination of shear and compression which leads to a variation in the deformation response.

The next chapter discusses the conclusions and scope for future work for further development of the meta-material and also the Unit Cell Synthesis Method.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

5.1. Conclusions

The work presented in Chapter 2, Chapter 3, and Chapter 4 has successfully addressed the research objectives that have been defined in section 1.5.

1. In Chapter 2, the Unit Cell Synthesis Method is described in detail as explained in [16]. A probable cause for the inability to obtain a feasible meta-material design using the existing method is determined. On careful analysis, a modification is proposed to the method which involves replacement of step 6 with a multi-objective optimization step. This change is proposed to ensure that all application specific requirements are taken into consideration while optimizing the meta-material design.

2. In Chapter 3, the meta-material design process is initiated using the design method with proposed modification to mimic the nonlinear compressive behavior of the existing tank track backer pads. A cantilever beam based UC geometry is initially conceptualized for preliminary analysis. On following the steps prescribed in the Unit Cell Synthesis Method, the concept UC is deemed feasible in the concept evaluation stage and multi-objective optimization is carried out to match the target response curve and minimize the stresses developed in the meta-material structure. Satisfactory results are not obtained from the results of the optimization set-up and it is decided to adopt a higher order connection configuration in accordance with the method. Therefore, an oval beam is added in series with the existing cantilever beam to construct the new UC. On performing concept evaluation and subsequent multi-objective optimization, a Pareto optimal design solution is chosen as the final meta-material geometry. This design solution satisfies all the

application specific requirements predetermined for the track pad application. The successful attempt at designing the meta-material serves as a good validation of the Unit Cell Synthesis Method and the modification suggested for improving the same.

3. The meta-material, designed using the Unit Cell Synthesis Method, is a result of optimization of the meta-material deformation response carried out using static finite element analysis. However, if it is to be implemented as an effective replacement of the elastomeric backer pads, its performance has to be gauged in a dynamic system which involves the interaction of the rolling tank road wheel with that of the track pad assembly. Hence, a dynamic finite element analysis is carried out in Chapter 4 and the deformation response of the meta-material pad is compared to that of the existing elastomeric pad. The findings of this comparative study proves that the meta-material can indeed be used as an effective replacement for the current pads. This study also proves the efficacy of the meta-material designed using the Unit Cell Synthesis Method by validating its performance against the application specific scenario.

5.2. Future Work

The scope for future work is divided into two parts i.e. further development and physical implementation of the meta-material track pad and improvement and validation of the Unit Cell Synthesis Method.

5.2.1. Meta-material Track Pad Design

Some of the future work that can be carried out on the meta-material track pad is discussed below:

1) Stress Reduction

Different UC designs may be considered with a different or higher order connection configuration to find alternate designs that match the target response curve and further minimize the stresses. Using the Unit Cell Synthesis Method, multiple feasible UC designs may be obtained that satisfy the same set of objectives but having different advantages over the others based on ease of manufacturing and maximum stress developed.

2) Fatigue Life Estimation

A fatigue life analysis may be carried out to determine the fatigue life of the meta-material track pad subjected to cyclic compressive loading. Since the pad undergoes approximately 53000 compression cycles during a 500 mile run, it is expected that, for a meta-material to last for about 2000 miles, it should be able to withstand at least 200000 compressive cycles. However, taking road obstacles and hazards into consideration, the fatigue life should be aimed at 400000 cycles. A fatigue life determination analysis will help give a good insight into the areas in the pad that would experience failure first and changes may be incorporated into the design accordingly.

3) Detailed Dynamic Analysis

A detailed 3D dynamic analysis should be carried out to accurately gauge the meta-material track pad deformation response. The road wheel suspension system should be modeled to accurately simulate the tank road wheel and track pad interactions. Different road terrains and obstacles may also be modeled to determine failure modes of the meta-material.

4) Design for Manufacturing

As seen in Figure 3.26, the meta-material track pad consists of gaps in its structure to account for deformation of the EFGs. During the normal tank operation, foreign bodies including stones, gravels etc. can enter these gaps and interfere with the functioning of the meta-material. Hence, it is important to cover all sides of the meta-material such that it prevents foreign bodies from entering the structure, protects the structure from external impact and at the same time, does not interfere with the deformation behavior of the meta-material.

Another aspect that needs to be taken into consideration is that of manufacturing tolerances. It is important to determine the effect of deviation of design variables and constitutive material property values on the overall performance of the meta-material. The manufacturing method and process control variables should be adjusted accordingly to take into account these factors.

5) Prototype Manufacturing and Testing

The meta-material track pad prototype should be ultimately manufactured and tested for its conformance with the computational results. Also, a static fatigue test with compressive cyclic loading may be carried out to determine the actual fatigue life of the track pad components. Another way of testing the fabricated component would be to insert it on to the actual tank tracks and test its durability either on testing grounds or on a test rig.

5.2.2. The Unit Cell Synthesis Method

Modifications are proposed to the Unit Cell Synthesis Method in Chapter 2 to improve the original method defined in [16]. Further improvements can be made to the method which are discussed in detail below:

1) EFG Repository Augmentation

Only three EFGs are considered till now having four different deformation behaviors as shown in Figure 2.2. An expansion of the EFG repository is required to offer a wide spectrum of nonlinear deformation behavior and increase the number of possible combinations and configurations. Also, EFGs should be identified that may be applicable when the meta-material is to be developed for pure shear loading conditions or a combination of tension, compression or shear.

2) Asymmetric UC or Aperiodic Meta-material

All the UC designs that were conceptualized in [16] and in this work consider symmetrical UCs and periodic tessellation of a single UC in the meta-material structure. It would be interesting to determine the effects of considering asymmetric UCs in the meta-material structure. Also, different UCs may be incorporated into an aperiodic meta-material to investigate if it imparts a higher order nonlinear deformation behavior typically represented by an 'S' shaped curve.

3) Automation of Method Implementation

The inclusion of the multi-objective optimization formulation as proposed in section 2.4 is one step towards the automation of the method as the designer is no longer required to evaluate all the optimal design points for their feasibility with respect to the

manufacturing and stress constraints. The method can be further automated by determining the force-displacement behavior of the EFGs and their combinations analytically as a function of their geometrical and material parameters. If such analytical functions are determined, Step 5 which involves concept evaluation can be entirely eliminated. Knowing these functions, and cross referencing it to the target curve, the ability of the EFG combinations to match the target curve can be easily determined, thereby eliminating the need to carry out a DOE or a sensitivity study in the concept evaluation step.

