DESIGN OPTIMIZATION AND SIMULATION OF DRY FRICTION CLUTCH PLATE

Project Report submitted to

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
MECHANICAL ENGINEERING

Submitted by

ANANTHU R MENON (MUT18ME014)

JOHN T ANIL(MUT18ME037)

OLICKAL JOHNSON FRED(MUT18ME049)

SIDDHARTH S NAIR(MUT18ME056)

Muthoot
Institute of Technology & Science

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL BOARD OF ACCREDITATION

(Validity June 2019 to June 2022)

MUTHOOT INSTITUTE OF TECHNOLOGY AND SCIENCE
VARIKOLI P.O, PUTHENCRUZ- 682308
JUNE 2022

MUTHOOT INSTITUTE OF TECHNOLOGY AND SCIENCE
VARIKOLI P.O, PUTHENCRUZ- 682308
DEPARTMENT OF MECHANICAL ENGINEERING

(Validity June 2019 to June 2022)

CERTIFICATE

This is to certify that the report entitled ***"Design Optimization and Simulation of Dry Friction Clutch Plate"*** *is a bonafide record of project work done by* ***"Ananthu R Menon (MUT18ME014), John T Anil (MUT18ME037), Olickal Johnson Fred (MUT18ME049), Siddharth S Nair (MUT18ME056)"*** *during the year 2021- 2022. This report is submitted to APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Mechanical Engineering.*

Dr. Pradeepmon T G
Associate Professor (Internal Guide)
Dept. of Mechanical Engineering

Dr. Manoj Kumar K
Associate Professor and
Head of Dept. of
Mechanical Engineering

ACKNOWLEDGEMENTS

First of all, we would like to thank GOD ALMIGHTY for giving us the opportunity to excel in our efforts to complete this project on time. We express our sincere gratitude to everyone who helped at the completion of our project.

We extend our sincere gratitude to Dr Neelakantan P.C, Principal, Muthoot Institute of Technology and Science for giving us an opportunity for doing the project.

To Prof.Dr. Manoj Kumar, Head of Mechanical Engineering Department, for his valuable support and encouragement during the work.

We wholeheartedly thank our project guide Asst. Prof. Dr Pradeepmon T. G, for his valuable advice and support in carrying out this project.

We would also like to thank Dr Praveen K M, Assistant professor, Dept. of Mechanical Engineering, and Mr Rony Thomas Murickan, Assistant professor, Dept. of Mechanical Engineering for their instructions, co-ordinations and suggestions for our project. We convey our sincere thanks to all other faculties in the department for their support and encouragement. We thank all our friends who helped us during this work. Once again, we convey our gratitude to all those people who directly or indirectly influenced our work.

ABSTRACT

The clutch is a critical component in the proper operation of automobiles. It is subjected to high pressure, torsional, and temperature loads, causing the friction lining of the clutch plate to wear out over time. The friction lining of the clutch plate has been chosen as the machine part to be optimized. Our goal is to reduce the mass of the friction plate and heat generation while increasing torque capacity. We have identified the inner and outer radius of the friction clutch plate as the design variables. To carry out this optimization process, we have chosen the PyMoo framework, a python-based programming framework that can give us the fundamental structure of an algorithm to find the best values for our design variables.

The geometric models of traditional and optimized friction clutch plates are created using Solidworks. Afterward, finite element analysis is done using ANSYS. The outcomes of the analysis are compared and evaluated.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

NSGA-II – Non Dominated Sorting Genetic Algorithm II

GA – Genetic Algorithm

FEA – Finite Element Analysis

SOEA – Single Objective Evolutionary Algorithm

PSO – Particle Swarm Optimization

q – Heat generated

μ – Coefficient of friction

p – Contact pressure

v – velocity of slip

T_c – Torque Capacity

W – Load applied

R1 – Outer radius of friction plate

R2 – Inner radius of friction plate

m – mass of friction plate

ρ – density

P – Power

CHAPTER 1
INTRODUCTION

Optimization is a very essential part of engineering design that enables engineers to develop more efficient and cost-effective parts. Optimization can be done using algorithms that are developed for the sole purpose of finding the most suitable solutions to our problems. An optimization algorithm is an iterative procedure to determine the most optimal solution to a maximizing or minimizing problem by searching an n-dimensional space of values defined by the range of values that the design variables can take. It involves restricting the search space in n-dimensions and finding the most viable solution within those bounds or constraints. The most viable solutions are then compared with each other and the most suitable solution can be selected from these solutions accordingly.

The friction clutch plate of a clutch assembly is the part which is in direct contact with the flywheel. It is coupled to the flywheel by the force acting on the friction plate by the pressure plate. The pressure plate exerts pressure by the use of a spring arrangement, which allows the force to be removed when a pedal is engaged. When the clutch is engaged, the rotating flywheel imparts frictional force to the friction plate, which causes some energy to be lost as heat. Our objective is to reduce this heat generated. But the main objective of the clutch assembly is to transfer torque from the flywheel to the transmission. In order to make sure that the principle objective is tackled effectively, we are also optimizing the torque capacity of the friction clutch plate.

1.1 SEQUENCE OF WORK

In this project, we have aimed to use optimizing algorithms to find the most optimal design variable for a dry friction clutch plate, which is predominantly used in automobiles to transfer torque from the engine flywheel to the transmission components. The design functions we have taken into consideration are, heat generated during slipping, the torque that can be transferred, and the mass of the friction plate. Since higher heat generation leads to lower efficiency, we minimize the amount of heat generated. Torque capacity is maximized since the clutch should be able to carry the entire load produced by the flywheel. The mass of the friction plate is minimized so that we can achieve weight and material reduction. The algorithm used is the Non-Dominated Solution Genetic Algorithm (NSGA-II), which is a multi-objective optimization algorithm derived from the single objective Genetic Algorithm (GA).

After obtaining the optimal design variables, we check the results by modeling the friction plate and then performing Finite-Element Analysis (FEA) on the new model. The models are created using SOLIDWORKS, and FEA analysis is done using ANSYS. The results of FEA are then used to compare the Von Mistres stresses of the traditional model and the new optimized model.

