



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ


TECHNISCHE UNIVERSITÄT
ILMENAU

Pontificia Universidad Católica del Perú

Escuela de Posgrado

Tesis de Maestría

Investigation and development of a flexible gripper with
adaptable finger geometry

Para obtener el grado de:

Master of Science (M. Sc.)

en Ingeniería Mecánica

Presentado por: Geancarlo Enzo Ramos Gómez

Fecha y Lugar de Nacimiento: 08/01/1989 Lima, Perú

Tutor Responsable (TU Ilmenau): Dr.-Ing. René Uhlig

Professor Responsable (TU Ilmenau): Prof. Dr.-Ing. habil. Lena Zentner

Professor Responsable (PUCP): Dr. Jorge Antonio Rodríguez Hernández

Fecha y Lugar: 14/03/2019, Lima

Declaration of Authorship

I confirm that this Master thesis is my own work and I have documented all sources and material used.

This thesis was not previously presented to another examination board and has not been published.

Signed: _____

Date and place: _____



Acknowledgements

First, I would like to thank God because he has given me all I needed, including the wisdom to fulfill this work. It is written in the bible that the wisdom is one of the gifts from the Lord, and through this experience, I have noticed how this verse has turned into reality in my life.

**All wisdom comes from the LORD,
and so do common sense
and understanding.**

**⁷ God gives helpful advice
to everyone who obeys him
and protects all of those
who live as they should.**

**⁸ God sees that justice is done,
and he watches over everyone
who is faithful to him. Proverbs 2:6-8**

Also, I would like to thank my mother, who encourages me to keep going and to trust the Lord every day.

Besides, I would like to thank Prof. Dr.-Ing. habil. Lena Zentner, who gave me the opportunity to write this work at the Compliant Systems Group. Finally, I would like to express my special thanks to Dr.-Ing. René Uhlig for his patient guidance and valuable support introducing me into the field of compliant mechanisms. Without his support, this thesis could not be successfully conducted.

Zusammenfassung

Das zuverlässige und schonende Greifen ist ein Hauptanliegen bei der Entwicklung von neuartigen Greifvorrichtungen. Je größer die Kontaktfläche zwischen dem Greifer und dem Greifobjekt ist, desto schonender und zuverlässiger ist der Greifvorgang. Um dieses Ziel zu erreichen wurden in den letzten Jahrzehnten zahlreiche Untersuchungen zu adaptiven passiven Greifern durchgeführt. Ein neuer Forschungszweig im Bereich selbstadaptiver Greifer sind Greifer mit nachgiebigen blattfederartigen Greifelementen (Greiferfinger). Die Funktionsweise basiert auf dem elastischen Ausknicken der Greifelemente infolge einer translatorische Antriebsbewegung

Die vorliegende Arbeit konzentriert sich auf die Verbesserung des Greifvorgangs, indem die Kontaktlänge zwischen den blattfederartigen Greiferfingern und dem zu greifenden Objekt deutlich erhöht wird. Um diese Aufgabenstellung zu lösen, muss eine geeignete Greifergeometrie für ein gegebenes Greifobjekt berechnet werden.

Die gezielte Berechnung der erforderlichen Greifergeometrie für ein bekanntes Greifobjekt ist nicht möglich. Daher wurde als Lösungsansatz die umkehrte Richtung gewählt. Für eine definierte Greifgeometrie wird die Gestalt des dazu passenden "idealen" Greifobjektes ermittelt und anschließend mit der Gestalt zu greifenden Objektes verglichen. Bei Gestaltabweichungen wird die Greifergeometrie iterative verändert, bis seine geeignete Greifergeometrie gefunden wurde. Im Rahmen der vorliegenden Arbeit wird zunächst die Ermittlung des "idealen" Greifobjektes behandelt. Es wurde ein Algorithmus entwickelt, der für eine vorgegebene Greifergeometrie die Gestalt eines runden bzw. elliptischen Objektes ermittelt. Der Algorithmus verwendet als Eingabedaten die Biegelinien der elastisch ausgeknickten Greiffinger unter Berücksichtigung unterschiedlicher Randbedingungen. Als Ausgabedaten liefert der Algorithmus die Gestalt des passenden Greifobjektes zurück. Für quadratische bzw. rechteckige sowie für dreieckige Objekte wurden unterschiedliche Greifgeometrien untersucht. Außerdem wird für quadratische und rechteckige Objekte das Lösungskonzept für die Entwicklung eines weiteren Algorithmus beschrieben.

In Kapitel 1 wird eine Klassifizierung von Greifern basierend auf der Anpassungsfähigkeit vorgestellt. In Kapitel 2 werden Lösungskonzepte, Modelle und Theorien vorgestellt. In Kapitel 3 werden Ablaufdiagramme der Algorithmen dargestellt. In Kapitel 4 wird die Entwicklung des Algorithmus für elliptische Objekte und deren Betriebsmodi beschrieben. In Kapitel 5 werden Greifgeometrien für quadratische bzw. Rechteckige sowie für dreieckige Objekte analysiert und die Ideen eines Algorithmus für quadratisch bzw. rechteckige Objekte beschrieben. In Kapitel 6 wird ein kurzer Überblick über die zukünftige Arbeiten.

Summary

Reliable and gentle gripping is a major concern in the development of new gripping devices. The larger contact surface between the gripper and the gripping object, the gentler and more reliable the gripping process. In order to achieve this goal, further investigations on adaptive passive grippers have been carried out in the recent decades. A new branch of research in the field of self-adaptive grippers are compliant leaf-spring-like gripping elements (gripper fingers). Its mode of operation is based on the elastic buckling of the gripping elements as a result of a translatory drive movement.

The present work focuses on improving the gripping process by increasing significantly the contact length between the compliant leaf-spring-like gripper fingers and the object to be gripped. In order to solve this task, a suitable gripper geometry for a given gripping object should be calculated

The specific calculation of the required gripper geometry for a known gripping object is not possible; therefore, this work aims in the opposite direction. For a defined gripping geometry, the shape of the matching “ideal” gripping object is determined and then compared with the desired object to be gripped. In case of a deviation in the size, the gripper geometry is iteratively changed until its suitable gripper geometry has been found. In the present work, the determination of the “ideal” gripping object is the first task to deal with. An algorithm has been developed to determine the shape of a round-elliptical object for a given gripper geometry. The algorithm uses as data input the bend lines of the compliant two-gripper finger under different boundary conditions. As data output, the algorithm returns the shape of the matching gripping object. For square-rectangular and triangular objects, different gripping geometries have been investigated. Furthermore, for square-rectangular objects, solution concepts for the development of an algorithm is described.

In chapter 1, a classification based on adaptability is presented. In chapter 2, solution concepts, models and theories involved are introduced. In chapter 3, process flow diagrams of the algorithms are presented. In chapter 4, the development of the algorithm for elliptical objects and its operation modes are described. In chapter 5, gripping geometries for square-rectangular and triangular objects are analysed and the ideas of an algorithm for square-rectangular objects are described. In chapter 6, a brief overview of the futur work is commented.

Contents

Zusammenfassung.....	i
Summary	ii
Contents	iii
List of Figures.....	v
List of Tables and Diagrams	vii
Chapter 1	1
State of the Art.....	1
1.1 Framework.....	1
1.2 Gripper's definition	1
1.3 Classification of end-effectors.....	2
1.3.1 Non-adaptive grippers.....	3
1.3.2 Adaptive grippers	7
1.3.3 Self-adaptive grippers	10
1.3.4 Self-Adaptive Passive Grippers	15
Chapter 2.....	27
Solution concept and theories	27
2.1 Solution concept	27
2.2 Mechanical models of the compliant gripper.....	28
2.3 Non-linear deformation theory	35
2.4 Least squares method.....	38
Chapter 3.....	41
Goal description and process flow diagrams	41
3.1 Goal description	41
3.2 Process flow diagram: round-elliptical objects	42
3.3 Process flow diagram: square-rectangular objects	45
Chapter 4.....	46
Algorithm for fitting elliptical objects.....	46
4.1 Module 0: non-linear theory- differential equations.....	46
4.2 Module 1: data input extraction and filtration of the range	47
4.3 Module 2 : function <i>ellipse_fit</i> - least squares method.....	49
4.3.1 Creation of the matrices using the filtered data input.....	49
4.3.2 Least squares iterations : ellipse parameters calculation	50
4.3.3 Evaluation of the semi-major axis of the ellipse.....	51

4.3.4	New tasks to solve.....	53
4.4	Module 3 : Task 1 – Location process.....	55
4.5	Module 4: Creation of the new geometry.....	56
4.5.1	Task 2 – Alignment of x_i values.....	56
4.5.2	Task 3 - calculation of the variable <i>differenceY</i>	56
4.5.3	Selection of the new y_i component for the new geometry.....	57
4.5.4	Variable <i>data_iter2</i> and evaluation of the ellipse major – axis.....	58
4.6	Exit-loop conditions.....	59
4.7	Resetting the variables.....	60
4.7.1	Resetting least squares method iteration variables.....	60
4.7.2	Resetting range variables.....	60
4.9	Operation modes.....	62
Chapter 5.....		68
Investigations of square-rectangular and triangular objects.....		68
5.1	Enhanced segments.....	68
5.2	Simulations: square–rectangular objects.....	69
5.3	Algorithm overview for fitting square–rectangular objects.....	71
5.4	Algorithm overview for fitting triangular objects.....	75
Chapter 6.....		80
Conclusions and future work.....		80
6.1	Conclusions.....	80
6.2	Future work.....	81
References.....		84
Appendix.....		87
Algorithm Variables.....		87
1.A	Module 0: Non- linear deformation theory algorithm.....	87
1.B	Module 1: Range Filtering Variables.....	88
1.C	Module 2: Least Squares Method.....	90
1.D	Module 3: Location variables.....	91
1.E	Module 4: Creation of the new geometry.....	92
1.F	Exit-Loop Variables.....	93
1.G	Flags.....	93
1.H	Loop Counters.....	93