4) Method Validation

So far, the Unit Cell Synthesis Method has only been implemented to design meta-materials that match a nonlinear compression curve. In order to validate the method, more case studies should be performed to match nonlinear target response curves in not only compression but also tension and shear. Further, attempts may be made to design a meta-material that completely matches the nonlinear deformation behavior of an entity by matching its uniaxial tension, pure shear and equi-biaxial tension curves simultaneously. If such a meta-material is successfully designed, it would serve as an ultimate replacement of the original target material and serve as the ultimate validation of the method.

REFERENCES

[1] Katz, H. S., and Wittig, J. T. J., 1986, "Development and Fabrication of Track Pads."

[2] Medalia, A. I., Alesi, A. L., and Mead, J. L., 1991, "Pattern Abrasion and Other Mechanisms of Wear of Tank Track Pads," Rubber Chem. Tech ology, **65.**

[3] Dangeti, V. S., 2014, "Identifying Target Properties for the Design of Meta-Material Tank Track Pads," Clemson University.

[4] Waldron, T., 2012, "Development of Laboratory Testing Apparatus and Fatigue Analysis for Tracked Vehicle Rubber Backer Pads."

[5] Lesuer, D. R., Santor, S. D., Cornell, R. H., and Patt, J., 1983, Field Evaluation of Tank Track Pad Failures.

[6] Ostberg, D., and Bradford, B., 2009, "Impact of Loading Distribution of Abrams Suspension on Track Performance and Durability," Proc. 2009 Gr. Veh. Syst. Eng. Technol. Symp. Fig., pp. 1–11.

[7] Gogos, C. G., and Andrews, R. D., 1978, A Study of Elastomeric Materials for Tank Track Pads.

[8] Lesuer, D. R., Zaslawsky, M., Kulkarni, S. V, Cornell, R. H., and Hoffman, D. M., 1980, Failure of Tank Track Pads, Livermore, California.

[9] Lentz, C. E., and Patt, J., 1982, Fabrication of T142 Tank Track Pads for Evaluation of a Rubber-Kevlar Composite Compound.

[10] Lesuer, D. R., Goldberg, A., and Patt, J., 1985, "Computer Modeling of Tank Track Elastomers," 32nd Sagamore Conf. July 22-26, 1985, lake Luzerne, New York.

[11] Goldberg, A., Chinn, D. J., and Brady, R. L., 1987, Studies on the Testing and

Analysis of T156 Tank-Track Shoes.

[12] Mars, W. V. (Endurica L., and Ostberg, D. (U. S. A. T., 2012, "Fatigue Damage Analysis of an Elastomeric Tank Track Component," Simulia Cust. Conf., pp. 1–14.

[13] Thyagaraja, N. B., 2011, "Requirements Determination of a Novel Non-Pneumatic Wheel Shear Beam for Low Rolling Resistance."

[14] Ashby, M. F., 2005, Materials Selection in Mechanical Design.

[15] Smith, D. . R., and Liu, R., 2010, "Metamaterials: theory, design and applications."

[16] Satterfield, Z. T., 2015, "Design of a MetaMaterial with Targeted Nonlinear Deformation Response," Clemson University.

[17] Czech, C., Guarneri, P., Gibert, J., and Fadel, G., 2012, "On the accurate analysis of linear elastic meta-material properties for use in design optimization problems," Compos. Sci. Technol., **72**(5), pp. 580–586.

[18] Fazelpour, M., and Summers, J. D., 2013, "A Comparison of Design Approaches to Meso-Structure Development," Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2013, pp. 1–10.

[19] Holzapfel, G., 2000, Nonlinear solid mechanics: a continuum approach for engineering.

[20] Mehta, V., 2010, "Design, analysis, and applications of cellular contact-aided compliant mechanisms," ProQuest Diss. Theses, **3436173**(August), p. 169.

[21] Choi, S.-K., and Patel, J., 2011, "Optimal Design of Cellular Structures under Random Fields," J. Eng. Des., **22**(9), pp. 651–668.

[22] Shankar, P., Fazelpour, M., and Summers, J. D., 2015, "Comparative study of optimization techniques in sizing mesostructures for use in nonpneumatic tires," J. Comput. Inf. Sci. Eng., **15**(4), p. 041009.

[23] Savic, D., 2002, "Single-objective vs Multiobjective Optimisation for Integrated Decision Support," Proc. First Bienn. Meet. Int. Environ. Model. Softw. Soc., pp. 7–12.

[24] Coello, C. A., and Christiansen, A. D., 2000, "Multiobjective optimization of trusses using genetic algorithms," Comput. Struct., **75**(6), pp. 647–660.

[25] "Abaqus Analysis User's Manual Version 6.14."

[26] Liang, Zhiyong; Wang, B., 2004, "Experimental design and optimization," Int. J. Nanosci., **293**(1-2), pp. 3–40.

[27] Elbeltagi, E., Hegazy, T., and Grierson, D., 2005, "Comparison among five evolutionary-based optimization algorithms," Adv. Eng. Informatics, **19**(1), pp. 43–53.

[28] "modeFRONTIER 4.4.2 User Manual."

[29] Fonseca, C. M., and Fleming, P. J., 1993, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," Icga, **93**(July), pp. 416–423.

[30] Tidbits, T., 2013, Effect of Stress Concentration on Fatigue Life.

[31] Francavilla, A., Ramakrishnan, C. V, and Zienkiewicz, O. C., 1975, "Optimization of shape to minimize stress concentration," J. Strain Anal. Eng. Des., **10**(2), pp. 63–70.

