COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

OPTIMAL DESIGN OF SERIAL LINK ROBOTIC MANIPULATORS VIA GENETIC ALGORITHMS

Gabriel Martins Franco Ramalho

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Mecânica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Mecânica.

Orientadores: Fernando Pereira Duda
Ramon Romankevicius Costa

Rio de Janeiro
Maio de 2019

OPTIMAL DESIGN OF SERIAL LINK ROBOTIC MANIPULATORS VIA GENETIC ALGORITHMS

Gabriel Martins Franco Ramalho

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA MECÂNICA.

Examinada por:

Prof. Fernando Pereira Duda, D.Sc.

Prof. Ramon Romankevicius Costa, D.Sc.

Prof. Amit Bhaya, Ph.D.

Prof. Daniel Alves Castello, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MAIO DE 2019

Ramalho, Gabriel Martins Franco

Optimal Design of Serial Link Robotic Manipulators via Genetic Algorithms/Gabriel Martins Franco Ramalho. – Rio de Janeiro: UFRJ/COPPE, 2019.

XIV, 87 p.: il.; $29,7$cm.

Orientadores: Fernando Pereira Duda
Ramon Romankevicius Costa

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Mecânica, 2019.

Bibliography: p. 83 – 87.

1. Robotics. 2. Optimization. 3. Hydroeletric. I. Duda, Fernando Pereira *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Mecânica. III. Título.

"...Você não fez mais que a sua obrigação."

-Ramalho, Angela

Acknowledgement

Without the help of others this thesis would not have been possible. I am immensely grateful to all that helped in so many ways.

First I would like to thank my teachers,Fernando Pereira Duda and Ramon Romankevicius Costa for there invaluable help throughout the entire thesis helping me with my doubts.

To Joao Monteiro and Rennan Freitas that would answer all my questions and help me with so my problems that appeared all through my work. Up to this day I still don't know how they don't hate me after annoying them every day.

To Matheus Di Vaio that suffered every single class with me and was with me no matter my mood, through darkness and through light. He that constantly suffered with my traitorous words "I need your help with this masters assignment, No Questions Asked" and would always do the work every single moment with a smile on his face. The ramens and insane dishes we invented have to continue!

To my Mother Angela that I don't give enough credit to and am always, always bothering although she doesn't deserve it. She, that stood by my side and helped me through every thing. The woman and person I look up to and is a symbol of strength that I truly love and admire even thought I will never say this directly to her. She is amazing and I cherish every moment with her and am grateful for her being in my life.

In memory of my Father Fernando that unfortunately couldn't be here in person to see me achieve one more step in my life. But I know that his is always with me in spirit. I know that this step specially makes him even more proud as he reverenced knowledge and to his final days he would do everything in his power to teach me what he could. Yes Father, you were a perfect dad I have absolutely nothing to complain about you.

To Thamiris Camara my betrothed and to whom I owe the smiles on my face every day. For the understanding that the countless hours awake away from bed at night where used and have culminated in to this thesis.

Finally to the two people that overall else are my Grandparents Joao and Cinira. They that treat me with so much warmth, caring and love that I cant expresse the joy I have to have them by my side.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROJETO ÓTIMO DE MANIPULADORES ROBÓTICO SERIAIS VIA ALGORITMOS GENÉTICOS

Gabriel Martins Franco Ramalho

Maio/2019

Orientadores: Fernando Pereira Duda
Ramon Romankevicius Costa

Programa: Engenharia Mecânica

Essa tese tem como proposta a criação de uma metodologia de projeto que otimiza manipuladores robóticos feitos para tarefas específicas e submetidos a restrições. Durante o desenvolvimento dessa metodologia um estudo de caso in situ será usado como base para as suposições e escolhas. Esse estudo de caso escolhido é o processo de revestimento a calor nas turbinas hidrelétricas de Jirau.

A metodologia criada utiliza o trabalho de Thomas Lambs como uma base e envolve a divisão do problema de otimização em duas partes: a otimização geométrica e a estrutural. As duas partes utilizaram da técnica de evolução diferencial acompanhado do método de otimização multivariável por peso.

A otimização geométrica utilizará o comprimento dos segmentos dos manipuladores, eixos de cada junta e coordenadas x, y e z da base do manipulador como os genes do método genético. Já a função multiobjetiva utilizará o comprimento, destreza, gradiente da destreza e área de trabalho útil para maximizar a função custo.

A otimização estrutural, por outro lado utilizará o diâmetro interno dos segmentos dos manipuladores como genes e otimizará o peso da estrutura e a deflexão do efetuador do manipulador.

Utilizando esse estudo de caso, o manipulador robótico foi otimizado e a metodologia testada com sucesso.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

OPTIMAL DESIGN OF SERIAL LINK ROBOTIC MANIPULATORS VIA GENETIC ALGORITHMS

Gabriel Martins Franco Ramalho

May/2019

Advisors: Fernando Pereira Duda
Ramon Romankevicius Costa

Department: Mechanical Engineering

This thesis proposes the creation of a project methodology that optimizes robotic manipulators for specific tasks and subjected to constraints. During the development of this methodology, a in situ case study will be used as a base for suppositions and chooses. This case study is the in situ hard-coating process of the Jirau hydroelectric dams blades.

The methodology created uses Thomas Lambs work as a base and involves dividing the optimization problem into two parts: a geometric optimization and a structural one. Both will use differential evolution in conjunction with a multi-objective weight method.

The geometric optimization uses the manipulator segments lengths, joint axis and x,y and z coordinates as gens and the multi-objective fitness function uses the length, dexterity, dexterity gradient and work area to maximize the cost.

The structural optimization, on the other hand, utilizes the inner diameter of the manipulator segments as gens and optimizes the weight of the structure and deflection of the end effector.

Using the case study, a robotic manipulator was optimized and the methodology tested with success.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In an energy-hungry society, each country looks at its natural resources to find ways to provide more energy for its population. These sources can generally be divided into two main categories: Renewable and fossil fuels. According to M. Pereira [13], the world has approximately 79% of its matrix in fossil fuel and only 18% of its matrix coming from renewable sources. This excessive use of fossil fuel has driven pollution to grow and become an alarming factor that endangers the world. This has driven most countries in a pursuit to transit its energy matrix from petroleum-based to clean and renewable energy sources.

As Brazil has an abundance of renewable energy sources that are economically viable, it has always focused its energy matrix on them. As an example, when it comes to electric energy, Brazil already has over 70% of its matrix being generated by hydroelectric plants [13].

While this renewable energy source is of invaluable help, its use translates to significant investments being made to expand the use of this energy source which, in turn, limits the diversification of the energy matrix [3]. The low diversity in energy sources tends to make any minor problems that can occur into a great pitfall as they favor the disruption of the entire energy distribution system. This makes it vital to analyze and fix every problem promptly. In hydroelectric power plants, especially plants on sediment filled rivers, one of the main problems is the extreme wear in the hydroelectric turbine blades caused by the impact of debris, sand abrasion, and the cavitation phenomena.

Cavitation is an event that occurs when a liquid is submitted to a dynamic-pressure reduction ate constant temperature that causes the liquids to, in a brief time in the order of a few nanoseconds, evaporate and create a vacuum cavity [14]. In flowing liquid, these cavities are subjected to a fluctuation of pressure that makes them collapse in an implosive manner damaging anything in its surrounding [1].

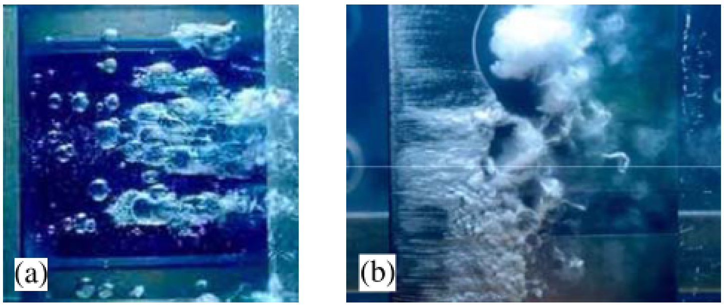


Figure 1.1: Types of cavitation: (a) travelling bubbles, (b) unstable attached sheet [1]

An obvious way to avoid premature wear of the turbine blades is to make the blades of a sturdy material where abrasion will not be a problem, and its mechanical strength will be higher than the cavitation forces [1]. The major drawbacks with this proposition are the difficulty in machining hard material and the high prices of particular materials, for example, the super duplex stainless steel series that have excellent mechanical proprieties but are extremely expensive [15].

To counter these drawbacks, thermal aspiration coating techniques can be used. These techniques consist of melting a hard material to a molten or semi-molten state that is then sprayed in high-velocity gas streams on to a soft metal substrate [2]. This process can be associated with coating an exposed outdoor surface with paint as a means of protection against the environment. This process allows the base material to be easy to manufacture, but still bestowing the structure with a hard coat finish that will lengthen its overall life. Another benefit of the method is the relatively small exposure of the base material to heated elements allowing the substrate material to maintain its structural proprieties as it is not subjected to thermal cycles.

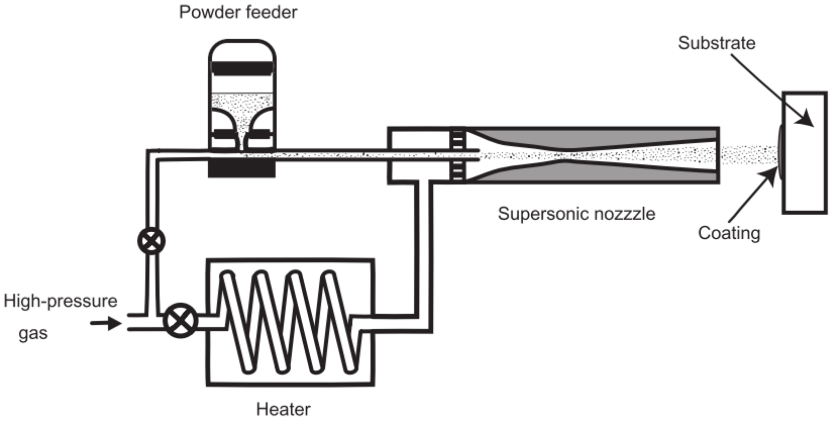


Figure 1.2: Cold sprat coating process example. [2]

Specifically in the hydroelectric turbine of Jirau, built in the Madeira river, the volume of debris brought by the water is large if compared to other plants and, consequently, the maintenance of each blade due to abrasion is high [3]. Before each blade was installed for the first time, it received a coating by the company Rijeza. The coating processes helped maintain the blades function up until now, but as time passes maintenance is required, thus causing a series of logistical problems. Up until the year of 2018, the maintenance consisted of a laborious process where each blade had to be removed, coated and placed back in its original location.

The coating processes demand a robotic arm for precision and costs more then R$320.00000 per turbine. While this cost can seem high, the biggest drawback is the disassembly, reassembly, and calibration of each turbine that ends up costing R$500.000,00 and can take up to 6 weeks [3]. As stated previously, when having an energy matrix that is highly dependent on hydroelectric energy, a six-week wait for maintenance can be detrimental to the entire country.

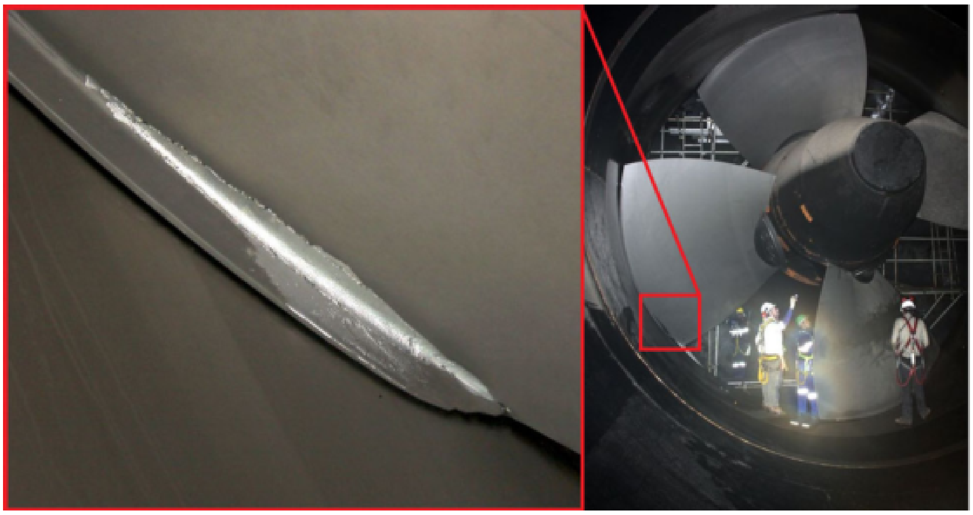


Figure 1.3: Erosion present in the hydroelectric turbine blade in Jirau [3]

1.2 Project EMMA

Project EMMA was created in a partnership between "ESBR - Energia Sustentável do Brasil" and LEAD laboratory of control and automation, engineering of application and development" of with the primary goal of reducing maintenance times for the hydroelectric turbine of Jirau. To accomplish this task, the team at LEAD decided to create, in situ, a robotic system capable of remotely applying the hardcoating process on each blade, and thus removing the need to disassemble, reassemble and calibrate each blade [3]. As the removal process is not needed this cost should be minimized or eliminated.

As stated previously, the turbine blades in the hydroelectric dam of Jirau, are subjected to cavitation and large amounts of derbies. This leads to wear that gradually alters the blade profile and reduces its efficiency and, in worst cases, augments the probability of mechanical failure [1]. To avoid this scenario, Rejiza, a company specialized in hard-coating processes utilizes the HVOF method to coat the surface of each blade.

The HVOF or "High-Velocity Oxygen Fuel" equipment is composed of: a spray gun where the feed material is partially melted and expelled; a feeder that takes, by pipe, a feed material, in powder form, from a reservoir; an electric power supply and finally a robotic manipulator [3].

The application of HOVF consists of feeding a specific hard material, dependent on the substrate metal and desired mechanical characteristics, to the combustion chamber situated in the spray gun[2]. In Jirau's specific case, the blade is made of stainless steel 420 and requires a tungsten carbide hard coat [3]. In the combustion chamber, a mixture of fuel and oxygen is injected and ignited. The hot gases almost instantaneously liquefy the feed material that is then expelled with pressures of up to 1Mpa from the nozzle [2].

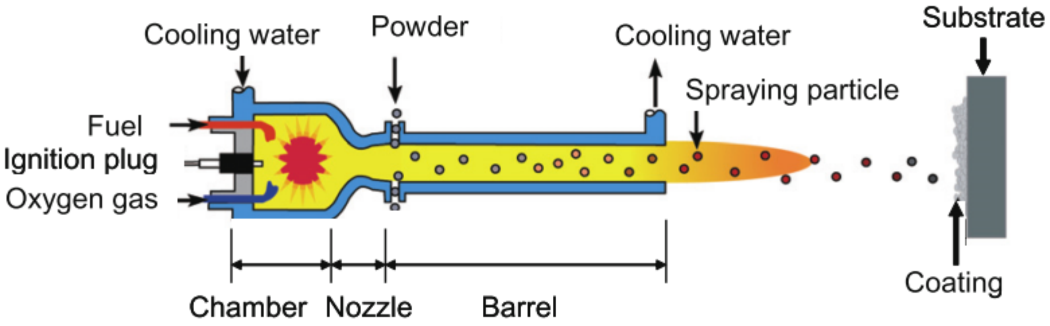


Figure 1.4: Cold sprat coating process example. [2]

The HVOF equipment weights 4kg, produce flames of up to 3000 ° C, a particle speeds of up to 1000 m/s and a recoil force of 15N. The spray has to be applied in regular coats with an accuracy of 3mm at a distance of 240 mm from the body that is being sprayed and at a 90 ° +- 60 ° angle in respect to the metallic surface plane [3].

To operate, this equipment is placed at the end of the robotic manipulator best suited for the size of the part to be hard coated, and a preplanned trajectory is then executed covering the desired structure. These robotic manipulators typically fall into two categories: Parallel and Serial types, where project EMMA utilizes, and this study will focus on the latter type.

Serial robots are classified by Pandilov [16] as "A robot is said to be a serial robot (fig.1.1 a) or serial (open-loop) manipulator if its kinematic structure takes the form of an open loop-chain". This means there is a linear sequence where each rigid body is composed of only one structure with joints in between each body similar to a human arm.

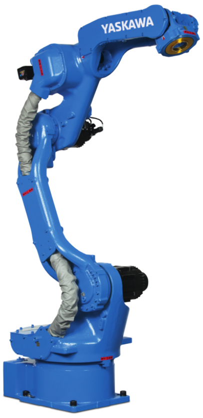


Figure 1.5: Example of serial type robotic manipulator chossen to be used in the in situ hard coating processes by the EMMA team. . [4]

When first analyzing the robotic manipulator to be used in project EMMA a series of industrial robots were considered, and robot MH-12 made by Motoman was chosen. This robot is a serial type robot with a 130kg weight, 1440mm range and a payload of 12kg [4]. This robot attended all primary criteria, where the most critical one was it could fit through an 80 cm hatch and had the most extended reach of the possible industrial manipulators. Although it presented many qualities, it still had a drawback of only being able to paint a small section of the blade at a time. In the following figure it is possible to see the numerous divisions that needed to be done to allow the robotic manipulator mh-12 to paint the entire hydroelectric turbine blade.

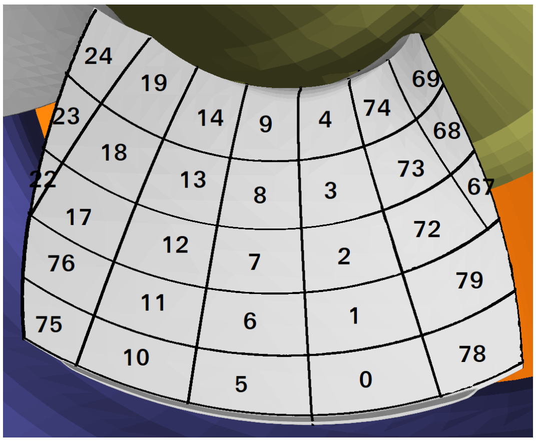


Figure 1.6: Division of the hydroelectric turbine blade for initial operation of the mh-12 robotic manipulator. The divisions are numbered using a special sequence used by the LEAD that helps indicate special requirements of each segment.

It is also possible to see a simple simulation of the robot doing the hard coating process in one of these divisions.

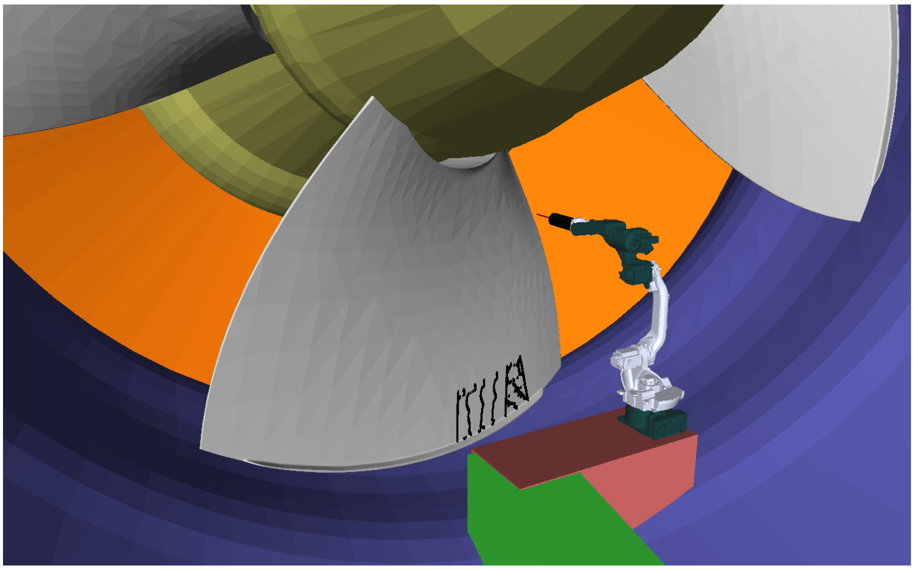


Figure 1.7: Example of serial manipulator hard coating a division of the turbine blade.

As the MH-12 is an industrial robot that has to serve in a variety of applications, it is not optimized for the hard-coating of the blades. In turn, this leads to inefficient coating times and higher costs as after each coated section the robot has to be moved and calibrated and every hour that the turbine is non-operational, thousands of Reais are lost.

As an extension of project EMMA, this thesis will expand on the robotic manipulators used and create a methodology using genetic algorithms that are proficient in designing an initial concept of an optimized robot capable of hard coating an entire side of a turbine blade. It needs to eliminate the previously described drawback of not reaching the entire turbine blade of the hydroelectric dam, while still being able to pass through the previously described hatch. This will theoretically speed the coating process leading to smaller maintenance times as it eliminates the need to move the robot so many times. The manipulator created at the end of this job will follow Lambs work [17] and will present an optimized initial solution that will not have all details such as bolt placement and detailed technical drawings that will have to be done subsequently by a team of engineers.

1.3 Objective

This thesis has the objective of creating a project methodology with the resources that will enable the development of an optimized robotic manipulator capable of reaching a designated distribution of points while still conforming to geometric and structural restrictions. While the method should be robust enough to be used in all cases, the specific problem presented is the case study of a hard-coating process done in situ, where there are ambient size restrictions, for example, a hatch where the manipulator has to pass through of 800mm in diameter. Other restrictions are the need to be as lightweight as possible for transportation while still being capable of handling the forces of the hard-coating process. The final objective is the capability of hard coating as much as possible of a hydroelectric turbine blade from one position while having the smallest size possible. As stated previously, the manipulator created at the end of this job will follow Lambs work [17] and will present an optimized initial solution that serves as a guide line and will not present all details such as bolt placement and detailed technical drawings that will have to be done subsequently by a team of engineers.

1.4 Literature Review

This section will review optimization techniques that can be used on robotic manipulators to optimize its form and structure. These techniques will be divided into two main groups: geometric and structural optimizations, which are composed respectively, of meta-heuristic and Parametric type optimization techniques.

The first category has, as the main objective, optimize the geometric structure of the robot, in other words, the lengths of each rigid body, in what plane a joint turns in respect with the previous joint and the overall (X, Y, Z) positioning of the base of the robot. This will be done by using an objective function that contains the overall length, dexterity, dexterity gradient and hard coating capacity.

The second type of optimization will be used to optimize the ridged structure of an already optimized robot geometry. The boundary conditions used will stimulate a reduction of weight while, maintain a minimum rigidity.

1.4.1 Meta-heuristic Optimization

Meta-heuristic based optimization is a class of methods created to attack real-world problems that traditional models like linear programming, dynamic, and non-linear programming could not cope with. These are problems with nonconvex and non-differentiable objective functions, objective functions that have many local maxima/minima [18] [19] [20]. As the present problem exhibits these traits of nonconvex with many local maximums and minimums, this class of method was chosen over the traditional, gradient-based methods.

Simulated Annealing

This method was motivated from the idea of annealing in solids where, if the material was cooled at high velocity the atoms would remain fixed in place with little time to move. In contrast, if the material were cooled slowly, the atoms would tend to achieve the state of minimum energy [21]. From a programming point of view, the objective is to find a global minimum, and the first operation is to choose numbers arbitrarily. They then are summed or subtracted a small random amount. In every cycle, the new number is applied in the function and compared with the previous result. If the new result smaller it becomes the new number, but if not a new set of evaluation happens where there is still a chance of it being accepted[21]. The most significant limitation of this technique is the number of generations that have to occur to achieve convergence.

Swarm Optimization

Swarm optimization is part of a group of techniques called swarm intelligence and was invented based on insect colony dynamics. The solving mechanics depend on an n-dimension space where the optimal solution is a point or a surface and the particles, or ”simulated” insects, try to arrive at this point. The main advantage of this technique is the impressive capacity of overcoming local optimal solutions compared to other meta-heuristic methods [22].

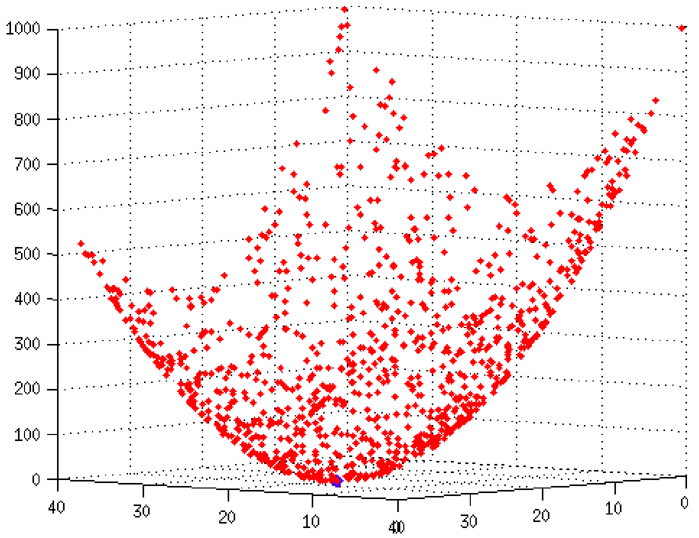


Figure 1.8: Example of scattered particles in a function optimization.