CHAPTER 2
LITERATURE REVIEW

The paper "**Multi-objective optimization of a spring diaphragm clutch on an automobile based on the non-dominated sorting genetic algorithm (NSGA-II)**" aims to optimize the average change in compression force of the spring and force of separation. Optimization process is carried out using the penalty function, Genetic Algorithm (GA) and a modified GA for multi-objective optimization problems called Non-dominated Sorting Genetic Algorithm (NSGA-II). From the results, it is concluded that the NSGA-II algorithm fairs better in almost all aspects of optimization since it is able to simultaneously consider both the objectives at the same time unlike the rest [1].

The paper "**Performance optimization of flywheel motor by using NSGA-2 and AKMMP"** discusses the application of NSGA-II and a new surrogate model called Adaptive Kriging Model based on Maximum Projection design to optimize the mass and torque density of flywheel motors. The result so obtained is found to be more efficient than existing designs with lower chances of errors recorded during the process [2].

The paper "**pymoo: Multi-objective optimization in python**" introduces a new python framework designed with multi-objective optimization as its main focus. It provides visualization tools, decision making controls and customization ability to help with the process of programming optimization algorithms. It also provides performance metrics that help us identify the convergence of solutions [3].

The paper "**A fast and elitist multi-objective genetic algorithm: NSGA-II**" introduces the new multi-objective algorithm for optimization. It lists the changes from the previous version of the algorithm which makes the new algorithm much faster and accurate. NSGA-II is then verified by solving some benchmark optimization problems by this algorithm [4].

The NSGA-II is introduced and developed with controlled elitism and dynamic crowding distance in the paper "**Improved non-dominated sorting genetic algorithm (NSGA)-II in multi-objective optimization studies of wind turbine**

blades.". The maximum power coefficient and the minimum blade mass are the optimization objectives for this method when applied to wind turbine blade design. When dealing with multi-objective, multi-variable, and multi-constraint optimization problems, this algorithm performs well in terms of convergence and robustness, giving a broad framework for multi-objective optimization design of wind turbines. Meanwhile, the optimization findings show that rather than a single optimum solution, which is typically achieved from conventional multi-objective optimizations, the Pareto-optimal solution set should be obtained. This presents a new concept for wind turbine multi-objective optimization design. Finally, a 5 MW wind turbine blade is designed using the enhanced NSGA-II [5].

The paper "**Multi-objective design optimization of three-phase induction motor using NSGA-II algorithm**" seeks to develop a MO that combines the NSGA-II method for power density minimization and efficiency maximization in three phases Squirrel Cage Induction Motor using nonlinear constrained optimization techniques. In order to tackle the multi-objective optimization problem of electric motor driving in a parametric manner, the Pareto-optimization technique is applied. It yields a set of ideal options from which the designer can select a suitable balance design depending on his or her preferences. In addition, to compare among Pareto-optimal solutions, various SOEA techniques such as Simulated Annealing, Tabu Search, and Genetic Algorithm are used. Simulation studies have been used to assess their performance using criteria such as Delta, Convergence, and Spacing. The results after optimization shows that When compared to the SA, TS, and GA findings from the initial model, the efficiency increases by 80% and the power density increases by 12 kW/kg following NSGA-II optimization. The NSGA-II performance measures provide the best feasible Pareto solutions [6].

In the paper "**Optimization of Shape and Design Parameters of the Rigid Clutch Disc Using FEM**", modal and steady state analysis were performed on a rigid clutch disc. These analyses were carried out in four design models out of which two are suggested models and rest are models present in the industry. Physical copies of the models are also created using 3D printing technology. It is found that the suggested models have lower mass and lower maximum stresses built up in it[7].

The paper “**Design Optimization of Friction Lining of a Clutch Plate**” aims to optimize the material used in manufacturing of friction clutch plates. The clutch plate of a TATA 609 CNG truck is taken and modeled in solidworks, after which FEA was conducted for heat, vibration and forces acting on the friction plate.It is seen that the removal of the outer diameter as a constraint helps significantly improve the heat conduction also helps reduce vibrations. The new materials used were also evaluated, which were found to have better frictional resistance[8].

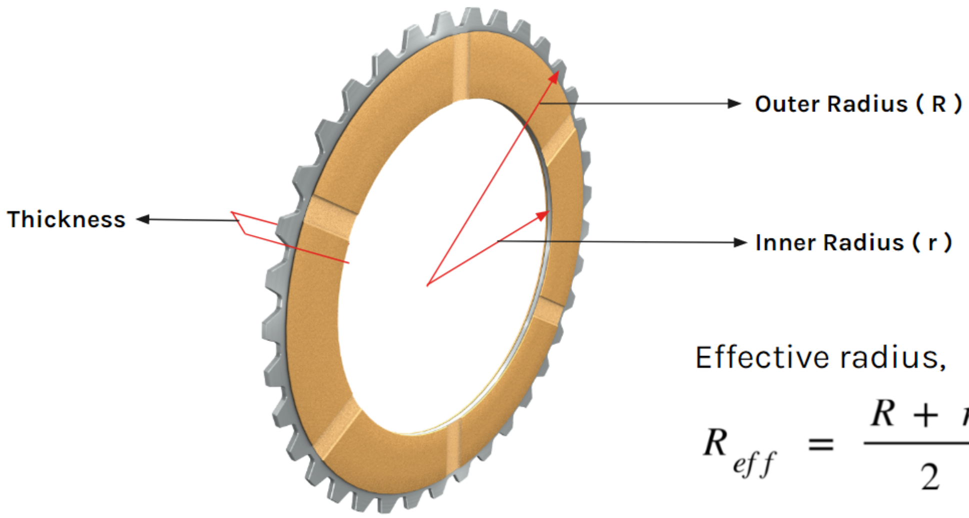
The numerical analysis and FEA of clutch is carried out in the paper “**Design and Structural Analysis of Single Plate Friction Clutch**”. The paper evaluates the design of the clutch of TATA IDITC 475 engine. The clutch wear using uniform pressure theory and uniform wear theory are found numerically and them FEA is done to find the maximum principal stress acting on the clutch plate. All the objectives are verified and favorable values are obtained[9].

In the paper “**Heat generation and Transfer in Automotive Dry Clutch Engagement**”, the heat transfer and generation of dry clutches are studied. The study investigates the frictional lining, and its material for frictional resistance and thermal properties in two cases. The two cases being a new clutch and a worn out clutch. The material properties are studied experimentally and details about its friction coefficient and microstructure are also discussed[10].