APPENDICES

APPENDIX A. PYTHON SCRIPT FOR CANTI-DUO DESIGN

```
### ABAQUS PYTHON INPUT SCRIPT FOR CANTI-DUO DESIGN ###

### By Neehar Kulkarni (neehark@g.clemson.edu) ###
### Department of Mechanical Engineering, Clemson University ###

###----------------------------------------------------------------------###

# -*- coding: mbcs -*-
# session.journalOptions.setValues(replayGeometry=COORDINATE,
recoverGeometry=COORDINATE)

### Defining Design Variables ###
Width  = 0.02050000  # Half Width of Unit Cell W
H1     = 0.00320000  # Height of Unit Cell H
Gap    = 0.00024855    # Gap g
Thick2 = 0.00111660  # EFG Thickness t2
Thick4 = 0.00205490  # ESG Parameter t3

Thick1 = 2*(Gap+Thick4)
H2     = 0.000
Thick3 = H1

BottomThickness = 0.0003 # Bottom Face Sheet Thickness Bot_T
TopThickness = 0.0017    # Top Face Sheet Thickness Top_T

Section_Thickness = 0.170 # Section Thickness

xdir = 3 # Number of Cells in x- direction
ydir = 3 # Number of Cells in y- direction

### Constitutive Material Properties ###
mat = 'Ti'  # Titanium Alloy
rho = 4820
E   = 102e9
v   = 0.32

### Load Cases ###
Press = [0.3817e6, 0.8284e6, 2.0632e6, 3.9327e6]

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 *

###-------------------------Meta-material Geometry Creation Begins---------------------###

### Half UC Geometry Construction in Abaqus CAE ###

mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)

#1
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, 0.0), point2=(
    Width-Thick4-Gap, 0.0))

#2
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[2])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, 0.0), point2=(
    0.0, -H1))

#3
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[3])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, -H1),
    point2=(Thick1, -H1))

#4
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[4])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Thick1, -Thick2),
    point2=(Width-Gap, -Thick2))

#5
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
```

```
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[5])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Thick1, -Thick2),
    point2=(Thick1, -(H1)))

#6
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[6])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap-Thick4,
H2+Thick3),
    point2=(Width-Gap-Thick4, 0.0))

#7
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[7])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap, -Thick2),
    point2=(Width-Gap, H2))

#8
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[8])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap-Thick4,
H2+Thick3),
    point2=(Width, H2+Thick3))

#9
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[9])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap, H2),
    point2=(Width, H2))

#10
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[10])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width, H2),
    point2=(Width, H2+Thick3))

mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[11])
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, 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__']


### Mirroring Half UC Geometry to form Entire UC ###
mdb.models['Model-1'].parts['Part-1'].DatumPlaneByPrincipalPlane(offset=Width,
    principalPlane=YZPLANE)
mdb.models['Model-1'].parts['Part-1'].Mirror(keepOriginal=ON, mirrorPlane=
    mdb.models['Model-1'].parts['Part-1'].datums[2])
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'])


### Single Tessellation of UC in y- direction with Half UC Width Shift ###
mdb.models['Model-1'].rootAssembly.LinearInstancePattern(direction1=(1.0, 0.0,
    0.0), direction2=(0.0, 1.0, 0.0), instanceList=('Part-1-1', ), number1=1,
    number2=2, spacing1=0, spacing2=H1+H2)
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-1-1-lin-1-2',
    ), vector=(Width-Thick1/2, 0.0, 0.0))


### Complete Tessellation of UCs in x- and y- direction ###
final = list()

for i in range(xdir):
 for j in range(ydir):
  if (j == 0 and i == 0):
   continue
  newline = "mdb.models['Model-1'].rootAssembly.instances['Part-2-1-lin-%d-%d'],"
%(i+1,j+1)
  final.append(newline)

#print final
finalline = "".join(final)
#print finalline

mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,
    instances=(mdb.models['Model-1'].rootAssembly.instances['Part-1-1'],
    mdb.models['Model-1'].rootAssembly.instances['Part-1-1-lin-1-2']), name=
    'Part-2', originalInstances=SUPPRESS)
mdb.models['Model-1'].rootAssembly.LinearInstancePattern(direction1=(1.0, 0.0,
    0.0), direction2=(0.0, 1.0, 0.0), instanceList=('Part-2-1', ), number1=xdir,
```

```
    number2=ydir, spacing1=Width*2-Thick1, spacing2=2*(H1+H2))
lastline = "mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,instances=(mdb.m
odels['Model-1'].rootAssembly.instances['Part-2-1'],"+ finalline +"),name='Part-3',
originalInstances=SUPPRESS)"
exec(lastline)


#### Top Face Sheet Creation ####
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(
    decimalPlaces=3)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(Width*2*xdir-(xdir-1)*Thick1-Thick1/2+Width, TopThickness))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-4', type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-4'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])


#### Bottom Face Sheet Creation ####
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(
    decimalPlaces=3)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(Width*2*xdir-(xdir-1)*Thick1-Thick1/2+Width, BottomThickness))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-5', type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-5'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])


### Small Part Creation for Side Edges of Tessellated Meta-material ###
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(
