Estimation of Contact Force on an Inflatable Structure Using Vision Sensors

António Pedro Pinheiro Carita Franco

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisor: Prof. João Carlos Prata dos Reis

Examination Comittee

Chairperson: Carlos Baptista Cardeira
Supervisor: Prof. João Carlos Prata dos Reis
Member of the Committee: Susana Margarida da Silva Vieira

November 2019

The noblest pleasure is the joy of understanding.

Leonardo da Vinci

Acknowledgments

To start off, I would like to thank Mr. Raposeiro for all the knowledge and help provided during the development stages of the prototype.

A special thank you to my supervisor Prof. João Reis as well, for believing in my ideas and for always being available to guide me.

To my co-workers at PSEM with whom I had the pleasure to work with and experience a side of engineering that can't be taught in classes.

To my colleagues for their companionship throughout these years at the IST, and for all the great memories that they I will keep from this journey.

Also, to my friends, for their many words of encouragement and unconditional support.

Finally, and most importantly, I would like to thank my family to whom I dedicate this work. For always believing in me and for letting me take my own path while providing me with everything that I could ever need.

Abstract

This thesis presents a study of force estimation on an inflatable robotic link using computational vision tools. The structure is intended to be used in medical applications to complement a robotic manipulator with passive safety features.

The inflatable structure has a truncated conical shape, and was designed to be easily deformed in order to enhance its sensibility to external forces. Its lateral surface was built using polyamide fabric coated with neoprene, a synthetic rubber, in order to make the structure airtight. Both the top and bottom surfaces were made of ethylene-vinyl acetate foam for soft contact surfaces.

An image processing algorithm was developed to infer the structure deformation from internal markings captured by a camera placed inside the prototype.

The results were then used as input in an artificial neural network that yielded the estimated force being exerted on the structure. The neural network was firstly trained by using data where both the input and output were known and then tested by comparing its performance against a exterior force sensor.

By analysing the performance of the prototype it was concluded that employing it under certain conditions displays satisfactory results, achieving a mean error of only 0.051 N, thus presenting a viable option for measuring forces.

Keywords

Soft Robotics, Tactile Sensor, Computational Vision, Artificial Neural Networks

Resumo

Esta dissertação de mestrado apresenta um estudo sobre estimação de forças exercidas numa estrutura insuflável. Para o efeito, foram aplicadas ferramentas de visão computacional em conjunto com a utilização de redes neuronais artificiais, uma técnica de *machine learning* em grande crescimento nos últimos anos.

O protótipo fabricado exibe uma forma cónica truncada, tendo sido projectado de modo a ser facilmente deformado, aumentando a sua sensibilidade a forças externas. A sua superfície lateral foi produzida utilizando tecido de poliamida revestido de neopreno, uma borracha sintética, de modo a ficar selado ao ar. As superfícies superior e inferior foram fabricadas através de espuma de etileno acetato de vinil de forma a constituírem superfícies de contacto com o exterior de consistência macia.

Em seguida foi desenvolvido um algoritmo de processamento de imagem que infere a deformação da estrutura através de marcações interiores capturadas por um câmara colocada dentro do protótipo.

Posteriormente utilizaram-se os resultados obtidos como *input* numa rede neuronal artificial que gera subsequentemente uma estimativa da força exercida na estrutura. A rede neuronal foi primeiramente treinada utilizando dados cujos resultados são conhecidos e em seguida devidamente testada.

Verificou-se, através da análise do desempenho do protótipo que a sua aplicação dentro de determinadas condições apresenta resultados satisfatórios, constituindo uma solução viável para a estimação de forças.

Palavras-Chave

Robótica Flexível, Sensor Táctil, Visão Computacional, Redes Neuronais Artificiais

Contents

List of Figures

List of Tables

Abbreviations

RGB - Red Green Blue

HSV - Hue Saturation Value

CMYK - Cyan Magenta Yellow Black

ANN - Artificial Neural Network

LMB - Levenberg-Marquardt Backpropagation

MSE - Mean Squared Error

LCD - Liquid Crystal Display

PLA - Polyactic Acid

FPS - Frames Per Second

NN - Neural Network

V - Volt

Pa - Pascal

List of Symbols

Greek Symbols

ω_i	Node input i weight
σ	Activation function
μ	Weight adjustment damping factor
α	Planified cone sector angle

Roman Symbols

I	Pixel intensity
R	Pixel red intensity
G	Pixel green intensity
B	Pixel blue intensity
Ib	Binary pixel intensity
T	Threshold
$\mathbf{C_x}$	X component of the centroid
$\mathbf{C_y}$	Y component of the centroid
$\mathbf{X_i}$	X component of the pixel i
$\mathbf{Y_i}$	Y component of the pixel i
e	Eccentricity
a	Semimajor axis length
b	Semiminor axis length
$\mathbf{x_i}$	Node input i
y	Node output
y	Node output
$\mathbf{e_i}$	Error generated from datapoint i
J	Jacobian matrix loss function
h	Flexible sleeve length
D1,**D2**	Truncated cone diameters
R1,**R2**	Truncated cone radii

Chapter 1

Introduction

1.1 Motivation

With the advances in robotics and the desire for an increasingly automated society there is also a rising interest in the development of soft robotics, in other words, the application of robotic concepts with compliant structures. This field studies the conception of robotics with highly flexible properties, constituting a bio-inspired solution to tasks with high dexterity requirements while being safe to interact with, an asset not frequently present in traditional rigid robots.

Regarding robotic manipulators, a sub genre of robotics that responds to the need of robots handling objects, it is frequently required to acquire tactile sensing information to successfully carry out tasks. For this requirement there is a extensive arrangement of solutions but only a few respond to this need using flexible structures.

With this necessity in mind comes the idea behind the conception of this dissertation, which is the design and manufacture of a flexible structure to be utilized in a robotics setting with the capability of estimating the force being exerted by it. It is intended for it to be employed as a robotic link placed before the robot's end effector to compose a flexible solution to the tactile sensing requirement. In order to achieve this objective computer vision techniques are integrated with a machine learning concept, namely, artificial neural networks to explore the reachable performances and robustness of the application of this solution.

1.2 State of The Art

1.2.1 Soft Robotics

Soft robotics is a field in robotics that focus on the design and production of robots made from flexible materials. It's mainly inspired by nature, from the behaviour of invertebrate animals such as worms and octopuses [1], to the motions of plants that utilize the manipulation of internal fluids [2]. Opposing to rigid robots, by having a flexible structure the motions of soft robots are difficult to model, composing a continuum mechanic problem [3]. Considering this, their behaviour is commonly modelled using finite element methods (FEM), also employing these approaches to validate experimental results [4].

In order to control the movement and generate the reaction forces of these soft robots it is necessary to have actuation systems adequate to their compliant nature, as per example:

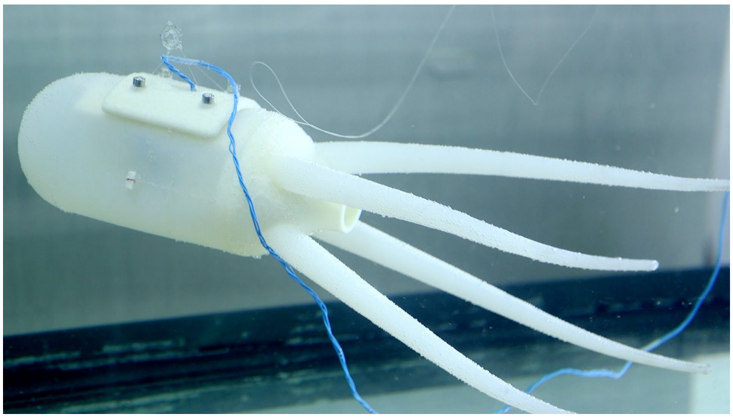
- **Electric Fields**, utilized by dielectric elastomer actuators (DEAs) to change their configuration. They are capable of high actuation rates and can produce high forces, yet their practical applications are limited by requiring high voltages [5];

- **Thermal Actuation**, employed in shape memory polymers (SMPs) or shape memory alloys, taking advantage of the thermal properties of these materials as a control system. These actuators are capable of strains of up to 1000% but are restricted to low response speeds and forces [6];

- **Pressure Variation**, by changing the internal pressure of flexible structures, commonly from the usage of compressed air, they apply forces to the environment and alter their shape in a similar manner as a muscle. Although they are highly practical to maintain desired configurations, they main constrain is the requirement of external sources of the utilized fluid [7].

Taking into account the compliant properties of these robots, they constitute versatile solutions to accomplish tasks and provide safety when working around humans. This last characteristic comes from their low risk of harmful interactions when operating near human personnel, and thus are often named intrinsically safe [8]. Considering this, soft robots are mainly employed in manufacturing scenarios and extend the implementation of robotics to medical fields. In the scope of manufacturing, the application of this technology promotes safety, given how the collision with their soft structure composes low risk of injury and therefore enabling their operation alongside human workers [9]. Regarding medical applications, soft robots can perform critical tasks in minimally invasive surgery (MIS) by adapting their shape to the human body structures and can also be employed in rehabilitation functions, aiding patients while allowing fluidity of motions due to their flexible nature [10].

In more recent years, soft robots have also been utilized in ocean exploration missions by having designs that mimic the morphology of aquatic animals, namely cuttlefishes (1.1) and lampreys [11].



Source: Youtube - Nature Video[1]

Figure 1.1: Soft Robot Inspired by an Octopus

Although the topic of soft robotics is relatively new, its firstly application dates back to 1963 with the development of pneumatic actuators to aid polio patients [12]. Despite the resulting prototypes were used as base for following robot designs, this subject only started to grow later on, being highly recognized nowadays by the robotic community. The research in this field is widely spread, from covering the application of this concept using a broad range of compliant materials to the implementation of different types of actuation systems, being pneumatic actuation the most popular one [13, 14, 15]. Different configurations of structures are also analysed, mainly from bio-mimicry, taking inspiration as per example the tentacles of octopuses [16], the trunks of elephants [17] and even caterpillars [18].

An highly important aspect regarding robotic manipulation is the measurement of the force being applied by the robot on the external surfaces and objects. This matter is usually solved by utilizing sensors placed on the robot, measurement of pressure or external vision systems [19]. Hence, this dissertation focuses on the development of a new approach to measure the force applied by a soft robot through integrated vision sensors that evaluate the deformation of its own structure.

[1] https://youtu.be/A7AFsk40NGE

1.2.2 Tactile Sensing

Tactile sensing technology relates to solutions and equipment capable of measuring information produced from physical interaction with the environment. As the name suggests it is equivalent of the human sense of touch, the only sensation present all across the human body, crucial for environment interaction and perception. Most designs are divided into sets of tactile elements, commonly named *tactels* or *taxels*, forming a grid to focus on spatial perception [20][21]. It is a broad concept covering the computation of force, torque, and surface shape, vital information for manipulation and grasping tasks. The process starts with the transduction of information using one of several sensing mechanisms, for instance: detection of the displacement of an elastic component, measurement of pneumatic pressure, piezoelectric effect, piezoresistive effect, capacitive-based sensing, etc [22].

Due to their high versatility tactile sensors are present in a wide range of applications, such as:

- Touchscreens, very commonly used in mobile phones and laptops;
- Performance testing in several automobile components like brakes, door seals, tyre footprints, among others;
- Tactile imaging, used in elastography by mimicking manual palpation through probes;
- Robotic manipulation of objects with unusual shapes, requiring higher precision, dexterity, etc.

Historical Evolution

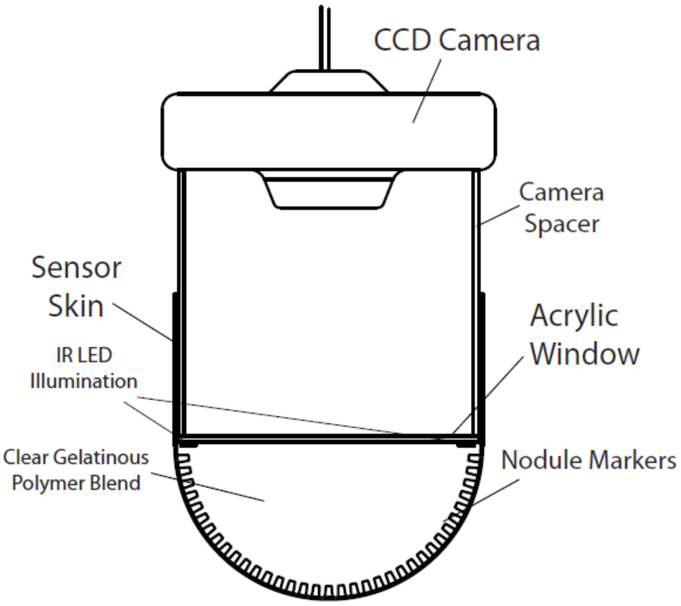
One of the earliest reported applications of tactile sensing technology dates back to 1973, at the Third International Joint Conference on Artificial Intelligence, where a tactile sensor was mounted on an anthropomorphic hand. Through the information obtained by this sensor a machine learning agent would learn how to recognize several shapes, namely a triangular prism, a square and a circle [23]. The chosen design had a total of twenty 3mm piezoelectric tactels scattered across five rows making the sensor small enough to fit in each of the artificial hand finger tips.

Following sensing devices showed great potential in industrial automation, the biggest sector in robotic object manipulation. Yet, applications specifically devised to this end went no further than experimental prototype stages [24] due to the wear that the sensors would be subjected to and the fact that computational vision tools constitute a much more simpler solution to perform the same tasks [25].

Subsequently it was shown that tactile sensing doesn't fit the industrial scenario very well, taking a turn from its initial intended purpose towards service robotics, food processing and medical applications with a strong emphasis on the latter [26].

As in many other fields, nature provides us with the best examples [27], and thus the advancements in the understanding of the human sense of touch lead to new solutions and improvements to tactile sensors.

Thus after publication of new theories about the nerve endings of fingertips and their structures in primates [28] new sensor designs inspired by these biological configurations were developed. The mechanism present in these sensors focus on the simulation of the skin mechanoreceptors that convert compressive strain into neural signals [29]. In 2009 this concept was utilized to build a tactile sensor that inferred force from the displacement of internal markers placed on a compliant contact surface [30]. Some years later prototype's were developed with similar morphology's in order to detect edges from sensor vision data [31][32](1.2).



Source: Article [31]

Figure 1.2: Vision Based Sensor

Lastly, a sensor was devised with a fingertip shaped soft structure, similar to the ones mentioned previously. It had two operation modes where one specialized in speed and the other in precision, in order to mimic the human reflexes [33].

1.3 Main Contributions

This dissertation has as main contribution the feasibility study of the estimation of force exerted by a robotic manipulator from the deformation of a soft link. In the scope of this work the following contributions are also achieved in regards to the application of computer vision and machine learning in the scope of soft robotics, namely:

- Testing the application of computer vision markers with blob analysis to infer the force being exerted in a soft structure;
- Integration of computer vision tools with the capabilities of neural networks in the field of force estimation;
- Analysis of the structure and parameters of neural networks in the scope of this work.

1.4 Thesis Outline

Chapter 1 starts with the description of the motivation, followed by an overview of the tactile sensing concept along with the historical context of its application using soft structures. Finally the main contributions and structure of the thesis is presented.

Chapter 2 will review all the theoretical concepts employed, defining the computer vision techniques and the artificial neural networks methodologies applied.

Next, in chapter 3 there will be a detailed description of the experimental setup, listing all its components and explaining the manufacturing process of the prototype.

Chapter 4 will present the obtained data from experimental trials up to the results achieved from the application of the concept explored in this work.

Lastly, chapter 5 conclusions are drawn over the accomplished objectives and recommendations over future work are made.

Chapter 2

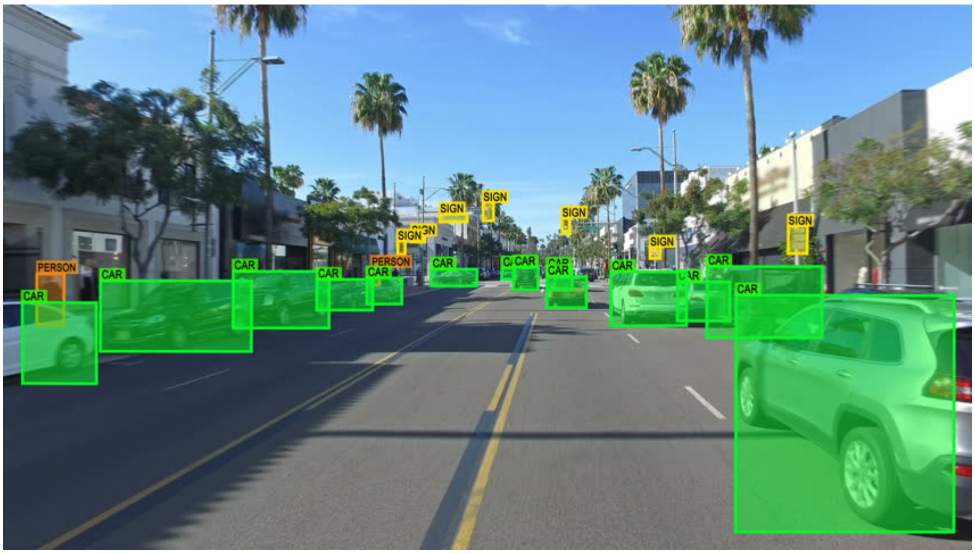
Theoretical Background

This section will cover the theoretical foundations applied across this work. First of all an overview on the concepts and methodologies used in the scope of computational vision, laying the theoretical knowledge necessary for the processing of images acquired from the camera device placed inside the prototype. Additionally and lastly, a synopsis over the field of neural networks, a key subject employed in the transformation of the data obtained by the tools of the previously referred domain into an estimation of the force being exerted on the prototype.