Figure 1.9: ref: https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/46985/versions/1/screenshot.jpg

Evolutionary Algorithm

Evolutionary Algorithms came from the concept of Darwin's evolution theory where only the fittest animals survive and pass their genes on to there offspring. In his theory, an environment with a given population will have its members strive for survival and reproduction. This leads to having the fittest pass their genes to the offspring in more quantity than the least favorable that end up, most of the time, being eliminated first [20, page 13]. The basic steps needed to make an evolutionary algorithm and that define there type are: Crossover: Determines how parents genes will be passed down, the obj: a random altering of the offspring, Fitness Function: the objective function that determines how well an individual is going and finally Selection: How the new generation will be chosen from all the offspring and parents. [19] [23]

1.4.2 Structural Optimization

After the geometric optimization is finished, the structure of the fittest individual has to be altered to reduce weight while maintaining a minimum rigidity. This type of optimization can be divided in to three main categories as determined by A. Rindi in his work "Static and Modal Topology Optimization of Turbomachinery Components" [6]"parametric optimization (which changes the size of the elements), shape optimization (which changes the structures shape), and topological optimization (which changes the topology of the structures, for example, when a shape splits into two, or it develops holes)." .

Shape Optimization

Shape optimization as the name suggests changes the structure shape [6] and has, as design variables, the scale factor of perturbation vectors. These vectors receive an initial configuration and utilize alternative grids to change the overall layout of the structure until an optimum candidate is achieved [24]. This transformation can be seen in figure 1.10.

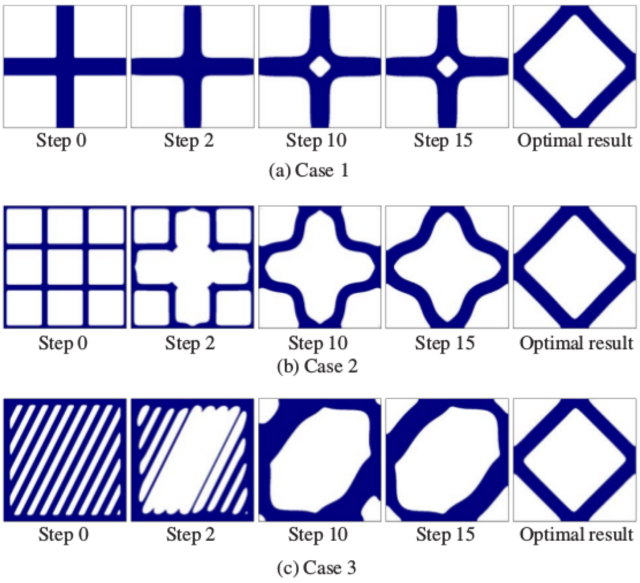


Figure 1.10: Three shape type optimization examples where different initial shape configurations end with the same optimal result[5].

Topological Optimization

This optimization alters the topology of the structure removing or adding material slowly changes the shape.[6]. In this method each element of the structure is considered a variable and by altering the density and the position of each element it is possible to find the optimal solution given a set of forces. [6] [25]

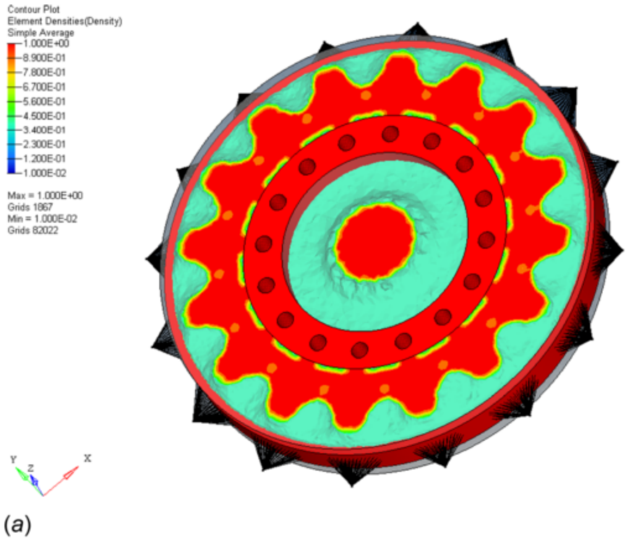


Figure 1.11: Example of a topological optimization where material is removed [4]

Parametric Optimization

Parametric optimization is done by altering the element sizes [6] thus is normally used in truss structures where the width of each beam is altered until the requirements are achieved. This type of optimization can also be used in three-dimensional cases, for example, changing the dimensions of a tube wall to withstand a force. [25]

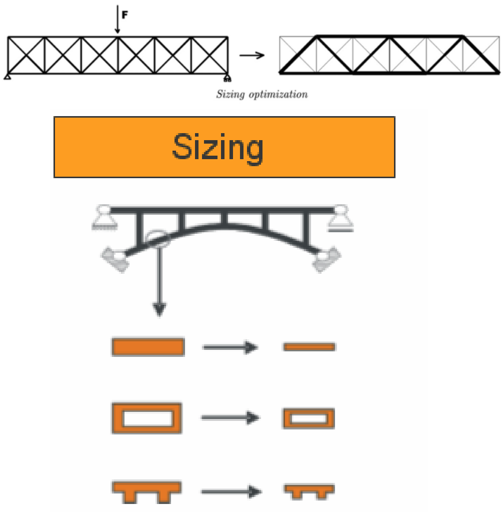


Figure 1.12: Representation of a truss structure with altering the width and example of tube altering dimensions due to optimization ref: [6] , http://carat.st.bv.tum.de/caratuserswiki/images/Design_variable_types.png

It is important to note that this method can be accomplished utilizing several different algorithms ranging from finite elements to gradient-based optimizations.

1.5 Proposed Solution

As seen previously the primary objective of project EMMA is to minimize the time spent on maintenance, as the cost of a single non-operational hydroelectric turbine is in the range of R$ 250.000,00 [3] per month. To accomplish this, the team responsible acquired a commercially available robotic manipulator capable of passing through the 80cm hatch to perform in situ hard-coating. Since the width of the blade is over 2.5m long, it would be necessary to use a manipulator with at least that length, but commercial manipulators with that size have bases larger than 80cm and excessive payloads for the problem at hand with over 20kg at the end-effector[26]. However, even with this change of procedure and use of in-situ operations, there is still an intrinsic time loss due to the inefficiency of a mass built product. As described earlier, the objective of this dissertation is to develop a Methodology capable of creating an initial design of an optimized robotic manipulator that can hard-coat an entire side of the turbine blade while still being able to pass through the 80cm hatch.

To accomplish this, the problem was divided into two main categories. As previously described in 1.4 the first section will be a geometric optimization utilizing a heuristic optimization method with the intention of reducing the size of the manipulator and the dexterity gradient while maintaining the ability of hard coating the as much of the blade surface as possible with the highest overall dexterity. As there are several possible methods inside the heuristic family of algorithms, other projects with similar objectives as this thesis where analyzed to determine the best suited for this particular case. In Soonwoong Hwang [11] work they describe the use of genetic algorithms and present promising finds with an improvement from the first models, which reduces the search domain to the evolutionary algorithms. To further narrow the choice the efficiency of the method and the overall results were observed, and according to Mohd Nadhirs work [19], that benchmarked a series of methods, Differential Evolution (DE) produced the best results, thus being selected as the candidate for the problem. This choice is further reinforced by other works that deem it as one of the most powerful stochastic methods available[27][28][29].

In the second segment, structural optimization is performed on the rigid links of an already geometrically optimized manipulator. This will be done to reduce the overall weight of the structure while still maintaining a minimum rigidity. To do this the parametric optimization method was selected. This method was chosen for a few reasons, the first being the ease of working both computationally and geometrically with cylinders, this permits more straightforward collision calculation due to the simplified geometry.It is also allows for the highest moment of inertia while having the lowest weight. Another reason is that the hatch already sets a boundary condition of the maximum radius of the ridged bodies ate 80cm so combining the two motives it is possible to determine a maximum dimension of the cylinders and later alter the wall thickness until the optimal solution is achieved. To utilize this method, the same Differential Evolution (DE) will be used to calculate the deflection and optimize the wall with each manipulator section independently. It is important to observe that the biggest contribution of the structural optimization will be the moment of inertia to weight ratio that will be calculated of each segment, thus allowing further alterations of the cylindrical body without much problem.

1.6 Dissertation Outline

This Dissertation is organized as follows:

Chapter 2 introduces the kinematic model and inverse kinematics necessary to efficiently use the genetic algorithms. It will also contain the analysis of the dexterity that will be implemented in the genetic algorithm. Finally, the forces, torques and natural frequencies present in the robot thus enabling a full physical review of a robotic model.

Chapter 3 presents an in-depth review of the Differential Evolution optimization algorithm where the mutation, crossover, parent selection, and fitness function will be explored and determined. It will also present a set of results obtained from the optimization process.

Chapter 4 addresses the structural optimization method and shows the results obtained by applying the Differential Evolution method and the structural deflection analysis.

Chapter 5 shows the final remarks and future works are suggested

Chapter 2

Robotic Manipulator Project and Kinematics

This chapter will be divided into two main sections. The first will focus on explaining the kinematics for a generic robot with 6 degrees of freedom. This includes explaining how kinematics models have developed and the fundamental ideas behind them. This will then be used as a base to explain inverse kinematics or ”IK” and how it will be used to map the angle of each joint as to position the end effector on any point in the hydroelectric turbine blade. Finally, the work of Soonwoong Hwang [11] will be studied to elaborate a methodology to use Robotic manipulability in the genetic algorithm. In this chapter dexterity and manipulability will be used as having the same meaning.

The second part will elaborate on project development stages and workflow methodologies to be used. This involves determining how products are developed and the methods that are used to speed up project creation. Then the kinematic representation from the first section will be used to determine a parametric equation for the forces, torques and natural frequencies of a generic robot. This extensive analysis was chosen as the use of genetic algorithms can develop any manipulator configuration; thus equations capable of couping with the immense number of possible kinematic configurations are needed. Finally, the physical representation of the manipulator will be studied with the intention of integrating the kinematic model and the structural optimization to produce more accurate results. This physical representation is made up of the motors that will be used as well as how they will connect with each structure.

2.1 Robotic Modeling

When working with complex fields of studies such as Robotics, there is a need for simplifying where it is possible. Usually, the intention of doing this can be put in two main categories, where the first one refers to the capability of human understanding and visualization. In this category, the simplification is done to allow a more straight forward and simplified approach to understanding the topic at hand and permitting a better visualization of what is essential. The second category applies to computers where simplification is needed in order to reduce the workload of data that can be considered irrelevant to the problem at hand. This is done due to the limited capacity of computational power and needs to obtain answers in a predetermined time frame.

In robotics, the first simplification that's done is by creating kinematic models that posses only the basic shapes and information necessary. In this thesis, this first step was done to create the simplified structure of the generic robotic manipulator. This model was then used to understand how the robot could change the angles in each joint and reach the desired point in space and what type of joint would be used to achieve this.

After understating the basic shape and applying geometric and configuration restrictions to the robot, the IK will be used to determine a point in space and then calculate the joint configuration to arrive in the point. As a means to create a more realistic model, it was necessary to expand on the inverse kinematics method and determine a way to reward the joint configuration that would allow the overall best trajectory. This meant not only looking at the robot but what it was capable of doing. This was done by analyzing the manipulability of each joint and its configuration to determine how this would influence the overall capacity of the robot to move from that position. Although there will be an overall explanation of kinematics, this thesis's prime focuses was not to develop Inverse kinematic methods but use already existing methods in conjunction with genetic algorithms. Due to this restriction, the python library TinyIK [30] was used to its fullest extent as a base for the inverse kinematics. Thus this section will give a broad overview of how each method is used and then focus and go into greater depths on how it was manipulated to work with the following chapter of genetic algorithms 3.

2.1.1 Kinematics

As stated previously, before being able to calculate the IK, manipulability and the designated coating area, it is first necessary to understand the core components that make a robotic manipulator. To do this, it is necessary to simplify the physical structure of the robot to more manageable parts. The first premises that will be used is that this robot will be fixed to a stable and not deformable structure. FThis is done to simplify the overall problem and allow that all studies be focused exclusively on the robot. The manipulator will be composed of six joints and seven ridged bodies connecting each joint as described in [7, page 58] and seen on 2.1

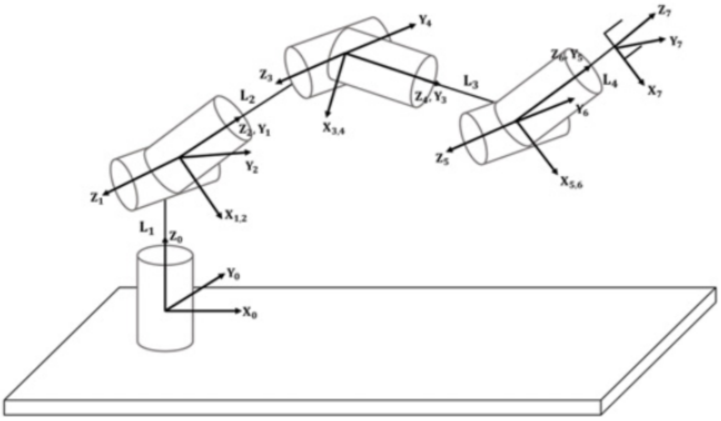


Figure 2.1: Figure representing a traditional kinematic structure.[7, page 59]

Rigid bodies are structural elements that will be considered, as the name implies, overall rigid and subjected to only minor deformations in the elastic region of the material. In this thesis, all bodies will be considered rigid and will only defer in there dimensions and shapes that will be defined by the genetic algorithm. In contrast to the simple and straight forward definition of rigid bodies, the concept of a joint is a mechanical structure that will allow two rigid bodies to move in between them thus there are at least five basic types of joints under two main categories: prismatic and revolute. The first type of joint provides a linear motion between to links with a single degree of freedom, while the second category permits a revolution type motion between two links. In this thesis all joints used will be considered revolute with there axis of revolution in one of the three Cartesian axis X, Y or Z when compared with the previous joint. After examining over 30 industrial application robots, it was noted that all used revolt joints without a single prismatic. Thus, as a result of the analysis, the criteria that were used to chose this type of joint was the frequency of use in industrial environments and applications similar to the ones that this thesis will be used for.

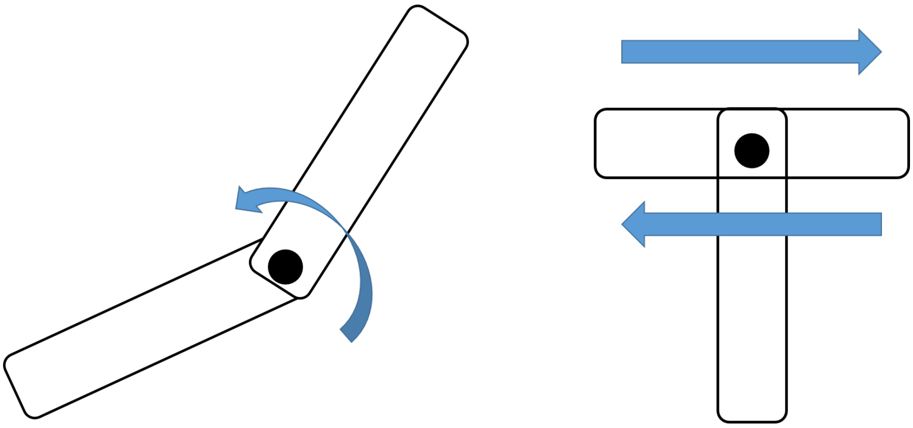


Figure 2.2: Example of prismatic and revolute joints

On chapter 3 it will be explained that although the analysis of the industrial manipulators show that all joints configurations have a base rotation in the ”z” axis followed by an ”x” axis rotation with the remaining joints altering from robot to robot, this thesis uses genetic algorithm no fixed configuration will be used, and all robots can change anyway the algorithm wants.

It is important to denote that as the manipulator moves the joint configuration may change for example an ”x” axis rotation, in the beginning, can become a ”Y” axis if turned 90 degrees. The following figure 2.3 allows for a simple understanding of the initial configuration and the method used to determine the joint revolution with respect to the previous joint.

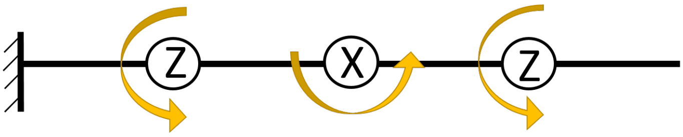


Figure 2.3: Example of method to determine initial joint configuration.

Thus it is important to reinforce that all ”X” and ”Y” joints following a ”Z” axis joint are the same so for example the configuration [Z, X, X, Z, Y] can be considered the same as [Z, Y, Y, Z, X] but configurations [Z, X, Y, Z, X] is different then [Z,Y,X,Z,X].

As demonstrated in figure 2.4 the robotic manipulator for this thesis was chosen to have 6 degrees of freedom thus needing six rigid bodies and six joints as previously stated. The configuration of these joints and bodies will be serial and anthropomorphic. This quantity of degrees was chosen as it is the bare minimum number of DOF that will allow movement and rotation on all axis, thus allowing the hard coating gun to maintain its orientation perpendicular to the coated surface if wanted.

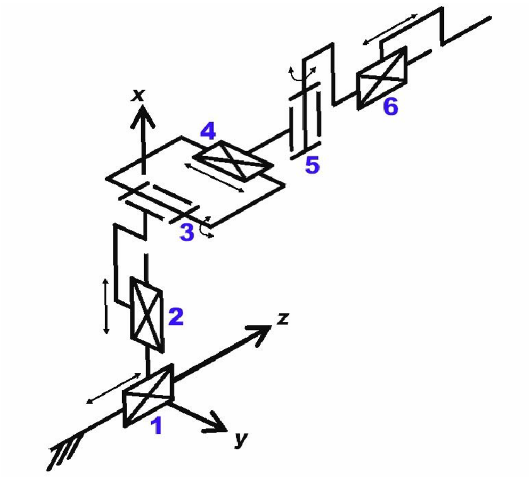


Figure 2.4: Kinematic representation of robot and axis design with six degrees of freedom[8]

2.1.2 Inverse Kinematics

When using a robot to hard coat the surface of the hydroelectric turbine, as specified in section 1.5 and seen in figure 2.6, it is of essential need to know the reach and position of the end-effector of the robot. This can be done using either direct or IK methods. Direct kinematics is a method where the joint angles are defined and the end-effector position calculated. As such this method would prove to be computationally expensive to apply when confronted with over 2000 desired points.

IK problems are the opposite of direct kinematics and consist of determining where in space the end-effectors has to be and then calculating the necessary angles for each joint. As IK is the primary method used to control robotic manipulators [31, page 58], plan trajectories and is more oriented to the problem at hand as such it will be the method of choice.

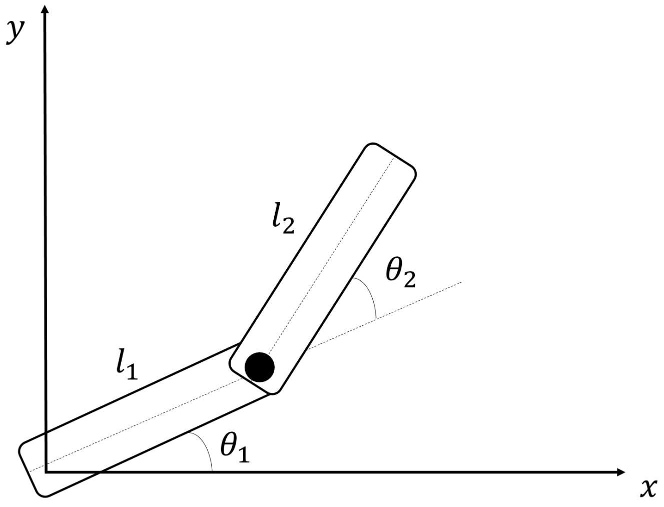


Figure 2.5: Example of a simple Inverse Kinematics

During previous studies the EMMA team used the simplified structures as demonstrated in figure2.1 in conjunction with programs such as OpenRave [32] and Matlab [33], to solve the IK of the robot and determine the total area that the MH-12 could cover. This application can be seen in figure 2.6 where the possible painting area is tested.

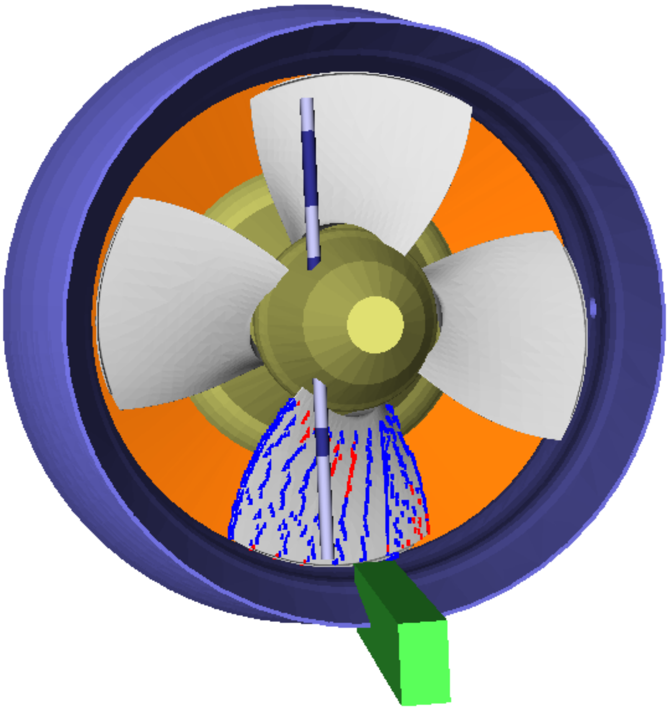


Figure 2.6: Example of covering area of robots

There are quite a few possible methods to solve an IK problem, for example, Jacobian transpose methods, Levenberg-Marquardt, pseudoinverse methods and others [31] with most of these methods utilizing gradient base approximations to determine the position of the end effector.

During the analysis of possible python libraries to be used, two with opposing characteristics were deemed as choices. The first library is OpenRave that possesses a complete analysis of the kinematics of each manipulator and thus can perform collision checks with the ambient and the robot itself. Other than possessing angle limits for each joint. The first problem that this library has is the impossibility of creating specific joint configurations as the use of genetic algorithms would require. The second is the methodology of calculating the inverse kinematics that is more suited on studying the full inverse kinematics of a robot then analyzing only a few points as it takes between 5 to 10 minutes to do the calculations [32]. The library TinkIK that was used as a base for this work, on the other hand, takes only a few milliseconds to accomplish a single inverse kinematic and has the versatility of being able to utilize all joint configurations. However, due to its simplicity, it did not initially posses limit angles for the joints nor the collision analysis that OpenRave has.

Since the creation of a kinematic algorithm is not in the scope of this thesis and the objective is to validate a method rather than propose an optimized robot, the chosen one will be TinyIK [30] as a base with a few modifications to its implementation to suit the needs of this thesis.

The first modification done was alter the optimization method used in the IK from a Newton-based optimization to a least square optimization with a multi-objective function. The two functions that were used are the minimum distance from the current location to the desired point, and the manipulability were the details describing the manipulability will be explained more in-depth furthermore in this chapter.

$$x = Target - PresentLocation$$

$$Manipulability = \frac{MinSingularValueofJacobian}{MaxSingularValueofJacobian}||$$

This flavor of optimization was chosen as it is an already built-in function of Scipy and permits the use of boundary conditions and multi-objective functions. The boundary conditions were used to creating limiting degrees for the joints where joints turning in the axial direction of the robot (Y) can perform 360 degree turns while joints in the other two directions (X, Z) can only go 120 degrees to each side. The need of two objectives during the inverse kinematics presented itself when manipulability was considered, as there are no restrictions regarding the overall angle that the robot can arrive at the desired point. Therefore there has to be a mechanism that obtains the configuration with the highest manipulability. This will be important during the optimization processes as the fitness function will not be optimizing an arbitrary configuration with an arbitrary dexterity, but an optimized configuration with, most probably, the highest dexterity for that position.

The way the least square optimization method works is by transforming the problem in to a optimization problem with iterations. This means every round the manipulator slowly gets closer to the desired point and at the same time tries to make the manipulability the highest possible for that distance. This is done by using direct kinematics to determine the closest distance to a point. This will be important during the application of this method as even if it does not give a correct answer, it will give the closest possible distance to a point.[7]

2.1.3 Differential Kinematics

Differential Kinematics is the kinematic methodology that gives the functional relationship between the end-effectors angular and linear velocities and the velocities in each joint. This can be represented by a matrix denominated Geometric Jacobian. The other Jacobian that can be expressed is the analytical Jacobian, which expresses the relationship of the end-effectors pose and the minimal representation in the operational space[7, page 105].

The use of the Jacobian is vital as it is used to find singularities, determine redundancies and deduce inverse kinematic algorithms. Any n-DOF manipulator can be described by the multiplication of the following homogeneous matrix for each joint:

$$T_e(q) = \begin{bmatrix} R_e(q) & p_e(q) \\ 0^T & 1 \end{bmatrix}$$

Where $R_e(q)$ is the rotational matrix, $p_e(q)$ is the position vector and $q = [q_1...q_n]^T$ is joint variable vector thus being that the position and the orientation of the end-effector vary as q varies. Deriving $p_e(q)$ and the angle we achieve the velocity vector as a function of the joint velocity:

$$\dot{p_e} = J_P(q)\dot{q})$$

It is also possible to formulate a relation of the angular velocity w_e and the joint velocity:

$$w_e = J_O(q)\dot{q}$$

Combining both results we arrive at the Jacobian: $J = \begin{bmatrix} J_P \\ J_O \end{bmatrix}$

In this thesis, the Jacobian will be used to determine the manipulability and its gradient for the end-effector.