    decimalPlaces=3)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(Thick1/2, Thick3))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Small Part',
type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Small Part'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])
```

```
### Assembly of Small Part ###
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Small Part-1',
   part=mdb.models['Model-1'].parts['Small Part'])
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Small Part-1', ),
   vector=(0, (H1+H2)*(ydir*2-1)+H2, 0.0))


### Assembly of Top Face Sheet ###
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-4-1',
   part=mdb.models['Model-1'].parts['Part-4'])
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-4-1', ),
   vector=(0, (H1+H2)*(ydir*2-1)+H2+Thick3-TopThickness, 0.0))


### Assembly of Bottom Face Sheet ###
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-5-1',
   part=mdb.models['Model-1'].parts['Part-5'])
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-5-1', ),
   vector=(0, -(H1+BottomThickness), 0.0))

### Creation and Assembly of Half UCs for Meta-material Edges ###
mdb.models['Model-1'].Part(name='Part-1-Copy', objectToCopy=
   mdb.models['Model-1'].parts['Part-1'])
del mdb.models['Model-1'].parts['Part-1-Copy'].features['Mirror-1']
mdb.models['Model-1'].Part(compressFeatureList=ON, mirrorPlane=YZPLANE, name=
   'Part-1-Copy-Copy', objectToCopy=
   mdb.models['Model-1'].parts['Part-1-Copy'])

for i in range(ydir):
 instanceline = "mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-Copy-%d',part=mdb.models['Model-1'].parts['Part-1-Copy'])"%(i+1)
 exec(instanceline)

for j in range(ydir):
 instanceline2 = "mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-Copy-Copy-%d',part=mdb.models['Model-1'].parts['Part-1-Copy-
Copy'])"%(j+1)
 exec(instanceline2)

for ii in range(ydir):
 translation = "mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-1-
Copy-%d',),vector=(xdir*2*Width-xdir*Thick1, %d*2*(H1+H2), 0.0))"%(ii+1,ii)
 exec(translation)
```

```
for jj in range(ydir):
 translation2 = "mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-1-
Copy-Copy-%d',),vector=(Width+Thick1/2, H1+%d*2*(H1+H2), 0.0))"%(jj+1,jj)
 exec(translation2)

appendline = list()

for ij in range(ydir):
 instanceappend = "mdb.models['Model-1'].rootAssembly.instances['Part-1-Copy-%d']," 
%(ij+1)
 appendline.append(instanceappend)

for ji in range(ydir):
 if ji== ydir-1:
  instanceappend2 = "mdb.models['Model-1'].rootAssembly.instances['Part-1-Copy-
Copy-%d']),'' %(ji+1)
 else:
  instanceappend2 = "mdb.models['Model-1'].rootAssembly.instances['Part-1-Copy-
Copy-%d'],'' %(ji+1)
 appendline.append(instanceappend2)

finalappend = "".join(appendline)

mergeline = "mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,instances=(mdb.m
odels['Model-1'].rootAssembly.instances['Part-3-1'],mdb.models['Model-
1'].rootAssembly.instances['Part-4-1'],mdb.models['Model-
1'].rootAssembly.instances['Part-5-1'],mdb.models['Model-
1'].rootAssembly.instances['Small Part-1'],"+finalappend+"name='Part-6',
originalInstances=SUPPRESS)"
exec(mergeline)

###----------------------Meta-material Geometry Creation Ends-------------------------###

### Material Assignment ###
mdb.models['Model-1'].Material(name=mat)
mdb.models['Model-1'].materials[mat].Density(table=((rho, ), ))
mdb.models['Model-1'].materials[mat].Elastic(table=((E, v),
   ))


### Section Assignment ###
mdb.models['Model-1'].HomogeneousSolidSection(material=mat, name=
```

```
    'Section-1', thickness=Section_Thickness)
mdb.models['Model-1'].parts['Part-6'].Set(faces=
    mdb.models['Model-1'].parts['Part-6'].faces.findAt(((0.0, 0.0, 0.0),
    )), name='Entire body set')
mdb.models['Model-1'].parts['Part-6'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-1'].parts['Part-6'].sets['Entire body set'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)


### Mesh Generation ###
mdb.models['Model-1'].parts['Part-6'].seedPart(deviationFactor=0.1,
    minSizeFactor=0.1, size=Thick2/6)
mdb.models['Model-1'].parts['Part-6'].setElementType(elemTypes=(ElemType(
    elemCode=CPE4R, elemLibrary=STANDARD, secondOrderAccuracy=OFF,
    hourglassControl=DEFAULT, distortionControl=DEFAULT), ElemType(
    elemCode=CPE3, elemLibrary=STANDARD)), regions=(
    mdb.models['Model-1'].parts['Part-6'].faces.findAt(((0.0, 0.0, 0.0),
    )), ))
mdb.models['Model-1'].parts['Part-6'].generateMesh()


###--------------------------------SET CREATION BEGINS--------------------------------###

### Bottom Face Set ###
bottom = list()
bottomlinefirst = "((2*Width, -H1-BottomThickness, 0.0),),"
bottom.append(bottomlinefirst)

bottomline="".join(bottom)

bottomfaceset = "mdb.models['Model-1'].parts['Part-6'].Set(edges=mdb.models['Model-
1'].parts['Part-6'].edges.findAt(" + bottomline + " ), name='Bottom face')"
exec(bottomfaceset)


### Left Face Set ###
left = list()
for i in range(ydir+3):
 if i == 0:
  partline_left = "((0.0 , -(H1)/2, 0.0),),"
 elif i == ydir+1:
  partline_left = "((0 , (%d*2-1)*(H1+H2)+H2+Thick3-TopThickness/2, 0.0),),"%(ydir)
 elif i == ydir+2:
```

```
 partline_left = "((0.0 , -H1-BottomThickness/2,0.0),),"
 else:
  partline_left = "((0.0, (%d*2-1)*(H1+H2) + H2 + H1/3, 0.0),),"%(i)
 left.append(partline_left)

leftline = "".join(left)
#print leftline
leftfaceset = "mdb.models['Model-1'].parts['Part-6'].Set(edges=mdb.models['Model-
1'].parts['Part-6'].edges.findAt(" + leftline + " ), name='Left face') "
exec(leftfaceset)


### Right Face Set ###
right = list()
for i in range(ydir+1):

 if i == ydir:
  partline_right = "((Width*2*%d + Width - (%d-1)*Thick1 - Thick1/2 , H2 + (%d*2-
1)*(H1+H2) + Thick3 - TopThickness/2, 0.0),),"%(xdir,xdir,i)
 else:
  partline_right = "((Width*2*%d + Width - (%d-1)*Thick1 - Thick1/2, H2 +
2*%d*(H1+H2) + H1/2, 0.0),),"%(xdir,xdir,i)
 right.append(partline_right)

rightline = "".join(right)