2.1 Computational Vision

Computational vision comprises in the study of how computers process and analyze the content of images in order to reach performances analogous to the human vision system. A trivial task such as the identification of a shape actually translates into complex procedures that are not that simple to simulate. Given this, the field of computer vision tries to emulate the complexity of vision perception through the deconstruction of imagery into symbolic information such as statistics instead of tackling the images as a whole.

It can be employed in a extensive range of applications, from relatively simple tasks such as enhancement of an photograph's quality to tougher implementations for instance video tracking. Another great example of its applications nowadays is the highly popular self driving cars. These are only possible thanks to high-end application of computer visions techniques such as image classification which focus on the identification of different objects present in an image.



Source: Shutterstock[2]

Figure 2.1: Example of Vision Recognition

Across this work several tools and methodologies from this field were employed with the goal of collecting information regarding the deformation of the prototype from images retrieved from a video camera device. These concepts will be reviewed by the order that they are applied.

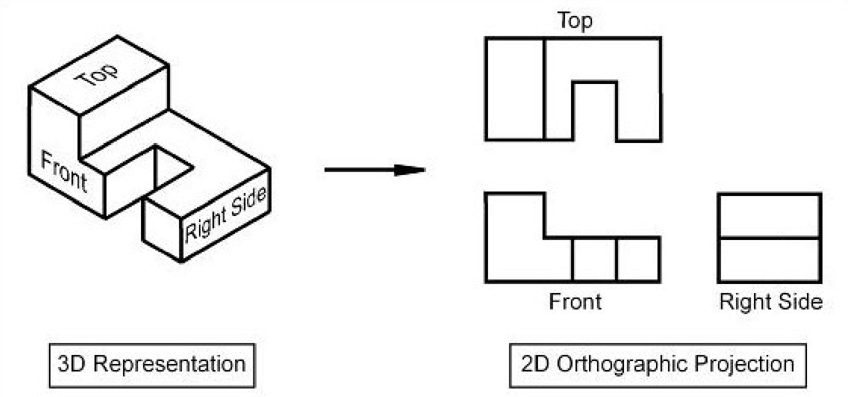
2.1.1 3D Projection

In computational vision 3D projection consists in the methods that transform a three-dimensional object into its two-dimensional representation. This must not be confused with perspective, which is a specific type of projection that aims to create a three-dimensional perception into the projected surface producing an illusion. It is a crucial step to obtaining vision data with a wide variety of approaches depending on the desired outcome, considering that they produce different representations of the same object. These procedures produce different outcomes by shifting the point of view from which the object is observed, that as result, generates different mappings from the three-dimensional space to the two-dimensional plane.

2.1.1.1 Orthographic Projection

When a camera's direction is parallel to one of the primary axis of the object, its captured representation is classified as an orthographic projection. In other words, is a type of projection absent of perspective distortion, as if there is no distance between the captured points and the camera's plane, resulting in having no perception of depth. By way of illustrating the process figure 2.2 presents the orthographic projection of an object from different points of view.

[2] https://www.shutterstock.com/pt/video/clip-33521563-autonomous-driver-less-car-computer-vision-object-detection



Source: CivilSeek[3]

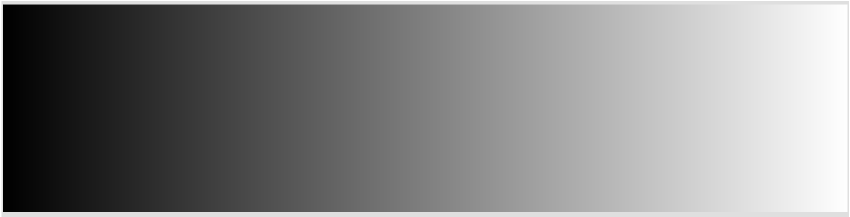
Figure 2.2: Object Orthographic Projection

Since in this form of projection the lengths between points are not foreshortened, in other words, their dimensions are not scaled according to depth, it provides accurate representations of the geometry of the perceived objects therefore it is commonly used in technical drawings. On the other hand considering this property, it is not possible to infer the actual distance between points if they are in different depths in the three-dimensional space.

In this work, by placing the camera in a central position relative to the axisymmetrical soft structure, the obtained images compose orthographic projections of the prototype's interior surface. This way, although it is not possible to directly measure the displacement, it allows the acquisition of relevant vision data about the prototype's deformation.

2.1.2 Greyscale Images

A greyscale image is the simplest form of representation of visual data taking into consideration that it only requires the intensity information for each pixel. The higher the value the brighter the pixel, so it covers every shade of grey, ranging from pure white to pure black colors (2.3).



Source: Pinterest[4]

Figure 2.3: Greyscale Color Gradient

[3] https://civilseek.com/orthographic-projection-drawing/
[4] https://www.pinterest.co.uk/pin/357473289158212718/?nic=1a

Its important to distinguish greyscale images, composed by shades of grey, from binary images which, as the name indicates, are only composed of two colors, black and white.

Taking into account how simple this imagery representation is, each pixel can be expressed by a single value that corresponds to its intensity. There isn't a global format for this numerical representation so the values range is arbitrarily chosen and therefore should be known before analysing a greyscale image.

2.1.3 Color Spaces

A color space is an abstract concept in which colors are reflected as tuples of numbers (typically triples or quadruples) to represent them mathematically. It can also be interpreted as an elaboration of a coordinate system where each point corresponds to a single color.

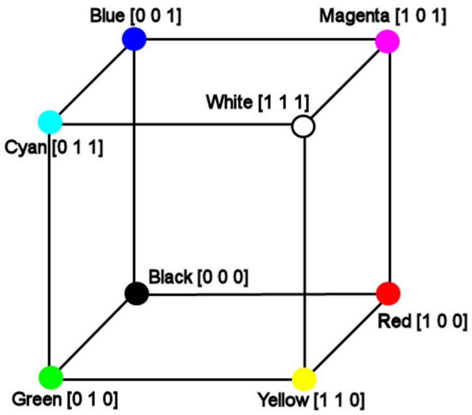
Since there isn't a globally defined color system and because a color space is roughly arbitrarily defined, many color spaces exist. In order to simplify their identification they are usually named after the names of the colors/components that compose the color model, as for example:

- **RGB** - Red, Green, Blue Color Model;
- **HSV** - Hue, Saturation, Value Color Model;
- **CMYK** - Cyan, Magenta, Yellow, Black Color Model.

2.1.3.1 RGB Color Space

RBG is a color model based on the addition of red, green and blue to create a wide range of colors. It is based in the human perception of colors given that the human eye has only sensitive receptors for red, green and blue.

It can be represented by a three-dimensional space where each axis corresponds to one of the three primary colors of this color model, therefore each RGB triplet constitutes an unique color. As can be noted from the image 2.4 at the origin vertex (0,0,0) is the color black resulting from the lack of any color and the opposite vertex (1,1,1) the color white derived in a similar manner.



Source: ResearchGate[5]

Figure 2.4: RGB Color Space Representation

RGB is one of the most popular color models and therefore on of the most used, being employed in digital camera, computer graphics, televisions, etc. One interesting application is an LCD display that can be thought as a grid with millions of small red, green and blue lamps spread across the screen, lighting up depending on the color of the pixel in that position.

⇒ RGB to Greyscale

In this work the method of processing the images begins with their acquisition through a video camera device that outputs them with a RGB color space. Images in RGB color mode can't be directly binarized, this is due to the fact that this procedure can only be applied to greyscale images. This way a conversion between RGB color model to greyscale is implemented, where the intensity of each pixel is calculated from a weighted sum specified in the equation 1.

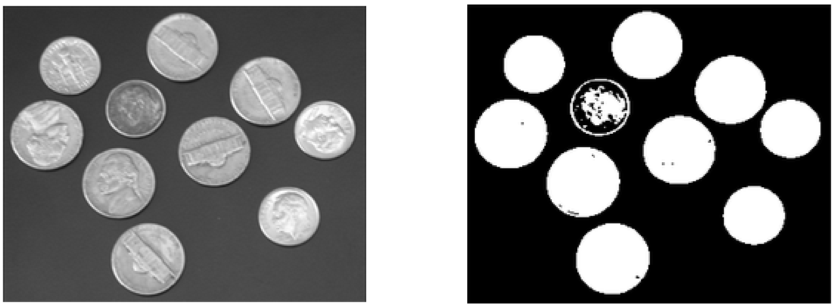
$$\mathbf{I} = 0.2989\,\mathbf{R} + 0.5870\,\mathbf{G} + 0.1140\,\mathbf{B} \tag{1}$$

The weights values come from the CCIR 601, a standard released in 1982 by the CCIR (International Telecommunication Union) [34] that takes into consideration the different human eye sensibility towards these three colors.

[5] https://www.researchgate.net/figure/The-RGB-colour-cube_fig1_304240592

2.1.4 Binarization

Binarization or tresholding is the transformation of images into their binary version, which as stated previously, is an image formed with only two colors, typically black and white. Taking this into account each pixel only has two possible values, 0 or 1, one associated with the foreground color and the other to the background (commonly white and black respectively). Given the output, tresholding its considered a method of image segmentation, in other words, a process that separates an image in multiple sections generally used to identify objects or their borders.



(a) Input

(b) Output

Source: MathWorks®[6]

Figure 2.5: Binarization

In terms of implementation it is executed by replacing each pixel by a white or black (**Ib**) one depending on how its intensity value (**I**) compares to the selected threshold (**T**). The next system of equations (2) translates this procedure.

$$Ib = \begin{cases} 1, \, if \, I > T \\ 0, \, if \, I \leq T \end{cases} \tag{2}$$

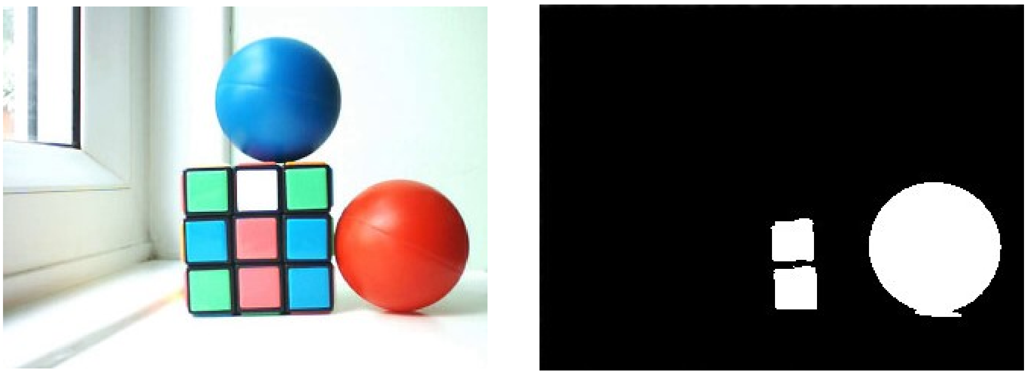
When the contents and conditions changes it is best to implement an automatic binarization algorithm that calculates the threshold for each image based on its distribution of intensity values. One of the most popular techniques is the Otsu method, this procedure computes an adaptive threshold that minimizes the weighted intra-class intensity variance, in other words, it estimates the threshold that reduces the variation of the intensity values as much as possible [35]. In the scope of this work a static threshold value can be used since that the conditions of the region captured by the camera will remain constant. For this purpose the threshold value can be assigned through a simple trial-and-error method, selecting the value that provides a robust distinction of the markers from the background.

[6] https://www.mathworks.com/help/images/ref/imbinarize.html

2.1.5 Blob Analysis

In the scope of computational vision, a blob is an area of an image in which some properties of the pixels are constant, which in the most simple case is regions where the pixels have the same color. Depending in the context in which it is applied, each cluster of pixels could represent objects present in images to sets of pixels that are distinct from their surrounding ones.

In this context blob detection consists, as the name indicates, in the identification of blobs in an image, dividing it into foreground (blobs) and background. By converting an image into this binary format, from a computational standpoint it is much simpler to retrieve the desired information given that it allows the processing of each picture element individually rather than the image as a whole. Blob detection methods can be employed as an initial step to complex algorithms, as for example, procedures that require the recognition of individual elements for supervision. These include, traffic monitoring, requiring the identification of each vehicle, or animal species control, from the detection and counting of animals from different species. To illustrate a simple example, image 2.6 presents the input (2.6a) and output (2.6b) of an algorithm that identifies red elements of an image. In accordance to the desired processing, the resulting blobs could then be counted or subjected for further classification, for instance, by their shape.



(a) Input

(b) Output

Source: CVSandbox[7]

Figure 2.6: Red Elements Detection

[7] https://www.http://www.cvsandbox.com/tutorials/blobs$_{processing/}$

Given these definitions, blob analysis is the extraction of informations of blobs following their detection. A wide variety of parameters can be chosen depending on the objective of the overall operation. In cases where many properties are available, a feature selection algorithm is usually implemented to select the ones that have a stronger correlation with the intended output. In the context of this thesis, considering that the target of this operation is to provide information related to the deformation of the prototype, the following parameters were measured:

- **Centroid ($\mathbf{C_{xy}}$)**, the most important factor since its strongly linked to the displacement;

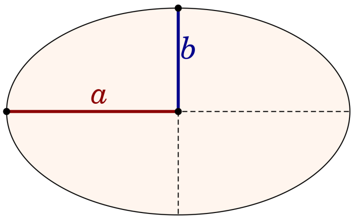
$$\boldsymbol{C_x} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{3}$$

$$\boldsymbol{C_y} = \frac{\sum_{i=1}^{n} Y_i}{n} \tag{4}$$

Where **n** is the number of pixels in the blob.

- **Eccentricity (e)**, by grading the shape in terms of how circular or elliptical it is, more data can be obtained relative to the warping of the structure. Its computed from the semimajor **a** and semiminor **b** axes of the blob considering it as an elipse (5).

$$\mathbf{e} = \sqrt{1 - \frac{b^2}{a^2}} \tag{5}$$



Source: Wikipedia[8]

Figure 2.7: Semimajor and Semiminor Axes

The output of this procedure is then used as input to the neural networks which will be covered in the subsequent section.

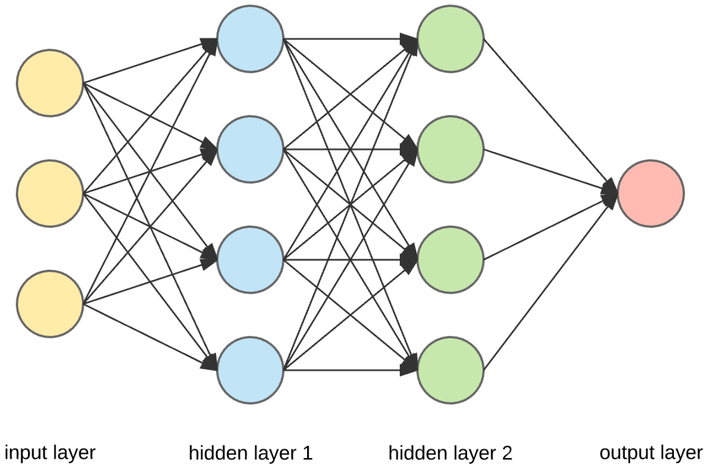
[8] https://en.wikipedia.org/wiki/Orbital$_{period}$

2.2 Artificial Neural Networks

Artificial neural networks consists is a methodology from machine learning inspired by biological neural networks through mimicking their structure and behaviour. It started in 1943 when Warren McCulloch and Walter Pitts devised a mathematical model of brain neural activity using logical calculus [36], from this concept developments over the years have deviated from biology but still maintaining the same structure as foundation. Nowadays ANNs reach performances in their target tasks that surpass even human abilities. One of the most popular examples is the AlphaGo, developed by DeepMinds, which in October 2015 won a match of the game Go against a professional player. Its latest successor, the AlphaZero, defeated other world-champion programs (which already ranked above the number 1 ranked human player) after only 24 hours of training [37].