CHAPTER 3

OBJECTIVES OF PROJECT

The design optimization of the friction clutch plate requires us to select some design parameters and identify the design variables for creating new models. These parameters and variables are used to define the optimization problem, which can be considered as the first major step in the design optimization process. This optimization problem is to be coded in some programming language to continue the process. Which is then further used to deploy the algorithm and find viable solutions.



$$R_{eff} = \frac{R + r}{2}$$

Fig 3.1

Design variables

In order to define optimization problem, we first identified the design parameters, which are:

Heat generated, $q = \mu * p * v_s$

Torque capacity, $T_c = ½ * \mu * W * (R_1 + R_2)$

Pressure, $p = W / A$

Mass, $m = \rho * \pi * (R_1^2 - R_2^2)$

The inner diameter and outer diameter of the friction clutch plate is chosen as the design variables since they are the geometric properties we will be able to modify without changes to material used. Since our objective is to minimize mass and heat generation, and maximize torque capacity, we can make use of the corresponding equations to write the objective functions as:

To maximize,

$$T_c = \frac{1}{2} * \mu * W * (R_1 + R_2)$$

$$m = \rho * \pi * (R_1^2 - R_2^2)$$

To minimize,

$$q = \mu * p * v_s$$

In order to get an optimum value for the design variables, we substitute values for all known parameters into the above equations. The values were obtained from the paper Design and Structural Analysis of Single Plate Friction Clutch authored by Mr. Vishal J. Deshbhratar and Mr. Nagnath U. Kakde[9]. The values used are actual parametric values found theoretically for a TATA 475 IDITC engine. The traditional clutch plate used in this engine has an outer diameter of 230mm and inner diameter of 200mm. Using this value, we can adjust our boundary condition for the optimization process near this traditional value. We have taken a range of 180mm to 210mm for inner diameter and 225mm to 250mm for outer diameter. After substituting known values, we got the equations as,

$$h = 49417.61 / (D_2 - D_1)^2$$

$$T_c = 843.75 * (D_1+D_2)$$

$$m = 0.01866 * (D_2^2 - D_1^2)$$

CHAPTER 4

DESIGN OPTIMIZATION

In order to get the variable values for optimized friction clutch plates, we need to use some search algorithm. A search algorithm is used to find the best solution to a problem, by identifying its convergence to some optimum value. A variety of such algorithms are present which are of many types. Some classifications are meta-heuristic algorithms, single objective algorithms, multi objective algorithms, single variable and multiple variable algorithms. These algorithms can either be tailored to one specific problem or can be used for a particular type of problem. Algorithms which are made to work on a variety of problems without any prior requirement on the type of problems, are called meta-heuristic algorithms. In our case, since we need to optimize three objective functions and two design variables, we chose a meta-heuristic multi-objective algorithm called NSGA-II.

4.1 NSGA-II

Non-dominated Sorting Genetic Algorithm – II (NSGA-II) is a modified version of GA which predominantly focuses on multi-objective optimization problems. It improves upon the traditional GA by sorting the solution according to their fitness of the individual and then sorting them. Tournament selection is then carried out from each group so that mutations of the fittest functions occur, thereby allowing us to obtain better solutions.

4.1.1 Principle

In this algorithm, every possible solution is determined to be non-dominant or not by comparing every solution with each other. Then the non-dominant solutions are placed in a front containing all the non-dominant solutions. The rest of the solutions are then grouped into fronts according to the value of non-dominants each solution has. After grouping the non-dominants, the value of every non-dominant not assigned to the front is subtracted by one. Then the solutions with no dominants are grouped to a second front. This process is repeated till all solutions are in fronts. These fronts are assigned to different levels according to the dominants present for each front, so that the top-level front will have the non-dominant solutions. Ultimately, this reduces the complexity of the algorithm, making it more resource efficient. In order to preserve the diversity of the search area, the earlier sharing parameter used in NSGA is replaced with density estimation and a crowd comparison operator [4].

In order to get an estimate of the density of the solutions near a solution, we have to calculate the average distance between two solutions on either side of the selected solution along each objective. In order for this crowding distance computation to work, we need to sort the population of solutions according to each objective function for each of the objective functions. After all this distance is computed for all solutions, we compare two solutions for their closeness to the other solutions. This is what we compare using a crowd comparison operator.

The crowd comparison operator helps to guide the algorithm towards a uniform pareto-optimal front. In order for this to work, two attributes of the individual are made use of. They are the non-domination rank and crowding distance explained above. Using the rank, we can select the individual with the lower rank during comparison. If both the individuals under comparison have the same rank, then the crowding distance is used as the secondary attribute to decide which individual is selected. Here, we prefer the solution from a less crowded region.

4.1.2 Working

A random parent is initially created. This population is then sorted according to the non-dominance. Each solution obtained is provided with a rank or fitness according to the non-dominants present. Then, an offspring population is created using normal GA rules, i.e binary tournament selection, recombination and mutation [5].

The population containing both parent and offspring is then selected and sorted according to nondomination. Due to the presence of parent and offspring generation, elitism is maintained here. Then only the fittest front is considered to produce the new generation. If the population in the first front is insufficient to produce the new generation, the second front is considered and so on. So, in this procedure, the weaker individuals are slowly sorted out and only the best solutions remain. The binary tournament selection occurring in these selections make use of crowd density and rank to ensure elitism is maintained.