#print rightline
rightfaceset = "mdb.models['Model-1'].parts['Part-6'].Set(edges=mdb.models['Model-
1'].parts['Part-6'].edges.findAt(" + rightline + " ), name='Right face') "
exec(rightfaceset)


### Top Face Set ###
top = list()
for i in range(xdir):
 partline_top1 = "(((%d+1)*Width*2 - %d*Thick1 - Thick1/2 - Gap/2 , H2 + (%d*2-
1)*(H1+H2) + Thick3, 0.0),),"%(i,i,ydir)
 partline_top2 = "(((%d+1)*Width*2 - %d*Thick1 - Thick1/2 + Gap/2 , H2 + (%d*2-
1)*(H1+H2) + Thick3, 0.0),),"%(i,i,ydir)
 top.append(partline_top1)
 top.append(partline_top2)

for j in range(xdir-1):
 partline_topmid = "(( 3*Width + 2*%d*Width, H2 + (%d*2-1)*(H1+H2) + Thick3,
0.0),),"%(j,ydir)
```

```
 top.append(partline_topmid)

partline_top_first = "((2*Width-Width/2 , H2 + (%d*2-1)*(H1+H2) + Thick3,
0.0),),"%(ydir)
partline_top_last = "((Width*2*%d + Width - (%d-1)*Thick1 - Thick1/2 , H2 + (%d*2-
1)*(H1+H2) + Thick3, 0.0),),"%(xdir,xdir,ydir)

top.append(partline_top_first)
top.append(partline_top_last)

partline_top_1 = "((H1/3 , H2 + (%d*2-1)*(H1+H2) + Thick3, 0.0),),"%(ydir)
partline_top_2 = "((H1/2+H1/3 , H2 + (%d*2-1)*(H1+H2) + Thick3, 0.0),),"%(ydir)

top.append(partline_top_1)
top.append(partline_top_2)
#Print top

topline = "".join(top)
#print topline

topfaceset = "mdb.models['Model-1'].parts['Part-6'].Surface(name='Top surface',
side1Edges= mdb.models['Model-1'].parts['Part-6'].edges.findAt(" + topline + "))"
exec(topfaceset)


### Set for Nodal Displacement Extraction ###
nodefordisp = "(( (Width*(xdir+1)-(xdir-1)/2*Thick1-Thick1/2) , H2 + (%d*2-
1)*(H1+H2) + Thick3, 0.0),)"%(ydir)
nodeline = "mdb.models['Model-1'].parts['Part-6'].Set(name='Node for disp', vertices=
mdb.models['Model-1'].parts['Part-6'].vertices.findAt(" + nodefordisp + "))"
exec(nodeline)

###---------------------------------SET CREATION ENDS----------------------------------###


### Step Creation ###
mdb.models['Model-1'].StaticStep(adaptiveDampingRatio=0.05,
   continueDampingFactors=True,initialInc=0.01, name='Step-1', nlgeom=ON,
previous='Initial',
   stabilizationMethod=DISSIPATED_ENERGY_FRACTION)


### Load Application ###
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
```

```
    distributionType=UNIFORM, field='', magnitude=Press[0], name='Load-1',
    region=
    mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].surfaces['Top surface'])


### BC Application ###
mdb.models['Model-1'].EncastreBC(createStepName='Step-1', localCsys=None, name=
    'Bottom fixed', region=
    mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].sets['Bottom face'])
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'Left face sliding', region=
    mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].sets['Left face'],
    u1=0.0, u2=UNSET, ur3=0.0)
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'Right face sliding', region=
    mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].sets['Right face']
    , u1=0.0, u2=UNSET, ur3=0.0)


### Field Output ###
mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].setValues(frequency=
    LAST_INCREMENT, rebar=EXCLUDE, region=
    mdb.models['Model-1'].rootAssembly.allInstances['Part-6-1'].sets['Node for disp']
    , sectionPoints=DEFAULT, variables=('UT', ))

mdb.models['Model-1'].FieldOutputRequest(name='F-Output-2',
    createStepName='Step-1', variables=('MISES', ), frequency=LAST_INCREMENT)

### Field Output for Interactive Use ###
# mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].setValues(variables=(
    # 'S', 'MISES', 'NE', 'LE', 'U', 'UT', 'RF', 'CF'))


### Job Submission for First Load Case ###
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='Cantiduo', nodalOutputPrecision=SINGLE,
    numCpus=1, numGPUs=0, queue=None, resultsFormat=ODB, scratch='', type=
    ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)

mdb.jobs['Cantiduo'].submit(consistencyChecking=OFF)
```

```
mdb.jobs['Cantiduo'].waitForCompletion()

o3 = session.openOdb(name='Cantiduo.odb')

session.viewports['Viewport: 1'].setValues(displayedObject=o3)

### Post-Processing ###
odb = session.odbs['Cantiduo.odb']
session.fieldReportOptions.setValues(printXYData=ON, printTotal=OFF,
printMinMax=OFF)
session.writeFieldReport(fileName='Output_file', append=OFF,
   sortItem='Node Label', odb=odb, step=0, frame=1, outputPosition=NODAL,
   variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))

session.fieldReportOptions.setValues(printXYData=OFF,printTotal=OFF,printMinMax=
ON)
session.writeFieldReport(
   fileName='Output_file',
   append=ON, sortItem='Element Label', odb=odb, step=0, frame=1,
   outputPosition=INTEGRATION_POINT, variable=(('S', INTEGRATION_POINT, ((
   INVARIANT, 'Mises'), )), ))


### Job Submission and Post-Processing for Rest of the Load-cases ###
for i in range(len(Press)-1):
 mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
   distributionType=UNIFORM, field='', magnitude=Press[i+1], name='Load-1',
   region=
   mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].surfaces['Top surface'])
 mdb.jobs['Cantiduo'].submit(consistencyChecking=OFF)
 mdb.jobs['Cantiduo'].waitForCompletion()


 o3 = session.openOdb(name='Cantiduo.odb')
 session.viewports['Viewport: 1'].setValues(displayedObject=o3)
 odb = session.odbs['Cantiduo.odb']
 session.fieldReportOptions.setValues(printXYData=ON, printTotal=OFF,
printMinMax=OFF)
 session.writeFieldReport(fileName='Output_file', append=ON,
   sortItem='Node Label', odb=odb, step=0, frame=1, outputPosition=NODAL,
   variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))
```

```
session.fieldReportOptions.setValues(printXYData=OFF,printTotal=OFF,printMinMax=
ON)
 session.writeFieldReport(
    fileName='Output_file',
    append=ON, sortItem='Element Label', odb=odb, step=0, frame=1,
    outputPosition=INTEGRATION_POINT, variable=(('S', INTEGRATION_POINT, ((
    INVARIANT, 'Mises'), )), ))
```