2.2.1 Network Structure

Due to their high versatility they are used in a variety of tasks that can be summarized into pattern recognition, classification, modelling and prediction. Depending on the objective the NN configuration will vary yet the basic structure will be the same, which is a set of interconnected nodes organized in layers. These layers can be of three types: input, output and hidden. Every net has one of each of the first two mentioned ones, since they work as inlet and outlet respectively. The hidden layers quantity vary from none to several layers, depending on the complexity of the tasks, yet an higher number of layers does not mean better performance, in matter of fact it usually results in the opposite effect.



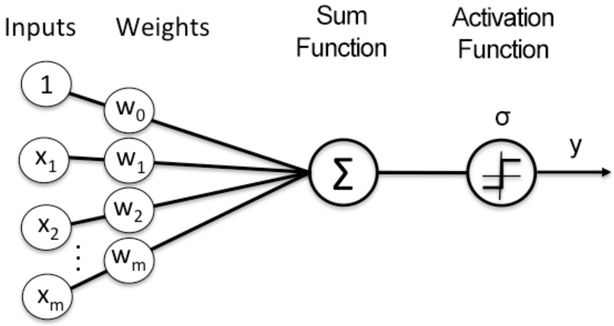
Source: Towards Data Science[9]

Figure 2.8: Example of an Artificial Neural Network

[9] https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

The nodes or artificial neurons, represented in the image 2.8 with circles, are the elementary units in an ANN. Just like a biological neuron it fires when it receives sufficient stimuli, which mathematically translates into producing a signal only if the sum of the weighted input values reaches a defined threshold. The following image and equation facilitate the comprehension of this process.

$$y = \sigma \sum \omega_i \, x_i \tag{6}$$



Source: Towards Data Science[10]

Figure 2.9: Artificial Node Example

The activation function σ defines in what condition does the node fire and the type of output that it produces given that it can be binary or continuous. Considering this several types of activation functions are available depending on the desired outcome, for example step functions (for binary outputs), linear functions (continuous), sigmoid functions (binary-continuous), etc.

Regarding the net architecture, in other words, the way that the nodes are connected between themselves, there are two possible types:

- **Feedforward Networks**, networks in which the neurons are only connected to neurons of the next layer, just like the net depicted in image 2.8. This structure is more appropriate for classification and regression;

- **Recurring Networks**, in contrast with the previous type this architecture allows connections between neurons of the same or previous layer. It is generally used in pattern recognition tasks.

[10] https://towardsdatascience.com/neural-representation-of-logic-gates-df044ec922bc

2.2.2 Learning

One of the most interesting aspects about artificial neural networks is their ability to learn, which is this scenario is defined as the improvement of the network's performance in a certain task by using sample data. Learning is an iterative process that adjusts the weights of the nodes connections in order the improve the overall result. This critical procedure requires some attention and sensibility towards the scope of the problem, because in similarity to other tuneable aspects of this methodology there aren't fixed optimal parameters for each problem, so solutions are obtained mostly through heuristics mixed with trial-and-error. Another important aspect is that when creating a new neural network the weights are generated randomly, so even if the same training data is used the net's performance will be different, reinforcing this tasks repeatability.

2.2.2.1 Learning Paradigms

Learning paradigms correspond to how the data is used to learn and the error that it yields, not to be confused with training algorithms which are responsible for the modification of the net's weights after the error is calculated. There are three major learning paradigms:

- **Supervised Learning**, this approach consists in using a set of inputs with the corresponding desired output. This way the error is quickly calculated and used as feedback on the quality of obtained solutions. On the other hand the overuse of training data can lead to overfitting which means that the net fits the training data too well, performing poorly when other data is used. This methodology is best suited for pattern recognition and regression types;

- **Unsupervised Learning**, this approach uses input data along with a cost function which helps with the estimation of the net's performance. Its main applied to clustering and filtering problems;

- **Reinforcement Learning**, is implemented when the problem has a inherent stochastic nature, making it impossible to know for sure what the outcome will be. In this scenario an agent has a set of possible actions with unpredictable results and the goal is to generate the best score, which is generated from the sum of the scores that resulted from each choice made. This is commonly applied to vehicle routing, management and video games.

2.2.2.2 Training Algorithms

As mentioned in the previous section, training algorithms take the calculated error and compute the adjustments necessary to perform to the nodes connection weights. These methodologies can be seen as applications of optimization theory and statistical estimation since they aim to obtain the optimal values from their estimation after using empirical data.

There is a vast list of training algorithms available to select, each with their own specific algorithm and suitability for particular tasks. Considering the nature of this work (regression - estimation) the Levenberg-Marquardt Backpropagation (LMB) algorithm was chosen given that it is suited for the training of theses types of neural networks and is one of the fastest methods, an important factor considering that this process is very time consuming.

The first step in training the network is to compute the performance of the current net, that is an estimation of how well the neural network performs the given task. In case of the LMB algorithm it is equal to the mean squared error **mse** (7).

$$mse = \frac{1}{n} \sum {e_i}^2 \tag{7}$$

Where **n** is the number of datapoints of the training set.

To start the weights adjustment its necessary to calculate the jacobian matrix of the loss function (8). This function expresses the difference between the net's output and the desired output from the data set so consequently the jacobian matrix indicates the variation of the error from changing the net's weights.

$$J_{i,j} = \frac{\partial e_i}{\partial \omega_j} \tag{8}$$

For i = 1,...,**m** and j = 1,...,**n**. Where **m** is the number of data set points, and **n** the number of tuneable weights in the net.

From this it is then possible to calculate the weights for the next iteration (9):

$$\omega^{k+1} = \omega^k - [J^T J + \mu J]^{-1} J^T e \tag{9}$$

Where $\boldsymbol{\mu}$ is a damping factor that is decreased after each successful iteration to increase the convergence speed and accuracy.

Chapter 3

Experimental Setup

An overview of the entire experimental setup used to collect data is made. Firstly a description of the manufactured prototype and its components combined with the reasoning behind the choice of their characteristics. Lastly a review over the force sensor used with its operation requirements.The following image shows the setup with the proper identification of each component.

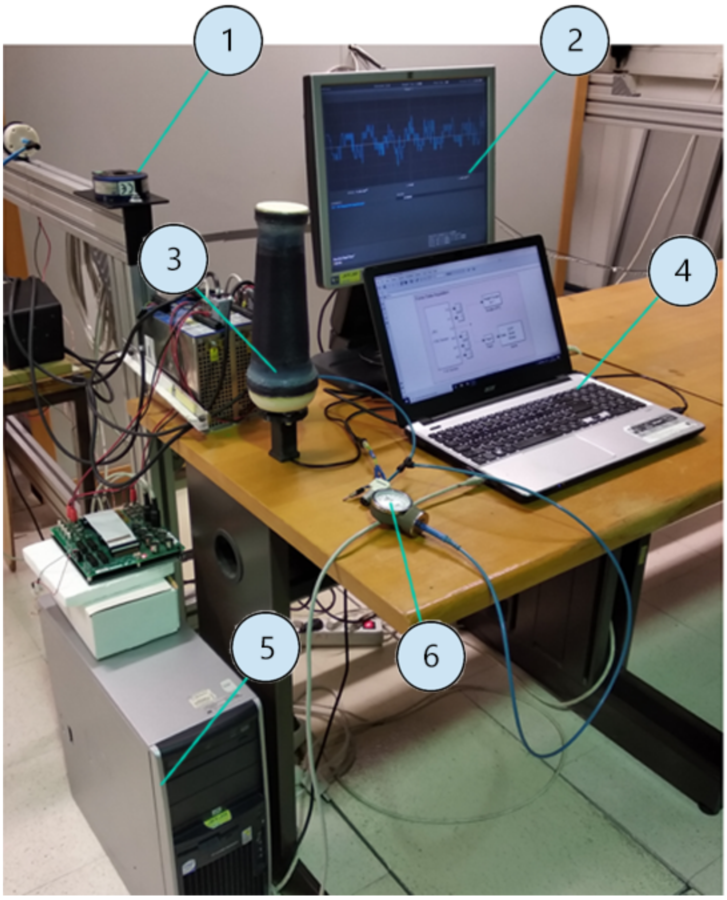


Figure 3.1: Experimental Setup

1 - JR3 Force Sensor; 2 - Target Computer Display; 3 - Prototype
4 - Host Computer; 5 - Target Computer 6 - Pneumatic Circuit

3.1 Prototype

In this section every integrant part of the prototype will be described. Several components were manufactured, so a detailed explanation of the construction process behind them will be made, justifying their main attributes and their fabrication process. The remaining components were bought henceforth their review will consist on an overview of their features and their respective function in the setup.

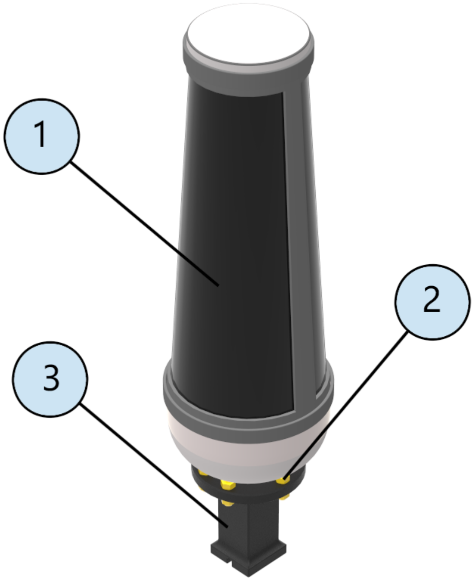


Figure 3.2: Prototype
1 - Soft Link; 2 - Fastening System; 3 - Prototype Handle

3.1.1 Image Acquisition

To capture images from inside the soft link an image acquisition device was required. No major sophistication is needed for this component as long as it the meet the following requirements:

- **Low resolution - 0.2 MP to 1.3 MP**, the bigger the resolution the more pixels the resulting image will have, meaning more data to process which would require a bigger processing power to calculate the force being exerted instantly. On the other hand, better resolution will result in an increase of the estimator sensitivity, this is due to the fact that smaller changes in the markers will be captured. The lower limit of this requirement is a resolution that can provide an image with enough quality so that all markers are well captured along with their respective boundaries. This sets how well the markers will be seen;

- **Wide Field of View**, this enables the camera to fully capture the markers when they are in a central position (null force being exerted on the estimator) and when they are diverted from this position (non null force being exerted on the estimator). It is very important to always capture the entire marker, otherwise it will result in lower of accuracy when calculating its correct statistics (centroid, eccentricity);

- **Incorporated Light Source**, since the soft link will be closed its inside will be completely dark, disabling the camera's image capturing capability. This way, to avoid having a separate light source that would only add unnecessary complexity, it is best to have the light source as a camera's feature;

- **Small Dimensions** $\varnothing \leq$ **90 [mm]** , considering that the device will be inside the soft link, its dimensions must be small enough to fit inside the foam proximal extremity while leaving some space to accommodate the camera's housing.

After considering the previously stated requirements with available equipment, a **Trust Spotlight Webcam** was chosen. It meets all specifications and is very straightforward to operate. It only needs to be connected to a computer USB port and it is ready to use, with no additional software required.

Source: Trust[11]

Figure 3.3: Trust Spotlight Webcam

Maximum Resolution	640 x 480
Maximum Framerate	25 FPS
Lighting	Yes
Height	53 mm
Width	53 mm
Depth	75 mm

Table 3.1: Trust Spotlight Webcam Features

[11] https://www.trust.com/en/product/16428-spotlight-pro-webcam-with-led-lights

3.1.2 Prototype Handle

This component provides a solid surface to help on the manipulation of the prototype, by having a small squared cross section that can be easily grabbed by a robotic arm gripper or by a human hand. It also acts as foundation to the soft link, proving a rigid base to where it will be fastened using bolts. Additionally a gap is needed throughout this element to serve as a pathway to the camera's electrical wires.

Due to its complex geometry the best option is to manufacture this component recurring to 3D printing technology. This process starts by designing the desired shape using a CAD software, which in this case was Solid Edge ST10® and before proceeding to print the dimensions must be scaled to account the thermal expansion of the printing material, in this instance PLA.

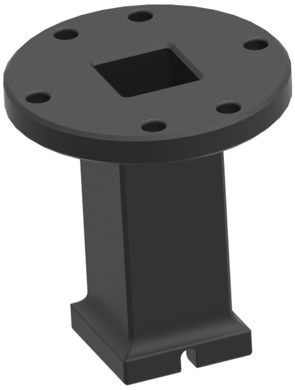


Figure 3.4: Prototype Handle

The component's material and rough finish provides a good surface to grip with low slippage, improving the prototype's handling.

3.1.3 Pneumatic Circuit

In order to pressurize the soft link an elementary pneumatic circuit was devised. This assemblage was not only to inflate but also to compensate for leaks by continuously pumping air into the designed soft structure.

This circuit is composed of a gauge pressure sensor (a), 3V DC air compressor (b) and the soft link (c - represented by an air reservoir), all connected with 4mm tubing. Figure 3.5 illustrates the described circuit.

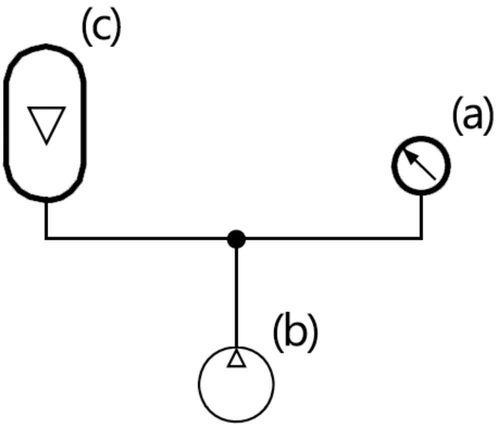


Figure 3.5: Pneumatic Circuit Representation

Due to the compressor dynamic this introduces noise into the measurement of pressure, this way, the tube that connects the pressure sensor to the tee is longer than the one connecting the soft link making the assembly act as kind of pneumatic low pass filter.

3.1.4 Soft Link

The soft link consists in the central part of the prototype. It includes all flexible components and its responsible for the harboring of the image acquisition device and the foam peripherals that will be in contact with the environment. Image 3.6 shows the entire set assembled with the identification of each component.

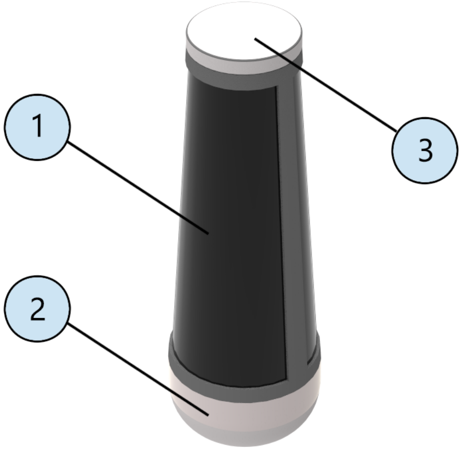


Figure 3.6: Soft Link
1 - Flexible Sleeve; 2 - Foam Proximal Extremity; 3 - Foam Distal Extremity

Due to the importance of the assembly of all constituents, the last subsection will be dedicated to how the entire set was mounted.

3.1.4.1 Foam Extremities

The extremities of the soft link needed to be made from a flexible and smooth material. They can't provide stiffness to the structure, since the softer it is, the higher the sensitivity will be, allowing the measurement of smaller forces. The top edge will also be in contact with the environment, so it must be made from a material that won't damage either equipment or people. The material must also be airtight to keep the interior of the soft link pressurized. Finally a light material is preferred, to make the entire prototype manipulation easier and preventing the structure from collapsing since the distal extremity is supported almost entirely by air pressure, so the heavier the material, the more pressure is needed.

Considering these conditions the best pick is a foam, so EVA foam, also know as Ethylene-vinyl acetate, was selected. There are three different types of this material, depending on the proportion of VA (Vinyl Acetate), where the higher the percentage the more dense it is. To make sure the air tightness condition is meet the EVA copolymer with an high proportion of VA was chosen ($>$ 60%).

⇒ **Distal Extremity**

The top extremity has a simple design where the main purpose is to have a shape that can fit in the top of the flexible sleeve and provide a good lateral surface that will be in contact with the surroundings. The whole extremity is made from two concentric discs joined together with appropriate glue. The smaller disc has ⌀ 60 [mm] and the other has ⌀ 70 [mm], both cut to shape from a plate of foam with 10 [mm] thickness.

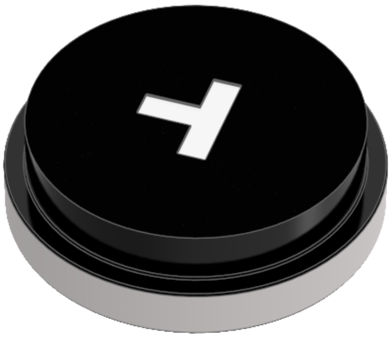


Figure 3.7: Foam Distal Extremity

Since the entire interior of the soft link that surrounds the markers has a black color, the disc portion facing inward must also have a dark coloration. Given that dark foam was not available this is achieved by painting the disc side facing inside with a non reflective black paint.