List of Figures

Fig. 1.1: (a) Connected differential mechanism [4] ; (b) Underactuated fingers [5].	2
Fig. 1.2: (a) Gripper using slider-crank mechanism; (b) Linkage actuation mechanisms. [10]	4
Fig. 1.3: (a) 4-Bar linkage gripper; (b) 16 Xboard–DC motor-worm. [9]	4
Fig. 1.4: (a) Permanent magnet; (b) Electromagnet [14]	6
Fig. 1.5: (a) Suction pads; (b) Suction force; (c) Shear force in cup. [15]	6
Fig. 1.6: (a) Air picker inflating; (b) Air gripper inflating. [16]	7
Fig. 1.7: (a) Festo learning-gripper [17] ; (b) Bellow of 2 degrees of freedom [18].	8
Fig. 1.8: (a) Degrees of freedom [19] ; (b) Final hardware [18].	9
Fig. 1.9: (a) Side view of the fingertip of finger F7; (b) finger F7. [18]	9
Fig. 1.10: (a) Sectional view electro-adhesion system; (b) electro-adhesive forces[20].	10
Fig. 1.11: (a) Pre-stretched elastomer membrane ; (b) Electro-adhesion forces [22].	11
Fig. 1.12: (a) Five-layer gripper; (b) Arrangement; (c) Finger open-close position. [22]	12
Fig. 1.13: (a) Voltage off position; (b) Voltage on position. [22]	12
Fig. 1.14: (a) Finray effect; (b) Forces on a fish tail and a two- finger-finray gripper. [24]	13
Fig. 1.15: Mechanical changeover parallel (left) and centric grip direction (right). [25]	14
Fig. 1.16: Three layers pneunets bending FEAs. [23]	15
Fig. 1.17: Gripper fitting capability, examples of gripper’s form adaptability. [27]	16
Fig. 1.18: (a) Chamber 1 - compressed air, Chamber 2- water; (b) Holding mechanism. [27]	17
Fig. 1.19: (a) Robotiq 2-finger adaptive gripper; (b) Encompassing grip; (c) Parallel grip. [28]	18
Fig. 1.20: (a) Internal gripping; (b) External gripping. [28]	19
Fig. 1.21: Equilibrium line on the two-finger, shown with no fingertip pads. [28]	19
Fig. 1.22: (a) Position 1; (b) Position 2 [29]	20
Fig. 1.23: (a) Basic mode; (b) Wide mode; (c) Pinch mode; (d) Scissor mode [29]	21
Fig. 1.24: (a) Closing and opening; (b) Encompassing and fingertip grips [29]	21
Fig. 1.25: Acting forces on the adhesive gripper [31]	22
Fig. 1.26: (a) Rest mode, motion, rest mode; (b) Bi-stable structure [31]	23
Fig. 1.27: Support structure. (A) rest position, (B) grasping, (C) lifting, (D) release [31]	23
Fig. 1.28: (a) Manduca Sexta prolegs; (b) Magnified view of crochets [33]	24
Fig. 1.29: (a) TPG, soft parts- blue, hard parts-red; (b) Fin ray effect [33]	25
Fig. 1.30: (a) Closed position; (b) Open position; (c) Magnified crochets [33]	25
Fig. 1.31: (a)-(b) Perpendicular payload 530 grams; (c) Parallel payload 240 grams [34]	26

Fig. 2.1 Concept of an adaptable gripper with two compliant gripper fingers	27
Fig. 2.2: Model 1 - Gripper finger in unloaded condition	28
Fig. 2.3: Model 1 - Gripper finger in loaded condition.....	29
Fig. 2.4 Model 1 - Gripper finger - free body diagram	29
Fig. 2.5: Model 2 - Gripper finger in unloaded condition	30
Fig. 2.6: Model 2 - Gripper finger in loaded condition.....	30
Fig. 2.7: Model 2 - Gripper finger - free body diagram	30
Fig. 2.8: Element A - Constant stiffness gripper finger	31
Fig. 2.9: Element B - Gripper finger with one enhanced segment.....	31
Fig. 2.10: Element C - Gripper finger with two enhanced segments	31
Fig. 2.11: Element D - Gripper finger with three enhanced segments.....	31
Fig. 2.12: Model 1 - Element A, grip condition – elliptical objects.....	32
Fig. 2.13: Model 2 - Element B, grip condition – rectangular object	32
Fig. 2.14: Model 2 - Element C, grip condition – square object.....	33
Fig. 2.15: Model 2 - Element B, grip condition – rectangular object	33
Fig. 2.16: Model 2 - Elements B and C, grip condition – triangular object.....	33
Fig. 2.17: Model 2 - Element B, grip condition – triangular object	34
Fig. 2.18: Rod under large deformation u ; (a) Unloaded condition; (b) Loaded condition [35]	35
Fig. 2.19: Differential element of the rod under external loads [35].....	36
Fig. 2.20: (a) Three-dimensional representation; (b) Two-dimensional representation [36]	39
Fig. 4.1: Data input – plot of the gripper bend lines – Model 1.....	46
Fig. 4.2: Data input – plot of the gripper bend lines – Model 2.....	47
Fig. 4.3: Data input filtration – Model 1	48
Fig. 4.4: Data input filtration – Model 2	48
Fig. 4.5: Ellipse plot first least squares iteration, $d < 0.15$	51
Fig. 4.6: Ellipse plot first least squares iteration, $d > 0.15$	51
Fig. 4.7: Ellipse plot – first least squares iteration with filtering of the semi-major axis	52
Fig. 4.8: Ellipse plot – first least squares iteration without filtering of the semi-major axis	53
Fig. 4.9: Ellipse centered at (X_c, Y_c) , axis lengths a' , b' , angle α [38].....	54
Fig. 4.10: Location process – finding the points $Pg_start_located$ and $Pg_end_located$	55
Fig. 4.11: Calculation of the row vector $differenceY = abs(yg) - abs(ye)$	56
Fig. 4.12: Geometrical representation of $data_iter2$ for the next least squares iteration.....	57
Fig. 4.13: Plots of the elliptical objects for the first and second least squares iterations	58

Fig. 4.14: Iteration $i = 1:882, n = 5$	61
Fig. 4.15: Iteration $i = 1:1733, n = 1$	62
Fig. 4.16: Operation mode 2 – range constant, $i = 1, n = 50$	65
Fig. 4.17: Operation mode 3 – range variable-, $i = 1:992$ (992 iterations);, $n = 1$	66
Fig. 5.1: Solution 1 for square and rectangular objects – inverted b.c	69
Fig. 5.2: Solution 2 for square and rectangular objects – symmetric b.c	69
Fig. 5.3: Solution 3 for square and rectangular objects – parallel grip	70
Fig. 5.4: Solution concept - rectangular objects algorithm	70
Fig. 5.5: Generation of the lines L1 and L3 by using a larger range (red)	73
Fig. 5.6: Fitting the four corners of the rectangle inside the gripper bend lines	74
Fig. 5.7: Generation of the lines L1 and L3 by using the enhanced segments	75
Fig. 5.8: Simulation with 2 different enhanced gripper fingers - Load condition.....	76
Fig. 5.9: Scheme of solution for triangular objects - Grip condition	76
Fig. 5.10: Rollover due to forces of the gripper bend line on the triangular object.....	77
Fig. 5.11: Free body diagram – Generic triangular object	77
Fig. 5.12 Equivalent systems – Finding the forces F_{k1} and F_{k1}'	78

List of Tables and Diagrams

Diagram 3.1: Process flow diagram, final goal overview	42
Diagram 3.2: Process flow diagram, algorithm <i>Elliptical Objects Fitting</i> – modules	43
Diagram 3.3: Process flow diagram, algorithm <i>Elliptical Objects Fitting</i> - input and outputs	44
Diagram 3.4: Process flow diagram – ideas for an algorithm for square-rectangular objects	45
Diagram 6.1: Process flow diagram, future work highlighted in red	82

Chapter 1

State of the Art

1.1 Framework

Human labors have always been in relationship with the acquisition of specific skills, talents, methods and experience. For the obtention and mastering of these new abilities, a huge amount of information is captured by the senses, processed by the brain and executed by the hand. As an analogy, as hands are the organs for manipulation, for robots, prehension tools or end-effectors are their equivalent. Furthermore, these prehension tools are normally known as “grippers”, which are the only parts of the robot having direct contact with the target to grasp. The main purpose of developing new gripper technologies for industrial robots consists on their inclusion in tasks normally done by human hands, generating more effective handling.

1.2 Gripper's definition

An end-effector is a device connected at the end of a robotic arm, which interacts with the environment to grasp objects like human hands for human manipulation. A specific definition for grippers is the following: *“Subsystems of handling mechanism, which provide temporary contact with the object to be grasped. They ensure the position and orientation when carrying and mating the object to the handling equipment. Prehension is achieved by force producing and form matching elements. The term “gripper” is also used in cases where no actual grasping, but rather holding of the object as e.g. in vacuum suction where the retention force can act on a point, line or surface”* [1].

The structure of an end-effector and the nature of programming and hardware that drives it, relies on its performing functions [2]. Grippers are the most functional end effectors which can be impactive (gripping force is achieved by the impact against the surface of the object from at least two directions), ingressive (penetration of the workpiece by the prehension tool), astrictive (direct contact is not needed at the beginning of the prehension, for example vacuum suction, electroadhesion, magnetism) or contigutive (direct contact without impactive methods, for example, chemical adhesion, thermal adhesion, surface tension) [3]. At the same time, they are based on different physical principles such as: friction,

vaccum, pneumatic, hydraulic electromagnetism, et al. They are utilized in applications such as automated assembly lines, observing radiation zones, minimal invasive surgery, and space exploration. In order to fulfill their task in different working environments, different materials are being used for their design as for example: piezoelectric crystals, shape memory alloys, magnetorheological fluids, carbon fiber and others [2]. Furthermore, the nature has been an inspiration for the development of new gripping mechanisms. Some of them are introduced and described in the following sections.

1.3 Classification of end-effectors

Robot end-effectors can be classified by their gripping mode (internal or external), working principle, adaptation, number of fingers, et.al. Nonetheless, because the present work aims to develop a two finger self-adaptive-passive gripper, a classification based on adaptability is presented. In order to accomplish a complete understanding of the gripper mentioned before. Different technologies of non-adaptive, adaptive and self-adaptive grippers will be classified and described, emphasizing a detailed explanation on two fingers- self-adaptive passive grippers.

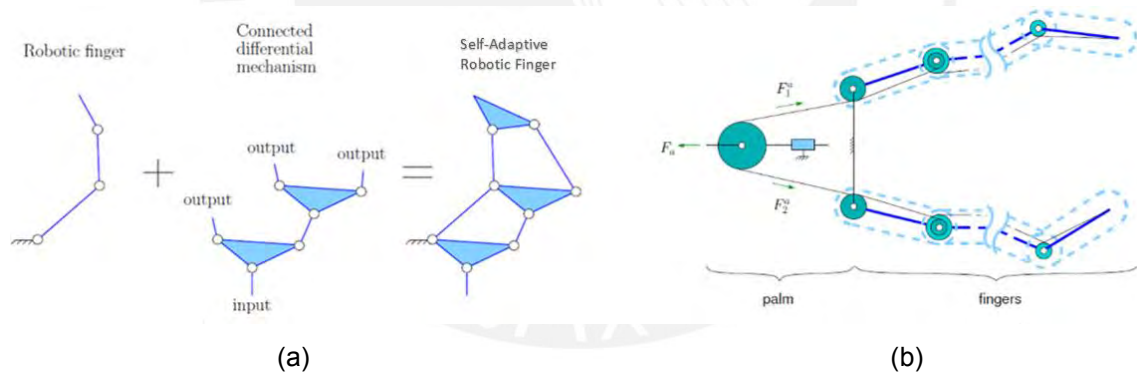


Fig. 1.1: (a) Connected differential mechanism [4] ; (b) Underactuated fingers [5].