2.1.4 Robot Manipulability

Robotic manipulability is the capacity a robot has to arbitrarily change position and orient its end-effectors in a point in space. There are several ways to accomplish this measurement, but the method that was chosen in this thesis was utilized by F.A. Lara-Molina, J. M. Rosario and D. Dumur [34] and is based on Yoshikawa's definition for robot manipulability [7]. This method was chosen due to its use in applications similar to the one developed in this thesis and consequently because it has been proven to work with optimization problems. [34]

Theory

To determine the dexterity of a manipulator according to Yoshikawa, it is first necessary to determine the position of each joint and then obtain the Jacobian for that desired configuration. With the jacobian determined, the Eigenvalues are calculated and the smallest and largest used. The smallest value was then divided by the biggest thus creating a value between 0 and one that compared the different radius of the ellipsoid as seen in the following equation.[7]

$$k = \frac{\sigma_{min}(J)}{\sigma_{max}(J)}$$

This manipulability can be better understood as a graphic in 2 dimensions where the disparity between the radius of the ellipsoid represents the "overall ease of motion " at a given position. Likewise, if a more in-depth analysis is needed, the vector size from the origin of the graphic to the edge of the oval will represent the manipulability in that direction.[7] pag[153]

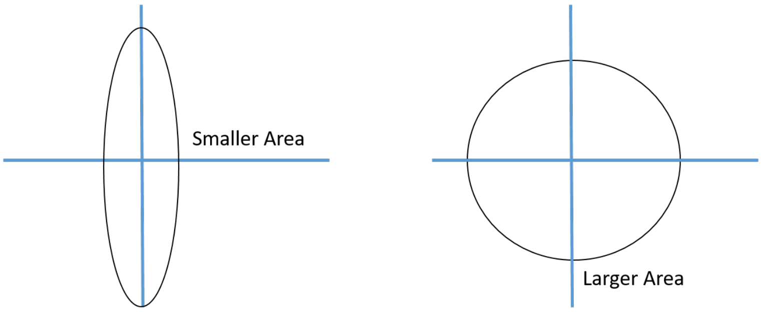


Figure 2.7: Representation of the area of manipulability

To utilize this dexterity there are a few ways that this can be approached, but all involve creating a Global Index that has the dexterity represented in some form.

The first global index that will be studied is the Global Conditioning Index or GCI. This index is always calculated by integrating the dexterity in the workspace W and dividing by said workspace. It is imperative to denote that this approach does not emphasize if the dexterity is linear or angular and utilizes only the largest and smallest value. The following equation demonstrates how this Index is calculated.

$$GCI = \frac{\int w^{1/k} dW}{\int w dW}$$

By having GCI utilize the largest and smallest eigenvalues regardless if they represent the linear or angular portion of the dexterity the equation is simplified. The problem this imposes is in the cases that one is more important than the other. One method of correcting this problem is by dividing it into its linear and angular portions and by utilizing singular-value decomposition there eigenvalues can be obtained. Then the same procedure described previously can be utilized to obtain the linear and angular dexterity.

$$k_{Linear} = \frac{\sigma_{min}(J_P)}{\sigma_{max}(J_P)}$$

$$k_{Angular} = \frac{\sigma_{min}(J_O)}{\sigma_{max}(J_O)}$$

The last dexterity index that will be studied is the Global Gradient Index that can be seen in Rosario's work [34] utilizing a Stewart-Gough Manipulator that was proposed with the objective of optimizing its workspace in relation to its overall size. This index works based on the concept that in the workspace other than just having a high overall dexterity, the robot dexterity also has to be uniform. This is done by obtaining the dexterity in all points in the workspace and then applying the gradient. After that, the highest value is chosen, as it represents the portion where the discrepancies in dexterity are the highest. With this, the objective is to have a work area with an overall gradient value of zero.

$$\Delta GGI = max\| \frac{\Delta 1}{k(J)} \|$$

An analogy to the gradient is the concept of temperature difference in a room, but in the present case, the difference is in dexterity capacity. To better visualize the following figure shows a 2-d image of temperature difference in a plate that is comparable to the case at hand were in blue there could be the lowest temperature or dexterity and in red the highest.

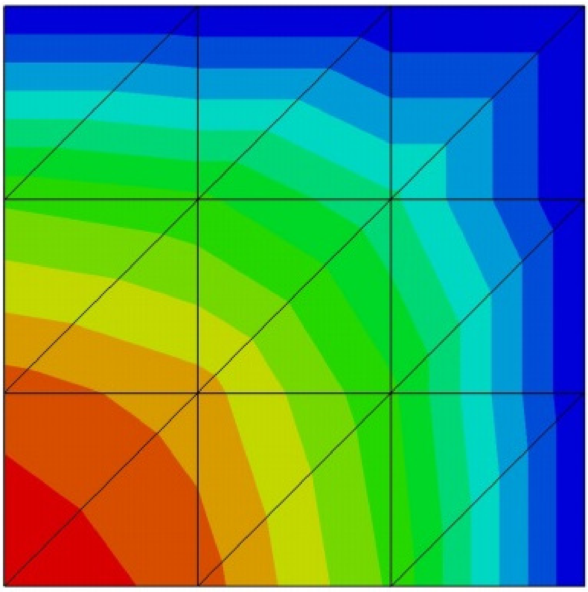


Figure 2.8: Representation of a gradient problem

The chosen method of representing the dexterity was combining both GGI and GCI. A more in-depth analysis of the correlation between these Indexes and the optimization method will be done in chapter 3. Chapter 3 will also explain why one method was chosen over the other while the computational concepts and application of the method will be explained in this chapter

Application

As stated previously the problem presented by this thesis, the main objective is to coat the hydroelectric blades entire surface. To accomplish the mapping of the surface of such a complex shape, 3d scanning techniques were used, whose outputs where coordinate points in X, Y and Z. This enabled the work area to be divided and discretized into individual points. The main benefit presented by this choice of using discreet points is the ability to place the robot in multiple ambients and have optimal solutions independent of the base position of the robot. An example of this is the Doris Robot [35] that always moves on tracks through an oil platform executing numerous tasks as using an inbuilt camera to observe changes in the environment or use a robotic arm to press buttons. With the discretization of space, button presses would be points in space, and thus an optimized robotic manipulator could be created for all tasks. On the other hand the main problem in this method is that it doesn't create a trajectory, so to accomplish an optimal robot for a determined trajectory there would have to be the creation of a trajectory optimization project for the end effector where the optimal dexterity for each moment was calculated in addition to the geometric optimization. As referenced previously this was done in a limited way where the optimal dexterity for each point was calculated, but no trajectory optimization was done.

To deal with the discretization stated previously, two possible outcomes for each desired point on the blade are proposed. Points that will be called "Red points" which have the property that cannot be reached by the end effector of the manipulator and "Blue points" that can be reached.

In the following figure, the manipulator is seen as a vertical blue and light blue sections, and the reachable points in blue and the none reachable points in red.

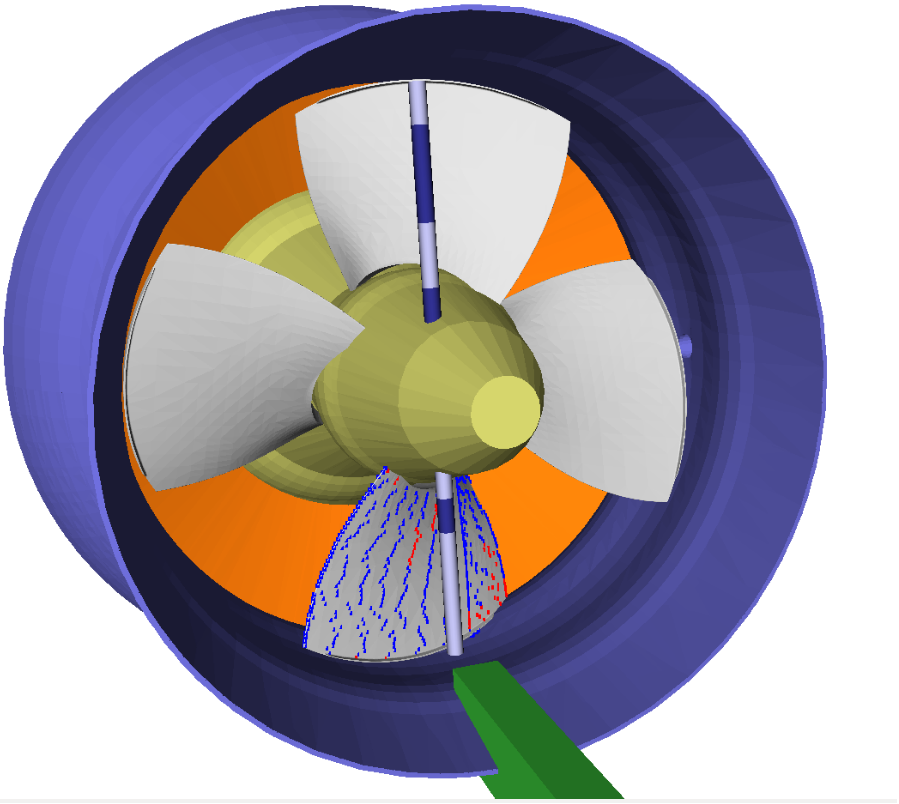


Figure 2.9: Representation of Red and blue points in the Hydroelectric turbine blade

To be able to use the dexterity and gradient dexterity with the genetic algorithm, it was necessary to create a global variable, where each robot had its own variable that represented the total uniformity in manipulability that the robot had. This was done by creating an array of points where only the blue points where analyzed, since the red points could not be reached and thus did not possess any manipulability. This allowed the algorithm to be faster as all red points were set aside and nor used nor calculated as they cannot be coated. The blue points, on the other hand, have a different manipulability for every possible configuration for that point.

As with the trajectory optimization stated previously, to create the best possible program it would be necessary for every point to undergo verification to determine which configuration permits the greatest manipulability, but due to the limited computational resources and time, only a single theoretically optimal configuration for each point was calculated. This, in turn, is used as that points "dexterity value" and is then subjected to the gradient with the other points as seen previously.

The gradient that was calculated is not the actual value as the spacing between points is different, but for the study at hand this does not pose a problem normally as the objective of the function is to go to zero which in turn reduces the discrepancies of the distances. It also does not pose a problem as the comparison of each manipulator is done in the same way, so both cases have the same overall scale of value.

2.2 Project Analysis

Typically during complex product development, it is necessary to estimate numerous variables in each step of the cycle. This occurs due to the co-dependencies of the variables where one feature cant be calculated before estimating another. This cyclical pattern leads to prolonged and expensive development times and leaves room for arbitrary impositions used to simplify the steps. In the first part of this section, the methods used to decrease cycle quantities studied addressed for the development of robotic manipulators.[17]

The second part of this section will work with the actual development of the manipulator, while still presenting innovative methods to cut down on cycle times and minimize the project development time. This is also where there will be the integration of the kinematic representation and the physical structure to be optimized alongside the studies of forces, torques, and possibly natural frequencies.

During the kinematic studies, all structural components between each joint were considered rigid thus the term formal "Rigid Bodies" was expectable to use. During the following stress and strain studies, the manipulator will be considered deformable making it unwise and confusing to utilize the term "Rigid Bodies" to address the structure between each joint; thus they will be called structural bodies.

2.2.1 Project Methodology

When commencing a mechanical project, it is necessary to understand the various steps that will take the project from a concept to the final product which is: synthesis, analysis, and evaluation [36]. These steps comprise the primary cycle and have the basic premise of always having various iterations coming closer to the end product as more cycles occur.

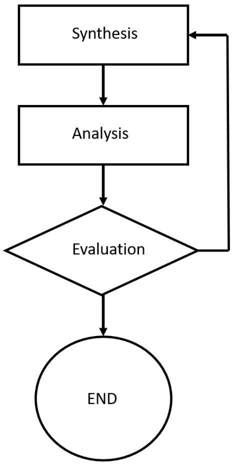


Figure 2.10: Representation of product design steps.

The first step of synthesis is where ideas are brainstormed, and solutions are found for the design problem. The analysis step is where the ideas of the previous steps are filtered and committed to. Finally, the last step of Evaluation is where the ideas are tested and approved or not. All rejected ideas are sent back to the synthesis step and rethought out.[36]

In projects that have multiple variables that influence one another, as the project in this thesis, the number of cycles can grow indefinitely and turn in to a problem especially if the objective is to end with an optimized project.

This perception that the cycles are interactive and in the smaller time frame and gaps led to a concept made by Evans, where the product cycle takes the form of a spiral. This format was created due to the need to estimate variables on every step as a way to estimate other variables. This transforms into a long cycle of estimations until the final product design is close to its end stages [9].

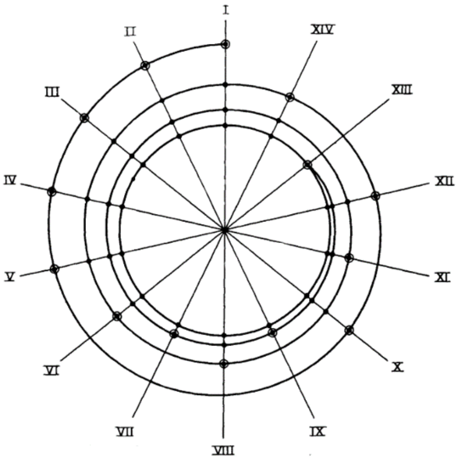


Figure 2.11: General design of the spiral diagram for a random design case [9]

One of the primary methods to reduce iterations is to fix values and consider them ideal and work around them. This may work but depends on the experience of the person arbitrarily choosing values and most of the time end up with less than satisfactory results or products that all work the same. This can be seen in robotics when analyzing manipulators from Motoman or Kuka where most of the robots have the same joint configuration and sometimes even motors altering them only in specific cases where the payload is significantly different.

To counter this problem and reduce iteration cycles, Lamb proposed using, for boat design, a simplified mathematical model but that was still trustworthy to create a first optimized product. This was done in a way so the algorithm would receive restrictions and a few initial values and produce an optimized first iteration that was already a solution close to the end product [17].

Following Lamb's methodology, this thesis will also create a primary optimization to cut down iteration cycles of the robotic manipulator utilizing genetic algorithms as the means to create the initial optimal solution. Due to the complexity of a robotic manipulator, there is also a need to simplify the iteration cycles inside the simplified solution thus chapter 4 will expand on Lambs work inside of the initial optimization cycle creating an objective, and small timed cycle but that still contains a valid result.

2.2.2 Physical Representation

In this subsection, the physical representation and model that is going to be used for the structural bodies and motors will be studied. The objective will be to find the design that approaches the final appearance while integrating the previous studies in kinematics.

Structural Bodies

Following the form used in the kinematic studies, the structural bodies will be represented as cylindrical billets of varying lengths but with an initial diameter of 700mm. This form was chosen for two main reasons: First, it is subjected to the same restrictions as the kinematic model, thus allowing the model to be as close to reality as initially possible. On the other hand, as it will be using the maximum dimension restrictions, it is possible to apply the structural optimization processes to eliminate the material from the rigid body.

When considering how the weight will affect the structure, it is important to denote that the length of the body is significant and has to be considered; thus the weight cannot be represented as a point weight in the middle of the body. This means that the weight of all structural bodies will be represented homogeneously through each structure. This representation can be seen in figure 2.12

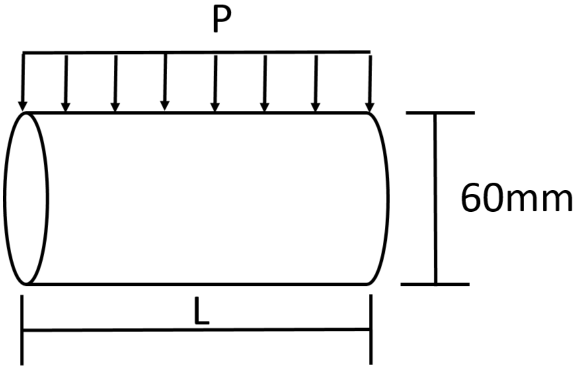


Figure 2.12: Representation of rigid body that will be used for force and torque analysis

This body will later be altered to receive the motors and to do this the overall length will be maintained, but the ends of each body will be swapped for the necessary geometry. This can be seen in the following figure:

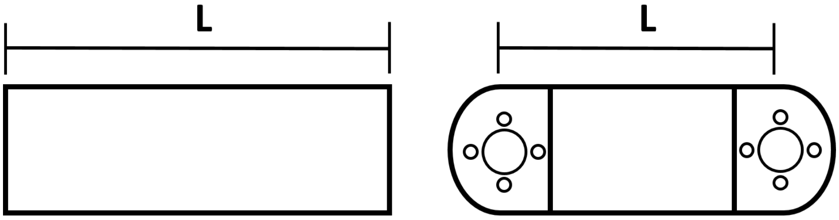


Figure 2.13: Representation of rigid body that will be used for force and torque analysis

Motors

Differently from the structural bodies, the motors depend heavily on the complete optimization cycle as the torque will significantly reduce with the reduction of the overall weight. Thus it will be necessary to represent the motors as simple cylinders of 150mm length and 150mm diameter. As this is the first representation and the motor is of small length compared to the structural bodies, the force will be considered a point force at the center of the body. This representation can be seen in figure 2.14. As a base value, all motors will be considered having 20kg as it is a conservative value that can be altered after optimization.

The motors and reduction used will be servo motors in conjunction with harmonic drives due to this combination being extensively used in all robotic and industrial applications. The throughout the use of this combination of motor and reduction system can be seen when analyzing the Motoman robots where most motors posses this configuration[37].

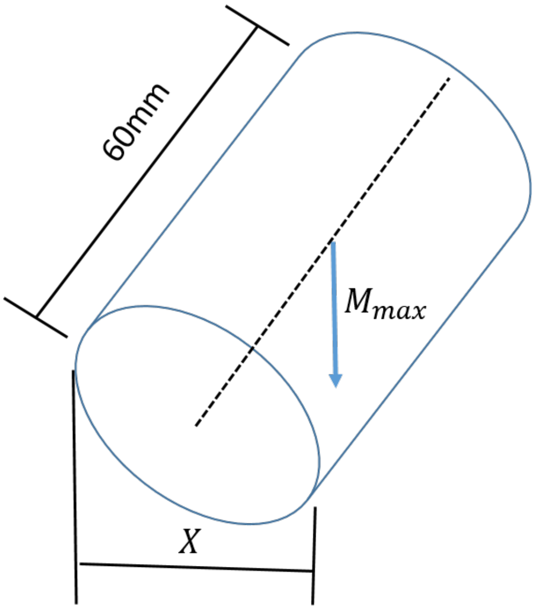


Figure 2.14: Representation of the motor that will be used for force and torque analysis

Harmonic Drives were developed in 1955 for aerospace applications where compact high-ratio transmissions were required. They are composed of three main parts: The wave generator, flexspline, and circular spline and work off of the concept that the wave generator will expand and contract the flexspline changing the position of there tooths each time it moves in relation to the circular spline. This allows for the compact but high gear ratio that is provided by moving a few degrees in every wave cycle. [10]

In the following figure, it is possible to see the division of each part and allows for better visualization of the motor and each of its components.

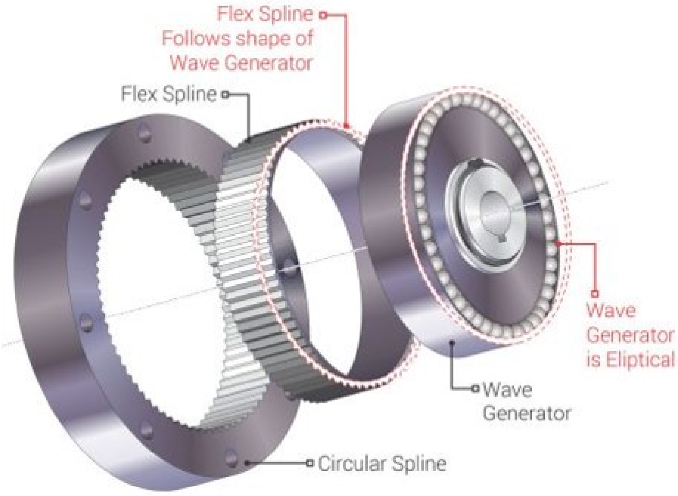


Figure 2.15: Representation of a harmonic drive[10]

2.2.3 Forces and Torques

To study the torques and forces present in the manipulator, it was first necessary to determine how each component would behave and create a model of this. In this chapter, the equations that determine the problem at hand will be created to obtain the angular deflection that the structure will be subjected to in a worst case scenario. This deflection will then be used as one of the variables in the Differential Evolution optimization technique.

Before determining the equations that will govern the problem at hand, it is necessary to elaborate a few definitions that will be used as a foundation to the problem. The first definition is the required maximum deflection for the end-effector that will be around chosen as 1.5mm at the farthest point situated at a distance of more than 2.5m from the base as the limiting factor, and boundary condition is not only the maximum stress, but the maximum deflection wanted.

The option of utilizing an approximate maximum deflection of the structure of around 1.5mm was chosen due to the width of the trail created by the hard-coating process being approximately 3mm and so the error due to the deflection was chosen to be half of the width as to always allow the application of the hard coating without any gaps as also containing few overlays. The following definition that has to be established is what will be considered as the "worst case scenario."

To define this case, it is necessary to lay out a few forces present during the hard-coating process. These forces usually are no more the 20N coming from the tip of the coating gun, and the weight of the gun is no more than 3kg. This means that the most prominent force that influences the structure is its own weight. Thus it is possible to define that the worst case will be when the structure is in the horizontal position as seen in figure 2.16.

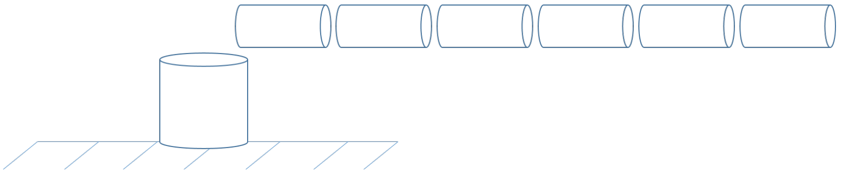


Figure 2.16: Representation of worst case scenario

This configuration was chosen with a static perspective in mind as, due to control techniques that have to be used during the hard-coating process, the acceleration and deceleration curve have to provide a slow and fluid change in direction. Thus it is implied that there will be little to no significant dynamic loads.

As described on subsection 2.2.2 the structural bodies will be represented as homogeneous bodies with a distributed weight. The motors will have point weights and will be positioned between each body. Considering all this, the worst case scenario will have a force distribution as seen on 2.17.

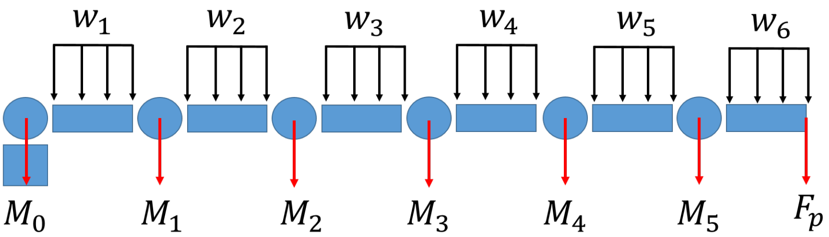


Figure 2.17: Forces applied by the bodies in the worst case scenario.

As it still is not possible to determine the torques at each rigid body and consequently the motors that will be used, the weight of all motors will be considered equal at first. This can be done without significant drawbacks as the weight that will be removed during the structural optimization process will most likely compensate for the added weight of the motors.

Before beginning the structural calculations, it is of common knowledge that the moments and forces applied to a chosen section are lower the closer to the end of the manipulator that they are located. Thus it is possible to deduce that the thickness of the tubes should grow the closer they are to the base. By Utilizing these deductions, it is acceptable to represent each segment of the manipulator being treated as having its own constant wall thickness. This will lead to the easier fabrication of the components and reduce the overall complexity of the manufacturing process.

As stated previously the forces applied can be broken down into two types as seen in figure 2.18 and 2.19 where one is a point force and another a distributed force.