Lastly a T-shaped marker was painted in the middle of the component with the intention to be used in future work, namely for a study focused on the estimation of torque. The shape was arbitrary as long as it is nonaxisymmetric to provide a good reference to compute the rotation of the distal extremity against the orientation of the camera.

⇒ **Proximal Extremity**

The bottom extremity has the crucial function of serving as a central link, to assemble all different sections of the prototype. It receives the air intake duct, holds the image acquisition device (and its housing) and fits into the flexible sleeve, closing out the flexible link.
In similarity to the distal extremity discs, it is also made from a foam plate with 10[mm] thickness but is reshaped by heating and then pressing it against a flattened semi spherical mold. This is only possible because of the thermoplastic properties of the EVA foam, that becomes moldable at certain elevated temperatures and solidifies after cooling down. After it has the desired shape some of the edge is carved out to form slot where the flexible sleeve will be fitted. Following this, six ⌀ 3 [mm] holes are made in an hexagonal pattern with help of a punch, to provide gaps where the bolts that assemble the soft link with the prototype handle go through. The bolts have an M5 size, the difference in diameter when comparing to the holes is to keep a tight fit, blocking the air from leaving the inside. Finally the side is punctured to make an opening for the air intake.

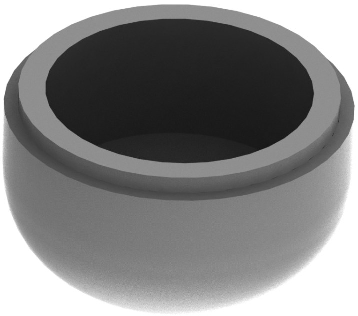


Figure 3.8: Foam Proximal Extremity

3.1.4.2 Flexible Sleeve

The flexible sleeve, placed in the middle of the soft link is the biggest and most complex component to manufacture of the entire prototype. Not only does it define nearly the entire structure stiffness but it also contains the computer vision markers, key elements to the acquisition of data.

In terms of shape it had to be able to allow the camera to capture as much of the markers as possible, since the bigger the surface it captures, the better the quality the data will have. In respect to the size, by increasing the length the more compliant it will be (increasing the sensitivity of the estimator) yet it also hinders the prototype manipulation. Considering the previously stated factors an hollow truncated cone with 230 [mm] of length was designed. The top and bottom orifices have $\varnothing$ 60 [mm] and $\varnothing$ 90 [mm] respectively.

The material needed to be a fabric capable of:

- Withstanding low pressure, 1 kPa - 7 kPa;

- Air impermeable;

- Available in a non reflective black color;

- Paintable surface.

Taking into account the stated constraints the flexible sleeve was produced using neoprene rubber padded with polyester, a fabric commonly used for aquatic sports suits. Its rubber side compromises as a good painting surface and can be extended without losing its air impermeability.
The entire truncated cone is made from a simple fabric sheet, but first it is necessary to cut the fabric from a template of the desired shape planified. The image 3.9 and equations 10 illustrate the planification process.

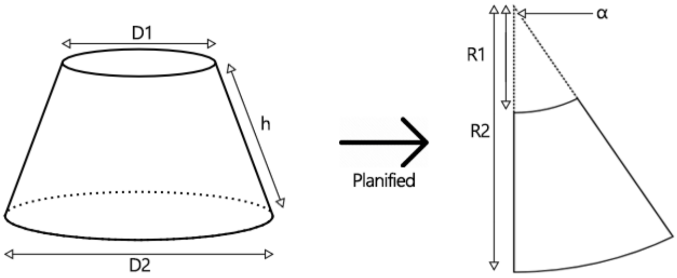


Figure 3.9: Truncated Cone Planification

$$\begin{cases} R1\,\alpha = \pi\,D1 \\ R2\,\alpha = \pi\,D2 \\ R2 - R1 = h \end{cases} \tag{10}$$

Where h = 230 mm, D1 = 60 mm and D2 = 90 mm, resulting in R1 = 460 mm, R2 = 690 mm and α = 0.41 rads

⇒ **Markers**

Before proceeding to close the shape it is necessary to paint the computer vision markers on to the fabric. Since it is intended to resort to image binarization to distinguish the markers from the background the select color was white, considering that is has the biggest contrast when comparing to the color of the flexible sleeve. The shape of these markers will depend on the type of image processing that will be applied, from this perspective the following options were considered:

- Randomly scattered dots, requiring the design of an interest point detector;
- Annuli across the cone's cross section, recurring to blob analysis to process the images.

Given the fact that an interest points detector uses an algorithm more computationally demanding, the second option was chosen.

The markers were placed as close to the top of the cone as possible, in the premises that the top section of the sleeve will be subjected to higher deformation therefore easier to detect the application of force. Three annuli were painted, equally separated from each other by 15[mm], but with different widths since the farther away from the camera, the smaller the area of the marker it will capture (3.10).

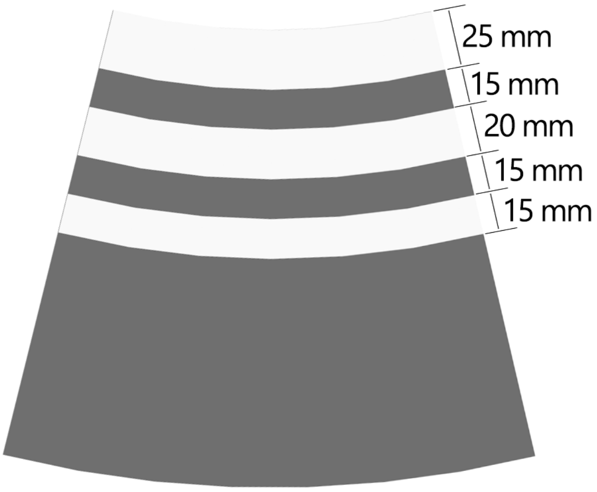


Figure 3.10: Painted Cone Planification

Acrylic paint was used on the account that when it dries it has a dull finish and is flexible, preventing the reflection of most of the camera's light and avoiding any cracks when the sleeve expands due to the air pressure.

Finally the lateral edges are joined together using contact glue and reinforced with a 10 [mm] strip of the same fabric, closing out any gaps between the two surfaces.

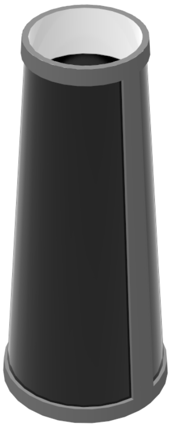
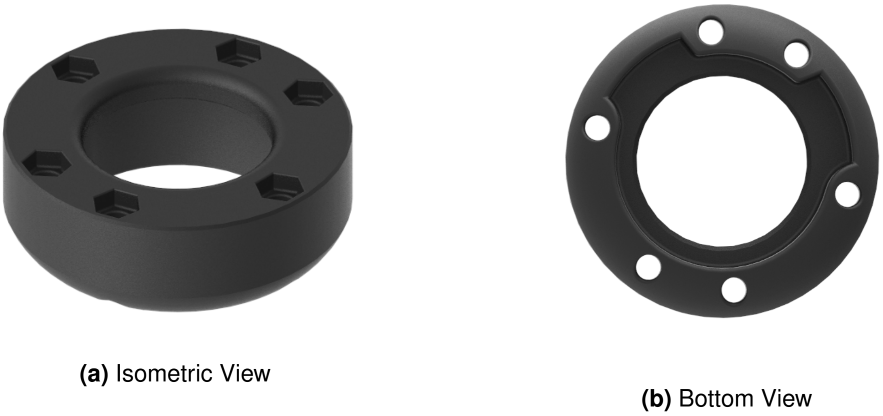


Figure 3.11: Flexible Sleeve

3.1.4.3 Webcam Cover

This component is utilized to secure the webcam to the bottom of the base foam extremity while also preventing it from rotating, ensuring its fixed position relatively to the soft link. In order to achieve this it is necessary that the cover accommodates to the camera's profile while being fastened to the prototype's handle.

In similarity to the prototype's handle, given its format this constituent was also 3D printed, using the same technique and material.



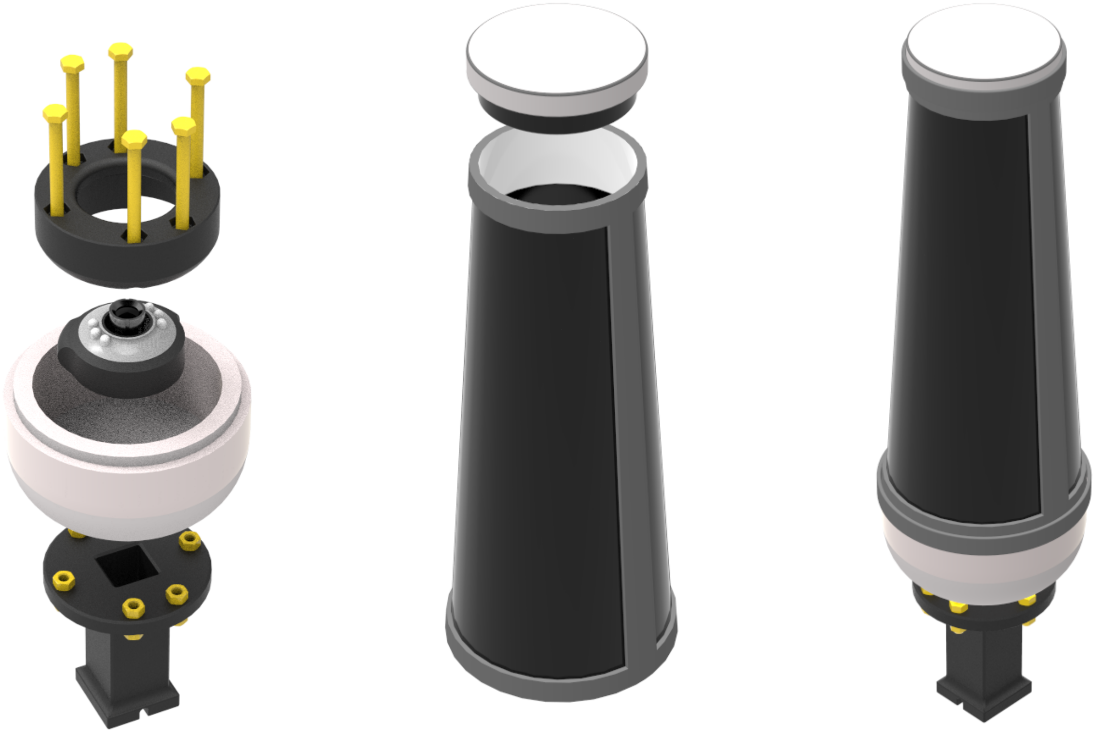
(a) Isometric View

(b) Bottom View

Figure 3.12: Webcamera Cover

3.1.4.4 Assembly

The final assembly of the soft link is a delicate process where the main focus is to ensure as much air tightness as possible. It started by fitting the webcam into its housing and then fastening this set to the foam proximal extremity using M5x50 bolts (3.13a). Following this, the foam distal extremity was mounted into the flexible sleeve, using contact glue to bind the two components (3.13b). Subsequently this previously assembled set was then fitted into the foam proximal extremity, closing out the soft link. Before the final step, 10 [mm] strips of fabric were glued to the contact surface between the foam elements and the flexible sleeve (3.13c). To finish the pneumatic circuit was connected and all edges between the different parts were covered with silicone to serve as a sealant against air leaks.

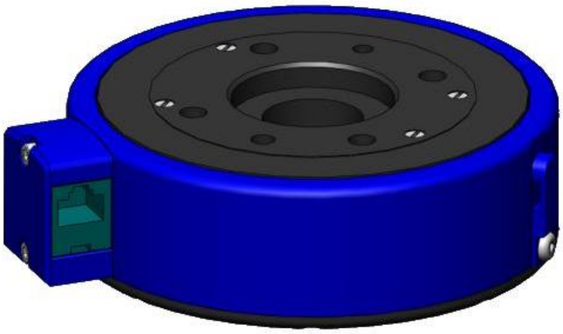


(a) Step 1 **(b)** Step 2 **(c)** Step 3

Figure 3.13: Prototype Assembly

3.2 Force Sensor

In order to obtain force data to subsequently train the neural network it was necessary to set a force sensor. The employed sensor was a JR3 multi-axis load cell (3.14), capable of measuring both forces and torques along three orthogonal axes (x, y and z) through strain gauge signals. The origin point used to calculate the torques is located at the center of the sensor and the z axis is aligned with its central axis.



Source: JR3[12]

Figure 3.14: JR3 Force Sensor CAD

Taking into account that the equipment requires specific hardware and software to read its output, a computer was already prepared, fully dedicated to this purpose. It was possible to interact with this computer through a LAN connection and by using MATLAB®[13] Simulink®[14], thereby permitting to send/receive data to/from any connected device. Considering that it exclusively ran the uploaded Simulink® model it was possible to acquire data at a frequency of 1000 Hz, a adequate rate for its intended application. This type of setting is commonly referred as **Target Computer**, on the other hand, the computer that interacts with it is referred as **Host Computer**.

3.3 Simulink® Models

This sections consists in the synopsis of the several Simulink® models designed and used throughout this work. Simulink® is a MATLAB® tool for simulation and analysis of systems with a wide range of applications.

In this work it was used to acquire the data from the force sensor as well as the camera. By employing its simulation capabilities it was also possible to calculate the neural network output in a testing scenario.

[12] https://www.jr3.com/products/force-torque-sensors

[13] MATLAB Version 1.7.0 (R2015b). Natick, Massachusetts: The MathWorks Inc., 2015

[14] Simulink Version 8.6. Natick, Massachusetts: The MathWorks Inc., 2015

3.3.1 Force Data Model

This model was utilized in order to receive the force measurements from the sensor and to send this data to the main computer. The image 3.15 shows the model and table 3.2 briefly describes each component.

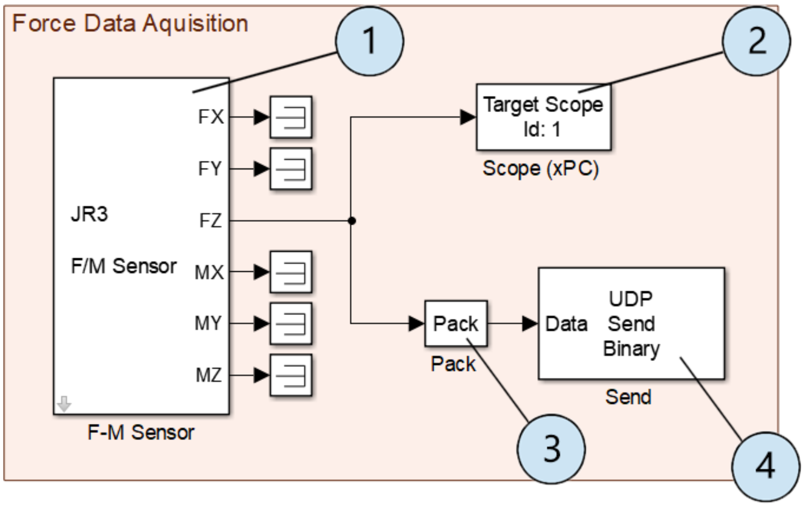


Figure 3.15: Force Data Acquisition Model

1	**JR3 Block**, allows the communication between the sensor and the model
2	**Data Scope Block**, plots the data into the computer's display
3	**Pack Block**, packaging of data into a binary output vector
4	**UDP Send Block**, transmission of data to a remote destination

Table 3.2: Force Data Model Blocks

3.3.2 Calibration Data Model

Variation of the vision data statistics will be utilized as input data for the neural network, thereby reference values from a state where null force is being applied were required.

While running this model collects the centroids and eccentricity values of each marker. Finally, after finishing it calculates the average between the obtained values and saves the result in a variable, enabling its usage in other Simulink® models.