Unlike adaptive and non-adaptive grippers, a self-adaptive gripper is able to adjust to the object's silhouette relying only on the mechanical arrangement or material flexibility when an actuating force is applied [6]. The objective of a self-adaptive mechanism is to place part of the control functions from the electronics to the mechanical layout of the system itself [4]. To set this goal, differential mechanisms and passive elements are embedded in a self-adaptive-passive gripper as exemplified in figure 1.1. The formers are two-degrees of freedom mechanisms that resolve a single input into two outputs and viceversa. The latters are utilised to kinematically constrain the system [5]. Since some degrees of

freedom (DOF) of these hands are not controlled, they are often referred to as “underactuated”. An under-actuated serial robot is defined as a manipulator with one or more unactuated joints. On the other hand, “under-actuated” or self-adaptive fingers use passive elements (normally springs) in the design of their unactuated joints, therefore it is recommendable to catalogue these joints as uncontrollable or passively driven[4]. In addition, self-adaptation do not absolutely require the insertion of specific passive elements, over and above, using inertial properties to fulfill the grabbing task is also a suitable option [4]. The gripper must have three mechanical features in order to achieve a passive adaptive grasp. First, the fingers must be able to curl as they close around the target. Second, they should close around the grasping object independently of each other during grasping. Finally, the fingers must allow inwards and outwards rotation, as well as closing and opening around the grasping object [7]. It is relevant to highlight that the huge advantage owned by passive grippers is that they only require power for grabbing and releasing the object, thus there is no energy consumption due to the grabbing function while transporting the piece; therefore, energy is saved.

1.3.1 Non-adaptive grippers

Non-Adaptive Grippers are designed to grasp a target such that its position is fully determined and does not change during the process. Unfortunately, a final product comprises different parts, therefore different shape-inadaptive-grippers are mandatory to succeed its completion. This type of gripper does not suit the form of the object when an actuating force is applied to the mechanism. For non-adaptive grippers there are different working principles such as: friction, pneumatic, hydraulic, electromagnetism, vacuum, inter alia. The gripping force is prefixed for the specific shape of the object and grasping task. The force can not be regulated during the grasping process, therefore if softer materials are handled, they can easily get broken or deteriorated. Neither underactuation nor passive elements are included in the design of these type of grippers. A huge disadvantage is the rigidity of the end effector, which leads to a reduced autonomy manoeuvring objects with different shapes and sizes. Therefore, it is mandatory to change the tool for every single target, consuming time and resources during grasping operations.

1.3.1.1 Mechanical-rigid-friction grippers

A mechanical rigid gripper is an end-effector that comprises of mechanical fingers which are driven by a mechanism to grip an object. The target is manipulated by the fingers by one of the following motions: pivoting or swinging, linear or translational. These gripping devices wraps around an object and rely on friction to secure its holding without smoothness and regulation of the gripping force while the process take place. In this specific type of rigid-mechanical grippers, neither under-actuation nor passive elements are incorporated, as a consequence, self-conforming is not possible. Friction between the gripper and the object will depend on two variables. The former is the type of surface: smooth or rough. The latter is the force which is pressing the surfaces together [8]. Fingers are either attached to the mechanism or are already integrated on it. Furthermore, in order to decrease the pressure applied to the object, deformable complaint rubber materials are used at the finger's surface which is in direct contact with the gripping target, increasing the friction coefficient [9].

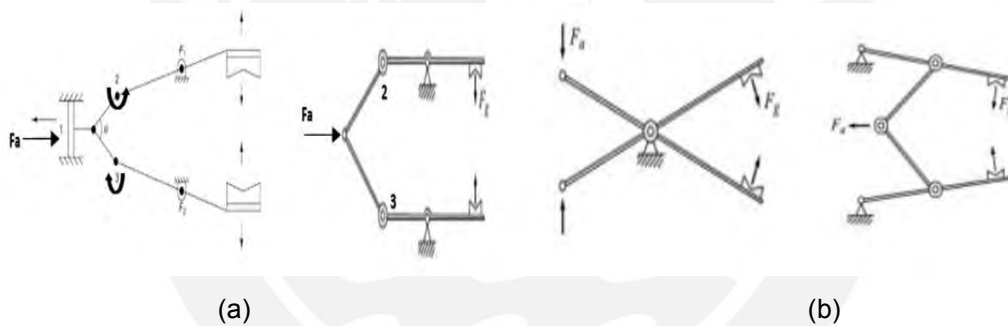


Fig. 1.2: (a) Gripper using slider-crank mechanism; (b) Linkage actuation mechanisms. [10]

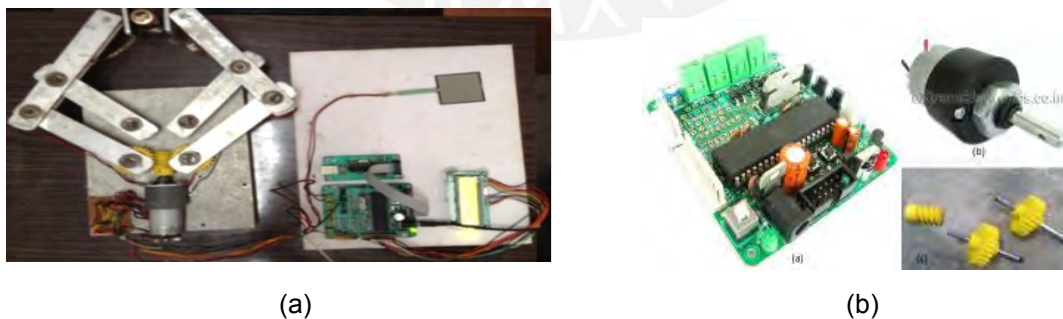


Fig. 1.3: (a) 4-Bar linkage gripper; (b) 16 Xboard–DC motor-worm. [9]

An example of linkage actuation is shown in figure 1.3. The gripper comprises four rectangular aluminium flats arranged according to a four bar linkage gripper configuration. The fingers are flat and rectangular. Holes are made in the bar for

fitting the bearings. The gripper is guided by using a worm and spur gear setting to convert the rotational motion to linear motion. To control the pressure, tactile force sensors are included. Finally, the driving force is given by a permanent magnet dc motor of 10 RPM, 2A -12V rating [9].

There are more types of mechanical grippers, which can be considered in this classification such as cam-actuated grippers, screw type grippers, translational and rotatory grippers. It is important to highlight that the use of more than two fingers will increase the degrees of freedom and greatly aid the adaptability of grippers. The key advantage for employing a three-finger grip consists on holding the object in three spots, making possible a tighter grip and the holding of spherical objects of different size maintaining the centre of the object at a specified position. Even though the manoeuvrability is increased, in most applications, two fingers are sufficient to hold the work piece; therefore, grippers with three or more fingers are not so frequently used. Maintenance for the bearings, energy loss because of friction in the joints, plus the lack of adaptability are the main trade-offs of these grippers which can reach high grasping forces.

1.3.1.2 Magnetic grippers

Magnet grippers can be configured by permanent magnets or electromagnets (figure 1.4). The formers do not need an external supply for grasping, once an object is grasped, there is an additional device called “stripper push” which separates it from the gripper. The latter include a control unit and a DC power, which can grasp magnetic objects. When placed in contact with an electromagnet, the object provides a flow path for the magnetic flux that completes the magnetic circuit. The object is held by the electromagnet due to the electromagnetic force generated by the circuit [11]. A disadvantage is that magnetic grippers only work with ferrous materials (Iron, Nickel and Cobalt). Furthermore, during the handling, the target tends to slip against the electromagnet surface if the tangential holding force in that direction exceeds the limiting force of static friction for the magnet-part contact. This component of the holding force relies upon the holding force normal to the electromagnet surface via the coefficient of static friction for the magnet-part material pair [12]. It is also important to emphasize that high temperatures demagnetize permanent magnets; therefore, an extent contact with a hot work piece will weaken them, reducing their cycle life. Most magnetic materials are relatively unaffected up to 100 degrees Celsius [13].

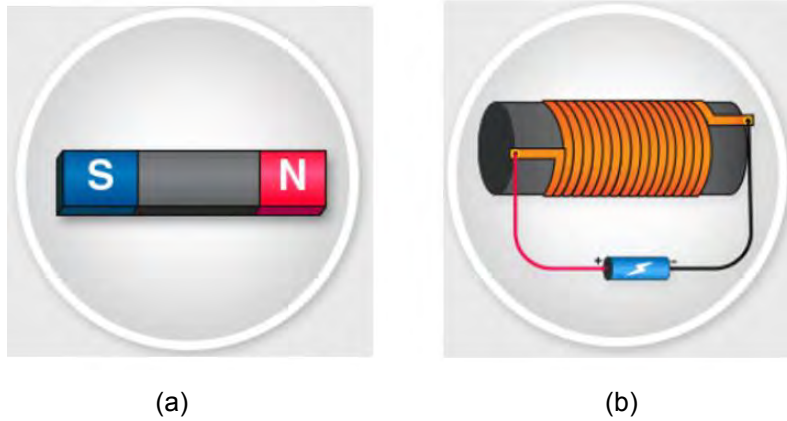


Fig. 1.4: (a) Permanent magnet; (b) Electromagnet [14]

1.3.1.3 Vacuum grippers

When the objects are too thin to be handled, vacuum grippers can hold them. Vacuum-grippers are suction cups, which are made of rubber and are connected through tubes with under pressure devices for lifting up items. In order to release them, air is pumped from the suction cups. The vacuum pressure can be generated by vacuum pumps, ejectors, suction bellows or pneumatic cylinders [15]. Vacuum grippers are well recognized for their capacity to manage objects of different silhouette, weight, and robustness. The capabilities before mentioned may put these grippers as a suitable option for variable industrial tasks such as food handling and others. Even though there is an increased autonomy regarding target's shapes, and self-locking in case of power failure, limitations on the gripping task appear when grasping curved objects. To ensure the suction force created by the negative pressure inside the cups, it is mandatory that the object's surface matches perfectly to that of the suction cups, otherwise perfect vacuum will not be achieved, thus a failure by grasping will take place. In addition, it is also necessary that the surface of the targets are clean and smooth and that their materials are not porous.