APPENDIX B. PYTHON SCRIPT FOR CANTI-OVAL DESIGN

```
### ABAQUS PYTHON INPUT SCRIPT FOR CANTI-OVAL DESIGN ###

### By Neehar Kulkarni (neehark@g.clemson.edu) ###
### Department of Mechanical Engineering, Clemson University ###

###----------------------------------------------------------------------------------------------###

# -*- coding: mbcs -*-
# session.journalOptions.setValues(replayGeometry=COORDINATE,
recoverGeometry=COORDINATE)

### Defining Design Variables ###
Width  = 0.02050  # Half Width of UC W
H1     = 0.00320  # Height of UC H
Gap    = 0.0002  # Gap g
Thick2 = 0.00117  # CB EFG Thickness t2
Thick4 = 0.00184  # ESG Thickness t3
Thick5 = 0.00111  # OB EFG Thickness t4
R1     = 0.01460  # OB Major Radius r1
R2     = 0.00040  # OB Minor Radius r2

Thick1 = 2*(Gap+Thick4)
Thick3 = H1
H2     = 0.000

BottomThickness = 0.0003 # Bottom Face Sheet Thickness Bot_T
TopThickness = 0.0017    # Top Face Sheet Thickness Top_T

Fillets = "NO" # If fillets are to be included in UC, type YES, else NO

# Fillet Radii
f1 = 0.00100
f2 = 0.00075
f3 = 0.00030
f4 = 0.00055

xdir = 3 # Number of Cells in x- direction
ydir = 3 # Number of Cells in y- direction

### Constitutive Material Properties ###
mat = 'Ti'  # Titanium Alloy
```

```
rho = 4820  # Density kg/m^3
E   = 102e9 # Young's Modulus N/m^2
v   = 0.32  # Poisson Ratio

Sectionthickness = 0.170  # Section Thickness

### Load Cases ###
Press = [0.3817e6, 0.8284e6, 2.0632e6, 3.9327e6]

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 *
from abaqusConstants import *

###----------------------------Meta-material Geometry Creation Begins-------------------###

### Half UC Geometry Construction in Abaqus CAE ###

mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)

#2
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, 0.0), point2=(
    Width-Gap-R1, 0.0))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[2])

#3
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, 0.0), point2=(
    0.0, -H1))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[3])

#4
```

```
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, -H1),
   point2=(Thick1, -H1))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
   addUndoState=False, entity=
   mdb.models['Model-1'].sketches['__profile__'].geometry[4])

#5
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Thick1, -Thick2),
   point2=(Thick1, -H1))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
   False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[5])

#6
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap, -(R2+Thick5)),
   point2=(Width-Gap, 0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
   False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[6])

#7
mdb.models['Model-1'].sketches['__profile__'].EllipseByCenterPerimeter(
   axisPoint1=(Width-Gap-R1, 0.0), axisPoint2=(Width-Gap, -R2), center=(Width-Gap,
0.0))

#9
mdb.models['Model-1'].sketches['__profile__'].autoTrimCurve(curve1=
   mdb.models['Model-1'].sketches['__profile__'].geometry[7], point1=(
   Width-Gap, R2))

#10
mdb.models['Model-1'].sketches['__profile__'].offset(distance=Thick5,
   objectList=(mdb.models['Model-1'].sketches['__profile__'].geometry[9],
),side=RIGHT)

#11
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Thick1, -Thick2),
        point2=(Thick1+1e-6, -Thick2))

#12
mdb.models['Model-1'].sketches['__profile__'].trimExtendCurve(curve1=
   mdb.models['Model-1'].sketches['__profile__'].geometry[11], curve2=
   mdb.models['Model-1'].sketches['__profile__'].geometry[10], point1=(
   Thick1+1e-6, -Thick2), point2=(Width-Gap-R1-Thick5,-1e-10))

#13
```

```
mdb.models['Model-1'].sketches['__profile__'].autoTrimCurve(curve1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[10], point1=(
    Width-Gap-R1-Thick5,-1e-10))

#14
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap-Thick4,
0+Thick3),
    point2=(Width, 0+Thick3))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[14])

#15
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap, 0),
    point2=(Width, 0))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[15])

#16
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width, 0),
    point2=(Width, 0+Thick3))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[16])

#17
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(Width-Gap-Thick4,
0+Thick3),
    point2=(Width-Gap-Thick4, -Thick5/8))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[17])

#18
mdb.models['Model-1'].sketches['__profile__'].trimExtendCurve(curve1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[17], curve2=
    mdb.models['Model-1'].sketches['__profile__'].geometry[9], point1=(
    Width-Gap-Thick4, -Thick5/8), point2=(Width-Gap-Thick4, -R2))

#19
mdb.models['Model-1'].sketches['__profile__'].autoTrimCurve(curve1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[9], point1=(
    Width-Gap-1e-10,-R2))

mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, 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__']


### Mirroring Half UC Geometry to form Entire UC ###
mdb.models['Model-1'].parts['Part-1'].DatumPlaneByPrincipalPlane(offset=Width,
    principalPlane=YZPLANE)
mdb.models['Model-1'].parts['Part-1'].Mirror(keepOriginal=ON, mirrorPlane=
    mdb.models['Model-1'].parts['Part-1'].datums[2])
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'])

### Fillets Creation in UC ###

if (Fillets == "YES"):
 mdb.models['Model-1'].ConstrainedSketch(name='__edit__', objectToCopy=
    mdb.models['Model-1'].parts['Part-1'].features['Shell planar-1'].sketch)
 mdb.models['Model-1'].parts['Part-1'].projectReferencesOntoSketch(filter=
    COPLANAR_EDGES, sketch=mdb.models['Model-1'].sketches['__edit__'],
    upToFeature=
    mdb.models['Model-1'].parts['Part-1'].features['Shell planar-1'])
 mdb.models['Model-1'].sketches['__edit__'].FilletByRadius(curve1=
    mdb.models['Model-1'].sketches['__edit__'].geometry[13], curve2=
    mdb.models['Model-1'].sketches['__edit__'].geometry[6], nearPoint1=(
    Width-Gap-1e-5, -(R2+Thick5-1e-15)), nearPoint2=(Width-Gap, -(R2+Thick5-1e-
10)), radius=f1)
 mdb.models['Model-1'].sketches['__edit__'].FilletByRadius(curve1=
    mdb.models['Model-1'].sketches['__edit__'].geometry[5], curve2=
    mdb.models['Model-1'].sketches['__edit__'].geometry[12], nearPoint1=(
    Thick1, -Thick2-1e-5), nearPoint2=(Thick1+1e-5, -Thick2), radius=f2)
 mdb.models['Model-1'].sketches['__edit__'].FilletByRadius(curve1=
    mdb.models['Model-1'].sketches['__edit__'].geometry[2], curve2=
    mdb.models['Model-1'].sketches['__edit__'].geometry[19], nearPoint1=(
    Width-Gap-R1-1e-5, 0), nearPoint2=(
    Width-Gap-R1+1e-5, -1e-5), radius=f3)
 mdb.models['Model-1'].sketches['__edit__'].FilletByRadius(curve1=
    mdb.models['Model-1'].sketches['__edit__'].geometry[19], curve2=
    mdb.models['Model-1'].sketches['__edit__'].geometry[18], nearPoint1=(
    Width-Gap-Thick4-1e-10, -R2+1e-5), nearPoint2=(Width-Gap-Thick4, -R2+1e-3),
radius=f4)
 mdb.models['Model-1'].parts['Part-1'].features['Shell planar-1'].setValues(