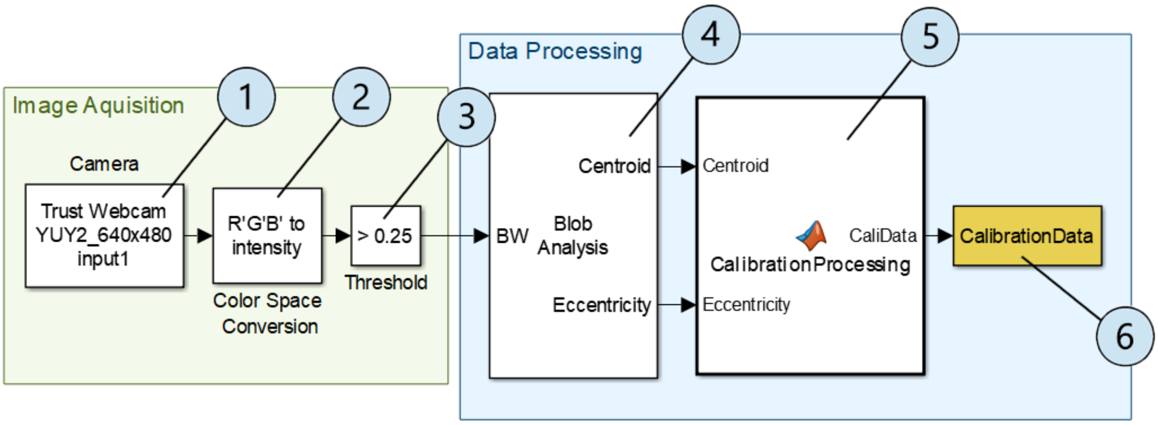


Figure 3.16: Calibration Data Model

1	**Camera Block**, outputs the video captured by the camera
2	**Color Space Conversion Block**, converts the image from RGB to greyscale
3	**Threshold Block**, binarization of the input image
4	**Blob Analysis Block**, outputs the statistics of connected regions (blobs)
5	**Calibration-Processing Block**, organizes the input into a structure
6	**To Workspace Block**, saves the input into the MATLAB® workspace

Table 3.3: Calibration Model Blocks

3.3.3 Vision Data Model

This model receives the signals from the **Force Data Model** and calculates the variation between the reference data (obtained from the **Calibration Data Model**) and the vision data statistics values. Both of these results are then outputed to the MATLAB® workspace to process posteriorly.

Its important that the force data and vision data are processed in the Simulink® at the same time to make sure that they are synchronized. This is crucial to make sure that the input and output data for the neural network match correctly otherwise it will be trained with erroneous values and hinder its accuracy.

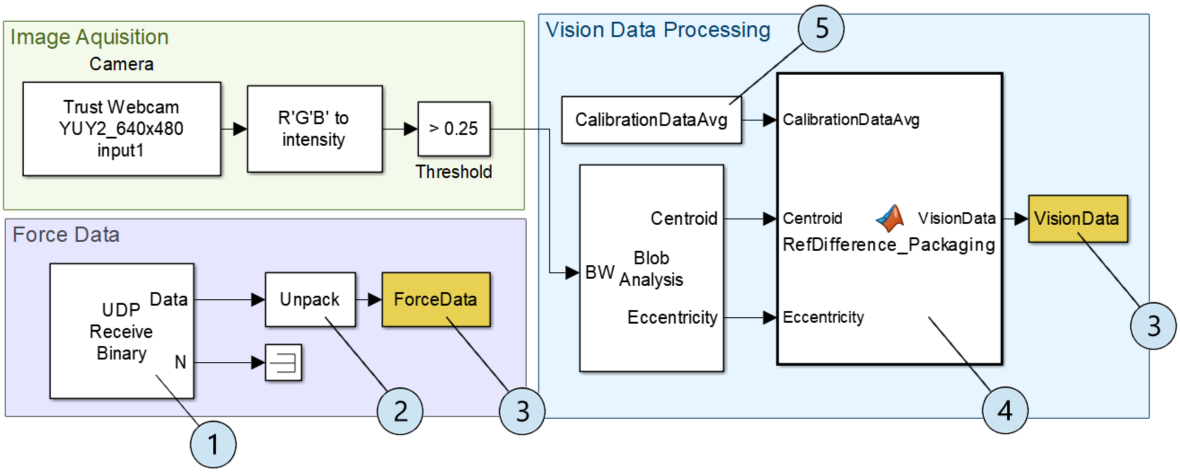


Figure 3.17: Vision Data Model

1	**UDP Receive Block**, reception of data from a remote destination
2	**Unpack Block**, unpackaging of data from a binary output vector
3	**To Workspace Block**, same as nº6 from table 3.3
4	**RefDifference-Packaging Block**, calculates the difference between the blob analysis output and the reference values
5	**Constant Block**, outputs constants values loaded from the MATLAB® workspace

Table 3.4: Vision Data Blocks

After the model finishes running it is necessary to process the data with the intention of preparing it for the training of the neural networks. Firstly, given the noise present in the signals produced by the force sensor it is necessary to filter them. This filtering is not directly implemented in the Simulink® because it would introduce a delay in the signal. Taking this into account, the MATLAB® function *filtfilt* was used since it filters the noise from a signal using a designed filter and then removes the time delay introduced by it. Image 3.18 presents the achieved results from employing a low pass FIR filter used throughout this work.

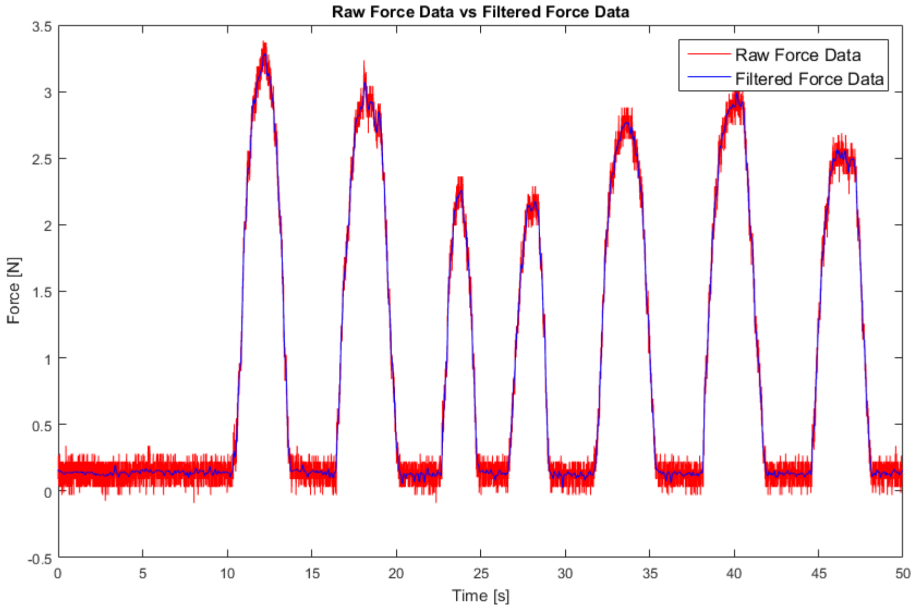


Figure 3.18: Filtered Force Data

As can be noted from the previous figure, the force sensor has a bias that is crucial to remove in order to improve the estimator performance. Considering that, a custom function was devised to estimate the present offset of the force values and then subtract it to the filtered data. The following image (3.19) shows the results of the implementation of this procedure.

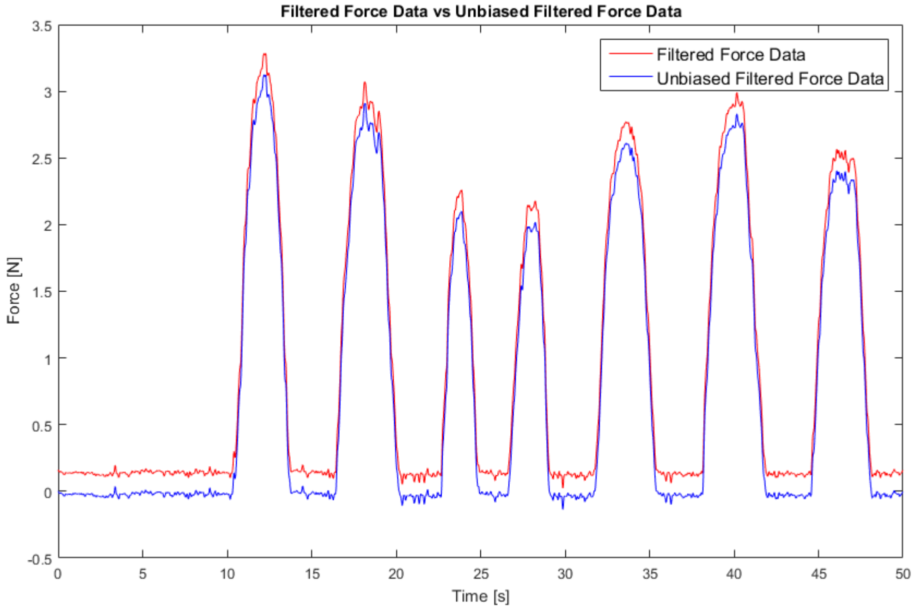


Figure 3.19: Unbiased Filtered Force Data

Finally both sets of data (force and vision) are inserted in a matrix with the appropriate format to use posteriorly. This processing methodology allows a faster collection of data and its uniformization from trial to trial.

3.3.4 Neural Network Testing Model

For the purpose of simulating the output of the developed neural networks using previously collected real data, this model was designed. Its main advantage is that it allows the study of the neural networks while skipping the data collection step, decreasing greatly the time consumed in the evaluation of their performance.

As stated previously it feeds the designed NN with formerly gathered data and afterwards saves its output into the MATLAB® workspace.

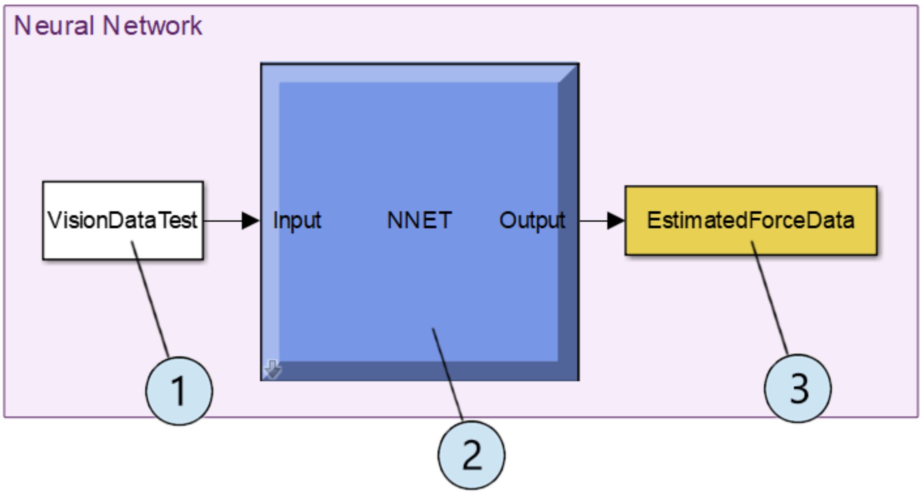


Figure 3.20: Neural Network Testing Model

1	**From Workspace Block**, similar to the constant block but changes the values in each instance of time
2	**Neural Network Block**, simulates the behaviour of the loaded neural network
3	**To Workspace Block**, same as nº6 from table

Table 3.5: Neural Network Testing Blocks

Chapter 4

Results and Discussion

In this chapter the obtained results are presented. Firstly the data acquired from the force sensor followed by the processing of the vision data from the images received from the camera to the corresponding values that the blob analysis outputs. Subsequently a research on the neural networks is made in order to design a net with the best performances. Given the prototype's structure, its behaviour will not be axisymmetrical so its necessary to explore two scenarios: with and without the rotation of the prototype when collecting the data.

4.1 Experimental Data

In this section the experimental data gathered from the JR3 sensor and the prototype is presented and interpreted. The procedures to obtain the data are described along with the cautions necessary to have to ensure its quality. As stated previously, two sets of data are obtained, one with rotation of the prototype as shown in the image bellow (4.1), and one without.

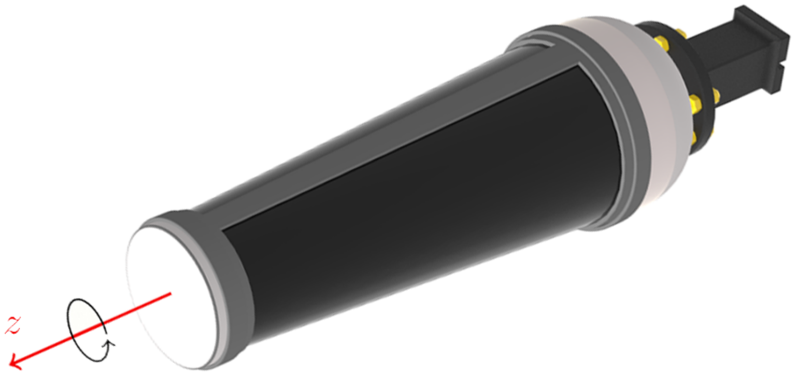


Figure 4.1: Prototype Rotation

4.1.1 Force Data

After having the target computer completely configured and connected to the host computer, the experimental setup was then prepared to acquire the force data collected from the JR3 sensor.

Following the establishment of the connection it was then possible to run the **Force Data Model** which ran without a stopping time in order to be always operational during the experiments. Due to the limitations of the sensor some pressure was applied to it before actually collecting data, this procedure helped stabilize its noise and bias which otherwise could change mid experiment and thus require additional processing before it was utilized.

As stated in chapter 3 the processing of the force values is only executed after the **Vision Data Model** finishes running, this is because filtering in real-time introduces a delay in the signal, adding a time shift to it and consequently desynchronizing the force data in comparison with the vision data. To avoid this, the signal was initially processed using the MATLAB® function **filtfilt** that filters the data and subsequently removes the introduced time delay. Given the high frequency noise present in the force values, as seen from the image 4.2, a lowpass FIR filter was applied. To finalize the processing of the force data, the present bias was removed by firstly estimating its value and then subtracting it to the filtered data.

The following image (4.2) presents a comparison between the raw data and the processed data.

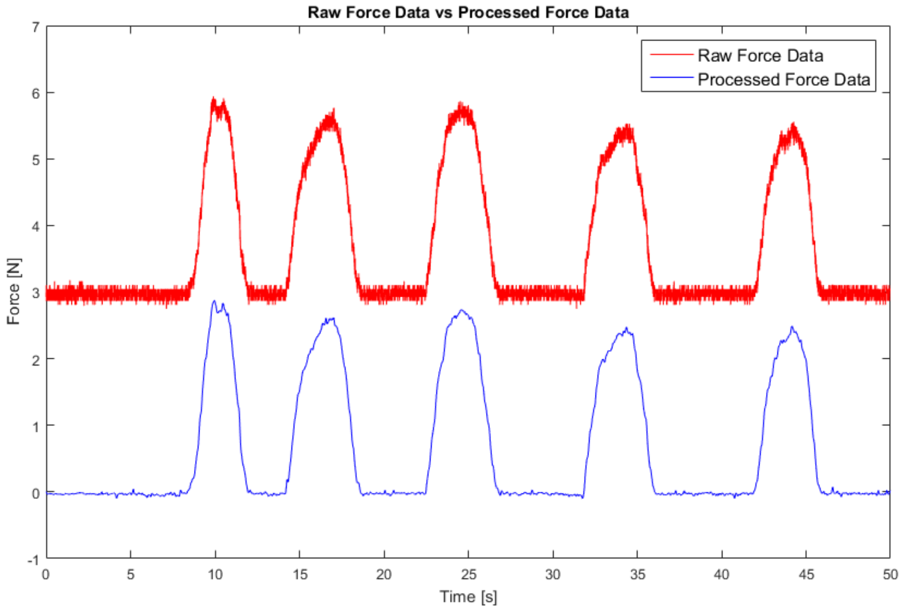


Figure 4.2: Raw vs Processed Force Data

4.1.2 Vision Processing

Having the force data being properly acquired it was then necessary to gather the information regarding the deformation of the prototype from images taken by the webcam.

It was crucial that the prototype was appropriately prepared by having it connected to the host computer and the pressure inside the soft link was stabilized after the air compressor was switched on. The compressor was running with 3.5V and the pressure stabilized at 4kPa.

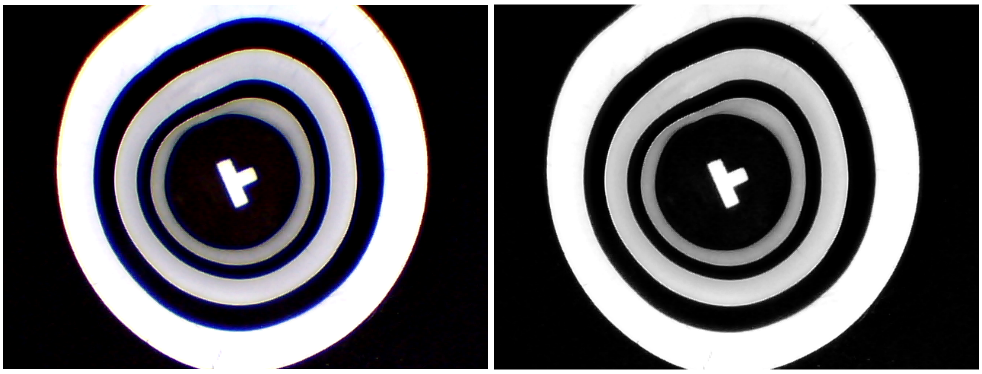
Following this step it was then possible to run the **Calibration Data Model** to obtain the reference values that were consequently used. Its important that when this model was being executed that the prototype remained immobile to avoid disturbances in the reference values.