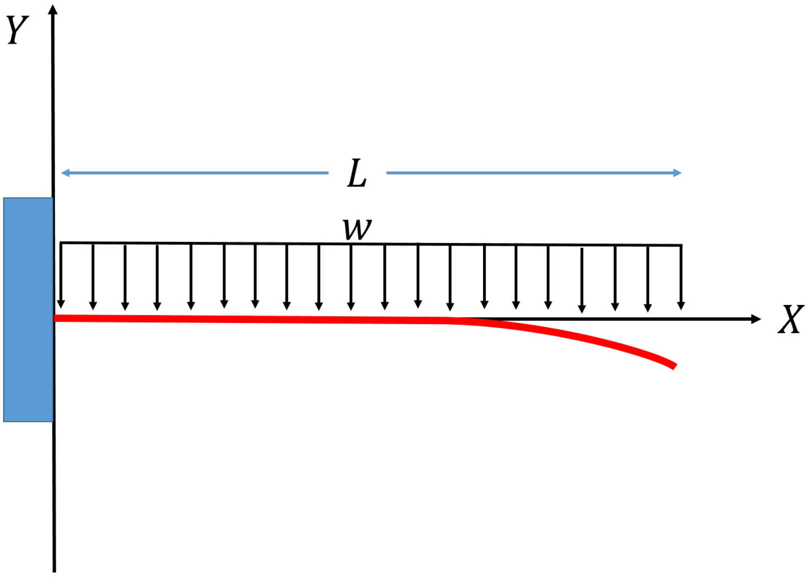


Figure 2.18: Example of a Distributed force

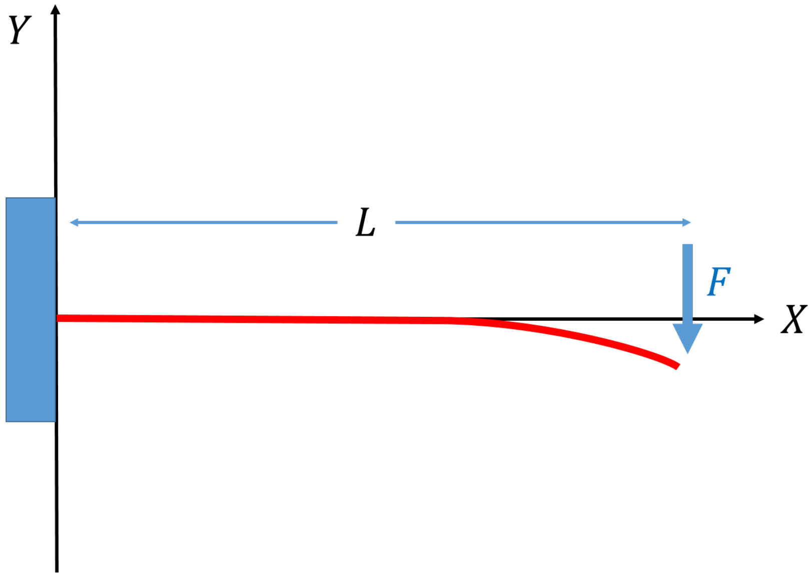


Figure 2.19: Example of a Point force

As the manipulator will have different characteristics throughout its length, the theorem of Castigliano will be used to deduce de maximum deflection of the beam. This theory permits that each force be treated separately and the total deflection summed at the end. Castiglianos theorem is based on the principle that de deflection is equal to de partial derivative of the deformation energy by that force $\delta = \frac{\partial U}{\partial F_i}$ and in the case present in this thesis it can be generalized as the following equation:[38]

$$\delta = \int_0^l \frac{1}{EI}\left(M\frac{\partial M}{\partial F}\right)$$

For the point forces, for example, the motors and force at the end effector, the moment M is defined by:

$$M = Fx$$

While the distributed load can be calculated by utilizing a moment defined by:

$$M = \frac{wx^2}{2}$$

The variable l is the length between the point and the force, E is Young's modulus of the material, I is the second area moment of a cylindrical billet which is defined by:

$$I_x = I_y = \frac{\pi}{64}(D^4 - d^4)$$

Finally, Fw is the distributed forces in N/m [38] and F is the point Force. w is the area density of each individual segment where $w = Area * Density$. The weight of each segment will be used during the calculations of the cost function and can be determined by:

$$M = \frac{\pi(d_o^2 - d_i^2)l\phi}{4g}$$

Where d_o is the outer diameter, d_i is the inner diameter, g is gravity, and finally, ϕ is the density of the material.[38]

After determining the overall stresses that the structure will suffer and optimizing the structure with the weight of the motor being equivalent, the motors will be chosen utilizing the product catalog from manufacturers whose catalog pages can be seen in the Appendix. The first choice of motors will represent an extremely conservative as stated previously were each motor has a total weight of 20kg. They will possess enough torque to move the optimized body and there own theoretically maximum weight. After this initial calculation, the algorithm will choose the possible motor replacement for the user to execute the changes manually.

As described previously, the analytical calculation for the maximum displacement caused by the weight of the motor, generated by a point force, can be described by the same equations as the force caused by the hard-coating gun. With the equations described previously, all forces described in image 2.16 can be calculated.

Finally, it is necessary to discuss the base structure that will support the weight of the rest of the manipulator. Theoretically, it would be necessary to optimize this part and consider the total deflection it would be subjected to. The weights of the motors and structure will produce a pure moment at the end of the beam, and thus it would be necessary to utilize the equation:

$$y_max = \frac{M_b l^2}{2EI}$$

Where M_b represents the sum of moments caused by the other structures of the manipulator.

Although this calculation is theoretically necessary, it will be seen in chapter 3 that the hight of the base structure will be minimal compared with the others thus eliminating the need for this type of calculation. Furthermore, the drastic change in the appearance of the base from the tubular one discussed previously to having the same one as in figure 2.20 reinforces that another type of deflection study should be done in its place. Thus it becomes more practical to utilize finite element programs to determine the deflection and deformations as most probably the structure will have a more complex shape.

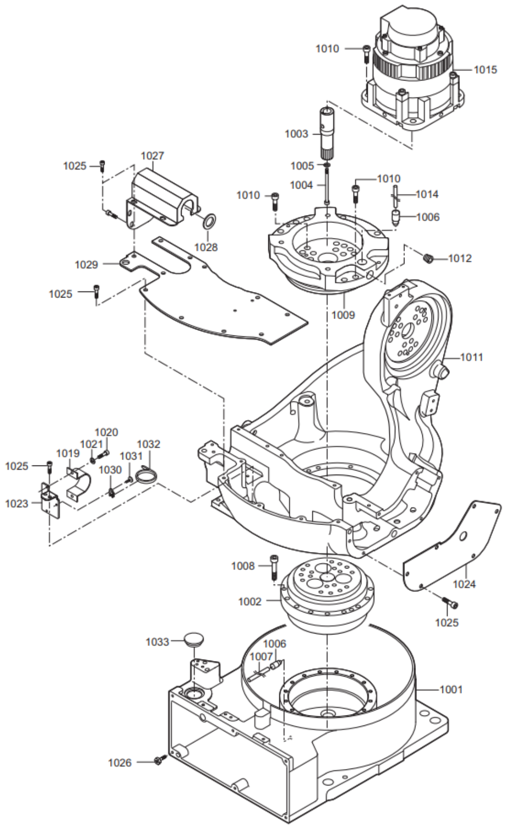


Figure 2.20: Representation of motor placement in base structural body

It is essential to point out that this project methodology uses heuristic optimization processes and allows for any configuration possible. This may lead to manipulator formats that were not considered or that diverged from the initial assumptions. One of these cases is the joints that rotate on the same axis as the structure. The motors in these joints will not be submitted to the same torques as the other motors, but they are still be considered equivalent to the other motors. It will be necessary for the user to analyze the results and chose the motor manually in these cases.

Finally, it is necessary to highlight the implications the impact of the structural optimization will have to reduce project times and helping project development. As the cylindrical format is a simples shape, not always will it be possible to use, this would lead to the proposed methodology losing its value. However, this is not the case, as this simple shape allows the user to create a relation between the moment of inertia of a cross section and the density to strength ratio of any material. This allows any format to be chosen for each segment of the manipulator provided that the relationship is met.

2.2.4 Natural Frequencies

In addition to the deformations suffered by the robotic manipulator due to structural stresses, any vibration can be the cause of inconsistencies and errors during the hard-coating process. In the stated problem, the main possible source of vibration is the hard coating gun, and thus a thorough analysis of the application process should be done. After various technical consultations with the company responsible for the hard-coating process in Jirau, Rijeza, all forms of vibration coming from the coating gun were shown to be negligible. This conclusion leads to the study of the natural frequencies of the body not being necessary and thus will not be conducted.

2.2.5 Buckling

Another one of the possibilities that may occur with the structure is self-buckling when all bodies are completely straight in a vertical position. In the present case of studies, this is less likely due to the outer dimension of the tube being relatively closer to the length of the overall structure then what would be considered as a thin beam. Although the possibility is small, a brief analysis should be considered. Furthermore, this will allow the methodology to be applied in other situations as it will be more robust.

To determine if the segments will suffer from self-buckling the equation determined by Greenhill [39] [40]in 1881 will be used as follows:

$$l_{max} = (7.8373\frac{EI}{\rho g A})^{\frac{1}{3}}$$

This equation was determined by differentiating the Euler-Bernoulli beam equation [41] where for Greenhill's equation there is ρ as the structure density, g the gravity, A the cross-sectional area of the beam, Young's modulus E, and finally I the second moment of area of the beam cross-section.

Chapter 3

Geometric Optimization

An evolutionary algorithm as its name suggests is a class of optimization techniques that have the concept of evolution and survival of the best individuals ate is core. All methods revolve around the idea of creating an environment that provides a limited amount of resources where individuals have to compete to survive thus creating a natural selection or utilizing the area jargon "survival of the fittest." With more individuals at higher fitness levels, the populations average fitness tends to grow [20].

As the biological concept of Evolution is to lose to be used directly in a mathematical point of view, it is necessary to break down the ideas to smaller fragments that can be treated mathematically. Thus the inner workings of the selection process can be reduced on four major phases. The first is to create a random population and evaluate all individuals utilizing a fitness function to have a selection method applied. With the chosen individuals, a crossover process occurs where characteristics are traded between individuals in some way. These characteristics are called genes and represent how the algorithm sees the individual. Finally, a mutation phase occurs where, as in nature, a small number of individuals are submitted to a random modification of there gens. After the new offspring are evaluated parents and offspring are analyzed, and a new selection occurs. [20]

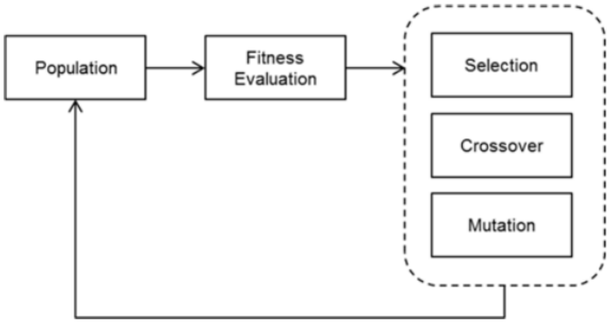


Figure 3.1: Example of the Evolutionary Algorithms cycle [11]

Due to there unique heuristic approach, these algorithms are mainly used when the cost function is nonlinear and non-differentiable and thus cannot be solved using typical gradient-based methods.

The difference between different evolutionary algorithms is simple details in the application and sequence of each of these phases. A simple example that shows this contrast is determining if the selection algorithm should include both parents and offspring or only offspring. Thus this chapter will focus on explaining and showing the development of the chosen evolutionary method along with each phase and how these phases were made to collaborate with de creation of a robotic manipulator.

3.1 Differential Evolution

Differential Evolution is a method inside the evolutionary algorithm categories that were developed with the primary objective of satisfying four criteria as described by R.Stone and K. Prince [42]:
"(1) Ability to handle non-differentiable, nonlinear and multimodal cost functions.
(2) Parallelizability to cope with computation intensive cost functions.
(3) Ease of use, i.e., few control variables to steer the minimization. These variables should also be robust and easy to choose.
(4) Good convergence properties, i.e., consistent convergence to the global minimum in consecutive independent trials."

To achieve this, the method was developed as a stochastic direct search method as other genetic algorithms. To allow for parallel computing the population receives stochastic perturbations that are independent of one another. To maintain objective (2) and (3) the self-organizing scheme utilizes two random individuals from the population to perturb a third individual and is done once for every individual. Finally, criteria (4) was determined empirically testing numerous times until there was sufficient data to characterize a global and good convergence. [42]

As all Evolutionary algorithms, this method utilizes the same basic structure but with minor changes. To maximize initial understanding of the method simplified sequences will be utilized to highlight the changes that ate made from traditional genetic algorithms.

(1-) Create a random population where the variables are continuous.
(2-) Analyze the initial population utilizing a standard fitness function
(3-) Each individual has cross-over done by analyzing the difference between each characteristic of two other individuals and adding this difference to there characteristics.
(4-) A greedy selection process occurs where the new modified individual and the previous one are compared, and the one with the highest fitness level is selected.

It is important to note that there is no direct mutation of the individual, although the cross over process resembles the mutation process since the method of incrementing the characteristic variables is discrete. Further details on each segment will be explained in the following sections.

As previously stated in chapter 1 the problem dealt with in this thesis possess a non-differentiable and complicated to analyze objective function with variables that represent the size of each segment of the manipulator, thus they are continues and positive larger than zero. With these characteristics, it is possible to determine that one of the more suited methods to use is precisely Differential Evolution.

In this thesis, the specific DE strategy used is described as DE/rand/1 [28]. This strategy can be briefly described as a Differential Evolution method that utilizes a random selection of the individuals during the mutation phase and utilizes one difference vector. Further explanations for each part will be given in the following sections.

3.1.1 Gene and Chromosome

To expand upon the evolutionary ideas, genetic algorithms possess what are called genes and chromosomes. Gens are nothing more then the variables altered in each generation also known as decision variables. In the problem at hand, these variables are the sizes of each segment of the manipulator, the position x,y,z of the base of the manipulator and the individual joint rotation plains with respect to a fixed coordinate system. The Chromosomes are merely the vector of variables or gens that will be used; thus each individual has there own chromosome with individual genes. Consequently, a group of individuals, each one with there own chromosomes, form a population 3.2 [20].

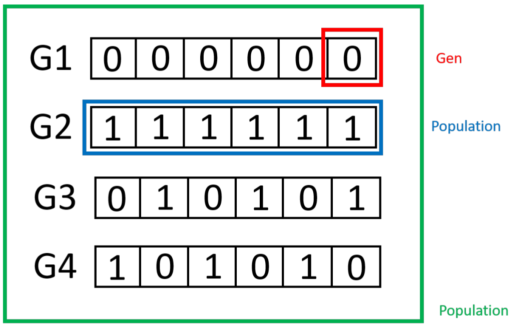


Figure 3.2: Example of Gens and Chromosomes

To determine the initial variable range of the gens that will be used on optimizing the manipulator, it is necessary to break down the problem into smaller parts. For the X, Y, Z of the robot base a 1.5m cube was used as a boundary condition, where from there, they would go to the optimal position inside the limits. This was done as the hydroelectric turbine is stationed inside a relatively confined space and thus not all positions would be possible for the base.

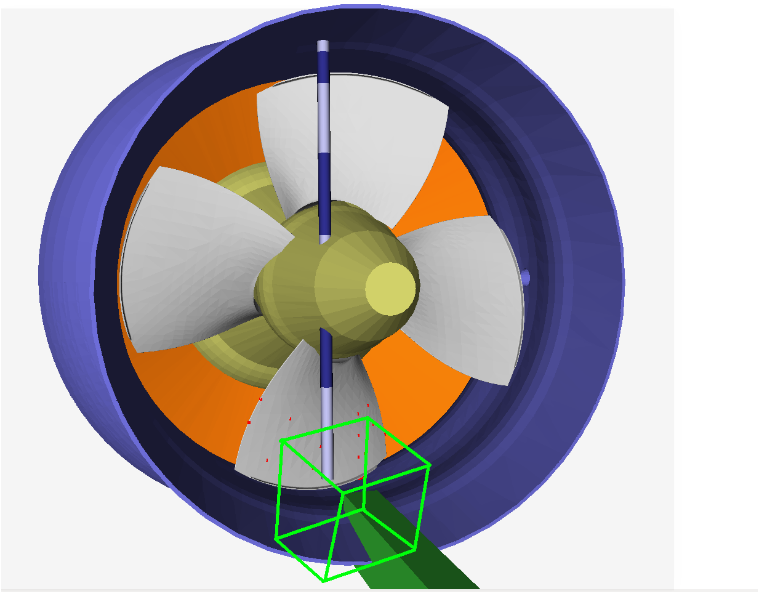


Figure 3.3: Demonstration of the cube used in the problem at hand.

For the link structure, the chosen initial limit conditions were 0.1m to 3m. The lower value was chosen as it is the smallest size possible to place a motor and still maintain a segment while the upper boundary was chosen exclusively to give the first generation of manipulators a wide initial configuration possibility. After generating the initial population the upper limit was removed, but due to the motor constraints explained previously, the lower boundary condition was maintained with a slight modification. This modification was the inclusion of a delta random variable between 1 and 1.5 that would multiply the lower boundary and allow the differential algorithm to evade getting frozen in the same variables and lead to premature convergence.

Beyond the base position and link length, the joints were left with all possibilities of x,y and z configurations. This was done to allow the algorithm to have total freedom of evolving. The results will be interesting to observe as all the industrial robots possess roughly the same configuration and it is not known the motives that lead the manufactures to choose this configuration, as only assumptions can be made.

This repeating configuration can be seen in the following table where multiple robots from two renowned companies in the robotics field were analyzed where, to simplify visualization, Axis-0 represents the base of the robot and Axis-5 represents the end effector.

KUKA						
Robo	Axis-0	Axis-1	Axis-2	Axis-3	Axis-4	Axis-5
KR 1000 titan	Z	X	X	Z	X	Z
Kr 470-2 PA	Z	X	X	X	Z	
KR 30	Z	X	X	Z	X	Z
KR 300-2 PA	Z	X	X	X	Z	
KR 360 FORTEC	Z	X	X	Z	X	Z
KR 500 FORTEC	Z	X	X	Z	X	Z
KR60	Z	X	X	Z	X	Z
KR CYBERTECH ARC NANO	Z	X	X	Z	X	Z
KR CYBERTECH ARC	Z	X	X	Z	X	Z
KR QUANTEC ultra	Z	X	X	Z	X	Z

MotoMan						
GP50	Z	X	X	Z	X	Z
GP35L	Z	X	X	Z	X	Z
GP180-120	Z	X	X	Z	X	Z
AR2010	Z	X	X	Z	X	Z
AR1440	Z	X	X	Z	X	Z
GP25-12	Z	X	X	Z	X	Z
MPX2600	Z	X	X	Z	X	Z
HC10	Z	X	X	X	Z	
MPX3500	Z	X	X	Z	X	Z
GP8	Z	X	X	Z	X	Z
MH250 II	Z	X	X	Z	X	Z
MH180-120	Z	X	X	Z	X	Z
MH110	Z	X	X	Z	X	Z
SIA5F	Z	X	Z	X	Z	X
ma3120	Z	X	X	Z	X	Z

Table 3.1: Table with the different types of commercial robotic manipulators and there axis of rotation on each joint.

As already stated in chapter 2, all axis rotations after a ”Z” rotation can and should be interpreted as both an ”X” or a ”Y” rotation depending on how the previous joint is positioned. Thus only ”X” rotations following one another that ate ”static” and will always possess the same axis of rotation.

3.1.2 Fitness Function

When dealing with an optimization problem, it is necessary to have an algorithm to govern and determine when a solution is better than another. This equation is called fitness function when leading with genetic algorithms in general due to the concept of an individual having to be ”fitter” or ”more adapted” in order to survive.

Genetic algorithms utilize the chromosomes as a base to create a fitness function that is typically represented by a characteristic that needs to be maximized or minimized. In this thesis, there are four characteristics that should be optimized: Manipulator size, Dexterity, Dexterity Gradient and capacity to hard coat the surface. Each and every one of these will be further described and discussed later on in this chapter. As there are multiple objectives in the optimization, it is necessary to utilize multi-objective fitness functions that will allow the creation of a Pareto front to determine the best possible robots.

In this section, the fitness function used to define the optimization criteria of the robots will be broken down and analyzed. All coefficients used will be explained and the motivation behind each variable elaborated.

Multi-Objective Fitness Function

Multi-objective fitness functions are used, as the name suggests when there is more than one objective in the fitness function. There are many different methods to accomplish this, but the method chosen is the weighted sum method. It was chosen due to the simplicity that it provides and the extensive studies that where conducted utilizing this method that prove that it is a suitable means of multi-objective optimization.

This method works by multiplying each variable by a constant that alters the weight of the variable, where the sum of all constants is equal to one[43].

$$F = \alpha_1 X_1 + \alpha_2 X_2 .... \alpha_n X_n$$

In the equation above, alpha represents the constant and X the variable. Although it was said that alpha is a constant, it is only a constant during each time the optimization program runs, as it is necessary to alter the value of all alphas in a discrete manner after each complete optimization attempt as only then will it be possible to create a Pareto front. The values of alpha were chosen to vary between 0.1 and 0.8 as a way to allow all values to be positive and different than zero making it obligatory that all variables will be present during the optimization. The value of all α is governed by the following equation where the sum of there values has to equal 1.

$$\alpha_1 + \alpha_2 + ... \alpha_n = 1$$

A solution is said to belong to the Pareto set of solutions if there are no other solutions that can improve on at least one of the objectives without making another solution worse [12]. This creates a surface where all solutions are optimal, and the desired answer depends on the objective of the user. This can be better understood by analyzing figure 3.4 where the entire solution space is denoted in black, and the Pareto front for maximization is shown in pink.

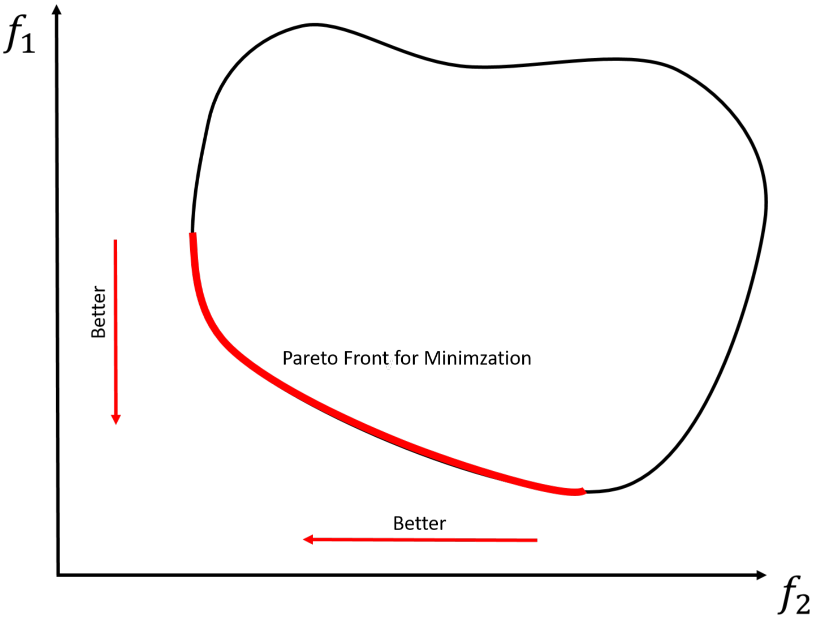


Figure 3.4: Example of Pareto Front in 2D [12]

This concept of the Pareto front can be used during the geometric optimization, but due to time constraints, the visual representation of its effect will be seen during the structural optimization process where the multi-objective fitness function will have its variables minimized.

Utilizing the concepts described above to achieve the objective of creating an optimized robot the equation chosen as the fitness function was:

$$F = \frac{\alpha_0}{0.4Size} + \alpha_1 * Paint_{porcentage} - \alpha_2 * GradientDextarity + \alpha_3 * Dextarity$$

Where the initial values of each variable in the fitness function can vary according to :

$$
\begin{aligned}
Size &\in [2,5;\infty] \\
Dextarity &\in [0;1] \\
GradientDextarity &\in [0;1] \\
PaintedPorcentage &\in [0;1]
\end{aligned}
$$

This is a maximization equation where the variables are manipulator size, Dexterity, Dexterity Gradient and percentage of the desired area that is painted. All are dependent on the gens and, co-dependent in between them. Further details of these variables will be described in the next subsections.

Manipulator Length

The manipulator length is a variable that has an initially a simple concept of being calculated by the sum of each individual length of the structural components. Yet it presents a few difficulties in implementation that need to be observed and tended to.

$$L_{total} = \sum_{n=1}^{7} L_n$$

A great example of these difficulties is that it is a value that needs to be minimized and so it cant be used directly in the fitness function. The other problem that appears is its value that can vary between 2.5 and ∞ which leads to an inconsistency compared to the other variables. The value could be corrected by normalization, but that would not solve the minimization aspect, so the chosen method was to normalize the values by multiplying by 0.4 and then inverting them.

$$Size = \frac{\alpha_0}{0.4L_{total}}$$

The value of 0.4 was chosen as 2.5m is the only boundary condition that is possible for the size of the manipulator, as it describes approximately the overall length of the desired work area. This also helps in altering the problem from minimization to maximization as 2.5m will become the maximum value of 1 and larger values will tend to zero.

Dexterity

As stated in chapter 2 there where two possible candidates that could be used for dexterity one being the Global Manipulability index (GMI) and the other the Global Gradient Index (GGI). A few of the qualities and problems where explained, but in this section, the effects on the genetic algorithm will be studied.

To determine the option that would be used all three methods were tested, and the first option that was ruled out was GMI with the dexterity divided into its angular and linear potion. This was done due to the overall benefits this permits being greatly overshadowed by the excess time it takes to calculate both dexterities. During tests that were conducted the time to calculate them was over 30% compared with a single dexterity calculation case. There was also a lack of efficiency in the results where they to be worse than the single dexterity cases.

Thus for the dexterity, the simple case was chosen by utilizing the following equation:

$$GGI = \alpha_3 * Dexterity$$

Finally, when analyzing the problem at hand, the objective is to hard coat all desired points, and thus the dexterity on every point should be minimum as the manipulator is being designed to arrive at only the determined points. This means that utilizing the gradient as a Global function is also very important to the problem since it is terrible to have a few points with high dexterity and the rest with a low one.

To better understand the Gradient, its value at zero can be represented physically by the manipulator that will be able to arrive at all points with the same manipulability. Thus the equation that will be used is as follows.

$$GGI = \alpha_2 * GradientDextarity$$

As seen in the equation, as with the dexterity, there was no need to apply any additional scaling factors other than the already described α.