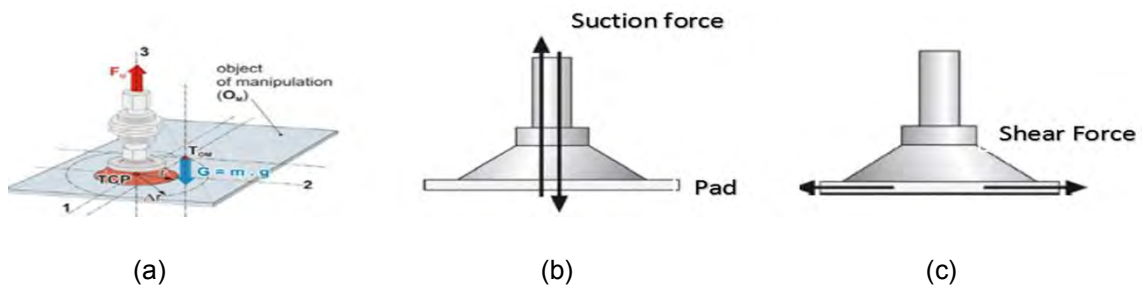


Fig. 1.5: (a) Suction pads; (b) Suction force; (c) Shear force in cup. [15]

1.3.1.4 Pneumatic expandable bladder grippers

A bladder hand is a specialized robotic end effector that can be used to pick up and move rod-shaped or cylindrical objects. The key component of the gripper is an inflatable donut-shaped or cylindrical sleeve that resembles the cuff commonly used in blood pressure measuring devices. Their flexible inflatable rubber shape enables them to grasp various fragile or sensitive items of different dimensions and hold them softly and securely. There are many applications and some typical ones include handling bottles, cylinders, photoconductor drums, or as exhaust and pump leak testing probes used in automated processes in factories or on assembly lines. Additionally, it is possible to classify them as “air pickers” and “air grippers” according to their grasping form. The formers insert themselves into the neck of a bottle, cylinder or similar object, pneumatically inflating their rubber sleeves to create an inner seal, allowing it to be gripped and moved. The latter locate themselves around the outside of the neck of an item, inflating to create a seal that grips around like a clamp before it is moved mechanically [16]. Their main disadvantages are: the required inflation time and autonomy (restricted to cylindrical objects).



Fig. 1.6: (a) Air picker inflating; (b) Air gripper inflating. [16]

1.3.2 Adaptive grippers

A system that can adapt itself to grip a wider variety of objects save time and cost if implemented correctly. The first task for adaptive gripping is the object detection. In order to track the piece, optical, acoustical, laser or tactile sensors are used [6]. In addition, it must carry the object firmly enough to avoid slip out of the grip, while preventing the destruction of the target due to an excessive gripping force. The gripper attempts to close the fingers around the object, and hold on to it in such a way that mimics the softness of a human hand. It is important to highlight that one of the main characteristic within the gripping process is the feedback given from to sensors to the gripper system. As a

conclusion, its adaptability relies on the retrieved information collected by the sensors each instant of time and do not depend on the mechanical system itself. In addition, there is an algorithm running throughout the process to complete the task and it has no passive joints, therefore no self-adaptation per se.

1.3.2.1 Adaptive learning gripper

The learning gripper from Festo looks like an abstract form of the human hand. Twelve pneumatic bellows actuators with low-level pressurisation drive the four fingers of the gripper.

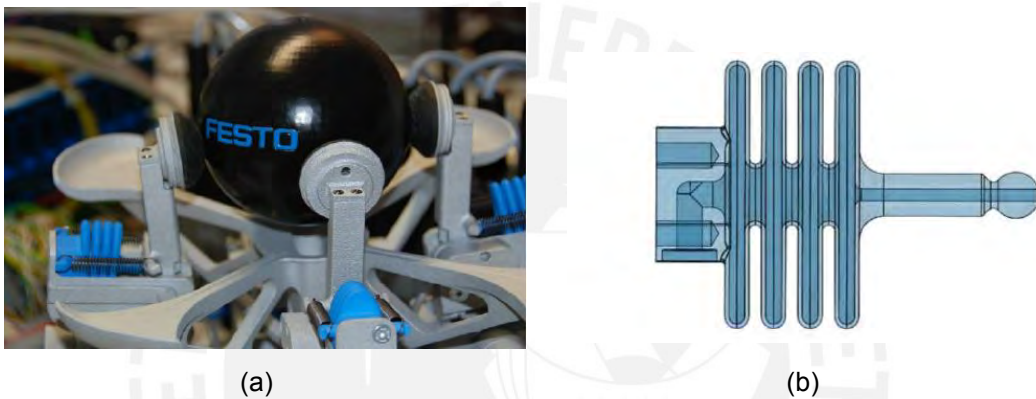
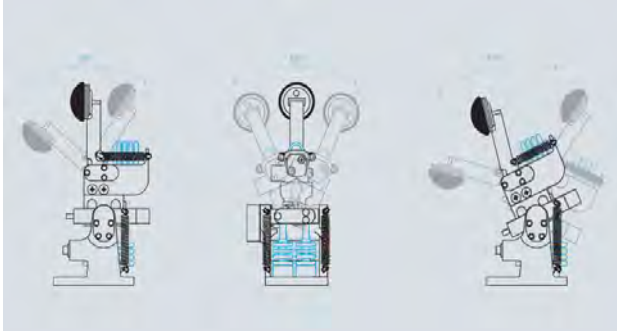


Fig. 1.7: (a) Festo learning-gripper [17] ; (b) Bellow of 2 degrees of freedom [18].

Due to the process of machine learning, it is capable to teach itself to fulfill complex actions such as gripping and positioning an object by the reinforcement learning method. The gripper maximize its capabilities entirely because of the received feedback on account of its previous actions. Specific actions are not provided to the system unlike in supervised learning. The learning system alternates its actions with the main aim of maximising feedback over time. As a result, the probability that a successful action is executed is increased and that an unsuccessful action is not repeated. In time, the learning algorithms develop a motion strategy based on this feedback. The actions required for each given status are learned by the gripper, knowing how to modify its motion so that it retrieves as much positive feedback as possible in order to carry out its task successfully [19].



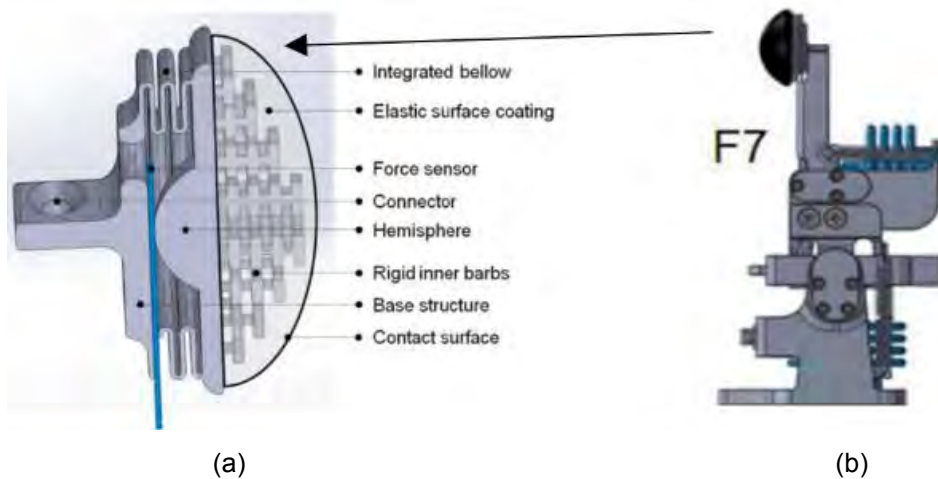
(a)



(b)

Fig. 1.8: (a) Degrees of freedom [19] ; (b) Final hardware [18].

The robotic hand is capable of lifting and rotating a geometric object, e.g. a sphere or a cube, into a desired final position. By hierarchical reinforcement learning, the required steps to accomplish the desired task are obtained. Furthermore, hierarchical reinforcement learning is divided into low-level and high-level learning. The former learns a set of basic manipulation strategies, which defines a high-level action. The latter learns to achieve a target orientation using the pre-learned high-level actions [17].



(a)

(b)

Fig. 1.9: (a) Side view of the fingertip of finger F7; (b) finger F7. [18]

Each of the four fingers have three pneumatic actuators. In contrast to classical pneumatic cylinders, the blue pneumatic actuators shown in figure 1.8 and 1.9 (b) can provide soft forces. The gripper comprises four crosswise-mounted pneumatic fingers and has twelve degrees of freedom leading to a high-dimensional state and action space. Possible applications of the gripper are the bin-picking tasks where the gripper grabs a detected object with arbitrary

orientation from a bin, catch it and rotates into the desired orientation. Finally, the gripper puts the object in the desired location on a conveyor belt.

1.3.3 Self-adaptive grippers

The main aspect of a gripper in order to be self-adaptive is its capability to adapt to the object shape without complex mechatronic actuator and sensor system. As long as a triggering force due to any energy source (compressed air, vacuum, electro-magnetic field, et al.) is applied, the fingers will automatically grasp softly the object's contour. In the same manner, for releasing the object, the actuating force must disappear. In contrast to passive grippers, energy consumption due to the triggering force takes place while the target is being transported.

1.3.3.1 Electro-adhesive self-adaptive gripper finger

An electro-adhesion system normally comprises four essential components: an electro-adhesive pad, a high voltage power source, a substrate and a control system as illustrated in figure 1.10 [20]. The electro-adhesive pad contains conductive electrodes embedded in dielectric materials, which are connected with high voltage power sources, usually in the range of kV. Electrostatic adhesion uses a set of conductive electrodes deposited inside a dielectric in an ordered way so that when a high voltage potential is triggered across the electrodes an electric field is created, which generate an adhesive force on both, conductive and non-conductive substrates. On conductive surfaces, electrons can migrate freely toward the positive electrodes and form electron holes under the negative electrodes turning them into a set of capacitors in which the "plates" are the electrodes and substrate. On non-conductive surfaces, the electric field polarizes the substrate's molecules.