Finally all requirements were met to execute the **Vision Data Model**. Due to computer processing power limitations this main Simulink® model was executed with a step-size of 0.1 seconds, meaning that it could only save data at a frequency of 10Hz. During this the prototype distal extremity was slowly and carefully pressed against the force sensor while maintaining a constant downwards inclination of 30° with the horizontal plane (4.1). It was important to keep this slope stable to prevent interference with the uniformity of the trials conditions. Furthermore the speed at which the experiment was conducted is not important in view of the fact that the results don't account for dynamical effects, this is because each data point was processed by the neural network individually.

The following sections will follow the flow of the vision data from its acquisition by the webcam up to its final state.

4.1.2.1 Webcam Images

The first step is capturing the inside of the flexible sleeve. For this purpose it was important that the camera's light source was switched on given that otherwise the inner section would be completely dark, rendering impossible to observe the markers. The obtained images were then converted from RGB color space (4.3a) to greyscale (4.3b).



(a) Webcam Output

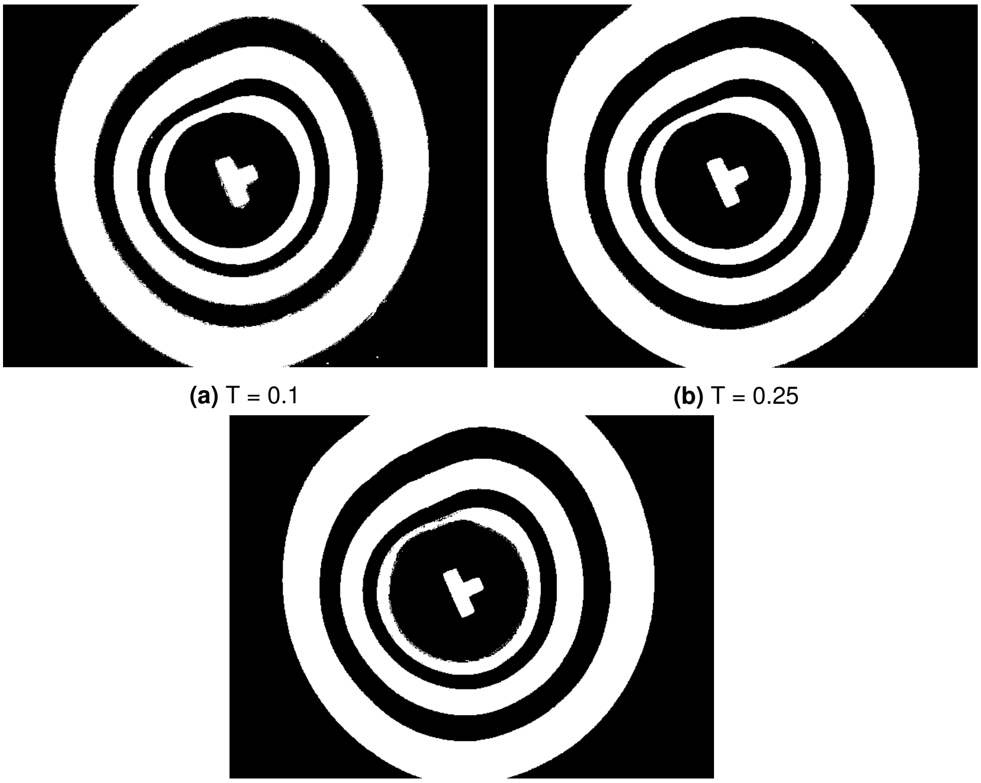
(b) Greyscale Image

Figure 4.3: Before and after color space conversion

There aren't many noticeable differences between the images considering that the major portion of the captured scenario already had dominantly black and white colors. Yet this process is important to rectify any noise in the image and to prepare it for the next step.

4.1.2.2 Image Binarization

In this stage the images were converted from greyscale to binary, but before, it was necessary to choose a threshold value for this transformation. Given that the light conditions inside the soft link don't change, this value could be a constant and therefore a simple trial-and-error method was enough to determine its value. The following figures illustrate this process (4.4).



(a) T = 0.1

(b) T = 0.25

(c) T = 0.5

Figure 4.4: Binarization Thresholds

As can be seen with a threshold of 0.1 introduces some noise around the markers perimeter and the 0.5 threshold doesn't properly capture the smaller annuli. Based on this **T = 0.25** was chosen.

4.1.2.3 Vision Data

The output from the prior section was then analysed to retrieve the statistical information of the connected regions (blob), namely the centroids and the eccentricities.

Only the blobs of the two smaller annuli were analysed considering that the third one isn't captured properly by the camera, avoiding the usage of inaccurate data. This is executed by checking an option in the blob analysis for it to not process any blob that is in contact with the border of the image. Given this condition there will be limits to deforming the prototype because if the remaining two markers reach the sides of the image they will not be processed.

Finally before saving the gathered data the reference values obtained previously were subtracted to output values of the blob analysis in order to only calculate the statistical variation triggered by the deformation of the prototype.

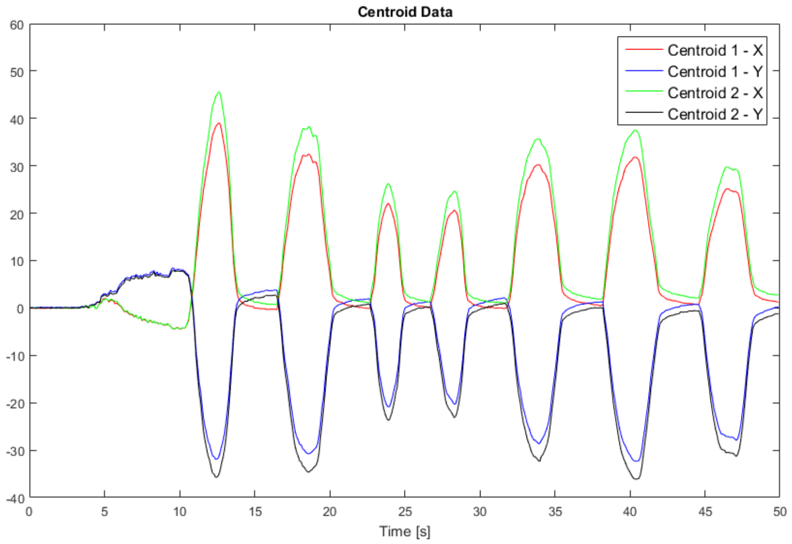


Figure 4.5: Centroid Data

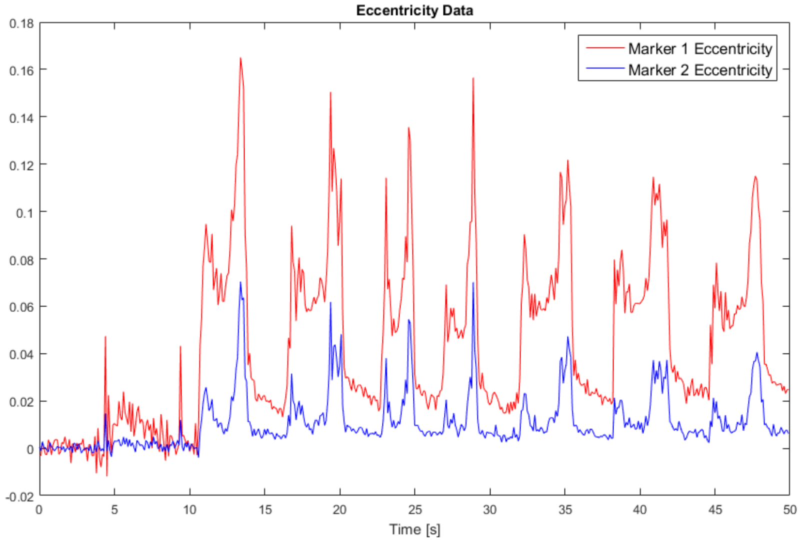


Figure 4.6: Eccentricity Data

4.2 Neural Networks

In the present section the results achieved from using experimental data are presented along with the executed experiments to obtain them. Firstly possible solutions for the design of the neural network for the no-rotation force estimation are explored. Subsequently the chosen structure is shown with the respective accomplished results. Finally the same procedure is repeated for the force estimation with prototype rotation problem.

Aside from the nodes weights, the only tunable parameter that varied in the chosen solution to each problem was the number of neurons in each layer. The remaining specifications remained constant, namely:

- **Learning Method**: supervised;
- **Net Architecture**: feedforward;
- **Input Layer Size**: 6, 4 inputs for the centroids and 2 for the eccentricities;
- **Output Layer Size**: 1, for the estimated force;
- **Output Type**: Continuous.

As mentioned previously, the neural networks performance varies slightly even if they have the same structure and the same training data, this is caused by the initial weights being randomly generated. So, even if the final results are somehow similar there is some stochasticity in the process, given this, it was necessary to perform the training process repeatedly, in order to estimate the real performance value of each specific net structure.

The training of the ANN was executed using the **train** command in MATLAB® that receives the input data, the output data and the training algorithm to produce the trained net.

4.2.1 Without Rotation

The first tackled issue was the force estimation when the direction of the applied force is kept constant ensuring that the relation between displacement-force doesn't vary. With the intention of studying this form of application, this was implemented in the trials by not rotating the prototype when pressing it against the force sensor. For clarification purposes image 4.7 portrays this operation, where **F** represents the force acting on the prototype.

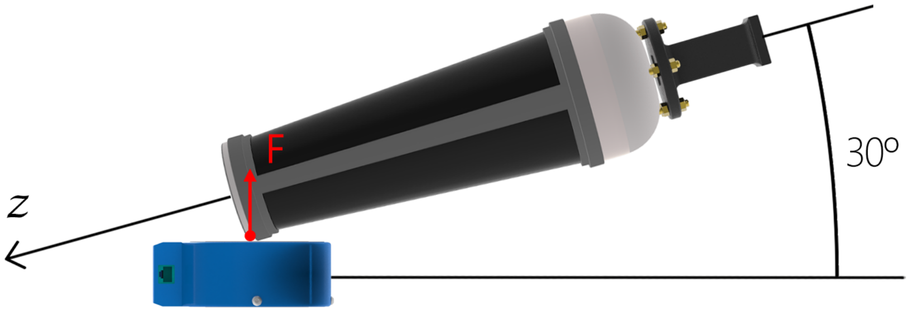


Figure 4.7: Procedure Without Prototype Rotation

From the perspective of the resulting vision data it can be reduced to a one-dimensional problem, where the absolute value of the displacement of the centroids would be enough to calculate the force being applied, although with less accurary. Considering this, a one-layered feedforward neural network is enough to solve this problem if designed with enough neurons.

The number of neurons in the ANN was calculated by creating batches of net's with the same structure, computing their error and finally, evaluating which design had the best performance. Taking into account that the measured performance is actually the mean squared error, then the lower the performance, the better the net is.

The batch processing consisted in generating a net, training it, saving the results and then repeating the process until the desired trials were achieved. When this occurred another batch processing would start with nets with an additional neuron in the hidden layer. For this first experiment this process tested net's with 1 to 40 neurons in the hidden layer.

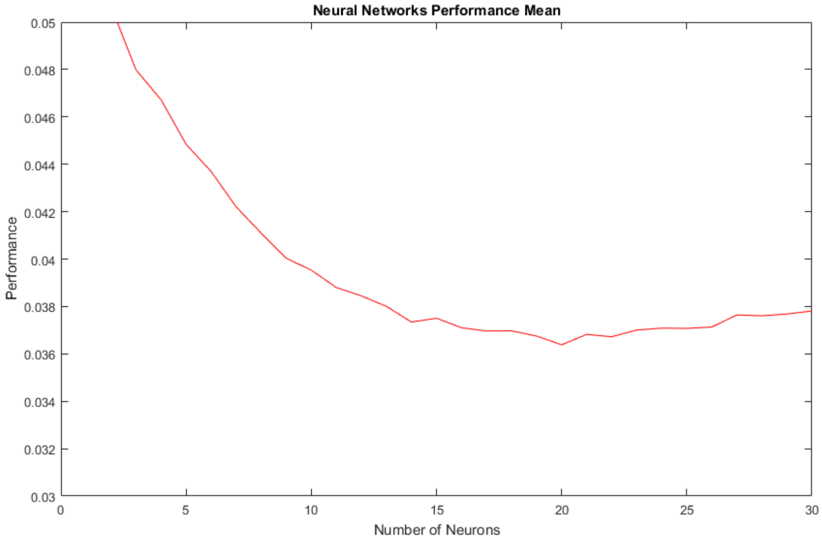


Figure 4.8: ANN Performance Mean

From 4.8 it can be concluded that the net's with best performances have between **15-25** neurons, with the minimum at **20** neurons with an average of **MSE = 0.0364**.

From the batch processing the net with lowest error was also saved and as expected it had **20** neurons and a performance of **MSE = 0.0297**. Using the **Neural Network Testing Model** this net was then tested and its output compared with the real values obtained from the force sensor.

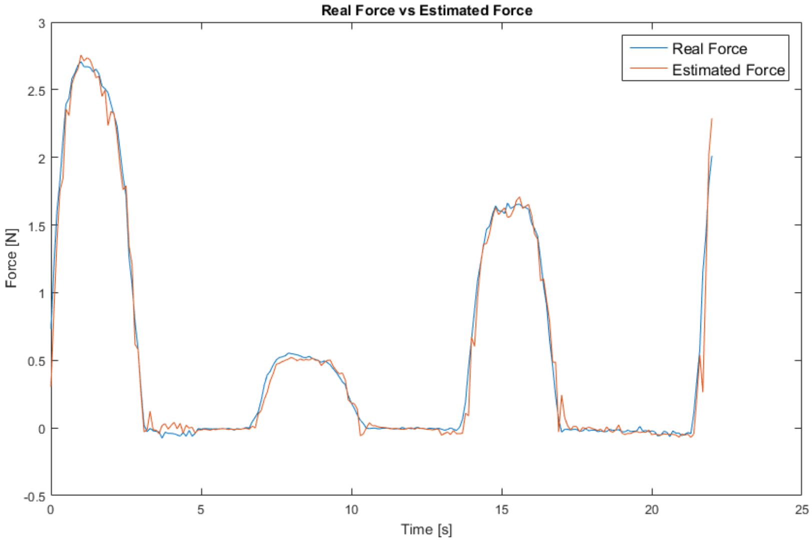


Figure 4.9: ANN Testing

As can be perceived, the results were excellent with the estimated force being very close to the measured force and mean error of **Ē = 0.0515 [N]**.

4.2.2 With Rotation

The force estimation when there is variation of the force direction composes a more complex problem, yet if solved it gives greater liberty when using the prototype in a real application scenario. In contrast with the trials of the previous section this setting was simulated by rotating the prototype when pressing it against the force sensor, as illustrated by figure 4.10.

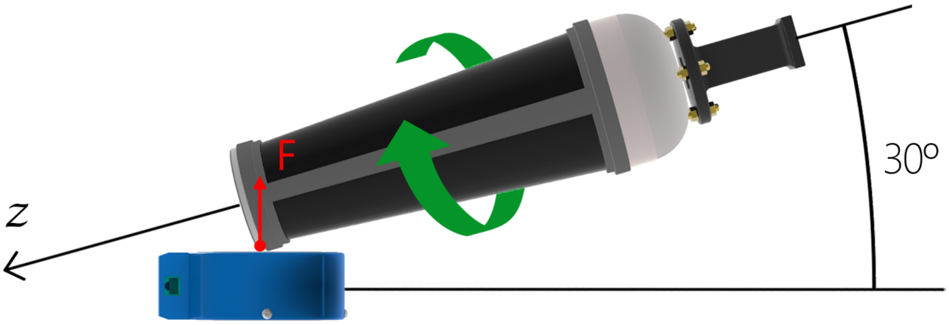


Figure 4.10: Procedure With Prototype Rotation

This problem presents higher complexity given that the relation between the centroid displacement and the force is no longer linear and depends on the direction of the displacement. This factor can be inferred by the combination of the centroid components along with their respective signs. Even if more difficult, the same procedure was used to compute the number of neurons for the ANN structure.

4.2.2.1 One Hidden Layers

As stated previously given the higher complexity of this matter, the more neurons are required to approximate the function that transforms the inputs into the outputs. Given this, the maximum number of neurons in the batch processing was increased to 100. Figure 4.11 presents the results.