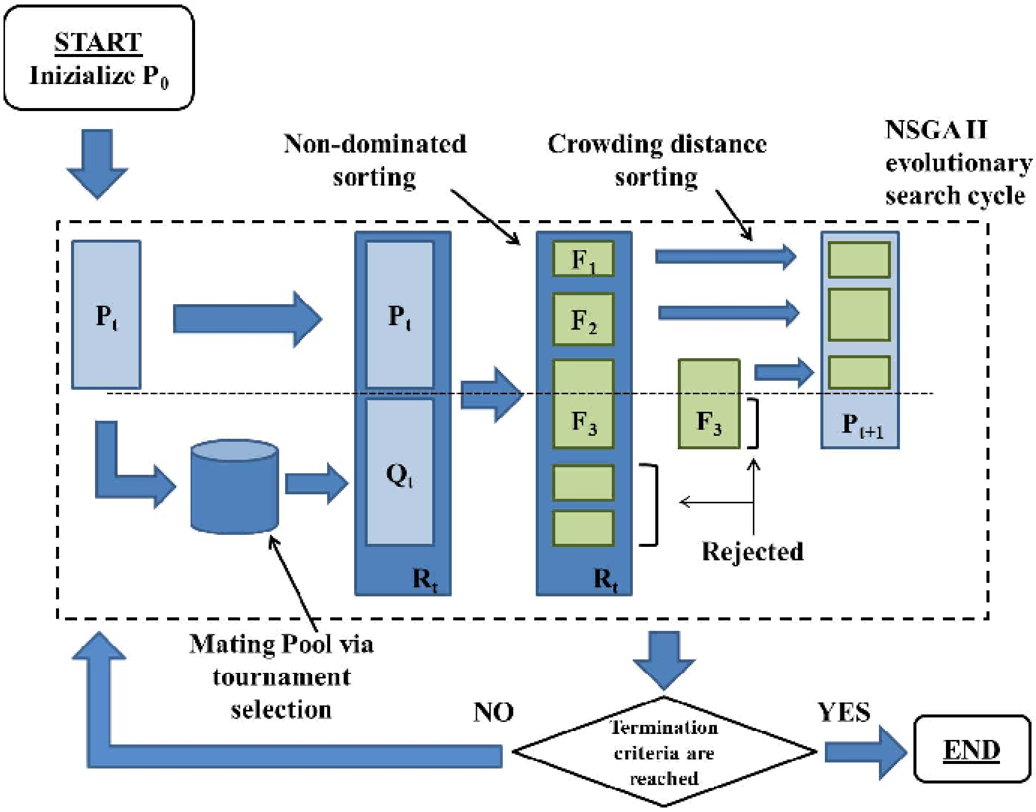


Fig 4.1 NSGA-II working

Source: Applying NSGA-II for solving the Ontology Alignment Problem

4.2 PYMOO FRAMEWORK

Pymoo is a python framework created with multi-objective optimization algorithms in mind. It provides tools and functions for executing optimization algorithms in python. A framework is a collection of packages or modules that allow developers to create applications faster without worrying about low level details.

Pymoo is coded in such a way that only minimization problems can be tackled using this framework. So, for maximization problems, we have to multiply the entire objective function with -1. Pymoo provides modules for input parameters, display options, results etc., which are some of the basic requirements for every optimization problem [3].

Pymoo also provides the structure for some well-known algorithms such as Particle Swarm Algorithm (PSO), Genetic Algorithm(GA) etc. This combined with the high customization abilities provided, we are able to modify algorithms based on our requirements. The various visualization techniques available in pymoo help us display the graph and plots of the optimization functions and convergence graphs without needing to write code for them from the base itself. Performance indicators embedded in pymoo helps us to attain the convergence of solutions and define fitness functions for the optimization problem.

4.3 METHODOLOGY

In order to optimize our problem using pymoo, we have to first define our problem as a class. This problem class will contain the equations for the objective functions, constraints and some other user defined values and limits. The user defined values include the number of variables present in the objective functions, the number of objective functions that are to be optimized, the number of constraints provided and the limits for each defined design variable. The objective function equations are then stored in a Numpy column stack. So, each objective function of the class will have its own values for coefficients and design variables.

After problem definition, we invoke the algorithm which is pre-coded into the pymoo framework. We can implement the algorithm by importing the required algorithm and then passing the defined problem class on to the algorithm. While invoking the algorithm, we provide some values that determine the iterations of the algorithm. In the case of NSGA-II, we pass the values for population size, number of offsprings per generation, the sampling method for initial selection of population, crossover function that determines the mating probability of parent solutions and mutation function that improves the performance of mutation in offsprings.

Now that the iteration process has started, we need a criteria to end the iteration, in order to do just that, a function called termination is called that stops the iteration at forty generations.

Now that algorithm and problem are defined, we can minimize the problem by importing the minimize function and invoking it using the defined algorithm, problem and termination criteria. When the minimize function is called, the iterations are run and all the solutions obtained are stored into new classes. The results are split into two classes, one for objective function and one for the design variables.

With the solutions at hand, we can now visualize the data and choose the optimal solutions by normalization.

4.4 OPTIMIZATION RESULTS

Optimization using the pymoo framework yields its results as a class containing the design parameter values and design variable values. In order to interpret this result in an easily understandable form, we make use of graphs and tables. This can be done using visualization methods present in pymoo itself.