Coating Percentage

Coating Percentage (C_p)as the name implies describes the percentage of the total area that can be hard coated by the robot. As stated in chapter 1 this should be considered a restriction rather than a variable as there is a need to coat the entire work area. In theory using the coating percentage as a restriction would work flawlessly but when applying a restriction as this one to the problem at hand, there would be a significant computational cost applied to robots that possess great dexterity and size but cant paint 100 % of the area and thus, due to the restriction, their fitness function would be zero. This would lead to the algorithm not knowing where to go as all robots would posses fitness zero.

In problems where there are no restrictions in space or collision the manipulator can be sized accordingly but in confined spaces, all robots would be stuck with the restriction; thus the genetic algorithm would no work.

As stated before the area of the hydroelectric turbine was discretized, similarly, as it is done in finite element problems, thus it is possible to choose the spacing between the points allowing for more precise calculations that lack speed, or fast analysis that posses limited mesh refinement. To analyze the effects that more points do to the calculations, a few cases of studies where done where the only difference between them was the number of points, which will be called mesh size for simplicity.

The following table contains 4 cases with different quantity of points and the time necessary to do one analysis of the inverse kinematics and the time it takes all 20 individuals to go through 4 generations.

Points	Time per manipulator(s)	Time per generation(m)	Total time(h)
27	20	x	x
41	30	9.05	0.83
78	50	20.6	-
101	75	23.35	-

Table 3.2: Table of average time for each simulation.

Observing this table, it is possible to see that the time grows linearly with the number of points. Thus the number of points that where chosen was 55. This quantity of points was chosen as it was fast enough to do one analysis of 150 generations and 40 individuals every two days while still having a sufficient number of points to scatter on the whole surface of the hydroelectric blade.

The following figure shows two different interactions, where the first image is a case with fewer points and the following image is a case where many points were tested.

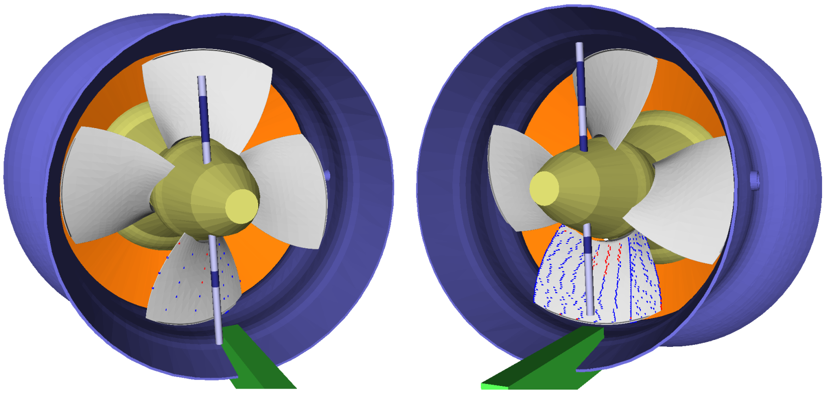


Figure 3.5: Example of point density obtained in Openrave

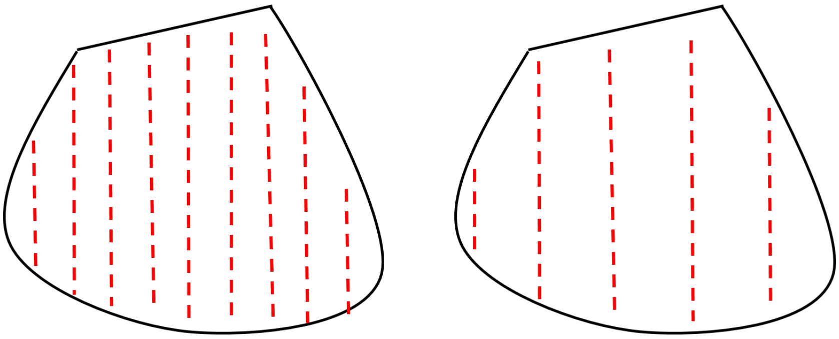
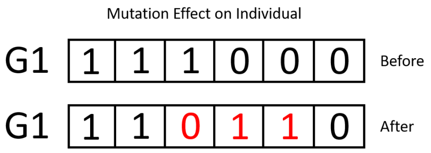


Figure 3.6: Example of point density simplified

3.1.3 Mutation

The mutation is stochastic in nature, and its normal use in evolutionary algorithms is to choose random gen that will be altered to another random value, as it is seen in nature where individuals mutate. This is more easily visualized in figure 3.7. [29]



Mutation Effect on Individual

G1	1	1	1	0	0	0	Before
G1	1	1	0	1	1	0	After

Figure 3.7: Example of Pareto Front in 2D [12]

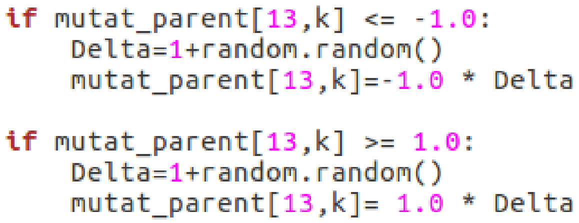
For Differential Evolution, the concept of crossover and mutation is slightly mixed. Normally the crossover process occurs before the mutation but in DE all individuals are mutated, and then a crossover process is done. The individuals that serve as a baseline for the mutation are called target vectors and are represented by $X_{r_1^i,G}$. These new mutated individuals are represented by $V_{i,G}$ where i varies according to the number of individuals in the entire population and will be called mutated vectors. [28]

To create the mutated individual differential evolution utilizes a difference vector of two random individuals to alter the target vector as seen in the following algorithm:

$$V_{i,G} = X_{r_1^i,G} + F(X_{r_2^i,G} - X_{r_3^i,G})$$
$$r_1 \neq r_2 \neq r_3 \neq i$$

Where X are random individuals chosen, G is the generation that all individuals are in, r_i are the random positions in the vector for the individuals, and F is a scaling factor for parameter controlling the difference vector which is defined by the user but will be self-adapting [28].

As described in chapter 2 the mutation was done followed by a randomizing of the lower and upper boundaries for the base position and the lower for the arm segment length for values that passed the limited region. The following line of code demonstrates how this was done.



```
if mutat_parent[13,k] <= -1.0:
    Delta=1+random.random()
    mutat_parent[13,k]=-1.0 * Delta

if mutat_parent[13,k] >= 1.0:
    Delta=1+random.random()
    mutat_parent[13,k]= 1.0 * Delta
```

Figure 3.8: Small fragment of code showing the method used to randomize the boundary conditions

This was necessary as the chosen optimization method does not handle very well having boundaries, especially when the variables tend to always arrive at these limits. When this happens, the algorithms get stuck at a broad spectrum of answers and does not tend to converge.

3.1.4 Cross-Over

When analyzing the crossover process in evolutionary algorithms, there is usually a random rupture [20] in the gen of two individuals, as shown in the following figure, where the point chosen for the division is stochastic.

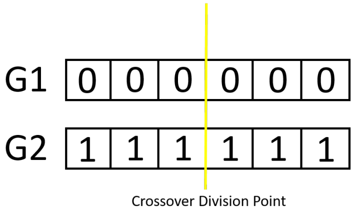


Figure 3.9: Division of the chromosome during a crossover process

After dividing the chromosome into two parts, a swap occurs, and each individual receives a new gen creating two new individuals.

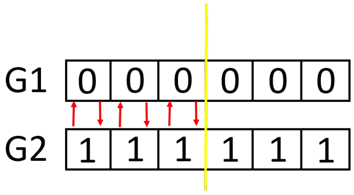


Figure 3.10: Crossover occurring

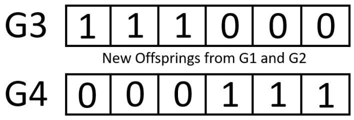


Figure 3.11: Example of two chromosomes after crossover process

For DE the cross over works by mixing the target and mutated vectors to generate a trail vector $U_{ij,G+1}$ that needs to be tested.

$$U_{ji,G+1} = \begin{cases} v_{ji,G+1}, & \text{if } r(j) \leq CR \text{ or } \quad j = r_n(i) \\ x_{ji,G+1}, & \text{if } r(j)\ CR \text{ and } \quad j \neq r_n(i) \end{cases}$$

$$CR\epsilon[0,1]$$

This process is done by creating a random number between 0 and 1 $r(j)$ and comparing it to a crossover constant (CR). There is another random number $r_n(i)$ that serves to force that at least one characteristic from the mutated vector is placed in the trailing vector. This is done as a means to avoid creating a parent generation that is exactly the same as the offsprings. This means the trail vector will only receive a gen from the target vector if $r(j) > CR$ or $j \neq r_n)(i)$. [28]

As Differential Evolution is used for continuous variables and the joints are discreet, a simple genetic algorithm with a simple mutation at 98 % as used for these variables.

3.1.5 Self Adapting-Variables

To create the self-adapting variables the work of [28] was used where these variables from the differential evolutions are F and CR. To create them four random numbers varying from 0 to 1 where created and compared with constants devised by them as follows:

$$F_{i,G+1} = \begin{cases} F_l + rand_1 F_u & \text{if } rand_2 < \tau_1 \\ F_{i,G}, \end{cases}$$

$$CR_{i,G+1} = \begin{cases} rand_3 & \text{if } rand_4 < \tau_2 \\ CR_{i,G}, \end{cases}$$

$$\begin{aligned} \tau_1 &= 0,1 \\ \tau_1 &= 0,1 \\ F_l &= 0,1 \\ F_u &= 0,9 \end{aligned}$$

The values for τ_1, τ_2, F_l and F_u were chosen based on the empirical data and studies done by Brest but showed the best global results in 23 benchmark experiments [28].

3.1.6 Parent Selection

The parent selection strategy in differential evolution is called a greedy selection and consists of always allowing the individual with higher fitness to survive while killing the lower valued individuals. This is done by directly comparing the target vector and the trailing vector that was created using the target vector as a base which can be seen as follows: [28]

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) > f(X_{i,G}) \\ X_{i,G} \end{cases}$$

This allows the population to remain constant and continuously growing in fitness.

3.2 Simulation Results

In this section, the results of the geometric optimization will be studied. First, each gen variable will be seen individually while simultaneously analyzing is standard deviations. Then the objective function will be broken down, and each variable will be discussed and compared with the gen variables to understand better and validate the results. To obtain the following data, two simulations were considered where one was done with 40 individuals at 300 generations and the other with 40 individuals and 600 generations. The data presented will be mostly from the 600 generation simulation, and only in the joint subsection, the 300 generation data will be shown as a means of comparison and a better understanding of the proposed algorithm and fitness function.

3.2.1 Manipulator Length Results

The first segment that will be studied from the manipulator can actually be considered the second segment. This is due to it being preceded by a joint which, in turn, has to be fixed on a stable base. As was shown in chapter 2 the real first segment is a massive block of material that has a totally different configuration when compared with the rest of the segments. For ease of understanding from this point on the stable base will be referred to as L0 or segment zero and will not have an optimization method applied to it.

When observing the following graphs obtained during the optimization of L1 for 600 generations, it is possible to see a convergence of the values around the 0.5m length. This convergence is further observed in the standard deviation where the value is steadily dropping after an initial growth where the variables scout all possible values before converging.

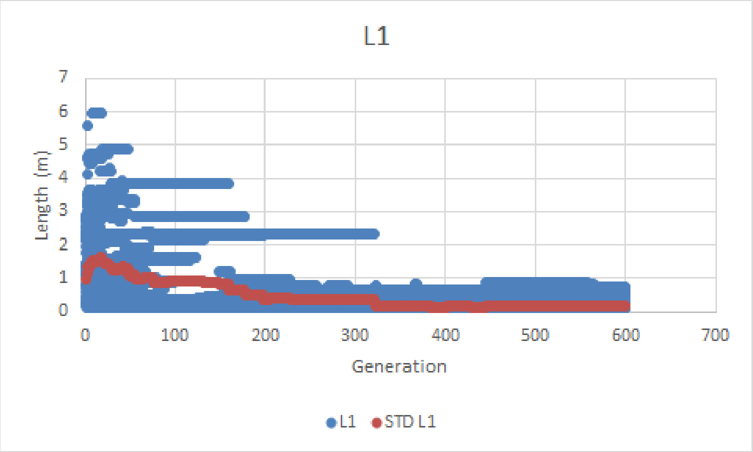


Figure 3.12: Analysis of the length from the first segment of the robotic manipulator

Similarly, with segment one, segment 2 or L2 has a converging graphic with a supporting standard deviation. When comparing this graphic with the previous one, there are very few differences between them.

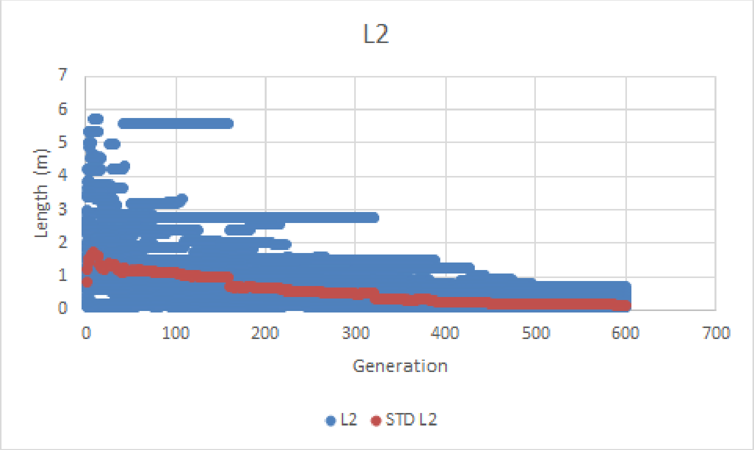


Figure 3.13: Analysis of the length from the second segment of the robotic manipulator

The following two segments present very similar graphics behavior with deviations also remarkably similar to the previous results.

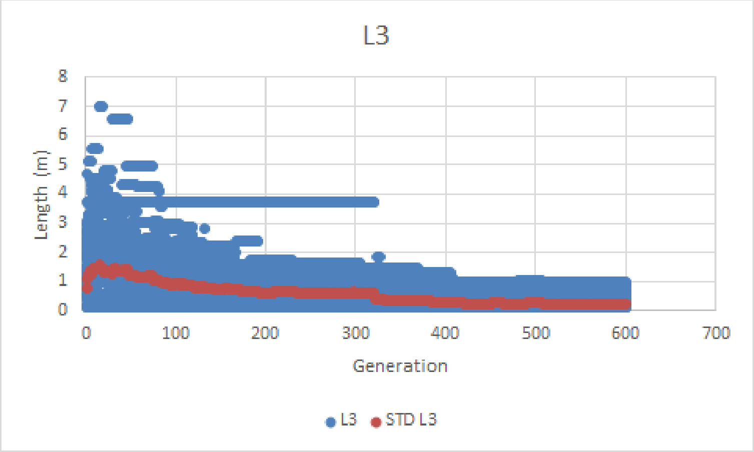


Figure 3.14: Analysis of the length from the third segment of the robotic manipulator

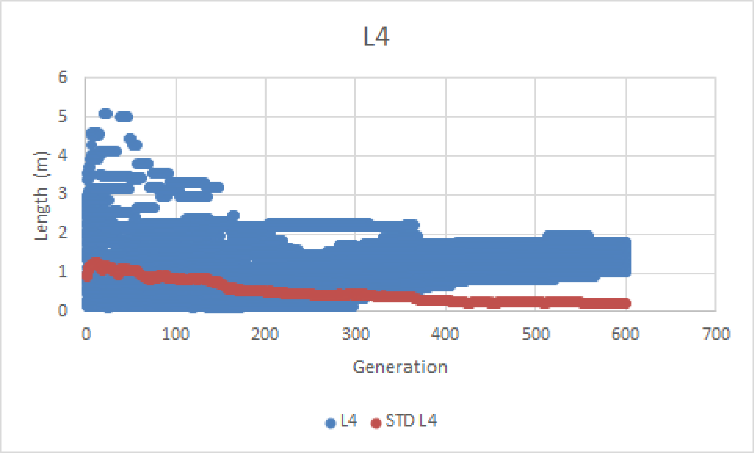


Figure 3.15: Analysis of the length from the forth segment of the robotic manipulator

The fifth segment posses the graphic that most shows a convergence coming from a very diversified base. This can also be seen in all other graphics, but only in this one a population of very different value survived for over 150 generations. This shows an in-depth analysis of the possibilities and relativity straight forward convergence process.

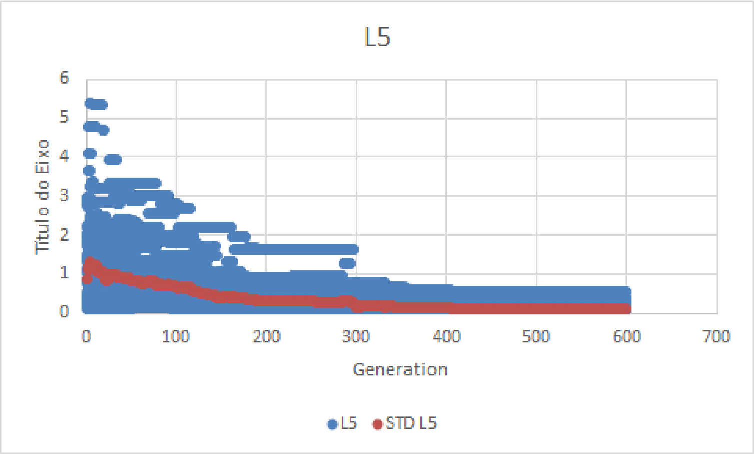


Figure 3.16: Analysis of the length from the fifth segment of the robotic manipulator

Finally, the Sixth segment, like all previous values has a well-behaved descent of the standard deviation and a narrowing of the overall values.

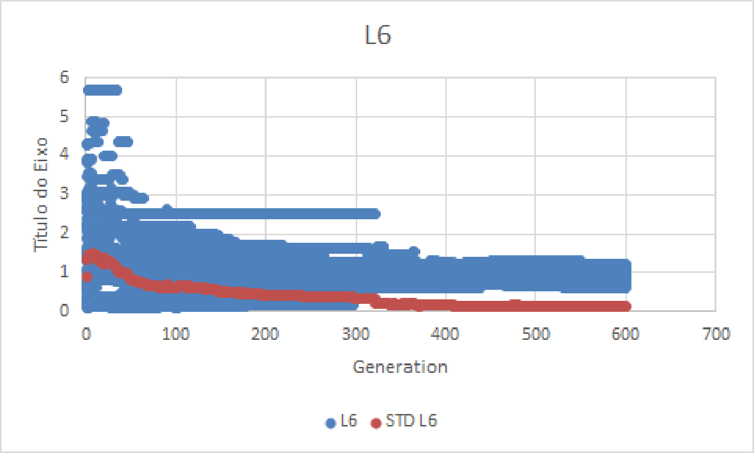


Figure 3.17: Analysis of the length from the sixth segment of the robotic manipulator

After observing all length graphics, it is possible to conclude that the overall lengths seem to be converging and to values that seem to be in the smaller range of the possible sizes. Thus the optimization of all variables appears to be working in minimizing their values as wanted.

3.2.2 Joint Results

Like the length values, all values for the joints already converged to a specific value, but a few took more generations to do so. For these variables, the values obtained for the 300 generation simulation will be presented and compared with the values obtained for the 600 generations as described previously.

This is not the case of the first joint as it has a possible value of 3, which correlates to the Z axis. The deviation graphic shows that it was one of the last joints to converge at over 300 generations.

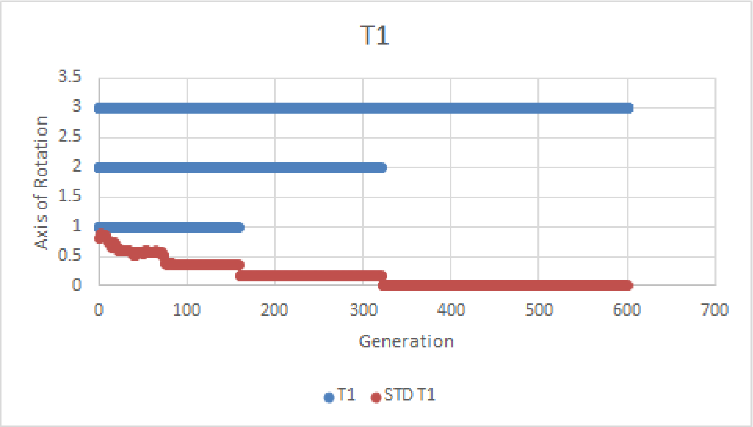


Figure 3.18: Analysis of the first joint of the robotic manipulator

When comparing this value to the other case with 300 generations, it is possible to see that it maintains the same behavior throughout all points, as for example sequence of elimination of the joints and the converged joint axis of rotation.

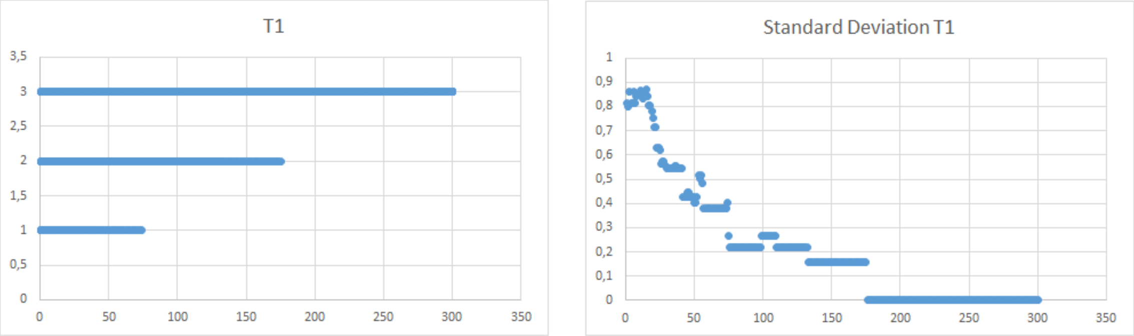


Figure 3.19: Analysis of the first joint of the robotic manipulator in the simulation with 300 generations

The second joint was one of the first joints to converge with only a few generations and did not show any difficulties in doing so.

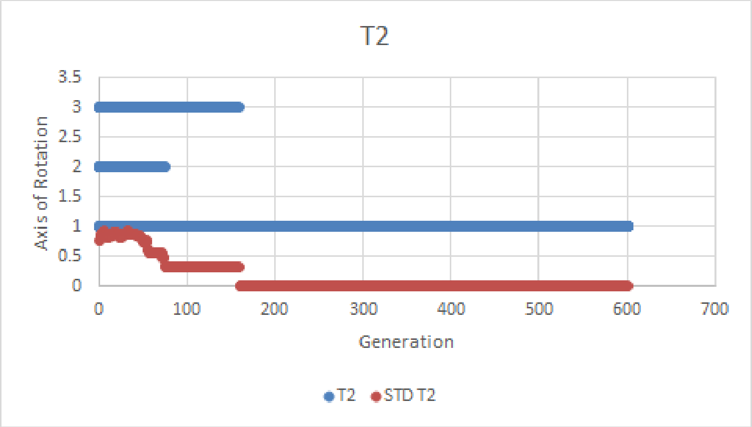


Figure 3.20: Analysis of the second joint of the robotic manipulator

The comparison with the 300 generation simulation it is possible to see the joints that were eliminated first changed, but the overall result was the same. It is also possible to see that the values for the standard deviation appear to be similar although the time it took to converge changed.

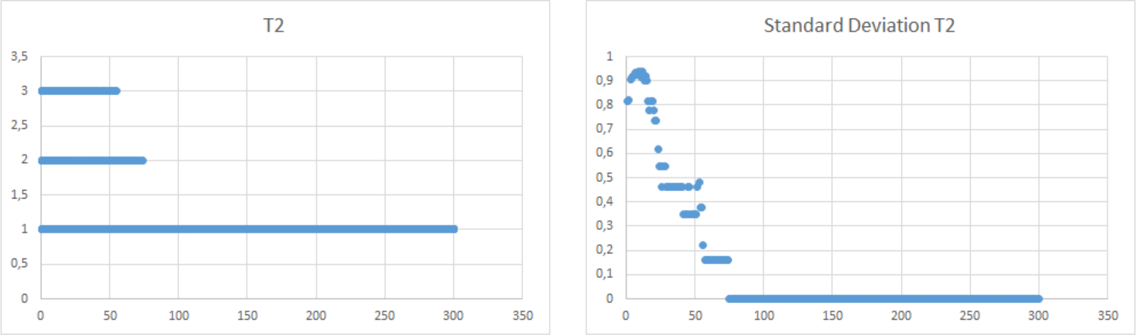


Figure 3.21: Analysis of the second joint of the robotic manipulator in the simulation with 300 generations

With a very similar graphic to the first joint, the third slowly decreased its divergence and converged at over 150 generations.

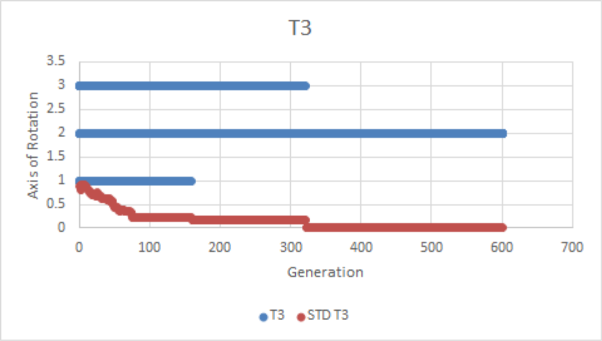


Figure 3.22: Analysis of the third joint of the robotic manipulator

The comparison of the third joint reinforces what was seen previously in the other joint cases where both the variables value and the deviation have the same behavior.