```

```
    sketch=mdb.models['Model-1'].sketches['__edit__'])
 del mdb.models['Model-1'].sketches['__edit__']
 mdb.models['Model-1'].parts['Part-1'].regenerate()


### Single Tessellation of UC in y- direction with Half UC Width Shift ###
mdb.models['Model-1'].rootAssembly.LinearInstancePattern(direction1=(1.0, 0.0,
    0.0), direction2=(0.0, 1.0, 0.0), instanceList=('Part-1-1', ), number1=1,
    number2=2, spacing1=0, spacing2=H1+H2)
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-1-1-lin-1-2',
    ), vector=(Width-Thick1/2, 0.0, 0.0))


### Complete Tessellation of UCs in x- and y- direction ###
final = list()

for i in range(xdir):
 for j in range(ydir):
  if (j == 0 and i == 0):
   continue
  newline = "mdb.models['Model-1'].rootAssembly.instances['Part-2-1-lin-%d-%d'],"
%(i+1,j+1)
  final.append(newline)

finalline = "".join(final)

mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,
    instances=(mdb.models['Model-1'].rootAssembly.instances['Part-1-1'],
    mdb.models['Model-1'].rootAssembly.instances['Part-1-1-lin-1-2']), name=
    'Part-2', originalInstances=SUPPRESS)
mdb.models['Model-1'].rootAssembly.LinearInstancePattern(direction1=(1.0, 0.0,
    0.0), direction2=(0.0, 1.0, 0.0), instanceList=('Part-2-1', ), number1=xdir,
    number2=ydir, spacing1=Width*2-Thick1, spacing2=2*(H1+H2))
lastline = "mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,instances=(mdb.m
odels['Model-1'].rootAssembly.instances['Part-2-1'],"+ finalline +"),name='Part-3',
originalInstances=SUPPRESS)"
exec(lastline)


#### Top Face Sheet Creation ####
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(
```

```
    decimalPlaces=3)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(Width*2*xdir-(xdir-1)*Thick1-Thick1/2+Width, TopThickness))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-4', type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-4'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])


#### Bottom Face Sheet Creation ####
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(
    decimalPlaces=3)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(Width*2*xdir-(xdir-1)*Thick1-Thick1/2+Width, BottomThickness))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-5', type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-5'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])


### Small Part Creation for Side Edges of Tessellated Meta-material ###
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.05)
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(
    decimalPlaces=3)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.0, 0.0),
    point2=(Thick1/2, Thick3))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Small Part',
type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Small Part'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])


### Assembly of Small Part ###
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Small Part-1',
    part=mdb.models['Model-1'].parts['Small Part'])
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Small Part-1', ),
    vector=(0, (H1+H2)*(ydir*2-1)+H2, 0.0))


### Assembly of Top Face Sheet ###
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-4-1',
    part=mdb.models['Model-1'].parts['Part-4'])
```

```
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-4-1', ),
    vector=(0, (H1+H2)*(ydir*2-1)+H2+Thick3-TopThickness, 0.0))

### Assembly of Bottom Face Sheet ###
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-5-1',
    part=mdb.models['Model-1'].parts['Part-5'])
mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-5-1', ),
    vector=(0, -(H1+BottomThickness), 0.0))


### Creation and Assembly of Half UCs for Meta-material Edges ###
mdb.models['Model-1'].Part(name='Part-1-Copy', objectToCopy=
    mdb.models['Model-1'].parts['Part-1'])
del mdb.models['Model-1'].parts['Part-1-Copy'].features['Mirror-1']
mdb.models['Model-1'].Part(compressFeatureList=ON, mirrorPlane=YZPLANE, name=
    'Part-1-Copy-Copy', objectToCopy=
    mdb.models['Model-1'].parts['Part-1-Copy'])

for i in range(ydir):
 instanceline = "mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-Copy-%d',part=mdb.models['Model-1'].parts['Part-1-Copy'])"%(i+1)
 exec(instanceline)

for j in range(ydir):
 instanceline2 = "mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-Copy-Copy-%d',part=mdb.models['Model-1'].parts['Part-1-Copy-
Copy'])"%(j+1)
 exec(instanceline2)

for ii in range(ydir):
 translation = "mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-1-
Copy-%d',),vector=(xdir*2*Width-xdir*Thick1, %d*2*(H1+H2), 0.0))"%(ii+1,ii)
 exec(translation)

for jj in range(ydir):
 translation2 = "mdb.models['Model-1'].rootAssembly.translate(instanceList=('Part-1-
Copy-Copy-%d',),vector=(Width+Thick1/2, H1+%d*2*(H1+H2), 0.0))"%(jj+1,jj)
 exec(translation2)

appendline = list()

for ij in range(ydir):
 instanceappend = "mdb.models['Model-1'].rootAssembly.instances['Part-1-Copy-%d'],"
%(ij+1)
```

```
 appendline.append(instanceappend)

for ji in range(ydir):
 if ji== ydir-1:
  instanceappend2 = "mdb.models['Model-1'].rootAssembly.instances['Part-1-Copy-
Copy-%d'])," %(ji+1)
 else:
  instanceappend2 = "mdb.models['Model-1'].rootAssembly.instances['Part-1-Copy-
Copy-%d']," %(ji+1)
 appendline.append(instanceappend2)

finalappend = "".join(appendline)

mergeline = "mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,instances=(mdb.m
odels['Model-1'].rootAssembly.instances['Part-3-1'],mdb.models['Model-
1'].rootAssembly.instances['Part-4-1'],mdb.models['Model-
1'].rootAssembly.instances['Part-5-1'],mdb.models['Model-
1'].rootAssembly.instances['Small Part-1'],"+finalappend+"name='Part-6',
originalInstances=SUPPRESS)"
exec(mergeline)

###-------------------------Meta-material Geometry Creation Ends-------------------------###


#### Material Assignment ####
mdb.models['Model-1'].Material(name=mat)
mdb.models['Model-1'].materials[mat].Density(table=((rho, ), ))
mdb.models['Model-1'].materials[mat].Elastic(table=((E, v),
    ))


### Section Assignment ###
mdb.models['Model-1'].HomogeneousSolidSection(material=mat, name=
    'Section-1', thickness=Sectionthickness)
mdb.models['Model-1'].parts['Part-6'].Set(faces=
    mdb.models['Model-1'].parts['Part-6'].faces.findAt(((0.0, 0.0, 0.0),
    )), name='Entire body set')
mdb.models['Model-1'].parts['Part-6'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-1'].parts['Part-6'].sets['Entire body set'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
```

```
### Mesh Generation ###
mdb.models['Model-1'].parts['Part-6'].seedPart(deviationFactor=0.1,
    minSizeFactor=0.1, size=(Thick5)/6)
mdb.models['Model-1'].parts['Part-6'].setElementType(elemTypes=(ElemType(
    elemCode=CPE4R, elemLibrary=STANDARD, secondOrderAccuracy=OFF,
    hourglassControl=DEFAULT, distortionControl=DEFAULT), ElemType(
    elemCode=CPE3, elemLibrary=STANDARD)), regions=(
    mdb.models['Model-1'].parts['Part-6'].faces.findAt(((0.0, 0.0, 0.0),
    )), ))