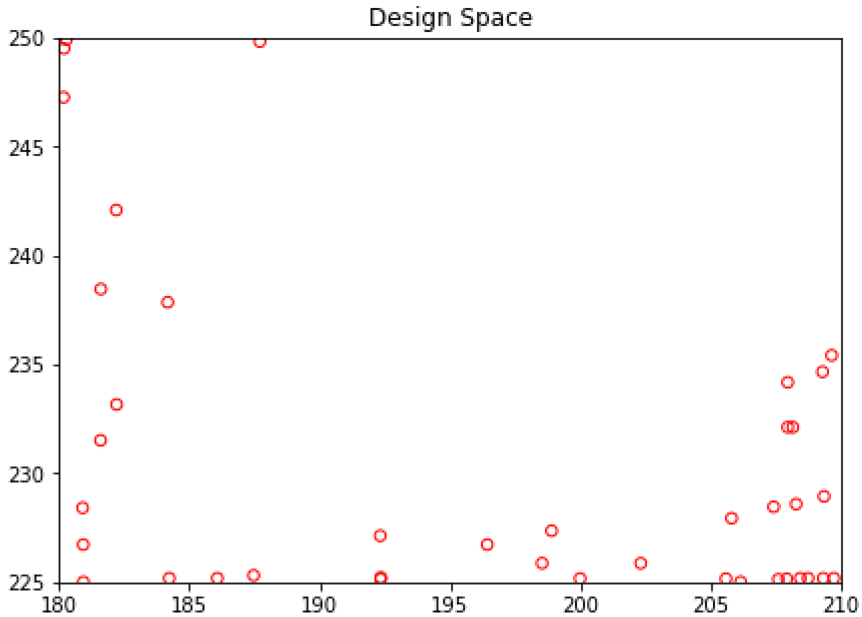


Fig 4.2 Design Space

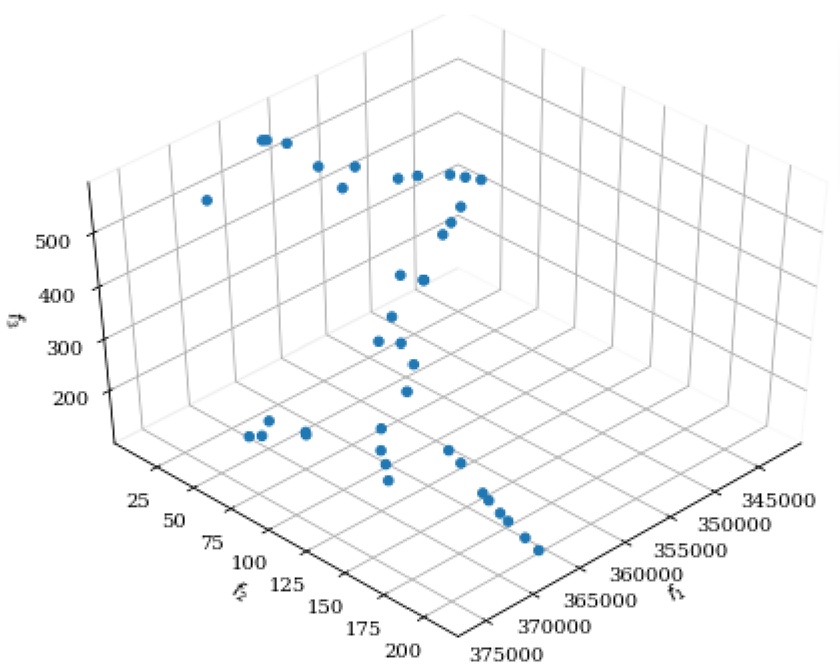


Fig 4.3 Objective space

In our case, we used a population size of forty and ten offspring in one generation and a total of forty generations. So, this will provide us with a search space of two hundred solutions. From these solutions, we can identify the best possible solution by providing weights to each objective function and then normalizing the results. In pymoo, weights are defined as an array with percent weight ordered respectively with objective functions. Since, we concentrate more on minimizing heat generation, a weight of 0.4 is given to the appropriate objective function and weights for torque capacity and mass is given as 0.3.

With weights given, compromise programming is done to apply the decomposition function to our problem, which breaks down the complex functions into easier chunks. This is done to obtain the results efficiently and to retrieve the best result from the solution class. This enables us to mark the best result in the objective space graph shown above (Fig 3.3) and also retrieve the objective function values and our required design variable values. In order to make the graph more readable, we can normalize the axes of the graph.

```
Best regarding ASF: Point
i = 23
F = [3.52255583e+05 4.59753814e+01 2.55407671e+02]
X = [192.35143549 225.13666276]
<pymoo.visualization.scatter.Scatter at 0x7f94f44f4ed0>
```

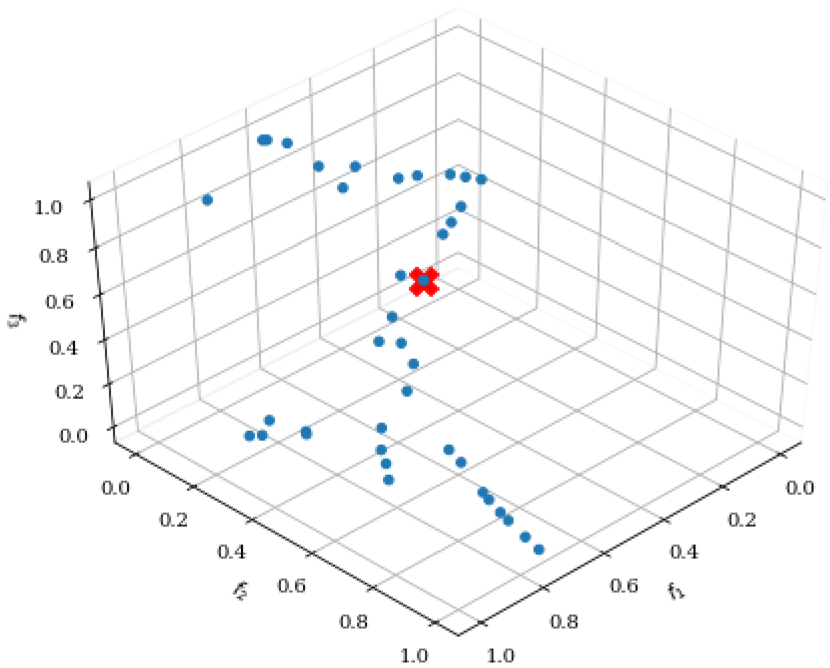


Fig 4.4

Normalized Objective Space and Best Solution

The above graph shows the normalized graph and the best solution is marked with a red cross. The corresponding values of design variables and parameters are given above the figure. The optimum value for design variables are obtained as 192.35mm for inner diameter and 225.13mm for outer diameter. The corresponding values for objective functions are also obtained as 0.352255W of heat generated, 459.7Nm of torque and mass of 255g.

CHAPTER 5
MODELING AND ANALYSIS

Now that we have obtained our optimized results for the inner and outer diameters of the friction clutch plate, we model the original design and the new obtained design in a CAD software and perform structural analysis on it in order to verify our results and find new gaps in the new design.

5.1 MODELING

The geometry of the friction clutch plate is a relatively simple design. It consists of a circular ring with six holes, evenly spaced, along it. These are provided for heat dissipation and also to fasten this friction lining to the pressure plate of the clutch assembly. The CAD models of the designs are modeled in a CAD software, SOLIDWORKS. The original model is designed with an internal diameter of 200mm and external diameter of 230mm, while the optimized model is designed with the design variable values obtained from optimization, which are, 192.35mm for internal diameter and 225.13mm for external diameter. Both the designs have six holes of 6mm diameter, evenly spaced out.