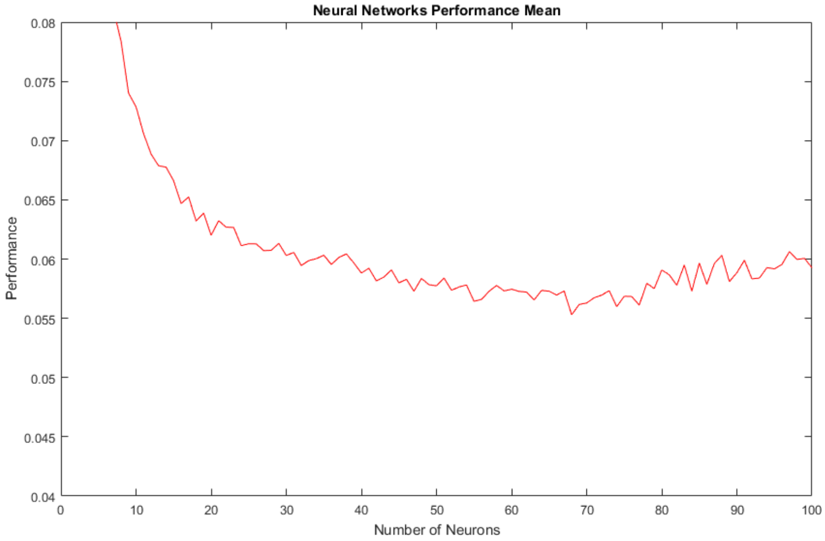


Figure 4.11: ANN Performance Mean

As can be noted, the ANN's with better performances have between **60-80** neurons, with the mean minimum at **68** neurons with an average of **MSE = 0.0567**. Altogether the best result was achieved by a net with **74** neurons and a performance of **MSE = 0.0487**, figure 4.12 presents the outcome of its testing.

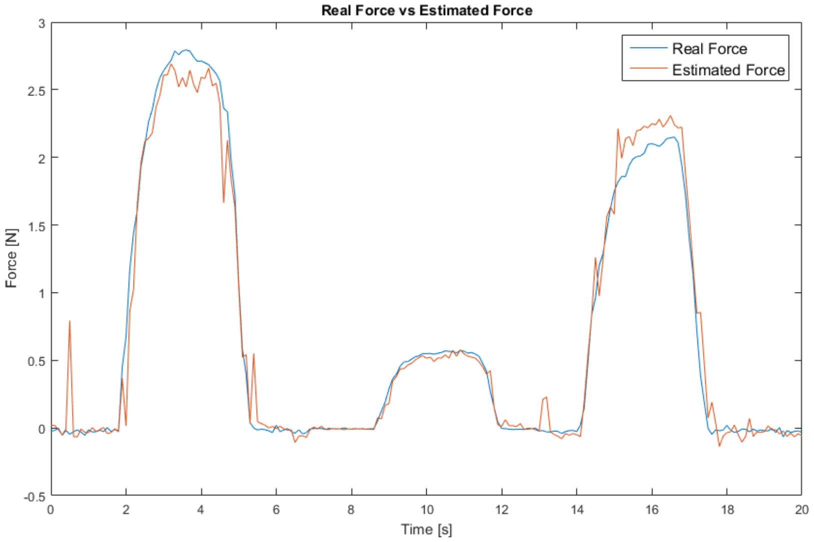


Figure 4.12: 1 Layer ANN Testing

Even though the obtained results were acceptable, with the purpose of trying to improve the problem was also approached using a neural network with two hidden layers.

4.2.2.2 Two Hidden Layers

To tackle this issue the same procedure can't be adopted to each hidden layer, considering that the net's performance isn't result of each layer individually but of both layers as a whole. With this in mind a surface plot was employed to examine the results of the mean performance analysis, with the X and Y axis as the number of neurons in the first and second layers respectively and the Z axis as the average performance value (4.13).

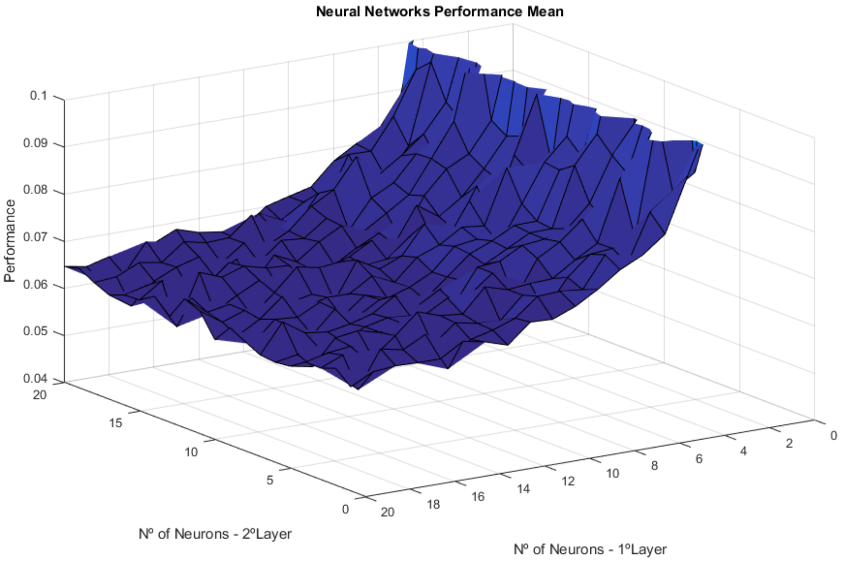


Figure 4.13: ANN Performance Mean

Taking into consideration how computationally demanding this procedure becomes with the increase of the number of layers and therefore more time consuming, the performance analysis was only performed to models where the number of neurons in each layer varied between 1-20.

From reviewing the results it was concluded that ANN's with two hidden layers performed worse with respect to net's with a single layer.

Furthermore from analysis of the graph, it was noted that the net's performance is independent of the dimension of the second layer, only changing when there is variation of the number of neurons in the first layer. In conformity with these results the best obtained ANN had a 16-1 structure (16 neurons in the first hidden layer and only 1 in the second). This means that the net behaved as if it only had a single hidden layer with 16 neurons, where the second layer behaved simply as an output layer that only added a scaling factor to the final result. In conclusion the second hidden layer incremented unnecessarily complexity to the neural network.

In order to illustrate these results, the following figure shows the output of the best net obtained (performance of $\mathbf{ANN_{performance} = 0.0612}$) against the real force values (4.14).

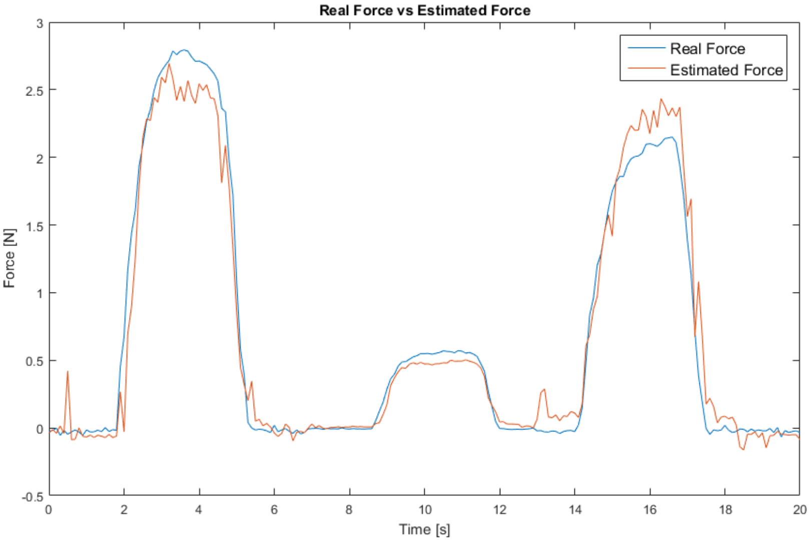


Figure 4.14: 2 Layer ANN Testing

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this final section conclusions are drawn on the reached results from the point of view of the work's success along with its real-life application and limitations. The prototype's building process is commented in addition to an overview of the computer vision techniques and artificial neural networks performance.

Primarily the prototype's conception required thorough planning and general craftsmanship. Although it physically performed as intended there was the issue of it not being totally airtight and it could be simpler by having an axysymmetrical behaviour.

With regards to the image processing procedure designed it was shown that it was appropriate to acquire the desired data from the images obtained from the webcam. Furthermore it was computationally light enough to be able to estimate the force in real-time, enabling its application in a real-case scenario where the exerted force must be measured promptly.

From the results analysis it was shown that artificial neural networks are a suitable approach in the scope of this work, being able to infer, with some degree of precision, the force values from relevant vision data.

The overall performance of the estimator can be separated into two sections, depending on its form of application:

- Without prototype rotation, for this type of implementation the prototype presented remarkable results proving to be an highly adequate solution. Table 5.1 shows the error of the obtained best results;

Maximum Error [N]	Mean Error [N]	Mean Percentage Error [%]
0.531	0.051	1.250

Table 5.1: Error Without Prototype Rotation

- With prototype rotation, its was shown that an ANN with a single hidden layer performs better than a net with more hidden layers for the input data used. Although the performance was lower when comparing to the previously mentioned application, as can be noted from table 5.2, the results were satisfactory but can be improved with further analysis.

	Maximum Error [N]	**Mean Error [N]**	**Mean Percentage Error [%]**
1 Hidden Layer	0.902	0.185	5.446
2 Hidden Layers	0.882	0.255	7.094

Table 5.2: Error With Prototype Rotation

As final conclusion, although this line of investigation still has a lot to improve, it was demonstrated that the estimation of force using soft structures combined with the application of computer vision and artificial neural networks techniques, presents a practicable solution.

5.2 Future Work

In the interest of pursuing the work initiated within the scope of this dissertation and to deepen the insight on force estimation by using computer vision with artificial neural networks, the following recommendations are made:

- Ensure the air tightness of the prototype. Although it was a major focus point of its manufacture some air leaks existed and despite the fact that the pneumatic circuit compensated for any air losses the overall prototype pressure wasn't constant. This factor influences directly the relation between force and deformation so any improvements on this side will be noticed in the increase of the estimator's accuracy;
- Increase the number of markers. By using more inputs in the ANN its results will possibly increase in precision;
- Perform a feature selection study with more inputs to verify which data contributes more for the prediction of the output;
- Study of the impact of the inclination. Using a gyroscope to measure the orientation and employing this information in order for the prototype to estimate the force correctly no matter the angle in which it approaches the touching surface;
- Widening the estimator's utility by analysing the measurement of torque using the variation of the semi-cross marker orientation.

Lastly, in addition to this propositions, for further validation of the viability of this prototype it’s crucial to analyse its performance in a real case scenario. This can be achieved by setting up the prototype in a robotic manipulator, executing a set of experimental trials that require feedback of the applied force and finally compare the obtained results against the desired specifications.

Bibliography

[1] S. Kim, C. Laschi, and B. Trimmer. “Soft robotics: a bioinspired evolution in robotics”. In: *Trends in Biotechnology* 31.5 (2013), pp. 287–294. ISSN: 0167-7799. DOI: https://doi.org/10.1016/j.tibtech.2013.03.002. URL: http://www.sciencedirect.com/science/article/pii/S0167779913000632.

[2] S. Li and K. W. Wang. “Plant-inspired adaptive structures and materials for morphing and actuation: a review”. In: *Bioinspiration & Biomimetics* 12.1 (Dec. 2016), p. 011001. DOI: 10.1088/1748-3190/12/1/011001. URL: https://doi.org/10.1088.

[3] I. Robert J. Webster and B. A. Jones. “Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review”. In: *The International Journal of Robotics Research* 29.13 (2010), pp. 1661–1683. DOI: 10.1177/0278364910368147. URL: https://doi.org/10.1177/0278364910368147.

[4] F. Schmitt, O. Piccin, L. Barbé, and B. Bayle. “Soft Robots Manufacturing: A Review”. In: *Frontiers in Robotics and AI* 5 (2018), p. 84. ISSN: 2296-9144. DOI: 10.3389/frobt.2018.00084. URL: https://www.frontiersin.org/article/10.3389/frobt.2018.00084.

[5] R. Pelrine et al. “Dielectric elastomer artificial muscle actuators: toward biomimetic motion”. In: *Smart Structures and Materials 2002: Electroactive Polymer Actuators and Devices (EAPAD)*. Ed. by Y. Bar-Cohen. Vol. 4695. International Society for Optics and Photonics. SPIE, 2002, pp. 126–137. DOI: 10.1117/12.475157. URL: https://doi.org/10.1117/12.475157.

[6] B. Jin et al. “Programming a crystalline shape memory polymer network with thermo- and photo-reversible bonds toward a single-component soft robot”. In: *Science Advances* 4.1 (2018). DOI: 10.1126/sciadv.aao3865. eprint: https://advances.sciencemag.org/content/4/1/eaao3865.full.pdf. URL: https://advances.sciencemag.org/content/4/1/eaao3865.

[7] D. Rus and M. T. Tolley. “Design, fabrication and control of soft robots”. In: *Nature* 521 (2015), pp. 467–475. DOI: http://dx.doi.org/10.1038/nature14543.

[8] H. Abidi and M. Cianchetti. “On Intrinsic Safety of Soft Robots”. In: *Frontiers in Robotics and AI* 4 (2017), p. 5. ISSN: 2296-9144. DOI: 10.3389/frobt.2017.00005. URL: https://www.frontiersin.org/article/10.3389/frobt.2017.00005.

[9] J. Rossiter and H. Hauser. “Soft Robotics - The Next Industrial Revolution?” In: *IEEE Robotics Automation Magazine* 23.3 (2016), pp. 17–20. DOI: 10.1109/MRA.2016.2588018.

[10] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario. “Biomedical applications of soft robotics”. In: *Nature Reviews Materials* 3.6 (2018), pp. 143–153. ISSN: 2058-8437. DOI: 10.1038/s41578-018-0022-y. URL: https://doi.org/10.1038/s41578-018-0022-y.

[11] *Soft robot to swim through Europa’s oceans*. http://news.cornell.edu/stories/2015/05/soft-robot-swim-through-europas-oceans. Accessed: 2019-10-27.

[12] V. L. Nickle, J. Perry, and A. L. Garrett. “Development of Useful Function in the Severely Paralyzed Hand”. In: *JBJS* 45.5 (1963). ISSN: 0021-9355.

[13] G. Robinson and J. Davies. “The parallel bellows actuator.” English. In: Robotica 98, Robotica 98 ; Conference date: 01-11-1998. 1998, pp. 195–200.

[14] R. Kornbluh, R. Pelrine, J. Eckerle, and J. Joseph. “Electrostrictive polymer artificial muscle actuators”. In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. Vol. 3. 1998, 2147–2154 vol.3.

[15] I. Boblan et al. “A Human-Like Robot Hand and Arm with Fluidic Muscles: Biologically Inspired Construction and Functionality”. In: *Embodied Artificial Intelligence: International Seminar, Dagstuhl Castle, Germany, July 7-11, 2003. Revised Papers*. Ed. by F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 160–179. DOI: 10.1007/978-3-540-27833-7_12. URL: https://doi.org/10.1007/978-3-540-27833-7_12.

[16] G. Immega and K. Antonelli. “The KSI tentacle manipulator”. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. Vol. 3. May 1995, 3149–3154 vol.3. DOI: 10.1109/ROBOT.1995.525733.

[17] M. W. Hannan and I. D. Walker. “The ’elephant trunk’ manipulator, design and implementation”. In: *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)*. Vol. 1. 2001, 14–19 vol.1. DOI: 10.1109/AIM.2001.936423.

[18] B. Trimmer et al. “Caterpillar locomotion: A new model for soft-bodied climbing and burrowing robots”. In: *Proceedings of the 7th International Symposium on Technology and the Mine Problem* (Apr. 2014).

[19] Z. Zhang, A. Petit, J. Dequidt, and C. Duriez. “Calibration and External Force Sensing for Soft Robots Using an RGB-D Camera”. In: *IEEE Robotics and Automation Letters* 4.3 (July 2019), pp. 2356–2363. DOI: 10.1109/LRA.2019.2903356.

[20] H. R. Nicholls and M. H. Lee. “A Survey of Robot Tactile Sensing Technology”. In: *I. J. Robotics Res.* 8.3 (1989), pp. 3–30. URL: https://doi.org/10.1177/027836498900800301.

[21] C. Chi, S. Xuguang, N. Xue, T. Li, and C. Liu. “Recent Progress in Technologies for Tactile Sensors”. In: *Sensors* 18 (Mar. 2018), p. 948. DOI: 10.3390/s18040948.

[22] P. Puangmali, K. Althoefer, L. Seneviratne, D. Murphy, and P. Dasgupta. “State-of-the-Art in Force and Tactile Sensing for Minimally Invasive Surgery”. In: *IEEE Sensors Journal* 8.4 (2008), pp. 371–381. DOI: https://doi.org/10.1109/JSEN.2008.917481.