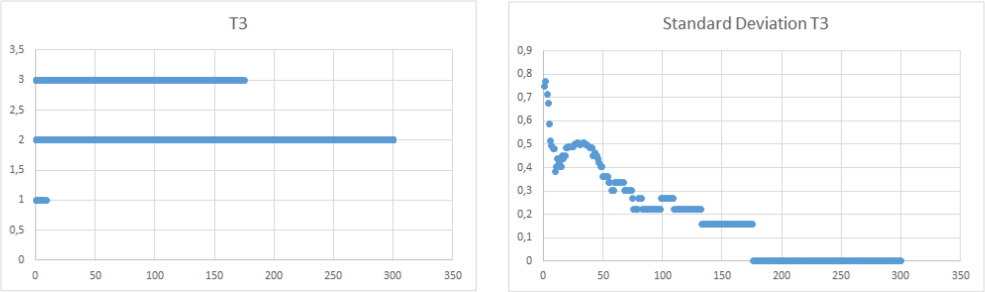


Figure 3.23: Analysis of the third joint of the robotic manipulator in the simulation with 300 generations

The fourth-gen presents the most significant difference between each of the simulations. For the 600 generation case, there was a rapid convergence with little to no problem. On the other hand, the 300 generation case still had not converged after 300 generations. While there is a big difference, it was possible to observe that the final convergence configuration was going to be the same for both cases as there were over 39 individuals with this gen as ”3” and only one individual still possessing this gen as ”2”.

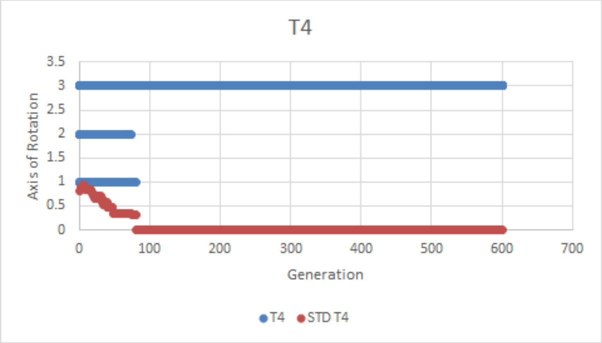


Figure 3.24: Analysis of the forth joint of the robotic manipulator

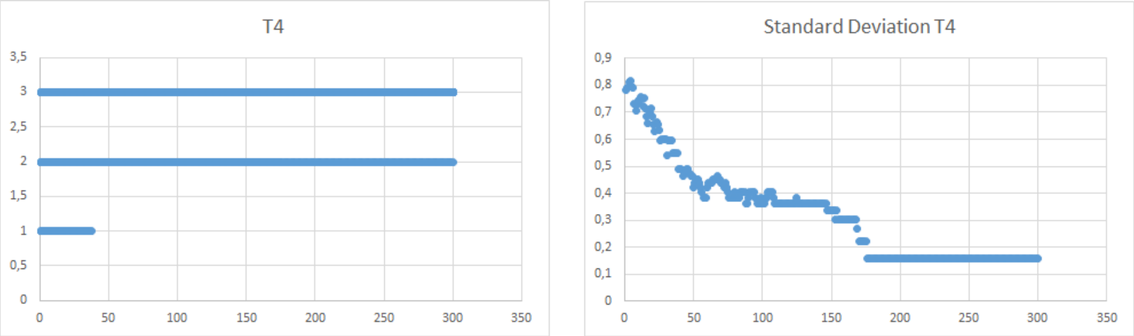


Figure 3.25: Analysis of the forth joint of the robotic manipulator in the simulation with 300 generations

The following joint has a strange behavior when analyzing the deviation where it first has a continuous and slow minimization, but at around generation 300, there is a step to the final convergence. This behavior can be due to the relations between the fitness functions variables, and at a certain point, a variable becomes dominant and thus stabilizing the number of individuals with that gen.

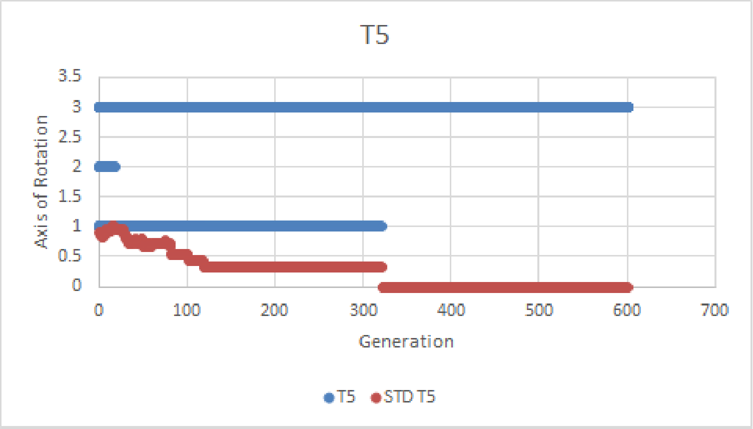


Figure 3.26: Analysis of the fifth joint of the robotic manipulator

The same behavior described previously can be seen with the 300 generation simulation, but with a major difference in the time, it took for this variable to converge with the 600 generation case taking over 6 times the amount of generations to converge.

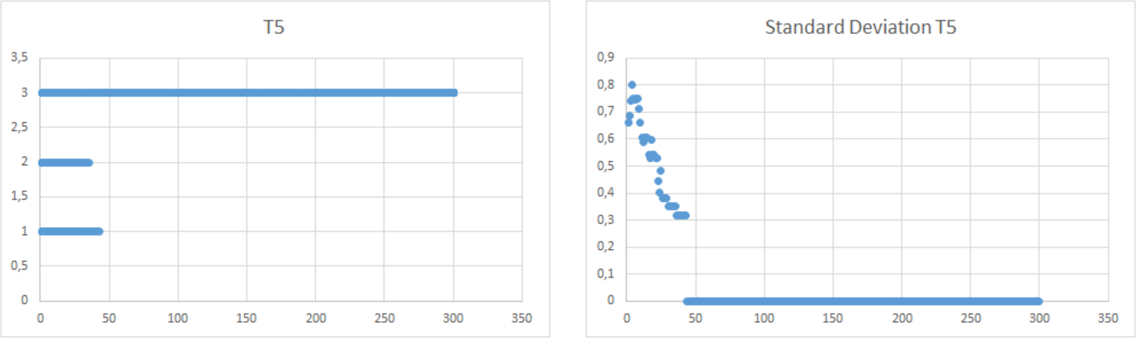


Figure 3.27: Analysis of the fifth joint of the robotic manipulator in the simulation with 300 generations

Finally, the last joint presents an alteration to the previous gens where the standard deviation has a continuous behavior, but that does not constantly minimize. This can be seen as a parabola with the maximum value being at around generation 270. Although it has this behavior, it finally converges at with a step at around generation 440.

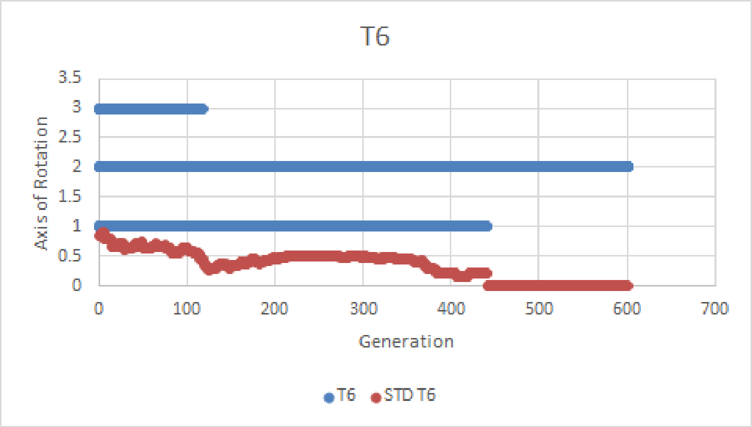


Figure 3.28: Analysis of the sixth joint of the robotic manipulator

Compared with the 600 generation case, the 300 generation case has a completely different behavior, converging without much problem in under 100 generations. The only discrepancy is the appearance of new gen at around 140 generations. This is most likely due to a mutated gene that was able to stay alive for a few generations before die out.

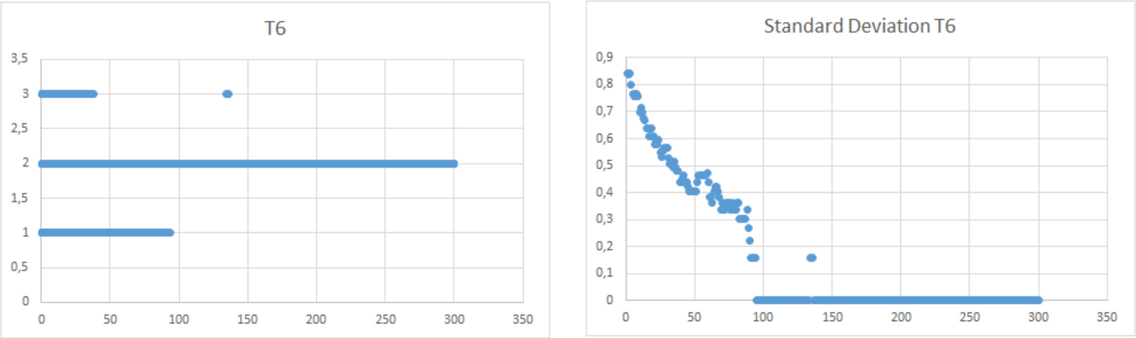


Figure 3.29: Analysis of the sixth joint of the robotic manipulator in the simulation with 300 generations

When comparing the values obtained between the two simulations explained previously, it is possible to see that the generated results present the same values for the joints. Thus it is possible to conclude that the algorithm converges to, at least, local maximums that are not dependent on the initial population and converge to the same values.

3.2.3 Base Coordinate Results

Different then the length optimization, the base coordinates do not have to converge to the smallest value, thus in the following graphics, the only necessity is to observe the variables converging to a single point, independently where they are in the 3-D space.

As seen in the following graphics both variables are converging, and there are no further observations to be made with regard to there development.

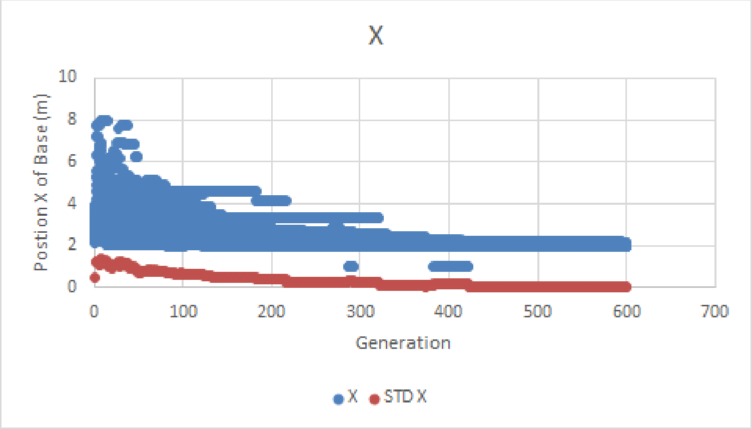


Figure 3.30: Analysis of the X position of the base of the robotic manipulator

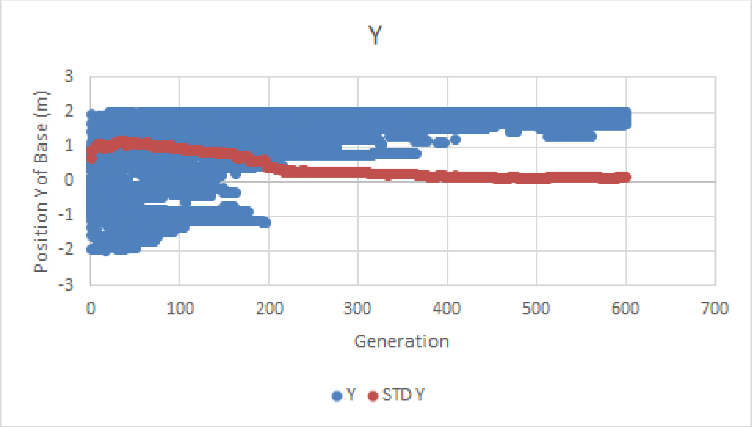


Figure 3.31: Analysis of the Y position of the base of the robotic manipulator

Out of the initial variables, the Z-axis probably has the worst convergence. This is likely due to the length variable have a direct correlation to this variable thus when one grows, the other can shrink in the same proportion and have the same overall value, this makes it difficult for the algorithm to converge. Although the convergence rate is slow, the standard deviation is lowering, demonstrating that there probably will have convergences if the algorithm was left to enough generations.

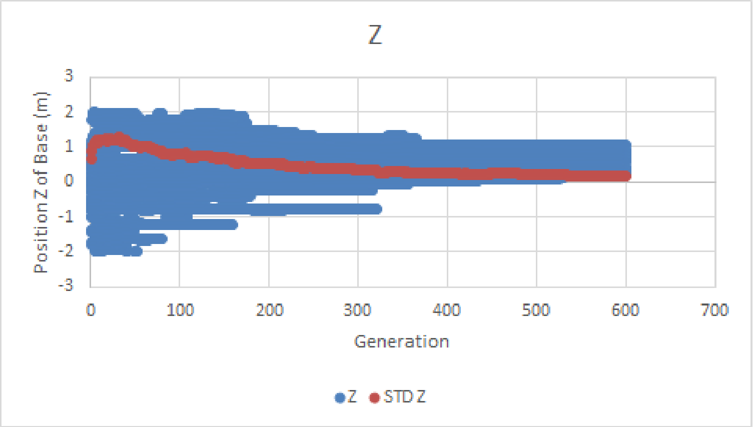


Figure 3.32: Analysis of the Z position of the base of the robotic manipulator

3.2.4 Fitness Results

With regards to the fitness function, it is necessary to observe that the values presented were made with α values of 1 for all variables, thus eliminating a dominant variable and obtaining equal importance for all variables. The standard deviation will be referred to in the figures as STD.

The first value to be observed is Cost. This value presents a gradual growth in the overall value and a standard deviation that slowly decreases in value, but never achieves a full convergence. This may be due to the limited amount of generations that were used or the erratic behavior of the chosen variables that all have the same α thus creating competing variables that do not converge in this limited amount of generations. Finally it is possible to observe a slight growth in the standard deviation at the end of the generations at around 250, but no conclusions can be made but, observing the gens it is possible to speculate that the appearance of individual colonies in the population can be responsible for this and, given enough time this deviation could disappear.

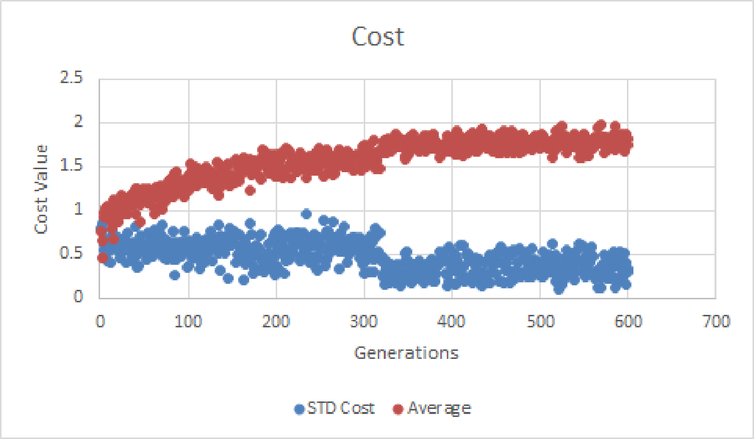


Figure 3.33: Analysis of the cost change in 600 generation with 40 individuals.

To better understand this graphic the maximum value in each generation was plotted with the standard deviation and it is possible to observe a continues and visible growth of the best individual until generation 300 where the function starts to flatten and stabilize.

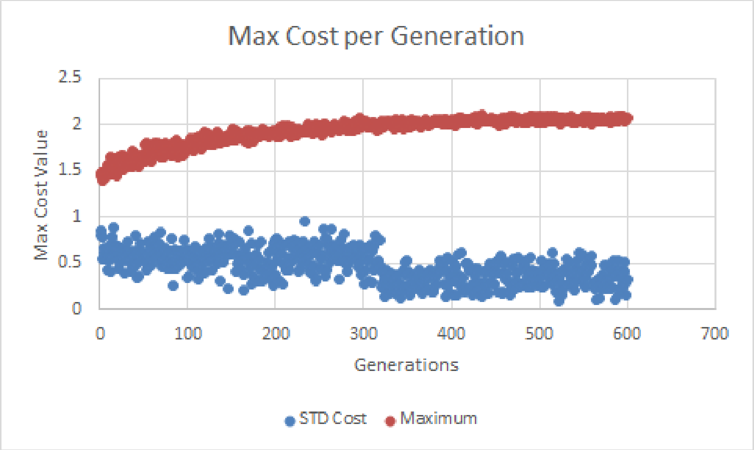


Figure 3.34: Analysis of the maximum cost change in 600 generation with 40 individuals.

The second value that was analyzed was dexterity, and out of all variables, although it has a steady growth in value, it also presents a growth in divergence. This is not something that should occur but is most likely due to the small number of points used in mapping the turbine blade, due to the small computational power available. This makes for an extremely discrete problem that presents more problems in converging. It is also possible to be a result of the interaction of the dexterity with its gradient as they are opposing values that have a direct influence in one another. It is also important to observe that the growth of this variable is the biggest out of all other variables going from initial values of around 0.05 to final values of over 0.4.

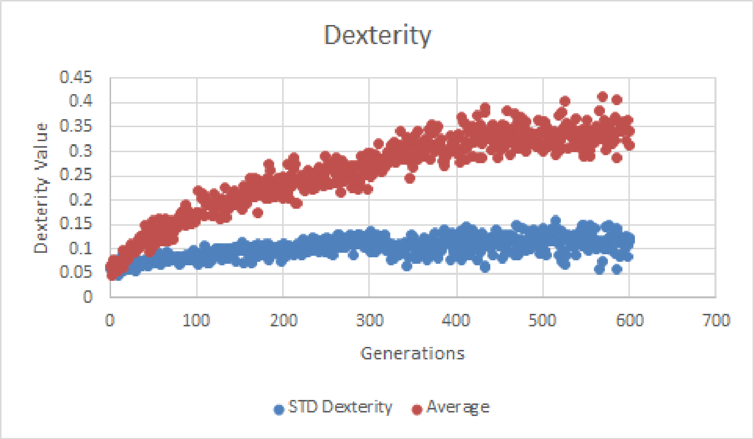


Figure 3.35: Analysis of the Dexterity change in 600 generation with 40 individuals

The following variable that was analyzed was the dexterity gradient. Observing these variables graphics, it is possible to see the concentration in the values and stability throughout all generations. During the implementation of this variable, there were a few cases that appeared where the manipulator had fewer than 2 points that could be reached. This caused the condition where no gradient could be calculated, thus it was necessary to implement, in these cases the max value for the variable which had value one. This can not be seen directly in the graphics but only in the steady decline of the standard deviation and concentration of the average of this variable.

It is also possible to observe that the values are systematic and regular. This may be due to the application of a built-in gradient method in Numpy. No further investigation was made as this should not affect the evaluation of the variable as a systematic decrease in the deviation is visible.

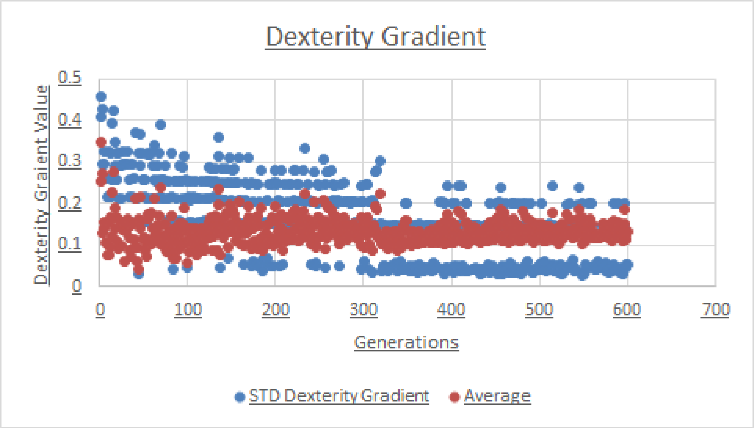


Figure 3.36: Analysis of the Dexterity Gradient change in 600 generation with 40 individuals

Out of the four variables, as this one was done in a discretized space, its graphic is harder to understand if not presented with the standard deviation by its side. With both graphics, it is possible to see that the value is converging and at the upper end of the painting percentages. This proves to be of great value as it shows that it is possible to paint the complete hydroelectric turbine with a relatively small robot.

There is also a direct relation in the standard deviation of this variable with the cost where the behavior of there graphics is the same.

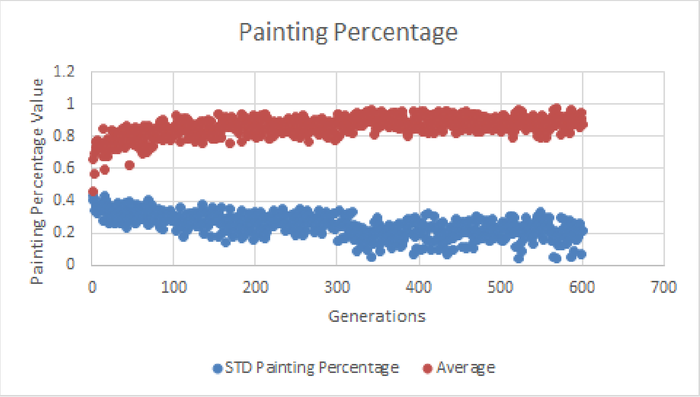


Figure 3.37: Analysis of the painting percentage change in 600 generation with 40 individuals

Finally, the variable that had the least problems in converging and having a well-behaved graphic is the length. This simple and straight forward behavior compared with the previous graphics is most likely due to the fact that this variable is not directly dependent on the inverse kinematics to calculate its value. All other values were completely nonlinear with very complicated and bad behaving equations while this variable is not submitted directly to these problems.

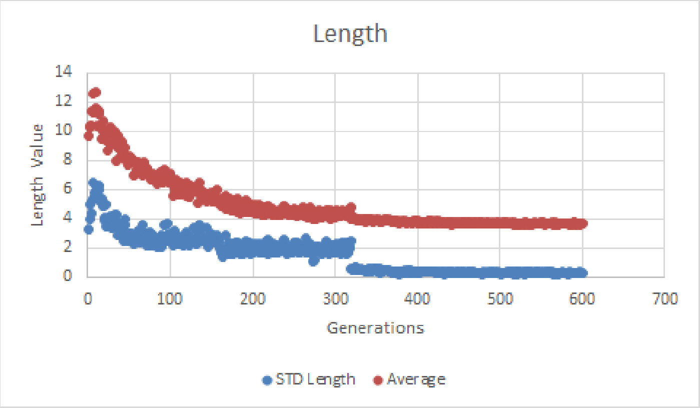


Figure 3.38: Analysis of the length change in 600 generation with 40 individuals.

To better evaluate the results presented in this chapter the worst manipulator from the first generation and the best individual from the last were used to illustrate how the algorithm developed. In the following figures, the blue points represent the points where the manipulators end effector can arrive at, and the red points show the points it cant. The robotic manipulator is shown in green where the black point represents the base position.

Then observing these figures the results of the optimization process are clear and can easily be observed. The number of red points decreased dramatically with only a limited number of red points. It is important to observe that there are over 200 points in these simulations where the algorithm had only 40 points to work with and even with this limitation the convergence is possible to observe.