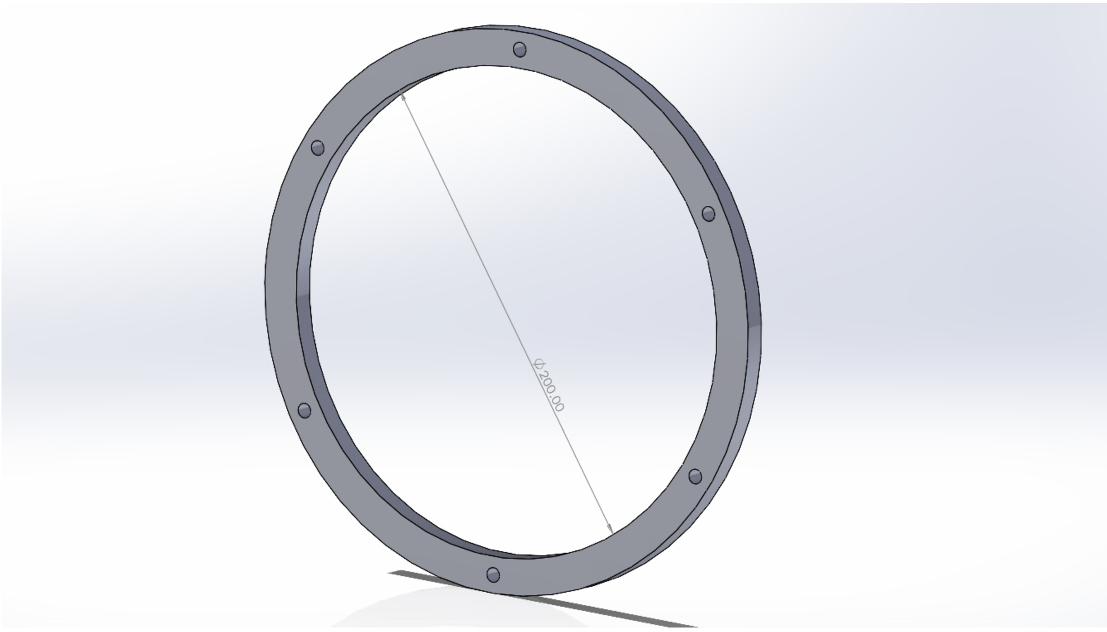


Fig 5.1

Original Truck Friction Clutch Plate

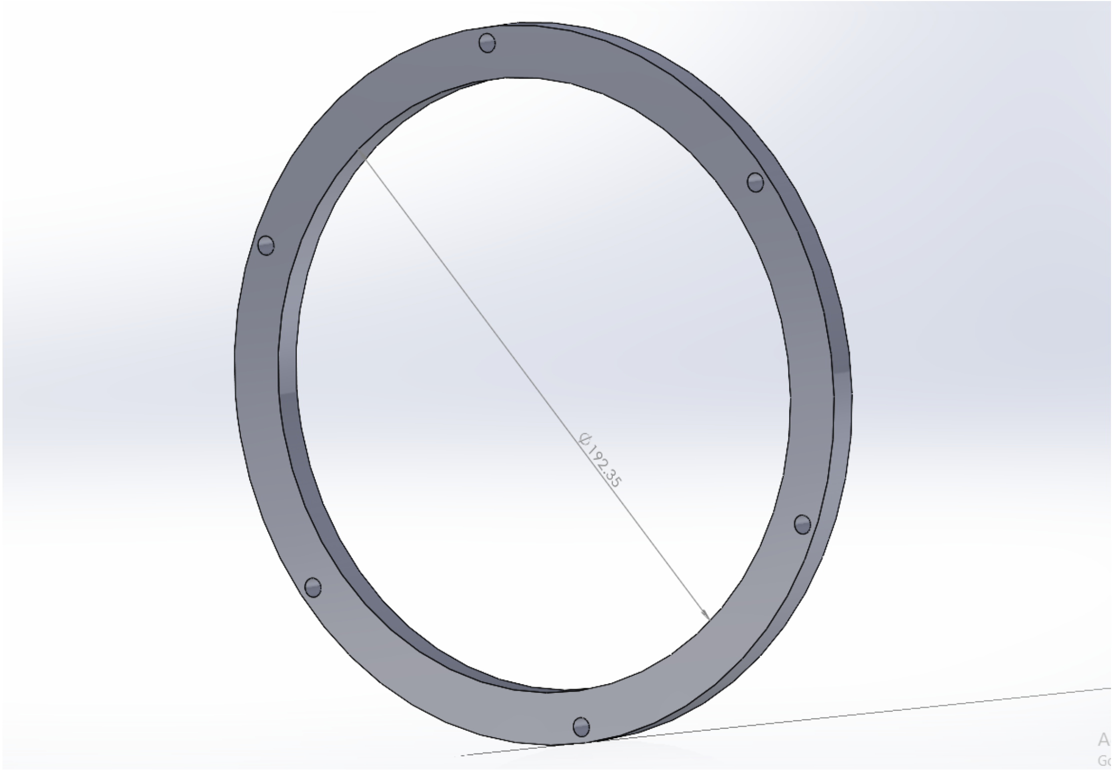


Fig 5.2

Optimized Friction Clutch Plate

5.2 STRUCTURAL ANALYSIS

With the modeling of both designs complete, we now perform Finite Element Analysis (FEA) on these designs to compare the structural characteristics of the designs. FEA is a method of dividing a surface or body into a finite number of elements and then computing the value of forces or other properties acting on each element, in order to simulate the working of the design under some specified condition. This is done with the help of multiple equations and calculations which are computed by the system processor. With the advent of high computing processors and ability to load large amounts of data into system memory, the simulation results obtained through FEA can be used to verify the performance and characteristics of the input designs. The software used for performing FEA is the ANSYS workbench.

In order to compare our optimized design with original design, we choose two parameters to be analyzed using FEA. The first analysis was done on both the models to find the amount of deformation that will be caused by a static pressure of 1MPa acting on the models. The second analysis was done to figure out the equivalent stresses or the Von Mistres stress along both the designs under the same load condition of 1MPa.