mdb.models['Model-1'].parts['Part-6'].generateMesh()


###-------------------------------SET CREATION BEGINS----------------------------------###

#### Bottom Face Set ####
bottom = list()
bottomlinefirst = "((2*Width, -H1-BottomThickness, 0.0),),"
bottom.append(bottomlinefirst)

bottomline="".join(bottom)

bottomfaceset = "mdb.models['Model-1'].parts['Part-6'].Set(edges=mdb.models['Model-
1'].parts['Part-6'].edges.findAt(" + bottomline + " ), name='Bottom face')"
exec(bottomfaceset)


#### Left face set creation ####
left = list()
for i in range(ydir+3):
 if i == 0:
  partline_left = "((0.0 , -(H1)/2, 0.0),),"
 elif i == ydir+1:
  partline_left = "((0 , (%d*2-1)*(H1+H2)+H2+Thick3-TopThickness/2, 0.0),),"%(ydir)
 elif i == ydir+2:
  partline_left = "((0.0 , -H1-BottomThickness/2,0.0),),"
 else:
  partline_left = "((0.0, (%d*2-1)*(H1+H2) + H2 + H1/3, 0.0),),"%(i)
 left.append(partline_left)

leftline = "".join(left)
leftfaceset = "mdb.models['Model-1'].parts['Part-6'].Set(edges=mdb.models['Model-
1'].parts['Part-6'].edges.findAt(" + leftline + " ), name='Left face') "
exec(leftfaceset)
```

```
### Right Face Set ###
right = list()
for i in range(ydir+1):

 if i == ydir:
  partline_right = "((Width*2*%d + Width - (%d-1)*Thick1 - Thick1/2 , H2 + (%d*2-
1)*(H1+H2) + Thick3 - TopThickness/2, 0.0),),"%(xdir,xdir,i)
 else:
  partline_right = "((Width*2*%d + Width - (%d-1)*Thick1 - Thick1/2, H2 +
2*%d*(H1+H2) + H1/2, 0.0),),"%(xdir,xdir,i)
 right.append(partline_right)

rightline = "".join(right)
rightfaceset = "mdb.models['Model-1'].parts['Part-6'].Set(edges=mdb.models['Model-
1'].parts['Part-6'].edges.findAt(" + rightline + " ), name='Right face') "
exec(rightfaceset)


### Top Face Set ###
top = list()
for i in range(xdir+1):
 small_first = "(( Thick1/4 + %d*(Width*2-Thick1) , H2 + (%d*2-1)*(H1+H2) + Thick3,
0.0),),"%(i,ydir)
 small_second= "(( 3*Thick1/4 + %d*(Width*2-Thick1) , H2 + (%d*2-1)*(H1+H2) +
Thick3, 0.0),),"%(i,ydir)
 mid_part = "(( Width-1e-3 + %d*(Width*2-Thick1) , H2 + (%d*2-1)*(H1+H2) +
Thick3, 0.0),),"%(i,ydir)
 top.append(small_first)
 top.append(small_second)
 top.append(mid_part)

topline = "".join(top)
topfaceset = "mdb.models['Model-1'].parts['Part-6'].Surface(name='Top surface',
side1Edges= mdb.models['Model-1'].parts['Part-6'].edges.findAt(" + topline + "))"
exec(topfaceset)


### Set for Nodal Displacement Extraction ###
if xdir%2 == 0:
 nodefordisp = "(( (Width*(xdir)-(xdir/2-1)*Thick1-Thick1/2 , H2 + (ydir*2-1)*(H1+H2)
+ Thick3, 0.0)),)"
else:
```

```
 nodefordisp = "(( (Width*(xdir+1)-(xdir-1)/2*Thick1-Thick1/2) , H2 + (%d*2-
1)*(H1+H2) + Thick3, 0.0),)"%(ydir)

nodeline = "mdb.models['Model-1'].parts['Part-6'].Set(name='Node for disp', vertices=
mdb.models['Model-1'].parts['Part-6'].vertices.findAt(" + nodefordisp + "))"
exec(nodeline)

###----------------------------------SET CREATION ENDS--------------------------------###


### Step Creation ###
mdb.models['Model-1'].StaticStep(adaptiveDampingRatio=0.05,
   continueDampingFactors=True,initialInc=0.01, name='Step-1', nlgeom=ON,
previous='Initial',
   stabilizationMethod=DISSIPATED_ENERGY_FRACTION)


### Load Application ###
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
   distributionType=UNIFORM, field='', magnitude=Press[0], name='Load-1',
   region=
   mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].surfaces['Top surface'])


### BC Application ###
mdb.models['Model-1'].EncastreBC(createStepName='Step-1', localCsys=None, name=
   'Bottom fixed', region=
   mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].sets['Bottom face'])
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
   distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
   'Left face sliding', region=
   mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].sets['Left face'],
   u1=0.0, u2=UNSET, ur3=0.0)
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
   distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
   'Right face sliding', region=
   mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].sets['Right face']
   , u1=0.0, u2=UNSET, ur3=0.0)


### Field Output ###
mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].setValues(frequency=
   LAST_INCREMENT, rebar=EXCLUDE, region=
   mdb.models['Model-1'].rootAssembly.allInstances['Part-6-1'].sets['Node for disp']
```

```
    , sectionPoints=DEFAULT, variables=('UT', ))

mdb.models['Model-1'].FieldOutputRequest(name='F-Output-2',
    createStepName='Step-1', variables=('MISES', ), frequency=LAST_INCREMENT)

### Field Output for Interactive Use ###
# mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].setValues(variables=(
    # 'S', 'MISES', 'NE', 'LE', 'U', 'UT', 'RF', 'CF'))

### Job Submission for First Load Case ###
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='Job-1', nodalOutputPrecision=SINGLE,
    numCpus=1, numGPUs=0, queue=None, resultsFormat=ODB, scratch='', type=
    ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)

mdb.jobs['Job-1'].submit(consistencyChecking=OFF)
mdb.jobs['Job-1'].waitForCompletion()

#mdb.saveAs(pathName='C:\Users\Neehar K\Desktop\FALL 2015\Research Fall
2015\Cantioval optimization\MultiObjective\cantioval.cae')
o3 = session.openOdb(name='Job-1.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o3)

### Post-Processing ###
odb = session.odbs['Job-1.odb']

session.fieldReportOptions.setValues(printXYData=ON, printTotal=OFF,
printMinMax=OFF)
session.writeFieldReport(fileName='output_file', append=OFF,
    sortItem='Node Label', odb=odb, step=0, frame=1, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))

session.fieldReportOptions.setValues(printXYData=OFF,printTotal=OFF,printMinMax=
ON)
session.writeFieldReport(
    fileName='output_file',
    append=ON, sortItem='Element Label', odb=odb, step=0, frame=1,
    outputPosition=INTEGRATION_POINT, variable=(('S', INTEGRATION_POINT, ((
    INVARIANT, 'Mises'), )), ))


### Job Submission and Post-Processing for Rest of the Load-cases ###
```

```
for i in range(len(Press)-1):
 mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, field='', magnitude=Press[i+1], name='Load-1',
    region=
    mdb.models['Model-1'].rootAssembly.instances['Part-6-1'].surfaces['Top surface'])
 mdb.jobs['Job-1'].submit(consistencyChecking=OFF)
 mdb.jobs['Job-1'].waitForCompletion()

 o3 = session.openOdb(name='Job-1.odb')
 session.viewports['Viewport: 1'].setValues(displayedObject=o3)
 odb = session.odbs['Job-1.odb']
 session.fieldReportOptions.setValues(printXYData=ON, printTotal=OFF,
printMinMax=OFF)
 session.writeFieldReport(fileName='output_file', append=ON,
    sortItem='Node Label', odb=odb, step=0, frame=1, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))

session.fieldReportOptions.setValues(printXYData=OFF,printTotal=OFF,printMinMax=
ON)
 session.writeFieldReport(
    fileName='output_file',
    append=ON, sortItem='Element Label', odb=odb, step=0, frame=1,
    outputPosition=INTEGRATION_POINT, variable=(('S', INTEGRATION_POINT, ((
    INVARIANT, 'Mises'), )), ))
```