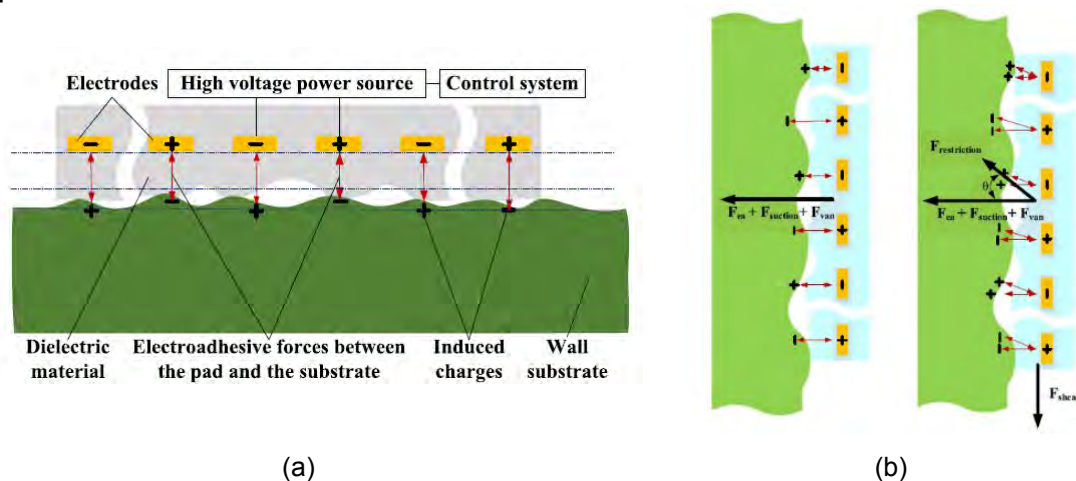


Fig. 1.10: (a) Sectional view electro-adhesion system; (b) electro-adhesive forces[20].

The adhesion force is proportional to the square of the electric field strength, which at the same time depends on the dielectric constant, voltage potential and electrode geometry [21]. Even though there are 33 variables affecting the electro-adhesive forces between the electro-adhesive pad and the substrate, pad geometry and voltage source type are the most important factors which determine the gripping capability for a substrate material and its release velocity [20]. Electro-adhesion can enable the pad to adhere to different substrate materials such as smooth aluminium plates and rough concrete surfaces, however the gripping force varies significantly on different substrate and dielectric materials, thus sufficient voltage should be applied to enable the pad to grasp different materials [20].

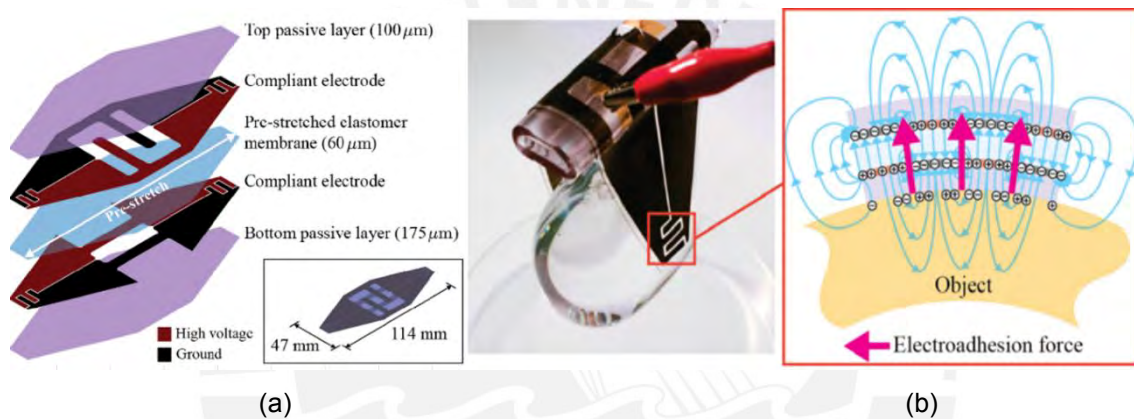


Fig. 1.11: (a) Pre-stretched elastomer membrane ; (b) Electro-adhesion forces [22].

A gripper developed by Shintake et al. is displayed in figure 1.11 where electro-adhesion and electrostatic actuation are combined in a single stretchable element. In this gripper, the electrodes of the dielectric elastomer actuators (DEA) are arranged with an interdigitated shape, obtaining finger actuation by thickness reduction and area expansion of the elastomer while forming out-of-plane electric fields; such fields penetrate into the object surface and lead to electro-adhesion forces by inducing surface charges on the object [22].

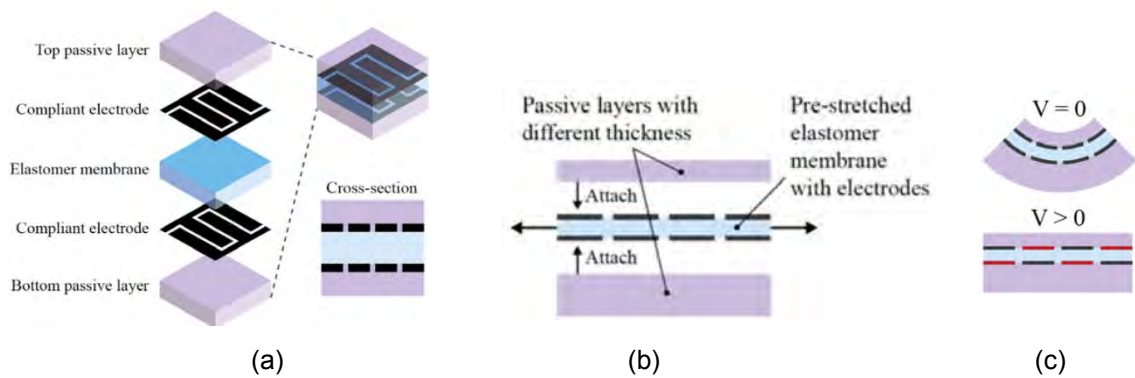


Fig. 1.12: (a) Five-layer gripper; (b) Arrangement; (c) Finger open-close position. [22]

When the voltage is turned on, the electrodes bend towards the object to be gripped, imitating muscle function, then the tips, which have a special electrode's arrangement, act like fingertips that softly conform to the shape of the object, gripping onto it with electrostatic forces. These electrodes can carry 54.7 times its own weight (1.5 grams) and no prior knowledge about the object's shape is necessary [23]. The electrode flaps consist of five layers as displayed in figure 1.12 (a), a pre-stretched elastomer layer sandwiched between two layers of electrodes, plus two outer layers of silicone of different thickness.



Fig. 1.13: (a) Voltage off position; (b) Voltage on position. [22]

When the voltage is off, the difference in thickness of the outer layers makes the flaps curl outwards. When the voltage is on, the attraction between the two layers of electrodes straightens out the membranes. This straightening of the membranes from a curled position mimics muscle flexion. At the tips of the flaps, the electrodes of each layer are designed for optimal electrostatic grip. These interdigitated electrodes, which look like two combs fitted together, create an electrostatic field that causes electro-adhesion

One important fact to take into consideration is the adhesion failure that take place when changing from grasping insulating materials to conductive materials as dielectric breakdown occurs when a high voltage is applied to conductive

substrates. For this reason, it is desirable to have an optimized and intelligent electro-adhesive system that come out with high forces when gripping different materials using lower voltages. Furthermore, unstable and unpredictable electro-adhesive forces appear while changing environmental factors such as temperature, humidity, air pressure, dust [23], therefore designing and implementing an electro-adhesive system that can output repeatable and reliable electro-adhesive forces in changing ambient environments is mandatory. Finally, a potential drawback is the residual adhesion force after removing the voltage. However, they are low for dielectric objects and last a few seconds [22].

1.3.3.2 Finray self-adaptive gripper finger

Festo's Multi-choice-gripper, which works under the finray principle will be introduced. The Finray Effect relies on the specific mechanical behaviour of the fish tail fin. When the fish swings its rear end and the tail fin, side pressure is generated to one flank of the fin causing the fin to curve inward, toward the side of the applied force as illustrated in figure 1.14 (b).

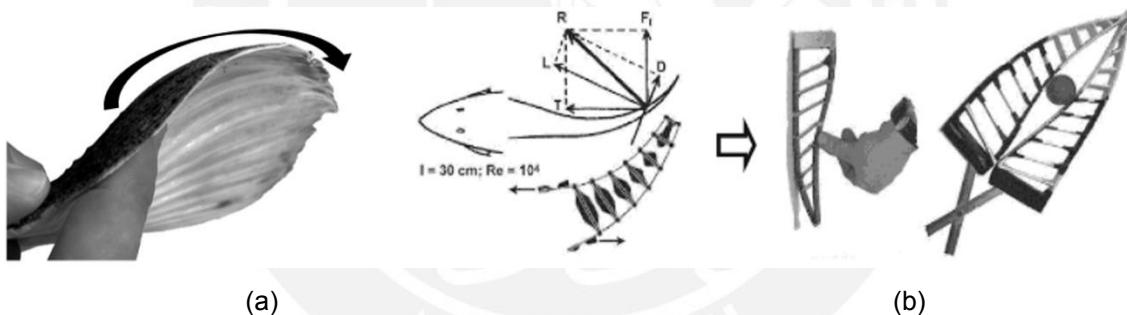


Fig. 1.14: (a) Finray effect; (b) Forces on a fish tail and a two-finger-finray gripper. [24]

The construction-principle of fish fins that counter-bend automatically, leads to self-fitting gripper fingers. The fine bone-braces between the covering surfaces of the tail fin are diagonally oriented. If the surfaces are shifted against each other during a fin beat, these oblique configurations cause the trailing edge's bending in direction of the beat. The water pressure normally overcomes this curvature-tendency, so that a functional bending results according to figure 1.14 (a). The described mechanism generates a contact-surface, constructed in a way that it snugles up automatically to a touched object.

Coming back to the multi-choice-gripper, due to the adaptive fingers with a Finray structure, this gripper is variable in terms of grasping direction as well as in adaptability to a wide variety of object shapes. Moreover, the grabbing process is very sensitive without additional sensor or control technology. In addition, its fingers can be switched over so that they can either grip in parallel or centric direction without requiring any conversion [25].

A combination of force fitting and form fitting is employed. The former involves grasping and holding objects using forces acting on a particular point or area. The latter is related to adaptation and less force exertion. The Multi-choice-gripper makes use of both force and form fitting type of closure by the use of fingers, which are adaptive. Two rotatable finger slots on the base body of the gripper enables it to change from parallel into centric grip mode as displayed in figure 1.15.