[23] G. Kinoshita, S. Aida, and M. Mori. “Pattern Classification of the Grasped Object by the Artificial Hand”. In: *Proceedings of the 3rd International Joint Conference on Artificial Intelligence. Standford, CA, USA, August 20-23, 1973*. 1973, pp. 665–670. URL: http://ijcai.org/Proceedings/73/Papers/072.pdf.

[24] M. Lee and H. Nicholls. “Review Article Tactile sensing for mechatronics—a state of the art survey”. In: *Mechatronics* 9.1 (1999), pp. 1–31. DOI: 10.1016/s0957-4158(98)00045-2.

[25] M. H. Lee. “Tactile Sensing: New Directions, New Challenges”. In: *I. J. Robotics Res.* 19.7 (2000), pp. 636–643. URL: https://doi.org/10.1177/027836490001900702.

[26] M. Tiwana, S. Redmond, and N. Lovell. “A Review of Tactile Sensing Technologies with Applications in Biomedical Engineering”. In: *Sensors and Actuators A: Physical* 179 (June 2012), pp. 17–31. DOI: 10.1016/j.sna.2012.02.051.

[27] A. Blandin, I. Bernardeschi, and L. Beccai. “Biomechanics in Soft Mechanical Sensing: From Natural Case Studies to the Artificial World”. In: *Biomimetics* 3 (Oct. 2018), p. 32. DOI: 10.3390/biomimetics3040032.

[28] A. Verendeev et al. *Comparative analysis of Meissner's corpuscles in the fingertips of primates.* 2015. URL: https://www.ncbi.nlm.nih.gov/pubmed/26053332.

[29] G. J. Gerling and G. W. Thomas. "The effect of fingertip microstructures on tactile edge perception". In: *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. Mar. 2005, pp. 63–72. DOI: 10.1109/WHC.2005.129.

[30] G. Obinata, A. Dutta, N. Watanabe, and N. Moriyama. "Vision Based Tactile Sensor Using Transparent Elastic Fingertip for Dexterous Handling". In: Feb. 2007. ISBN: 3-86611-283-1. DOI: 10.5772/4771.

[31] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter. "Development of a tactile sensor based on biologically inspired edge encoding". In: *2009 International Conference on Advanced Robotics*. June 2009, pp. 1–6.

[32] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter. "Tactile edge detection". In: *SENSORS, 2010 IEEE*. Nov. 2010, pp. 2593–2598. DOI: 10.1109/ICSENS.2010.5690181.

[33] E. Knoop and J. Rossiter. "Dual-mode compliant optical tactile sensor". In: *2013 IEEE International Conference on Robotics and Automation*. May 2013, pp. 1006–1011. DOI: 10.1109/ICRA.2013.6630696.

[34] W. Fischer. *Digital Video and Audio Broadcasting Technology. A Pratical Engineering Guide*. Springer, 2008, p. 81.

[35] D. Liu and J. Yu. "Otsu Method and K-means". In: *2009 Ninth International Conference on Hybrid Intelligent Systems*. Vol. 1. 2009, pp. 344–349. DOI: 10.1109/HIS.2009.74.

[36] W. S. McCulloch and W. H. Pitts. "A Logical Calculus of the ideas immanent in nervous activity". In: *Bulletin of Mathematical Biophysics* 5.1 (1943), pp. 115–133. URL: https://doi.org/10.1007/BF02478259.

[37] D. Silver, T. Hubert, J. Schrittwieser, and D. Hassabis. *AlphaZero: Shedding new light on chess, shogi, and Go*. 2018. URL: https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go (visited on 08/21/2019).

Appendix A - Additional Results

The following figures illustrate the shape of the collected data from the two types of trials executed. Firstly, figures (A.1), (A.2) and (A.3) for experiments without prototype rotation and subsequently figures (A.4), (A.5) and (A.6) for experiments with prototype rotation.

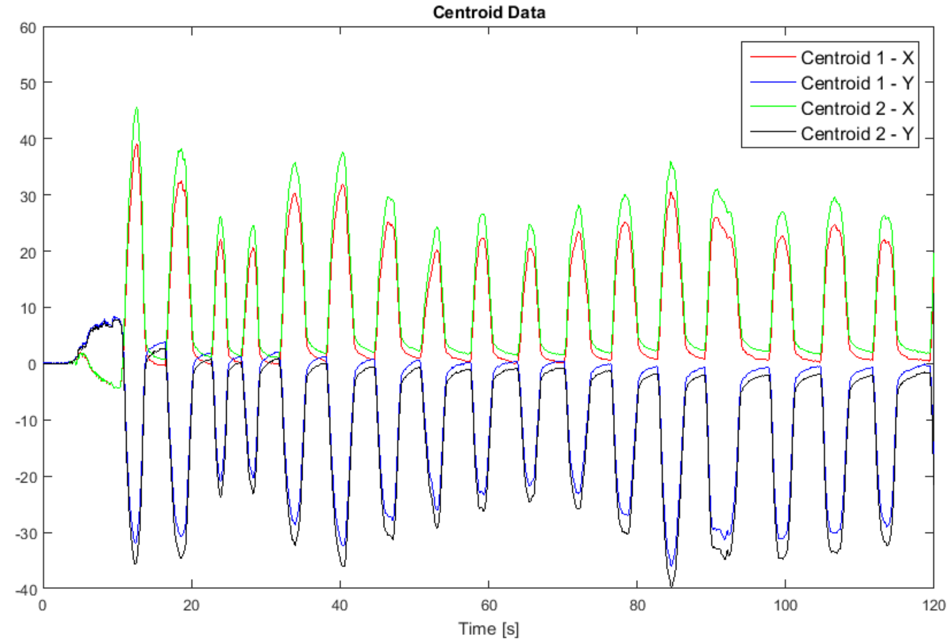


Figure A.1: Centroid Data Without Prototype Rotation

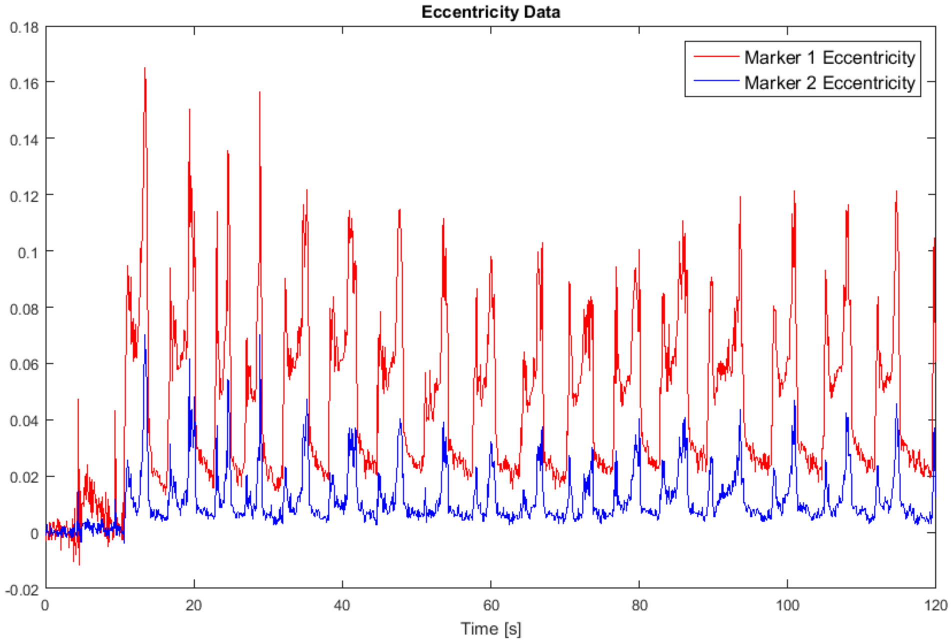


Figure A.2: Eccentricity Data Without Prototype Rotation

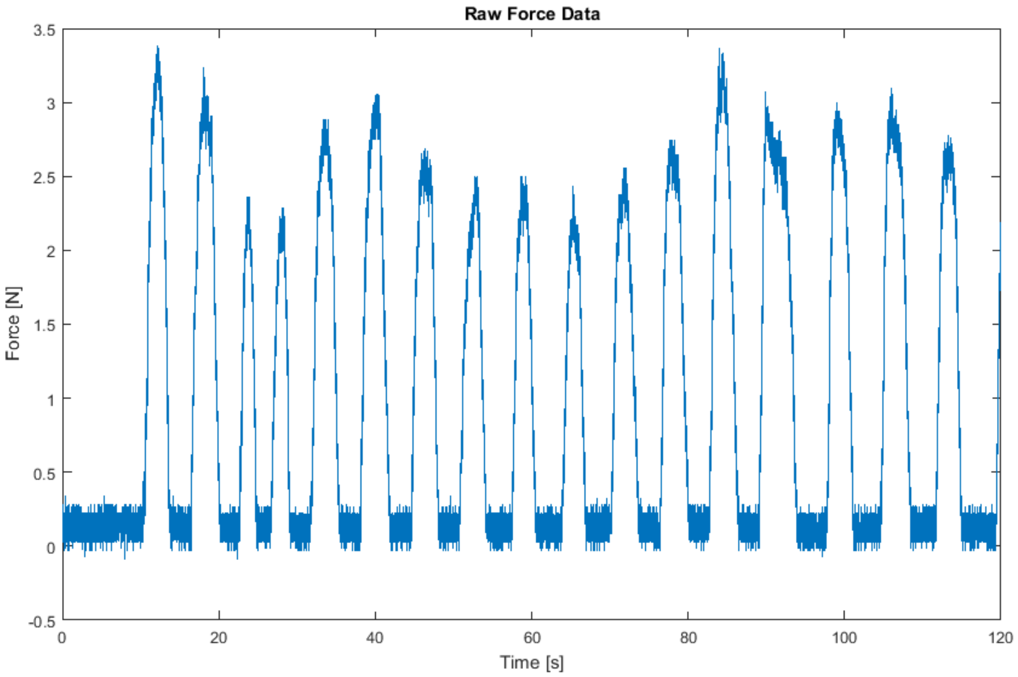


Figure A.3: Unprocessed Force Data Without Prototype Rotation

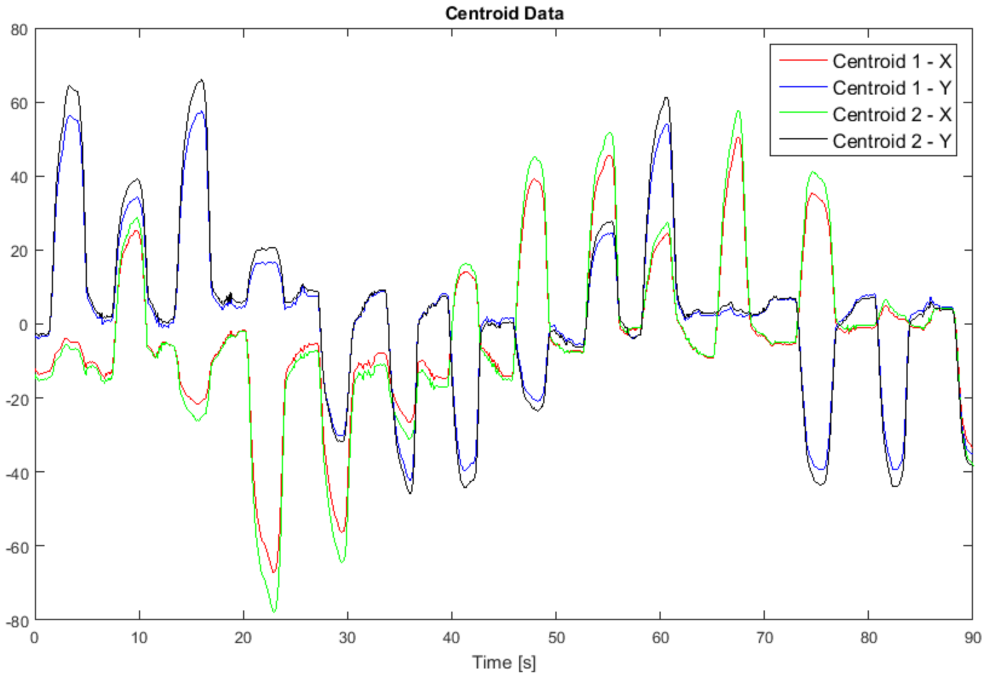


Figure A.4: Centroid Data With Prototype Rotation

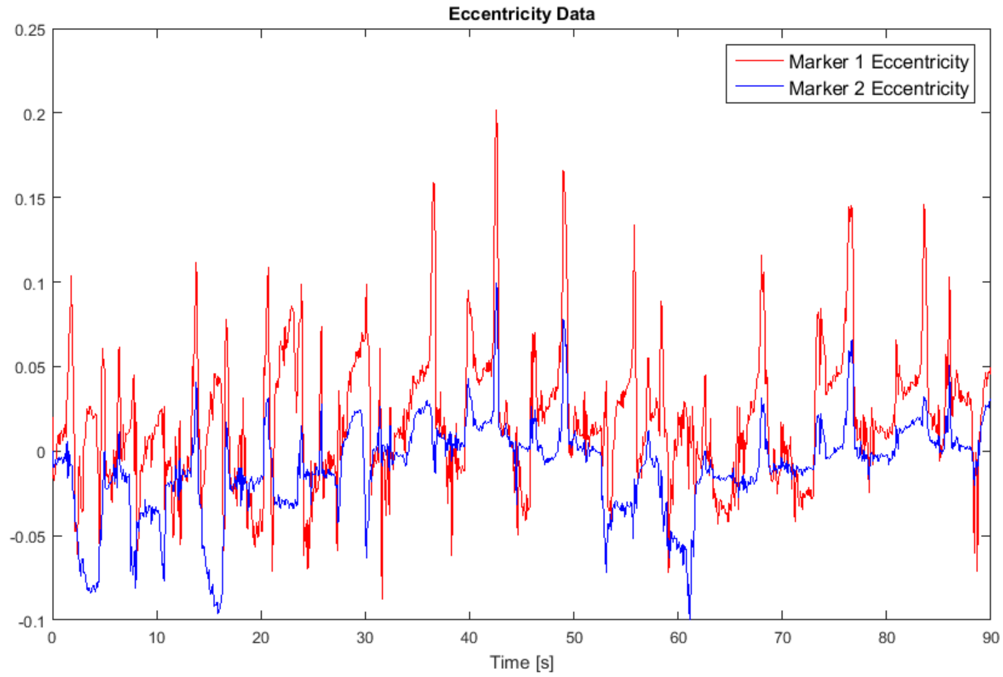


Figure A.5: Eccentricity Data With Prototype Rotation

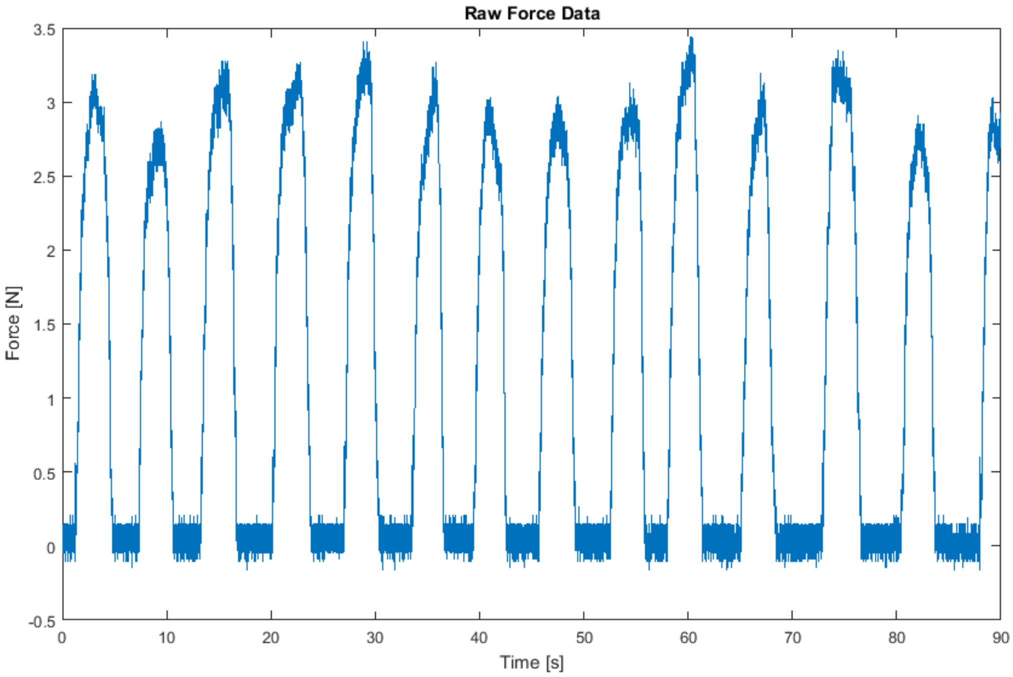


Figure A.6: Unprocessed Force Data With Prototype Rotation