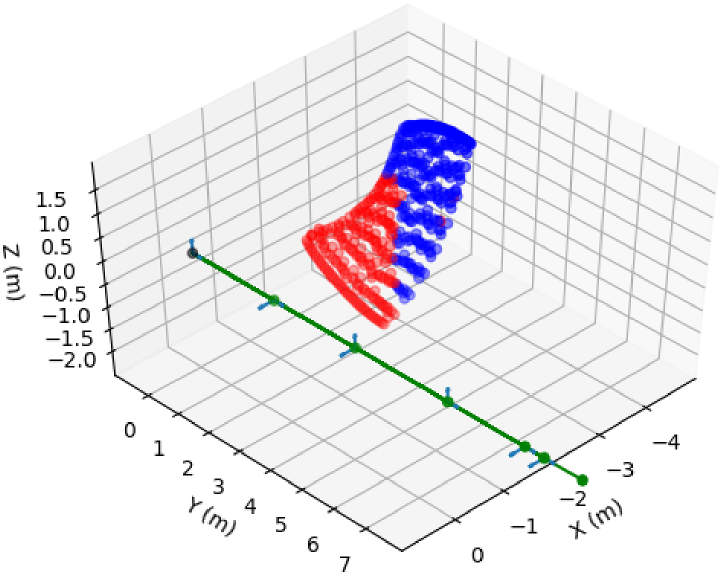


Figure 3.39: Representation of the performance of the worst robot from the first generation

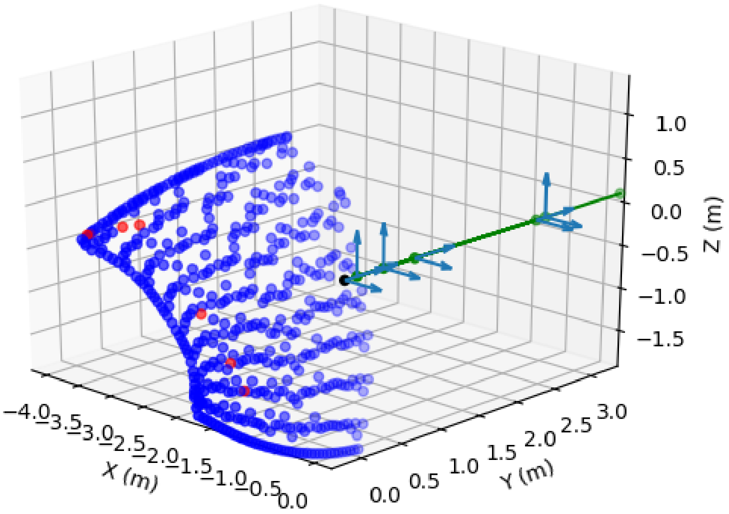


Figure 3.40: Representation of the performance of the best robot from the last generation

For these following figures, the arrows indicate the plane of rotation of each joint, the green points the point where the joints are located and the black point the initial manipulator position. Thus it is possible to observe that the worst robot has an axis rotation, from the base of: ("X","Z","Y","X","Z","Z") and the best robot has a rotation of:("Z","X","Y","Z","Z","Y").

From these figures, it is possible to observe the decrease in size of the manipulator demonstrating the optimization of this variable visually.

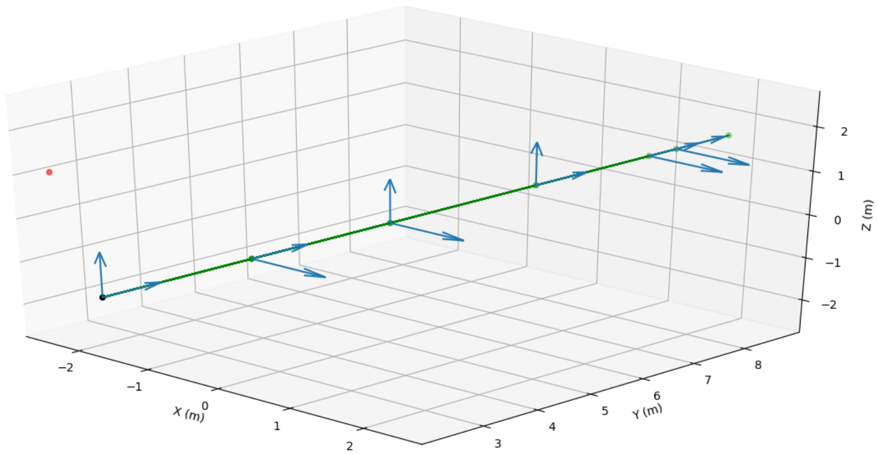


Figure 3.41: Physical representation of the worst robot from the first generation

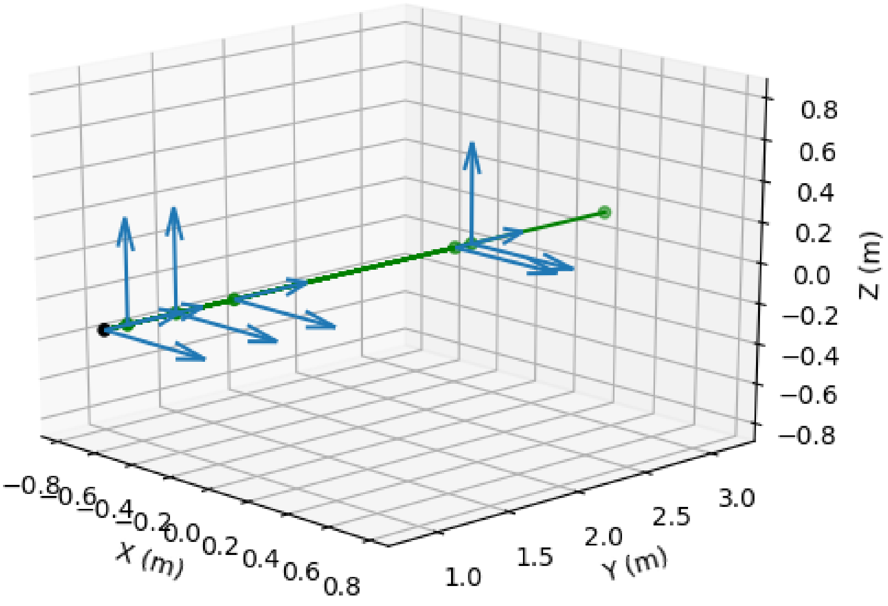


Figure 3.42: Physical representation of the best robot from the last generation

Chapter 4

Structural Optimization

In the recent years, structural optimization has developed and flourished due to the increase in computational capacity and programs that are powerful and simple to use in such a way that allows the users to carry out complex optimization task without extensive formal training. This has permitted the entire branch of structural optimization to merge and become a mainstream technique.[44]

In this thesis structural optimization was deemed necessary due to the large size that is required of the manipulator to allow it to accomplish the designated task as seen in chapter 3. This length causes extreme moments at the base of the manipulators and forces the use of bigger motors, both of which can be minimized if the structure is light2. Although it would be possible to use special motors made just for this robot as the substantial loss in money allows for more significant investment to solve the problem, this thesis will value options that use out off the self components when possible thus reinforcing the need of structural optimization.

As explained in chapter 1 there are many ways to optimize a structure ranging in complexity and overall objective. The chosen optimization group is the parametric optimization that focuses on changing the dimensions of components, for example, the size of the walls of a tube as seen in chapter 1.

Inside this optimization category, there are again many ways to deconstruct the problem to achieve the optimal solution. As the method of differential evolution has already been used and has proven results in the geometric optimization task, that is the tool that will be used.

In this chapter the optimization process will be discussed, followed by the structural alterations that will have to be made to allow motor placement and finally the simulation results will be explored.

4.1 Parametric Optimization

As stated previously, parametric optimization will be used to optimize the manipulator structure, but simply optimization is a vague term and needs further refinement to determine what aspects are fundamental to this case.

In the problem at hand, it is possible to see two critical criteria that could cause problems: weight and rigidity. To optimize these two that is, to reduce the weight but increase the rigidity of the structure, could be considered the initial problems to handle. However, this leads to a problem where the function wants to maximize one variable and minimize another leading to a series of potential problems. To counter these problems, the rigidity will be substituted by the deflection.

This change in objective allows the cost function of the optimization problem to search for the minimum of the deflection and weight, thus creating: $Cost = deflection + Weight$ where the minimum of the function is the objective.

As this thesis presents a methodology in optimizing robotic manipulators and uses the jirau blade hard-coating as a case study, the weight and deformation alone in this chapter will not be of greater use in creating a robust methodology. However, if combined with the relation between the moment of inertia of each cross-section and the density to strength ratios, it becomes a powerful and versatile tool, as engineers will have a strong base to create any cross-section desired.

Before continuing with the structural problem, it is necessary to analyze the possibility of applying structural optimization into the geometric optimization equation. This possibility was studied and only with a bigger view of the problem at hand was it possible to arrive at a satisfying conclusion:

Initially, the geometric optimization problem involves four variables, thus applying a fifth would increase the complexity of the problem, but this alone does not prevent the use of a structural variable, as the algorithm is sufficiency robust to withstand another variable. The second problem is the use of inverse kinematics that increases the time used by the optimization process up to four days to arrive at an answer. This would only increase if there where the need for taking a structural optimization cycle inside every robot created by the geometric problem. Finally and more importantly the geometrical optimization is based on discrete points and not a trajectory as discussed previously. This means that no real movement between points is done and with that, no joint configurations can be analyzed to determine the forces applied and torques for each moment.

Differential Evolution

To create the Differential Evolution optimization, the same steps of chapter 3 will be followed but naturally with other variables. The first change is the variables for the gens, where the internal diameter of each segment will be one gen. These variables were chosen due to the increase to the moment of inertia caused by having the largest outer diameter allowed; thus the outer diameter is fixed and the length of each section being predefined in chapter 3.

As described previously the fitness function will be a minimization one where the deflection and weight are wanted. Finally, as seen in chapter 2 the equations used to calculate the deflection and weight will have to be determined and applied in the optimization methodology.

As previously done there will need to be a constant α that will be used to generate a Pareto front.

Weight

The first and primary objective is to minimize weight. To calculate the weight of the structure the following equation of a tube was used.

$$M = \frac{\pi(d_o^2 - d_i^2)l\phi}{4g}$$

Where d_o is the outer diameter, d_i is the inner diameter and gen, g is gravity, and finally, ϕ is the density of the material with a value of $2700kg/m^3$ [38]. This value refers to aluminum, and it was chosen as the initial candidate as it is commonly used in the industry for this use. This value can be altered at will to any isotropic material.

The previous equation gives the weight of each section. Thus it is necessary to sum the weight of each part to obtain the total weight of $M_{Total} = \Sigma M_i$.

To create the variable for the fitness function, it will be necessary to normalize the value so it can vary between 0 and 1 and have a similar effect to the optimization problem as the deflection. To do this, it is indispensable to define the largest weight that a structure can obtain. For this case, the structure needs to be a cylindrical billet and not a tube, and as the outer diameter is fixed, it is possible to use the following equation to determine the total weight of the structure:

$$M_{max} = \frac{\pi d_o^2 l\phi}{4g}$$

Utilizing these two equations, it is possible to create the normalized value for the weight $M_{Normalized} = \frac{M_{Total}}{M_{Max}}$. This value will be the one used in the fitness function.[38]

Deflection

To determine the Deflection, Castigliano's theorem will be used and the equations determined in chapter 2 will be condensed in to the following equation:

$$\delta = \int_{0}^{l1} \frac{1}{EI}\left(M\frac{\partial M}{\partial F}\right) + \int_{l1}^{l2} \frac{1}{EI}\left(M\frac{\partial M}{\partial F}\right) + ... + \int_{l_n}^{l_{n+1}} \frac{1}{EI}\left(M\frac{\partial M}{\partial F}\right)$$

As done with the weight, it will also be necessary to normalize the deflection value. To do this the same structure made up of solid aluminum billets used in the maximum weight equation will be used and the same force configurations used in chapter 2 will be referenced and then summed to obtain the maximum deflection.

Before determining the deflections, it will be necessary to define the moment of inertia that will be used in all cases. For a billet the moment of inertia is defined by [38]

$$I = \frac{\pi}{64}(d_o^4 - d_i^4)$$

Thus for the maximum deflection for the point force at the end of the manipulator is:

$$y_{maxPoint} = \frac{Fl^3}{3EI}$$

For the deflections caused by the weight of the motors the sum would be:

$$y_{maxMotor} = \Sigma\frac{Fa^2}{6EI}(a - 3l)$$

Finally, the deflection for a distributed force, which is the representation of the weight of the structure is:

$$y_{maxWeight} = \frac{wl^4}{8EI}$$

These would be the logical and easy way to calculate the deflection of the structure, but due to the discrete nature of the structure where at every manipulator section, there is a new and unique cross-section and inner diameter, castiglianos theorem has to be used in all cases. For this, the equation of the point load and the motor are combined into one equation. This equation has the same appearance as stated previously.

Finally the weight of each body segment follows the same theorem of castigliano with the following equation:

$$\delta = \frac{w * x^2}{2} * \frac{-x}{2} * \frac{1}{E * I}$$

This equation is for the deflection of the first part of the structure and has to be summed with six other values regarding the weight of each segment.

Thus the maximum deflection will be $Y_{TotalMax} = y_{maxPoint} + y_{maxMotor} + y_{maxWeight}$ and consequently the normalized deflection will be determined by:[38]

$$Y_{Normalized} = \frac{Y_{Total}}{Y_{TotalMax}}$$

4.2 Structural Alterations

In chapter 3 the total lengths of each structural segment was determined. These values will then pass through the structural optimization to determine the width of each tube wall. In this section, a proposed alteration to the ends of the structure will be made with the intent of fixing the motors to the manipulator's bodies and vis-versa. The propositions made have the intent of briefly explaining the proposed approach to fix the motor and not determine the minor details of the structure.

As seen in chapter 2 first proposed solution is to remove the end of each segment and substitute it for a standardized structure as seen in the following figure.

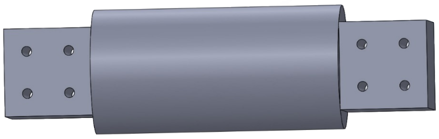


Figure 4.1: Rigid body representation that will be used in the optimization simulation

This initial concept is a simple representation of what is already done in other robots that utilize harmonic drive motors. In figure 4.2 it is possible to see that the motor is mounted on the base structure with the harmonic drive in between the base and next arm which means that the arm is only fixed on the reduction system.

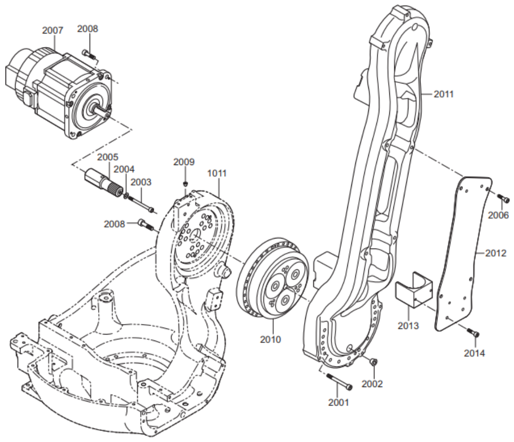


Figure 4.2: Representation of motor placement in the base of the structural body. Ref: techinal data bulletin of the Mh-12 received by the EMMA team.

To further analyze the possibilities of motor placement in different sections of a robotic manipulator, figure 4.3 as inspected.

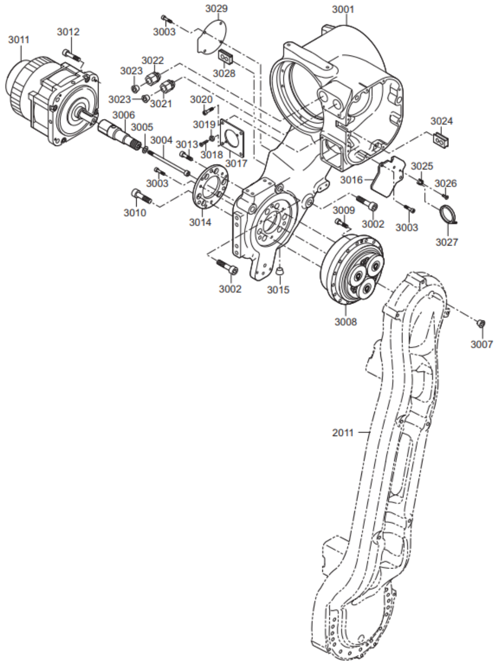


Figure 4.3: Representation of motor placement in repetitive segment of structural body. Ref: techinal data bulletin of the Mh-12 received by the EMMA team.

As seen in figure 4.2 the motor base is mounted on one structure while the gearing mechanism is sandwiched between two parts that move between them. This enforces the concept that the harmonic drive will take all the loads and are built to withstand these forces. Furthermore, it is possible to deduce by comparing the images, that the structure that will be created will have circular features that will locate the motor and gearing system while having large fillets that unite and reduce tension points between the tubular body and motor placement bracket.

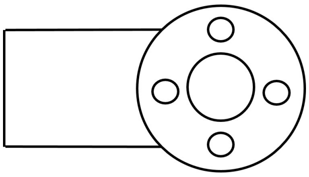


Figure 4.4: Final representation of motor placement in repetitive segment of structural body

To conclude the general motor mounting structure a catalog of harmonic drive motors can be seen in the appendix with mounting hole characteristics that will allow for further references when selecting a motor and a mounting configuration.

Just like the automatic calculation of segments inner diameters, the structural algorithm will be able to automatically chose the first option in motors by observing a pre-made list of motors. It will take into consideration the lightest motors will produce most torque from the list. A simple schematic can be seen as follows:

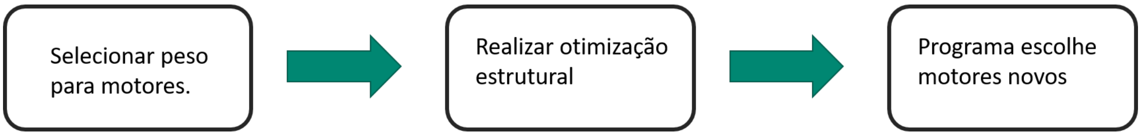

Selecionar peso para motores.
Realizar otimização estrutural
Programa escolhe motores novos

Figure 4.5: Simplified sequence representing the motor optimization process.

The following example shows the algorithm choosing two of the initial motors from the list that attend to the fundamental needs of the manipulator. In the example the first value refers to the maximum torque, the second to the weight of the motor, the third column represents the continuous stall torque, the torque is the motors name, and finally, the last column represents the manufacturer.

$$[204, 6.0, 140.0, u'CanisDrive - 25A - ULSIE', u'HarmonicDriveAG'],$$

$$[802, 13.2, 335.0, u'CanisDrive - 40A - ULSZE', u'HarmonicDriveAG'],$$

$$[], [], [], []]$$

These motors had their weights then introduced again in the algorithm, and a new result appeared with three motors being chosen.

$$[204, 6.0, 140.0, u'CanisDrive - 25A - ULSIE', u'HarmonicDriveAG'],$$

$$[372, 7.8, 216.0, u'CHA - 32A160', u'HarmonicDriveAG'],$$

$$[1180, 21.0, 736.0, u'CHA - 50CSZE160', u'HarmonicDriveAG'],$$

$$[], [], []]$$

The list with motors that were included in the appendix is just a small example of the actual list of industrial motors that can be obtained and so larger motors with higher gear ratios were not included, and thus cant be automatically selected currently.

4.3 Simulation Results

In this section, the results obtained from the optimization process will be analyzed and discussed. Furthermore, a simple program was created to evaluate the buckling of the optimized population. This will allow for a more in-depth review of what was done and better validated the structural results.

4.3.1 Parametric Optimization Simulation

During the structural optimization three different types of information where being considered: Function cost, deflection, and weight, where the first is composed of the two others. As seen in chapter 3 an optimization process should have all results converging to a single answer while having the cost function being minimized.

Before analyzing in depth the results obtained by the algorithm, a study was carried to determine the robustness of the algorithm and the sensibility of the method. To accomplish this, the number of generations and the number of individuals was altered, and the cost function analyzed. Due to the speed in which the method finalizes and not being applied to real-time simulations, the time was not taken into the equation.

In the following table, 12 different alterations were made where the first only the size of the population was altered then the same size of populations were tested for different quantity of generations.

	Generation	Population Size	Minimum Cost
Case 1	200	10	0,17678500298
Case 2	200	25	0,1767850328
Case 3	200	50	0,176785062
Case 4	200	80	0,1767850617
Case 5	150	10	0,176785955
Case 6	150	25	0,1767850703
Case 7	150	50	0,1767868822
Case 8	150	80	0,1767854897
Case 9	100	10	0,1767999309
Case 10	100	25	0,1767967552
Case 11	100	50	0,176831106
Case 12	100	80	0,1768950292

Table 4.1: Table with a robustness test to determine the effects that population size and number of generations cause to the cost function.

All tests made had a total duration of fewer than 10 seconds each and arrived at relatively similar answers. The minimum cost was chosen in the population of the last generation, and it was possible to observe convergence to almost the same value as changes only started to occur at $1 * 10^{-4}$; furthermore, it was possible to see that the less generation, the worse the overall cost became. It is also possible to see although the results degraded, the standard deviation of the smallest results were 0,000032618. This means the optimization method is robust for the specified case of structural optimization.

Manipulator Length Results

Before observing the results of the fitness function, it is necessary to observe the behavior of the inner diameter of each segment, as this will allow the variables in the fitness function to be better understood.

The studies were conducted with a generic model of the robot where each length segment had 1meter, and the motors had the same 20kg weight. This model was chosen as it reflexes approximately the overall size of the geometric manipulator optimized case while still having a geometric symmetry to compare the results between them better.

The first diameter, as expected, has the smallest inner diameter. This is due to it having to support the weight of the entire structure while its own weight being of the smallest impact on the deflection of the structure. In the standard deviation of this part is possible to see a total convergence of the results.

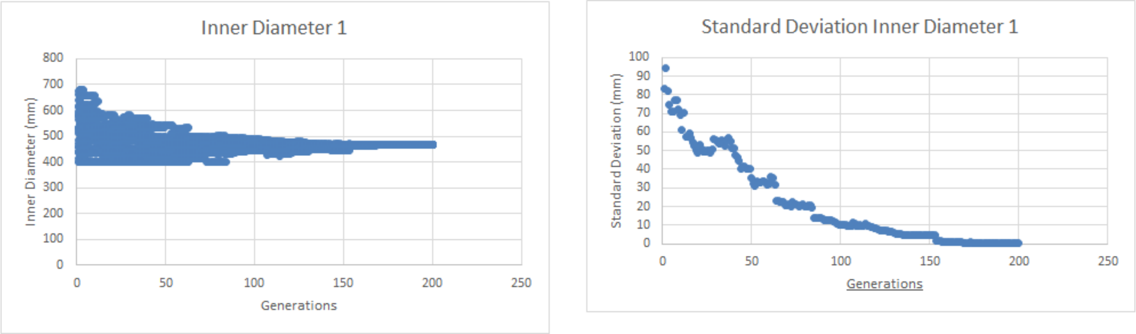


Figure 4.6: Analysis of the inner diameter from the first segment of the robotic manipulator

The following two inner diameters, present the same overall proprieties of the first while having a slightly larger inner diameter as predicted due to the motives explained previously.

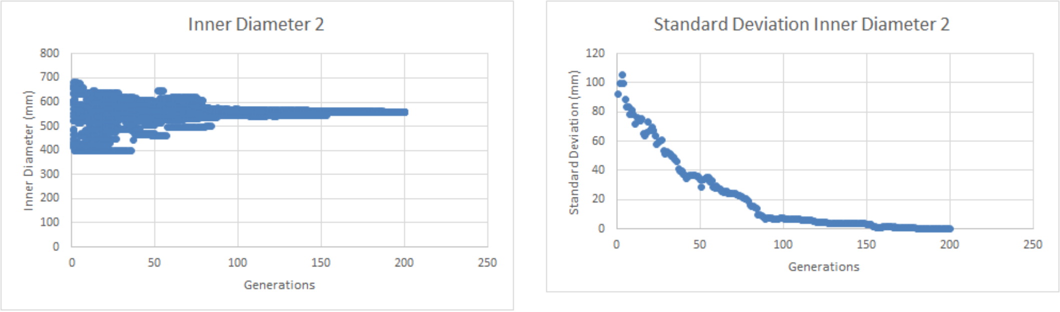


Figure 4.7: Analysis of the inner diameter from the second segment of the robotic manipulator

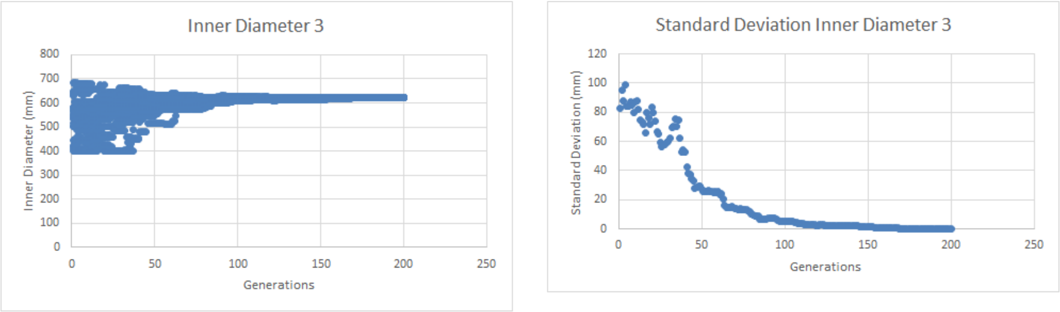


Figure 4.8: Analysis of the inner diameter from the third segment of the robotic manipulator

The fourth inner diameter has largely the same development than the previous two inner diameters, being the only difference a substantial discontinuity in the standard deviation. Although the difference, there are no consequences and the convergence is the same as all other genes.

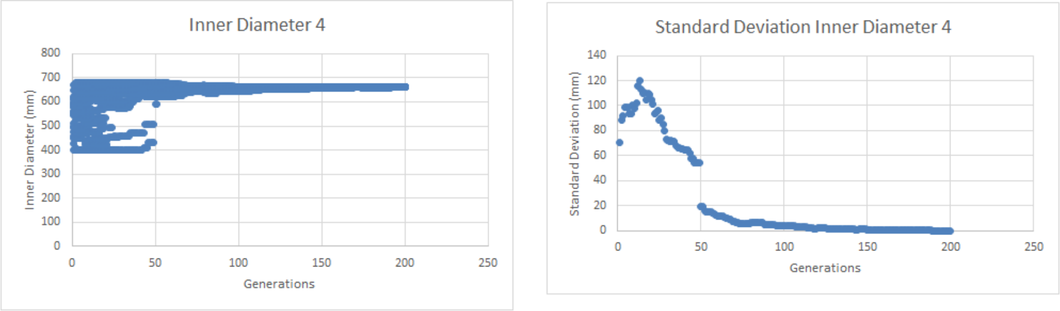


Figure 4.9: Analysis of the inner diameter from the forth segment of the robotic manipulator

In the case presented, with the restrictions previously described, the last two values, although following the same concept of the previous diameters as a converging standard deviation, had slightly the same inner diameter. This is due to the tip load generated by the hard-coating gun being merely 70N and the weight of the motor being only 200N. Had these values been higher there would have been a divergence in the inner diameter value.