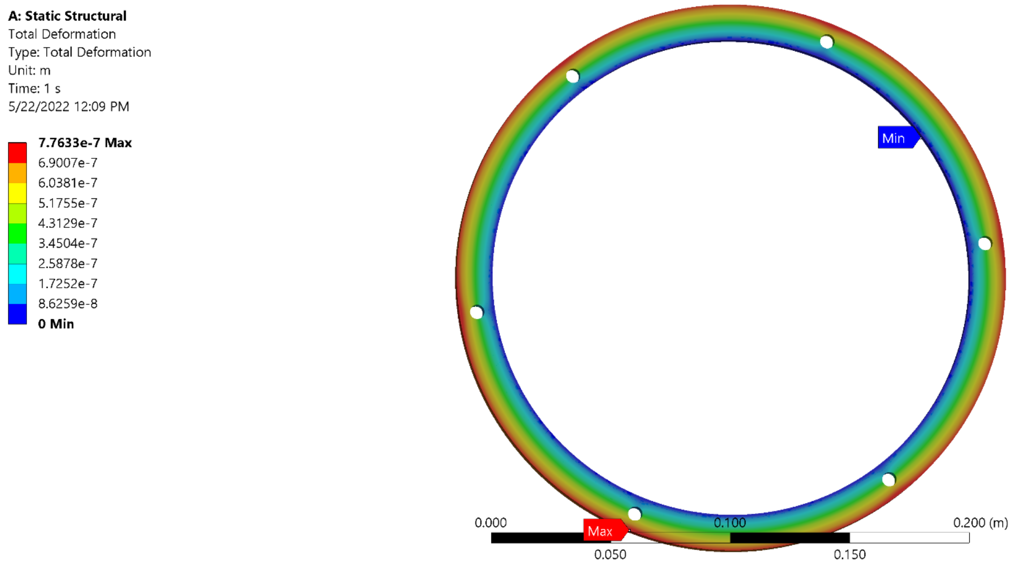


Fig 5.3

Total Deformation of Original Design

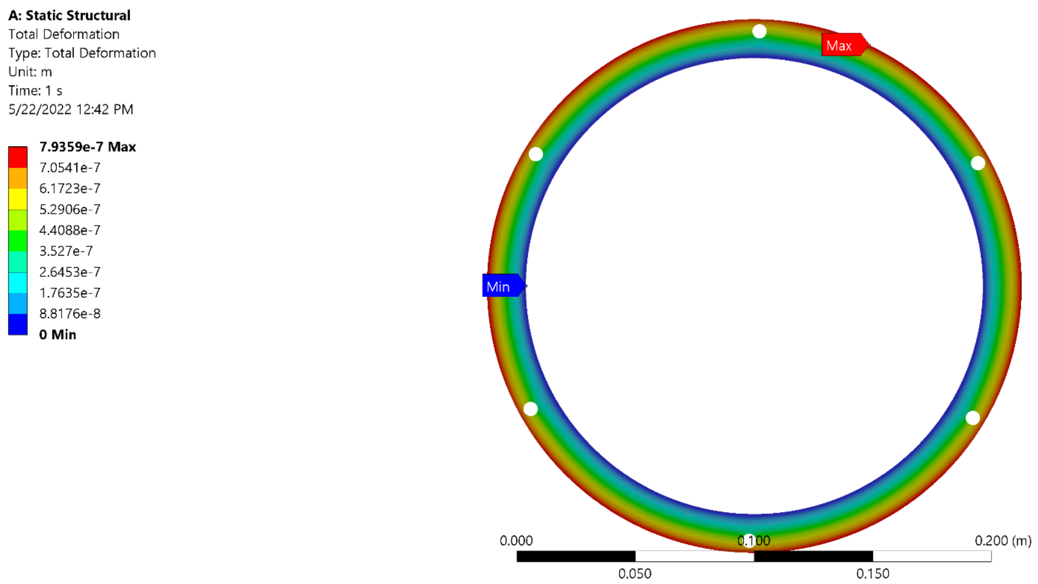


Fig 5.4

Total Deformation on Optimized Design

From the above two figures (Fig 5.3 and Fig 5.4), we can see that the maximum deformation of the optimized design has a slightly higher value than that of the original design. This can be attributed to the fact that the new design has a higher amount of area between the external and internal diameters. Since one of our objectives was to reduce the mass of the friction clutch plate, we must understand that even though the new design has a worser value of deformation, it still is only marginally greater than the deformation of the actual design. So, for a lower amount of mass, we are able to get a deformation that is very close to the deformation seen in the actual design.