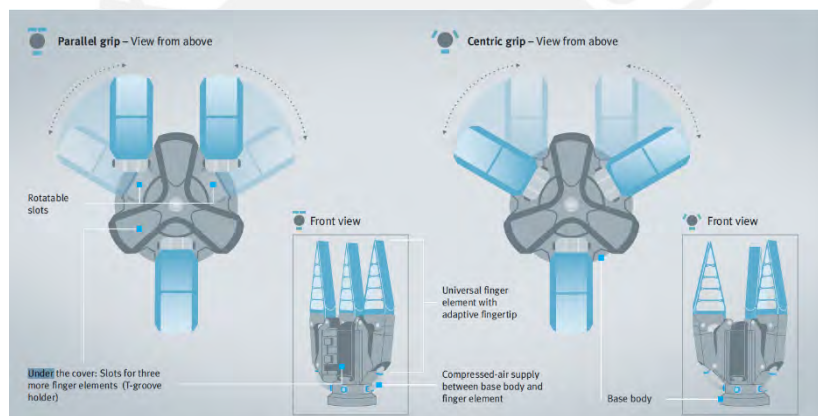


Fig. 1.15: Mechanical changeover parallel (left) and centric grip direction (right). [25]

A pull-push bar transfers the force to the holders located on the two finger elements, which in turn can be rotated. These holders change the finger position, either all the fingers are directed towards a central point (centric grip), or alternatively two of the fingers are arranged next to each other while the third finger performs the function of the opposable thumb to enable a parallel grip.

1.3.3.3 Fluidic elastomer self-adaptive gripper fingers (FEA)

FEAs can generate high forces, which are proportional to the pressure of the fluid and to the surface area where the active pressure is applied. Fluidic base systems have large strokes, only limited by the maximum stress that the material can withstand. Bending FEAs can reach bending angles around 300° in a reaction time of 0.05 to 1.0 seconds [23]. The reaction time depends on the pressure-flow

rate due to the pump or compressor, on the internal volume of the chamber and on the stiffness of the elastomer. Coordinated fluid pressurization of the actuator's chambers can produce multi-degree-of-freedom bending and can be used as fingers to create soft robotic grippers [26]. Furthermore, different architectures are available for FEAs; the most common ones include elongated elastomeric chambers with the addition of reinforcing fibres and layers, bellows-like structures, and tube-like tentacles. The moulding process, dealing with the elastomer in its liquid state, offers feasibility to add functional elements in the actuator such as fibres, inextensible layers, porous materials, variable stiffness elements, adhesion and strain sensors [23].

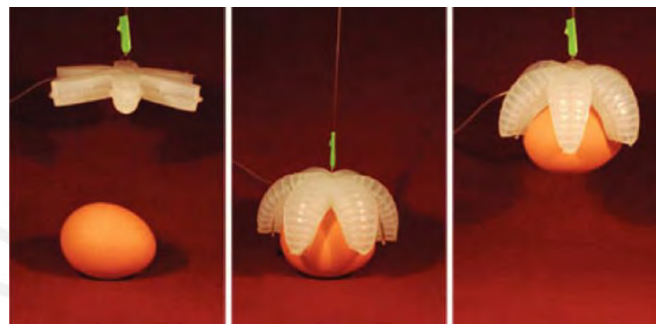


Fig. 1.16: Three layers pneunets bending FEAs. [23]

As a resume, elastomeric materials (normally rubber) with a number of built-in fluidic channels, the inclusion of composites (elements with different stiffness) and the pressurization rate management, bending or extension for soft gripping actuation is obtained. These grippers are able to conform to any object relying only on the stiffness of the materials embedded in them. One of the main advantages is that it is not necessary for the gripper to know the exact position of the object, therefore a lot of numerical computation is avoided. On the other hand, energy consumption during transportation takes place. They can be implemented efficiently in packaging where extreme high and low temperatures are present, such as warehouses or freezers in food industry.

1.3.4 Self-Adaptive Passive Grippers

As described, the objective of a self-adaptive mechanism is to delegate part of the control tasks from the electronic board to the mechanical layout of the system itself, avoiding an algorithmic procedure. Hence, for self-adaptive robotic hands, some degrees of freedom have to be uncontrolled electronically. However, these motions must be carefully predicted and studied in such a manner that the achievement of the desired closing sequence is accomplished; otherwise, they

can lead to erroneous behaviours. Since some degrees of freedom (DOF) of these hands are not controlled, they are often refer to as “under-actuated”. An underactuated serial robot is defined as a manipulator with one or more unactuated joints. On the other hand, “under-actuated” or self-adaptive fingers use passive elements (springs commonly) in the design of their unactuated joints. It is therefore recommendable to catalogue these joints as uncontrollable or passively driven instead of unactuated. In contrast to usual under-actuated manipulators, in self-adaptive robotic systems the actuation torque or force is distributed to each joint of the system. Passive grippers use energy only for grasping and releasing, but not for the transportation of the target.

1.3.4.1 Flex Shape Gripper

The chameleon is able to catch a variety of different insects by putting its tongue over the respective prey and securely enclosing it. Once the chameleon has its prey in its sights, it lets its tongue shoot out like a rubber band. Just before the tip of the tongue reaches the insect, it retracts in the middle, whilst the edges continue to move forwards. This allows the tongue to adapt to the shape and size of the respective prey and firmly enclose it. The prey sticks to the tongue and is pulled in like a fishing line. Flex Shape Gripper of Festo uses the principle presented above to grip the widest range of objects in a form-fitting manner, this means that the gripper can easily adapt to the form of the object by the use of its elastic silicone cap, picking up several objects in a single gripping process and put them down together.



Fig. 1.17: Gripper fitting capability, examples of gripper’s form adaptability. [27]

The gripper comprises of a double-acting cylinder. The former chamber is filled with compressed air whilst the latter one is constantly filled with water. This second chamber is fitted with elastic silicone moulding, which equates to the chameleon's tongue. The volume of both chambers is designed so that the deformation of the silicone part is counterbalanced. The piston, which closely separates the two chambers from each other, is fixed with a thin rod on the inside of the silicone cap.



Fig. 1.18: (a) Chamber 1 - compressed air, Chamber 2- water; (b) Holding mechanism. [27]

During the gripping process, a handling system guides the gripper across the object so that it reaches the target with its silicone cap. The top pressurised chamber is then vented, therefore the piston moves upwards by means of a spring support and the water-filled silicone part pulls itself inwards. At the same time, the gripper is guided by the handling system deeper across the object. In this way, the silicone cap wraps itself around the object to be gripped, which can have any silhouette, coming up with a tight form fitment. The elastic silicone allows a precise adjusting to a wide range of different geometries thanks to its high static friction, which builds a strong holding force. Energy is required only for holding and releasing steps, but not for transportation. The flexible quality of the compressible compressed air simplifies the coordination between the handling system and gripper during the grip stage. The force and the deformation of the silicone part can be set very precisely with the aid of a proportional valve, allowing several objects to be gripped at once in a single procedure [27].

Adaptable grippers can assume a significant role being used in any facility where multiple objects with a range of different shapes are handled at the same time, for example: in the service robotics sector, assembling tasks or when handling small parts. Moreover, they can also be utilised in flexible production plants, handling different components in one procedure without the necessity of changing the gripper. In addition, a suitable application could be sorting fruits and

vegetables or other objects with irregular shapes. Once the gripper is operating, it is capable to perform various tasks. This functional integration is an excellent instance of how systems and components themselves can be adapted to wide production scenarios.

1.3.4.2 Two-finger self-adaptive passive robot gripper

The two-gripper finger of Robotiq is designed for industrial applications to quickly pick, place and handle a large range of objects of varying sizes and shapes. It comprises of two articulated fingers, each having two phalanxes as displayed in figure 1.19. Five points of contact can be obtained between an object and the gripper, two on each of the phalanxes plus one on the palm. The fingers are under-actuated, which allows the adaptation of the fingers to the object's silhouette, simplifying the control of the gripper.

There are two grabbing possibilities: encompassing or parallel grip, both shown in figure 1.19 (b) and (c) respectively. The closing or opening is done via the "go to requested position" command, which is an input to the gripper. It is important to highlight that whether the fingers close to produce an encompassing or fingertip is decided at the gripper level automatically, depending on the object's geometry and its relative position with respect to the gripper.

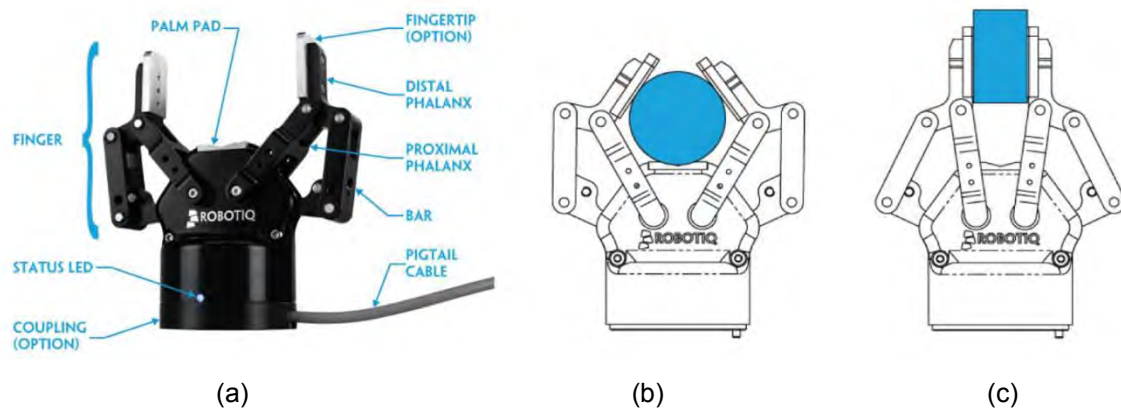


Fig. 1.19: (a) Robotiq 2-finger adaptive gripper; (b) Encompassing grip; (c) Parallel grip. [28]

Furthermore, internal gripping is also possible by applying pressure with the outer side of the fingers when handling hollow parts.

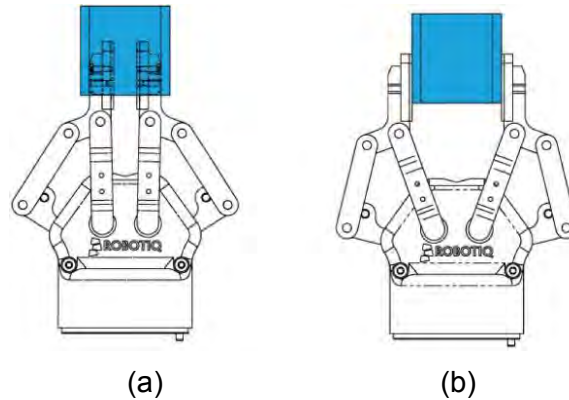


Fig. 1.20: (a) Internal gripping; (b) External gripping. [28]

On the other hand, the gripper equilibrium line is the gripping region that separates the encompassing grip from the parallel grip. When gripping an object close enough to palm of the gripper, encompassing grip will take place and the fingers will close around the object, unless its size or shape is not suitable. If gripped above the equilibrium line, the same object is picked up in a parallel grip by the fingertips and the fingers with a parallel motion performs the closing action. Figure 1.21 illustrates the encompassing grip region, equilibrium line, and the parallel grip region on the gripper.