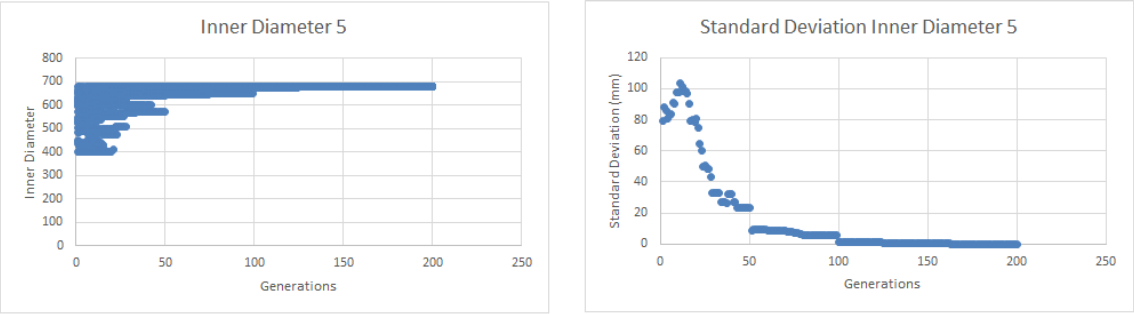


Figure 4.10: Analysis of the inner diameter from the fifth segment of the robotic manipulator

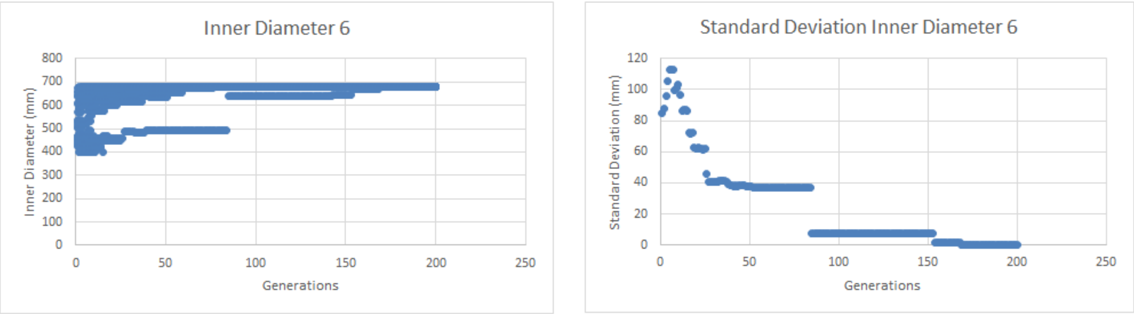


Figure 4.11: Analysis of the inner diameter from the sixth segment of the robotic manipulator

Thus it is possible to conclude that the inner diameter parameters have a behavior when converging that is expected and the information obtained analyzing these variables is invaluable to creating a robotic manipulator.

Fitness Results

Analyzing the three figures 4.12, 4.13 and 4.14, the first observation is that there is a convergence of all individuals at 150 generations, thus it is possible to conclude that there would only be a need to use 150 generations to achieve de same answer that was done with 200 and it is possible to use only 100 generations to achieve similar results.

Up until this point, all algorithms had their termination criteria the number of generations, but if done by applying a tolerance value for the standard deviation of that generation as a termination criterion, the number of generations would be minimum.

To do this, a termination criteria should be applied where one option is to determine that the standard deviation of the cost function should be higher than 0.0001 and for any value lower than that the simulation should stop. This method will be applied for further analysis, but for comparison with previous cases, the termination of the algorithm will be maintained at 200 generations.

The following figure represents the behavior of the cost function, where it is possible to see a broad base that ensures that all the possible population volume is being analyzed and thus permitting that most possible options are considered and the best chosen. It is also possible to see that the standard deviation tends to zero after only 80 generations demonstrating a relatively fast convergence to, most probably, a global optimum, or if not, a local optimum that is close to the global.

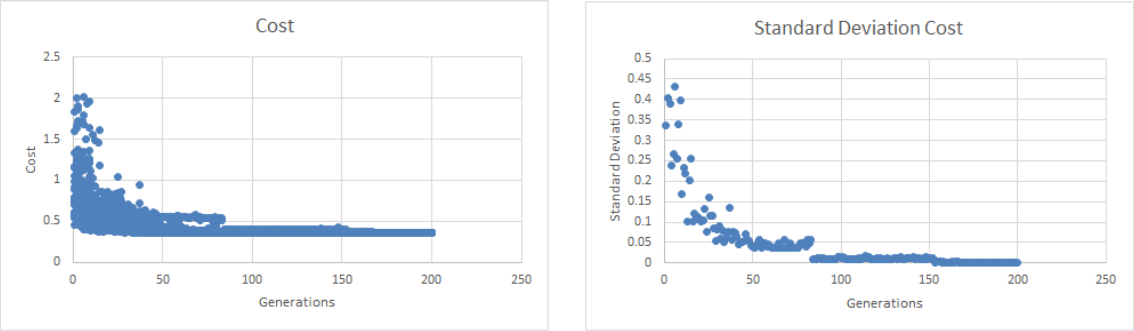


Figure 4.12: Analysis of the cost change in 250 generation with 25 individuals

As stated previously, the motors were considered with 20kg and placed in between each segment. There was also the consideration of equal segments of 1m in length. The following figures show the weight function with a converging behavior where, as the cost, the base has a throw range that gradually converges to an optimal answer.

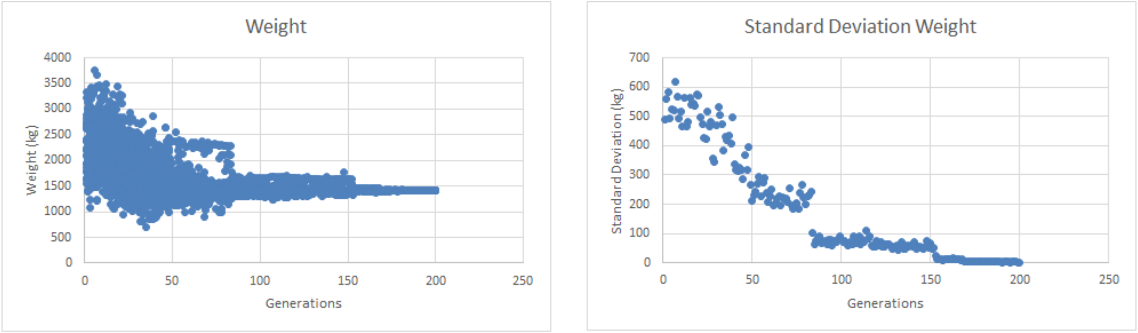


Figure 4.13: Analysis of the weight change in 250 generation with 25 individuals

Finally, the deflection is demonstrated, where, as the previous cases, the convergence is easily demonstrated by the graphics. For this deflection, it is essential to remember that all values are put in meters and the multi-objective function was chosen without variable weights with the objective of a clean and not a tendentious behavior of every one of the variables.

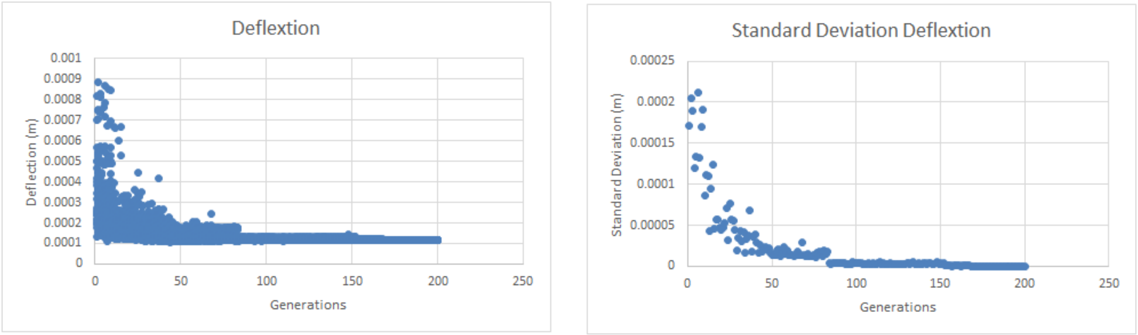


Figure 4.14: Analysis of the deflection change in 250 generation with 25 individuals

To finalize the study of the methodology, the Pareto front has to be determined but due to this thesis being the development of a methodology the chosen values for α should not be resolved as this is the job of the engineers to determine what they want more and not of the method.

Although the actual values will not be set, a simple theoretical example will be held to illustrate the jirau manipulator case and how the engineers should study the correct values for α.

First, it is necessary to recall the restrictions described in chapter 2. These restrictions are that the robot should be lightweight, but most importantly the total deflection cant is above 1.5mm. Then the Pareto front graphic is analyzed as follows.

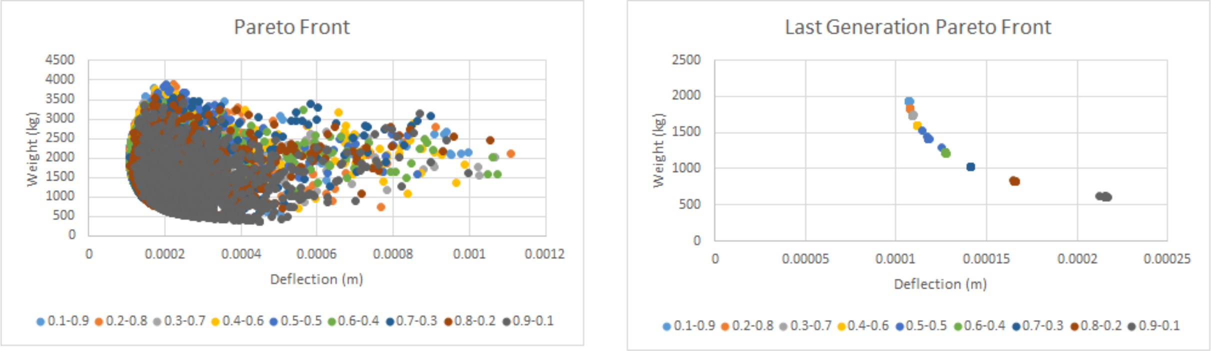


Figure 4.15: Analysis of the Pareto Front in 150 generation with 25 individuals

At first glance, it is possible to observe a Pareto front has a deflection of 0.8 mm for all necessary cases for α. This means the restriction in chapter 2 are complied with, and the weight will be the major choosing factor where the lightest possible population will be chosen, and thus series 9 with a 0.1 value for the α of deflection and a 0.9 value for the weight α will be chosen.

To finish the method the motors, that at the beginning of this simulation had the value of 20kg used as a base parameter, have to have their weights altered to a more realistic case. To do this the moment applied between each segment has to be calculated and then a motor chosen from the database that can be seen in the appendix.

Buckling Results

To be able to use greens equations as described in chapter 2, it is necessary to understand how it will be applied in the present case. First, the concept of greens equation is that a single cross-section beam is studied and its maximum length before buckling due to is own weight is determined. As this thesis presents a multi cross-section beam, the cross section that will have the biggest weight should be used, as all values below that will be supported and it presents a worst-case scenario. For the case at hand, this will be the base of the structure with around 450 of inner diameter and 700 of outer. Utilizing the following equations show in chapter 2:

$$l_{max} = (7.8373\frac{EI}{\rho g A})^{\frac{1}{3}}$$

It is possible to determine that the beam would have to have around 276756 m and thus the 6m structure will not buckle under its own weight.

In the case at hand, the buckling was foreseen as not critical, but as the methodology should be as robust as possible, the buckling will continue to be studied.

Chapter 5

Conclusions

As seen in each of the results, this thesis presents a robust methodology utilizing a differential evolution optimization in conjunction with a multiple objective weight function where the robotic manipulator had mostly converging results.

The geometric optimization has shown an increase in the objective function of over 45% from the first generation, and the structural optimization achieves a reduction of its objective function of over 40% its initial value. This means from a random combination of genes the final result was able to alter these values in a way that resulted in an improvement of 45% when observing the cost function.

With the buckling studies and the use of the moment of inertia in the structural optimization, it is possible to determine that the methodology is flexible and can be used in most cases that a robotic manipulator has to be optimized for a specific task.

On top of this, the methodology has increased robustness as it can address any quantity of degrees of freedom wanted. Furthermore, the concept of dexterity is further studied in this methodology to certify that the optimization of this parameter is done, rather than the optimization of a random dexterity result obtained by a simple Newton inverse kinematics equation.

5.1 Future Work

As further work, a motor database could be created to automatically chose the first example of a motor to be used in each segment of the manipulator. With this database, an optimized fixture system for each motor could be incorporated into the calculated optimized manipulator, and the structure analyzed to create a seamless combination of the two. There could also be the implementation of different types of joint combinations like prismatic and linear joints.

This structure analysis could also use the results present in this thesis to develop a structure that could incorporate vibration dampening if necessary and thus the implementation of natural frequencies in the problem.

Furthermore, the implementation of self-collision and an environmental collision could be incorporated into the problem to allow for complex environments and lead to further development of trajectory implementation in the problem.

Finally, the incorporation of a trajectory could be used to optimize the worst-case scenario of the manipulator and thus creating a more optimized and lightweight structure. This would involve also including a dynamic load calculation for a more robust equation.

Bibliography

[1] ESCALER, X., EGUSQUIZA, E., FARHAT, M., et al. "Detection of cavitation in hydraulic turbines", *Mech. Syst. Signal Process.*, v. 20, n. 4, pp. 983–1007, 2006. ISSN: 08883270. doi: 10.1016/j.ymssp.2004.08.006.

[2] KURODA, S., KAWAKITA, J., WATANABE, M., et al. "Warm spraying - A novel coating process based on high-velocity impact of solid particles", *Sci. Technol. Adv. Mater.*, v. 9, n. 3, 2008. ISSN: 14686996. doi: 10.1088/1468-6996/9/3/033002.

[3] FREITAS, R. S., SILVA, G. A., SOARES, E. E., et al. "State of the Art and Conceptual Design of Robotic Solutions for In Situ Hard Coating of Hydraulic Turbines", *J. Control. Autom. Electr. Syst.*, v. 28, n. 1, pp. 105–113, 2017. ISSN: 21953899. doi: 10.1007/s40313-016-0287-6.

[4] APPLICATIONS, G. "Motoman mh1 2, mh1 2f", pp. 2–3.

[5] YAMADA, T., IZUI, K., NISHIWAKI, S. "A Level Set-Based Topology Optimization Method for Maximizing Thermal Diffusivity in Problems Including Design-Dependent Effects", *J. Mech. Des.*, v. 133, n. 3, pp. 031011, 2011. ISSN: 10500472. doi: 10.1115/1.4003684. Disponível em: <`http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1450003`>.

[6] RINDI, A., MELI, E., BOCCINI, E., et al. "Static and Modal Topology Optimization of Turbomachinery Components", *J. Eng. Gas Turbines Power*, v. 138, n. 11, pp. 112602, 2016. ISSN: 0742-4795. doi: 10.1115/1.4033512. Disponível em: <`http://gasturbinespower.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4033512`>.

[7] SICILIANO, B., SCIAVICCO, L., VILLANI, L., et al. *Robotics: Modelling, Planning and Control.* 2009. ISBN: 9781846286414. doi: 10.1007/978-1-84628-642-1. Disponível em: <`http://books.google.com/books?hl=en{&}lr={&}id=jPCAFmE-logC{&}oi=fnd{&}pg=`

PR8{&}dq=Robotics+-+Modelling+Planning+and+Control{&}ots=3TMi0jIsuu{&}sig=VoMpDiIASZGcrAg7juLabQszoqE>.

[8] FISCHER, G. "Robotic Prostate Biopsy in Closed MRI Scanner", , n. February 2008, 2008. Disponível em: <http://oai.dtic.mil/oai/oai?verb=getRecord{&}metadataPrefix=html{&}identifier=ADA482672>.

[9] EVANS, J. H. "Basic Design Concepts", *J. Am. Soc. Nav. Eng.*, v. 71, n. 4, pp. 671–678, 1959. ISSN: 15593584. doi: 10.1111/j.1559-3584.1959.tb01836.x.

[10] TAGHIRAD, H. D., BELANGER, P. R. "Modeling and Parameter Identification of Harmonic Drive Systems", *Dyn. Syst. Meas. Control*, v. 120, n. December 1998, pp. 439–444, 2016.

[11] HWANG, S., KIM, H., CHOI, Y., et al. "Design optimization method for 7 DOF robot manipulator using performance indices", *Int. J. Precis. Eng. Manuf.*, v. 18, n. 3, pp. 293–299, 2017. ISSN: 20054602. doi: 10.1007/s12541-017-0037-0.

[12] NGATCHOU, P., ZAREI, A., EL-SHARKAWI, A. "Pareto Multi Objective Optimization", *Proc. 13th Int. Conf. on, Intell. Syst. Appl. to Power Syst.*, pp. 84–91, 2005. doi: 10.1109/ISAP.2005.1599245. Disponível em: <http://ieeexplore.ieee.org/document/1599245/>.

[13] PEREIRA, M. G., CAMACHO, C. F., FREITAS, M. A. V., et al. "The renewable energy market in Brazil: Current status and potential", *Renew. Sustain. Energy Rev.*, v. 16, n. 6, pp. 3786–3802, 2012. ISSN: 13640321. doi: 10.1016/j.rser.2012.03.024. Disponível em: <http://dx.doi.org/10.1016/j.rser.2012.03.024>.

[14] KNAPP, B. Y. R. T. "Cavitation and Nuclei", *Asme*, v. 80, pp. 1315, 1958.

[15] SATO, Y. S., NELSON, T. W., STERLING, C. J., et al. "Microstructure and mechanical properties of friction stir welded SAF 2507 super duplex stainless steel", *Mater. Sci. Eng. A*, v. 397, n. 1-2, pp. 376–384, 2005. ISSN: 09215093. doi: 10.1016/j.msea.2005.02.054.

[16] PANDILOV, Z., DUKOVSKI, V. "Comparison of the Characteristics Between Serial and Parallel Robots", *Fascicule*, v. 1, pp. 2067–3809, 2014.

[17] LAMB, T. "A Ship Design Procedure", *Mar. Technol.*, 1969.

[18] GEEM, Z., KIM, J., LOGANATHAN, G. V. "A New Heuristic Optimization Algorithm: Harmony Search", *Simulation*, v. 76, n. 2, pp. 60–68, 2001. ISSN: 00375497. doi: citeulike-article-id:2632511\rdoi:10.1177/003754970107600201. Disponível em: <`http://dx.doi.org/10.1177/003754970107600201`>.

[19] AB WAHAB, M. N., NEFTI-MEZIANI, S., ATYABI, A. "A comprehensive review of swarm optimization algorithms", *PLoS One*, v. 10, n. 5, pp. 1–36, 2015. ISSN: 19326203. doi: 10.1371/journal.pone.0122827.

[20] EIBEN, A., SMITH, J. *Introduction to Evolutionary Computing*, v. 2. 2004. ISBN: 3540401849. doi: 10.1162/evco.2004.12.2.269. Disponível em: <`http://download.springer.com/static/pdf/526/bok{%}253A978-3-662-44874-8.pdf?originUrl=http{%}3A{%}2F{%}2Flink.springer.com{%}2Fbook{%}2F10.1007{%}2F978-3-662-44874-8{&}token2=exp=1497015555{~}acl={%}2Fstatic{%}2Fpdf{%}2F526{%}2Fbok{%}25253A978-3-662-44874-8.pdf{%}3ForiginUrl{%}3Dhttp{%}25`>.

[21] RUTENBAR, R. A. "Simulated annealing algorithms: An overview", *IEEE Circuits Devices Mag.*, v. 5, n. 1, pp. 19–26, 1989. ISSN: 87553996. doi: 10.1109/101.17235.

[22] KOKASH, N. "An introduction to heuristic algorithms", *Dep. Informatics Telecommun.*, , n. August, pp. 1–8, 2005. doi: 10.1.1.105.8050.

[23] KONAK, A., COIT, D. W., SMITH, A. E. "Multi-objective optimization using genetic algorithms: A tutorial", *Reliab. Eng. Syst. Saf.*, v. 91, n. 9, pp. 992–1007, 2006. ISSN: 09518320. doi: 10.1016/j.ress.2005.11.018.

[24] LEIVA, J. "Structural Optimization Methods and Techniques to Design Efficient Car Bodies", *Vanderplaats Res. Dev. Inc*, 2011. Disponível em: <`http://bodyandassembly.com/wp-content/uploads/2012/01/G-Edited-Leiva-Paper.pdf`>.

[25] LEIVA, J. P. "Topometry Optimization : A New Capability to Perform Element by Element Sizing Optimization of Structures", *Genesis*, , n. September, pp. 1–15, 2004.

[26] REACH, E., LARGE, F. O. R., BENEFITS, K. E. Y. "Mh50 ii-20", .

[27] DAS, S., SUGANTHAN, P. N. "Differential evolution: A survey of the state-of-the-art", *IEEE Trans. Evol. Comput.*, v. 15, n. 1, pp. 4–31, 2011. ISSN: 1089778X. doi: 10.1109/TEVC.2010.2059031.

[28] BREST, J., GREINER, S., BOSKOVIC, B., et al. "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems", *Ieee Trans. Evol. Comput.*, v. 10, n. 6, pp. 646–657, 2006. ISSN: 1089-778X. doi: 10.1109/tevc.2006.872133.

[29] QIN, A. K., HUANG, V. L., SUGANTHAN, P. N. "Differential evolution algorithm with strategy adaptation for global numerical optimization", *IEEE Trans. Evol. Comput.*, v. 13, n. 2, pp. 398–417, 2009. ISSN: 1089778X. doi: 10.1109/TEVC.2008.927706.

[30] LANIUS. "Tinyik: a simple and naive inverse kinematics solver for Python". 2017. Disponível em: <`https://github.com/lanius/tinyik`>.

[31] BUSS, S. R. S. "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods", *Univ. California, San Diego, Typeset Manuscr. ...*, v. 132, n. 4, pp. 1–19, 2004. ISSN: 0306-4522. doi: 10.1016/j.neuroscience.2005.01.020. Disponível em: <`http://math.ucsd.edu/{~}sbuss/ResearchWeb/ikmethods/iksurvey.pdf{%}5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.73.8406`>.

[32] DIANKOV, R., KUFFNER, J. "OpenRAVE : A Planning Architecture for Autonomous Robotics", *Robotics*, , n. July, pp. –34, 2008. doi: CMU-RI-TR-08-34. Disponível em: <`http://www.ri.cmu.edu/pub{_}files/pub4/diankov{_}rosen{_}2008{_}2/diankov{_}rosen{_}2008{_}2.pdf`>.

[33] TOOLBOX, T. R. "A Robotics Toolbox", *Robotics*, , n. March, 1996. doi: 10.1109/100.486658.

[34] ROSÁRIO, J. M., DUMUR, D. "Multi-Objective Design of Parallel Manipulator Using Global Indices", *Open Mech. Eng. J.*, v. 4, pp. 37–47, 2010. ISSN: 1874155X. doi: 10.1109/AIM.2011.6026996.

[35] GALASSI, M., ROYROY, A., DE CARVALHO, G. P., et al. "DORIS - A mobile Robot for Inspection and Monitoring of Offshore Facilities", *An. do XX Congr. Bras. Automática*, pp. 3174–3181, 2014.

[36] KOBERG, J., &. B. J. *The universal traveler: A soft systems guide to creativity, problem-solving and the process of reaching goals.* 1991.

[37] INSTRUCTIONS, M. "INSTRUCTIONS", v. 00, pp. 1–103.

[38] BUDYNAS, R., NISBETT, K. *Elementos de Máquina de Shogley.* 2015.

[39] GREENHILL, A. G. “No Title”. In: *Proc. Cambridge Philos. Soc.,*, pp. 65–73, 1881.

[40] DUAN, W. H., WANG, C. M. “Exact Solution for Buckling of Columns Including Self-Weight Exact Solution for Buckling of Columns Including Self-Weight”, v. 9399, n. January 2008, pp. 1–5, 2014. doi: 10.1061/(ASCE)0733-9399(2008)134.

[41] BUDYNAS, R., NISBETT, J. *Shigleys Mechanical Engineering Design.* ISBN: 9780073529288.

[42] STORN, R., PRICE, K. “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”, *J. Glob. Optim.*, v. 11, n. 4, pp. 341–359, 1997. ISSN: 09255001. doi: 10.1023/A:1008202821328.

[43] CARAMIA, M., DELL’OLMO, P. “Multi-objective Optimization”, *Multi-objective Manag. Freight Logist.*, pp. 11–36, 2008. ISSN: 07384602. doi: 10.1007/978-1-4614-6940-7_15.

[44] VANDERPLAATS, G. N. “Structural optimization for statics, dynamics and beyond”, *J. Brazilian Soc. Mech. ...*, v. XXVIII, n. 3, pp. 316–322, 2006. Disponível em: <`http://www.scielo.br/scielo.php?pid=S1678-58782006000300009{&}script=sci{_}arttext`>.