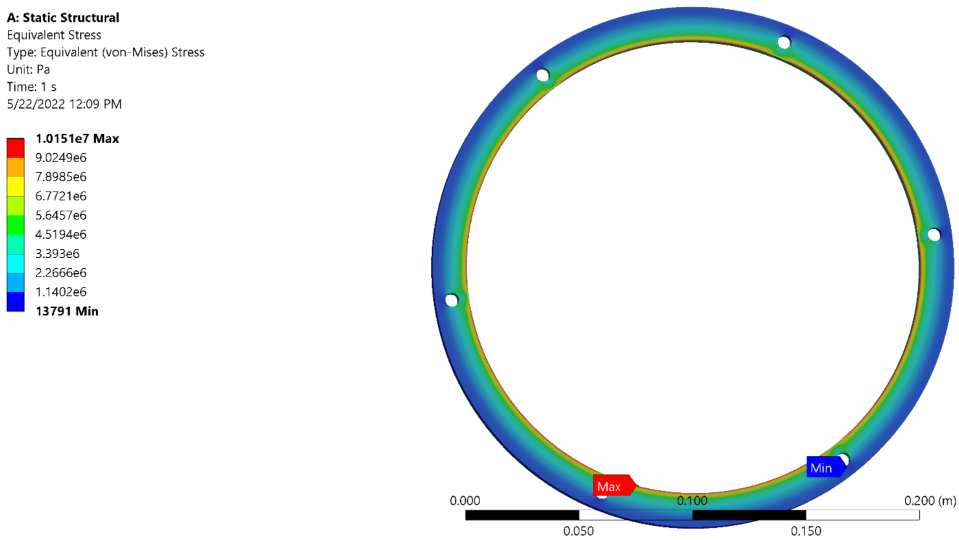


Fig 5.5

Equivalent Stress Formed in Original Design

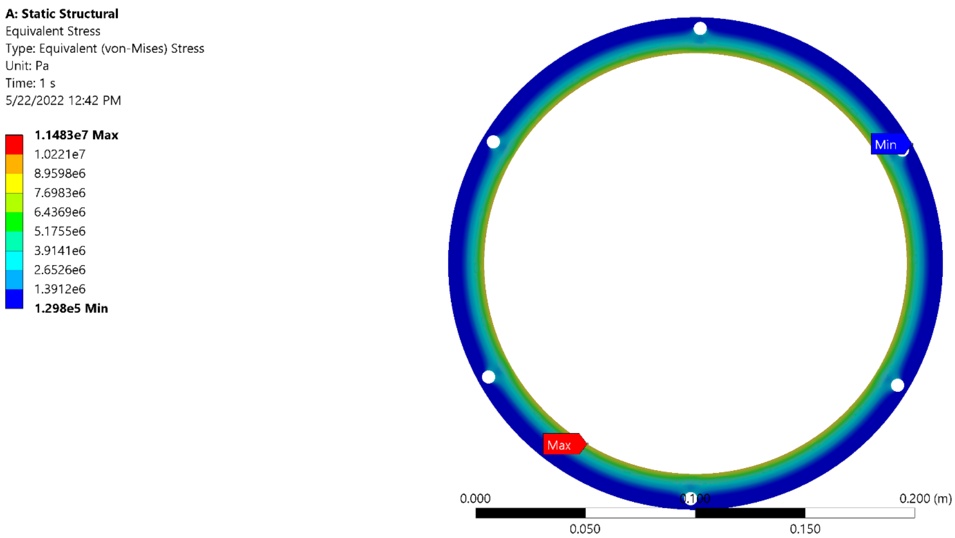


Fig 5.6

Equivalent Stress Formed in Optimized Design

In the above figures (Fig 5.5 and Fig 5.6), we can see the equivalent stresses formed in the original design and the optimized design. Same as in the case with total deformation, the maximum value of stress in the optimized design seems to be greater than the value seen in the original design. But, comparing the minimum values of stresses in both designs, we can see that the optimized design has a lower value than that of the original design. So, we can understand that the added stress has become more evenly distributed in case of the optimized design.

CHAPTER 6
RESULTS AND DISCUSSION

Through the optimization of the internal and external diameters of a friction clutch plate, we were able to improve some characteristics of the clutch. The optimization done in order to decrease the heat generation and mass, and increase torque capacity gave us satisfactory results. Our new design had a heat generation of 45.97J which is lower than the original design, which had generated heat of 54.9J with the same speed and time of slipping. This shows a 16% decrease in heat generated by the friction clutch plate while slipping.

The torque capacity of the new design is found to be 352.25Nm, which is lower than the torque capacity of the original design which could handle 363 Nm of torque. The mass of the new design is found to be 225g, while the original model had a mass of 240.7g. This shows a decrease of 6% of mass between the original and optimized models. The negative result for torque capacity can be due to the weightage given to its equation during the compromise programming.

The FEA results show us that the new design has a better stress distribution than that of the original design although the original design has a lower maximum value for the Von Mistres stresses.

CHAPTER 7
CONCLUSION

From the project work carried out, we have learnt to convert an engineering problem into an optimization problem. It has also enabled us to understand the various types of algorithms used to search for solutions. We could also understand how an optimization algorithm works, and how we can implement the same algorithm using python code.

We were able to convert the different parameters concerning the working of a dry friction clutch plate into an optimizing problem consisting of three objective functions. The optimized values so obtained were further used to model CAD designs and FEA was done on it. The optimization results were satisfactory and even though the FEA results were not entirely positive, the overall result is satisfactory.

This project has also helped us to become familiar with the characteristics and some parameters of the automotive clutch assembly.

CHAPTER 8
FUTURE WORK

Since we have only used one algorithm to optimize the friction clutch plate, we are not able to understand the suitability of application of NSGA-II algorithm to this particular problem. So, optimization of the same problem can be done using other multi-objective meta-heuristic algorithms. With this, we can also compare the different algorithms used and study the behavior of the algorithms with respect to the problems.

Another scope of expanding the project is through the thermal analysis of the optimized and original models.

REFERENCES

[1] Zhou, Junchao, Chun Wang, and Junjun Zhu. "Multi-objective optimization of a spring diaphragm clutch on an automobile based on the non-dominated sorting genetic algorithm (NSGA-II)." *Mathematical and Computational Applications* 21, no. 4 (2016): 47.

[2] Bu, Jian-Guo, Xu-Dong Lan, Ming Zhou, and Kai-Xiong Lv. "Performance optimization of flywheel motor by using NSGA-2 and AKMMP." IEEE Transactions on Magnetics 54, no. 6 (2018): 1-7.

[3] Blank, Julian, and Kalyanmoy Deb. "pymoo: Multi-objective optimization in python." IEEE Access 8 (2020): 89497-89509.

[4] Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation 6, no. 2 (2002): 182-197.

[5] Wang, Long, Tong-guang Wang, and Yuan Luo. "Improved non-dominated sorting genetic algorithm (NSGA)-II in multi-objective optimization studies of wind turbine blades." Applied Mathematics and Mechanics 32, no. 6 (2011): 739-748.

[6] Ranjan, Soumya, and Sudhansu Kumar Mishra. "Multi-objective design optimization of three-phase induction motor using NSGA-II algorithm." In Computational Intelligence in Data Mining-Volume 2, pp. 1-8. Springer, New Delhi, 2015.

[7] Abdullah, Oday Ibraheem, Josef Schlattmann, and Emir Pireci. "Optimization of shape and design parameters of the rigid clutch disc using FEM." *FME transactions* 41, no. 4 (2013): 317-324.

[8] Virmani, Kartik, Tanishq Madhogaria, and P. Baskar. "Design optimization of friction lining of a clutch plate." *Materials Today: Proceedings* 46 (2021): 8009-8024.

[9] Deshbhratar, Vishal J., and Nagnath U. Kakde. "Design and structural analysis of single plate friction clutch." *international journal of engineering research and technology* 2, no. 10 (2013).

[10] Gkinis, Theofilos, Ramin Rahmani, Homer Rahnejat, and Martin O'Mahony. "Heat generation and transfer in automotive dry clutch engagement." *Journal of Zhejiang University-SCIENCE A* 19, no. 3 (2018): 175-188.