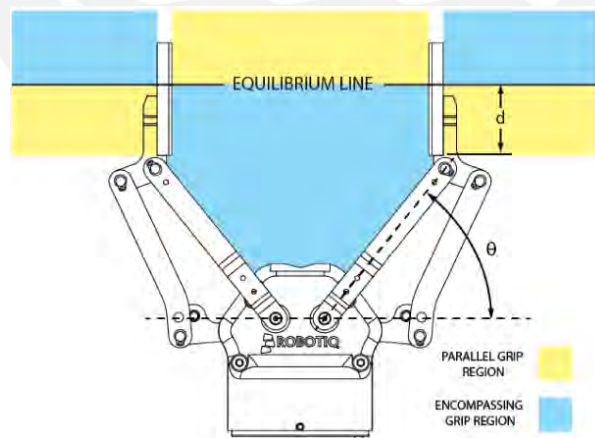


Fig. 1.21: Equilibrium line on the two-finger, shown with no fingertip pads. [28]

1.3.4.3 Three-finger self-adaptive passive gripper

The Three-gripper finger of Robotiq has three articulated fingers arranged as follows: finger A in front of finger B and finger C. Each finger has three phalanges as illustrated in figure 1.22 (a), therefore the gripper is able to engage up to ten points of contact with an object, three points on each of the phalanges plus one point on the palm. The fingers are under-actuated, allowing them automatically adaptation to the target and simplifying the control of the gripper.

The gripper can perform two different types of movement. The first determines the operation mode, changing the orientation of fingers B and C as shown in figure 1.23 (b). The operation mode is determined firstly by the use of the grip as a function of the size or shape of the target being gripped and by the task that has to be done. The modes of operation are: basic, wide, pinch and scissor mode.



Fig. 1.22: (a) Position 1; (b) Position 2 [29]

The basic mode is the most versatile, thus best suited for objects that have one dimension longer than the other two. On the one hand, the wide mode is optimal for gripping round or large objects. On the other hand, the pinch mode is used for small objects that have to be picked precisely, being capable of only gripping objects between the distal phalanges of the fingers. Finally, the scissor mode is used primarily for tiny objects. Even though this mode is less powerful than the other three, it is high precise. In addition, it is not possible to surround an object in scissor mode, fingers B and C move laterally towards each other while finger A remains still.

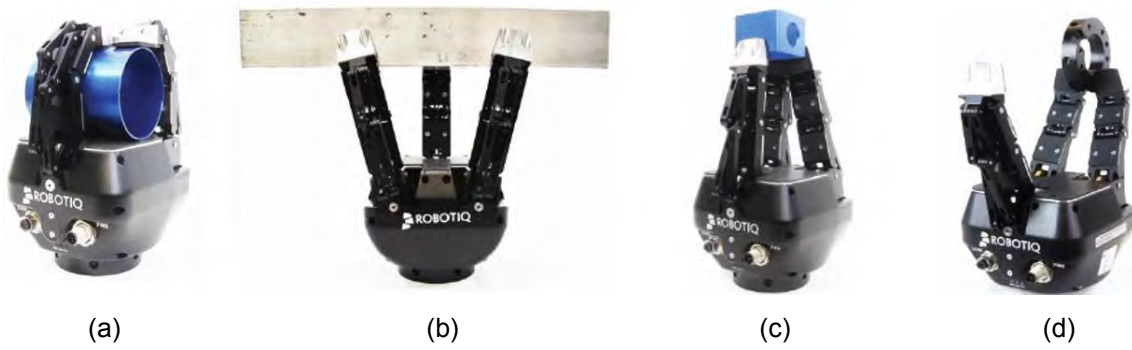


Fig. 1.23: (a) Basic mode; (b) Wide mode; (c) Pinch mode; (d) Scissor mode [29]

The second movement determines the closing and opening of the fingers as shown in figure 1.24 (a). This action is performed with a single input from the user. Each finger is not controlled independently; the gripper itself closes each finger until it reaches a stable configuration on an object or against the gripper's palm. It is precise to stand out that a user can specify the relative speed at which the fingers are closed and the force that is applied to an object.

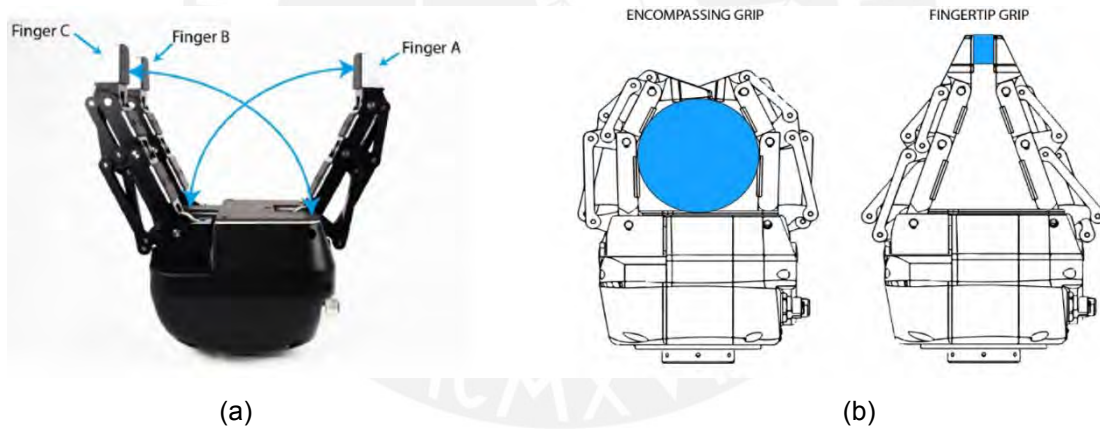


Fig. 1.24: (a) Closing and opening; (b) Encompassing and fingertip grips [29]

1.3.4.4 Adhesive self-adaptive passive gripper finger

When adhesive grippers grip an object, adhesive shear forces are built along the surface of it in the directions of the local tangents using controllable fibrillar adhesives. The resultant of these forces grasps the object as shown in figure 1.25 (b). Furthermore, because no active squeezing is required, the gripper can passively and dynamically grasp objects, i.e. it is able to catch thrown items without active control [30].

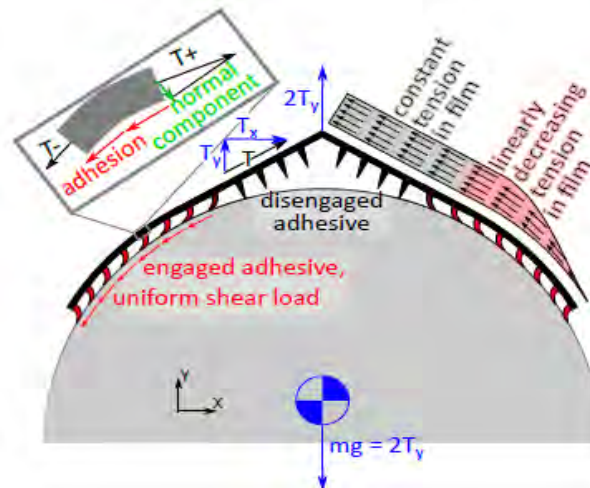


Fig. 1.25: Acting forces on the adhesive gripper [31]

As illustrated in figure 1.25 (b), approaching the micro-wedge adhesive within a few microns of a surface enables the tips of the wedges to engage with the surface. An important characteristic of these type of grippers is the controllability, which means that, as shear loads increase as shown in figure 1.26 (a), so does the contact area, thus the adhesive capacity. The adhesive augment the contact area on the surface, causing additional adhesion. When there is no shear stress, the contact area decreases. As a conclusion, in order to grasp an object, the pair of opposed adhesives need only be laid on the surface. At this moment, just the end part of the tip of each wedge is captured within the surface, thus there is no need to press them into the surface. Finally, as displayed in figure 1.26 (b), the removal of the shear load permits the wedges to go back to their initial position, assuring smooth release from the surface.

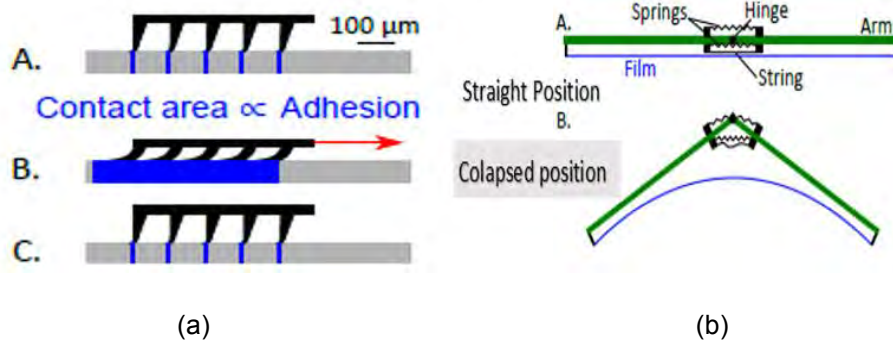


Fig. 1.26: (a) Rest mode, motion, rest mode; (b) Bi-stable structure [31]

The main forces acting on the gripper are shown in figure 1.25. The Y-component of the tension in the film is equal and opposite to one-half of the weight of the object, while the x-component is cancelled internally. The tension remains constant in the length of the film where the adhesives are disengaged but linearly decreases to zero in the region with adhesion. The adhesion is roughly constant across the length of the contact patch. Finally, when a small section of the engaged film is evaluated, T_+ in the direction towards the center of the gripper, is slightly larger than T_- . Both have a small normal component that faces toward the center of the object, which helps to pull non-engaged sections of the adhesive into contact [31]. The second key component of the shear adhesion gripper is a bi-stable support structure. It is crucial that the film is laid onto the surface with minimal crinkling which decreases contact area and thus adhesion, so initiating a grasp with taut film is suitable. It is also important to ensure that the film adjusts to the curved surface during the grasp. To achieve both of these design requirements, a support structure must be designed so that it is stable in both a straight position and in a collapsed position [31].

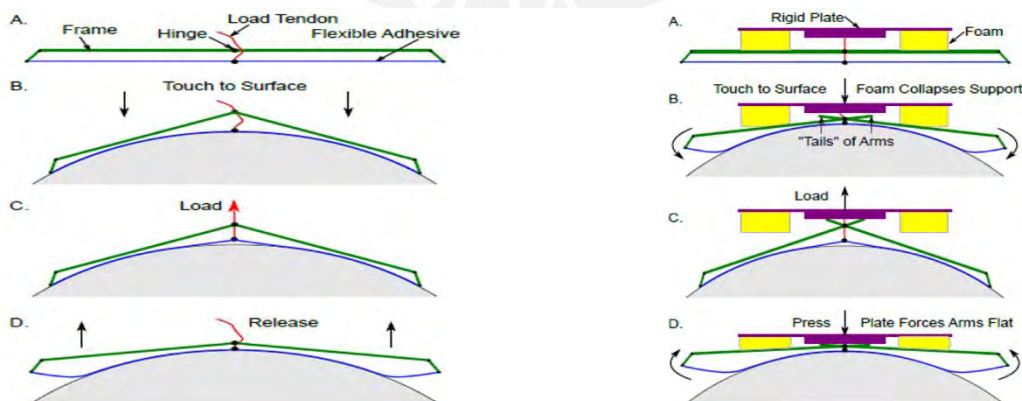


Fig. 1.27: Support structure. (A) rest position, (B) grasping, (C) lifting, (D) release [31]

As sketched in figure 1.27, in order to carry up a curved object, the gripper is first conveyed into contact with the object and has force applied to both arms to collapse the support structure. Once the structure has collapsed and the film has contacted the surface, the gripper can be loaded to lift the object. After the accomplishment of the manoeuvring, the gripper can be released by lifting the two arms to return the gripper to its straight configuration. Adding a plate, two pieces of foam, and “tails” on the ends of the arms enables the gripper to release without external actuation as displayed. During contact with the object, the foam gently presses on the extents of the arms, causing the support structure to collapse. Loading occurs without interference from the foam. To release, the object is set down, and the rigid plate is pressed into the tails of the arms while the soft foam compresses. This action makes the support to become straight again and releasing the object. The foam used is viscoelastic and remains compressed long enough to allow the pressing force to be removed [31].

1.3.4.5 Self-adaptive passive gripper finger (Tufts passive)

One step forward in the developing of Finray Effect Grippers is the inclusion of the *Manduca sexta* principle, which make use of passive gripping as a method of energy conservation. The prolegs of *M. sexta* use grappling hook-like crochets to dig into the substrate they are gripping. To release from the substrate, the animal pushes the crochets down, rotates them outwards, and then retracts the crochets [32].

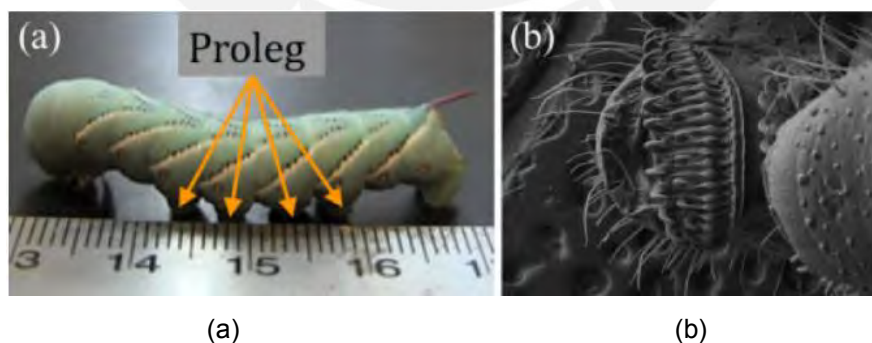


Fig. 1.28: (a) *Manduca Sexta* prolegs; (b) Magnified view of crochets [33]

Passive grippers minimize power consumption, which is suitable for power-limited environments such as those where robots are energized with batteries or

solar power systems or lack from a stable electric power grid access. Additionally, passive gripping provides a safety operation securing the grip although a power-loss event.

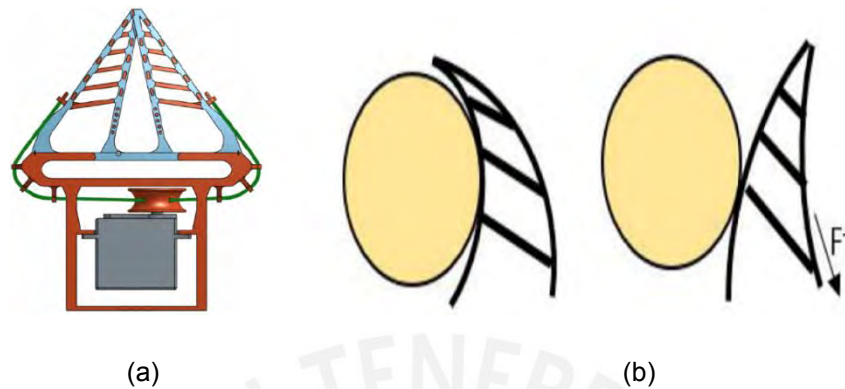


Fig. 1.29: (a) TPG, soft parts- blue, hard parts-red; (b) Fin ray effect [33]

The design criteria for the TPG focused on minimal power consumption plus handling and conforming to a wide range of objects without deteriorating or crashing them. The combination of soft and hard materials used in the model provides the gripper the dexterity to accomplish the goals before mentioned, nonetheless it is mandatory to limit undesired deformations without trading-off the benefits of using soft materials [34]. The TPG is a gripper incorporating soft and hard components that can be three-dimensional printed as a single part and has limited additional steps to attach the motor-tendon actuating system, which comprises on a pulley mounted on a servo, attached to the gripper. The tendon system represented in green in figure 1.29 is attached to the pulley and to anchor points on the gripper. In order to open and close the gripper, the servo is rotated clock- and counter-clockwise.

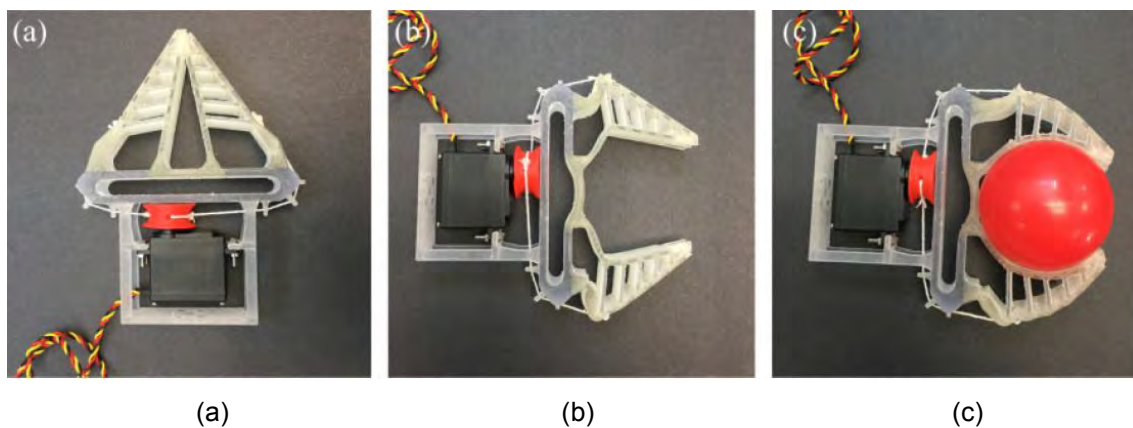


Fig. 1.30: (a) Closed position; (b) Open position; (c) Magnified crochets [33]

The finray - geometry is comprised of two beams connected to each other at one end (tip) with crossbeams of increasing length spaced between the tip and the base. The connections between beams and crossbeams are flexible. The deformation caused by a point force applied to the side of one of the beams give rise to deformation of the entire structure, causing the tip to bend toward the force, resulting in a grasping action. Altering the angle of the crossbeams with respect to the base creates a preferred bending direction [33], reducing the required motor power in order to actuate the gripper. As a conclusion, combining the finray effect and modifying it, the insertion of soft materials plus the manduca sexta effect resulted in an improved passive gripper. The TPG is capable of picking up objects of different silhouettes and sizes, especially objects that are convex and concave. Even though thin-light objects represent higher grasping difficulties for the TPG, it is possible to carry them on due to its fingernails. The payload of the gripper is measured in two separate situations: the gripper perpendicular to the ground and the gripper parallel to the ground as represented in figure 1.31.

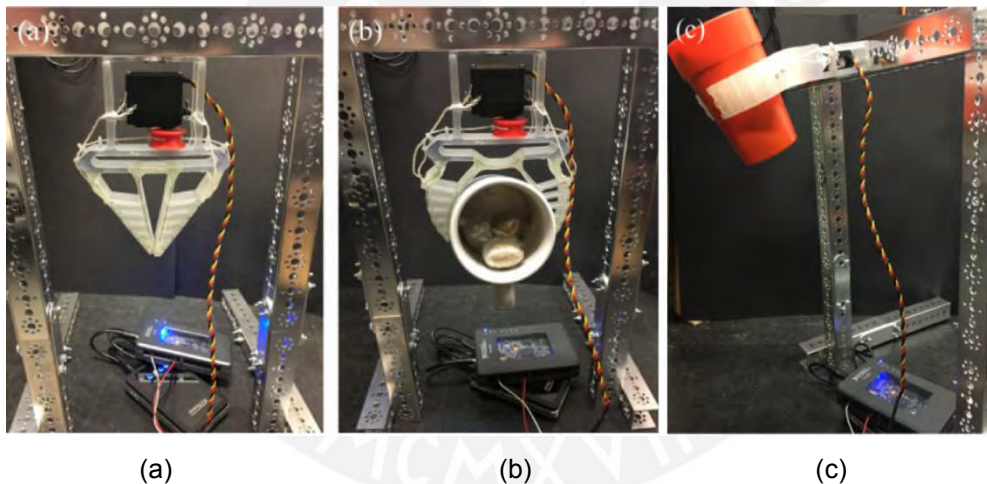


Fig. 1.31: (a)-(b) Perpendicular payload 530 grams; (c) Parallel payload 240 grams [34]

Chapter 2

Solution concept and theories

The task of this thesis is to develop compliant adaptive grippers, which fits different object shapes properly. Properly means that the contact length between the gripper fingers and the object should be increased. The object shapes to be investigated are the following: round, elliptical, square, rectangular and triangular. Hereafter, the solution concept will be presented, together with the mechanical models and theories involved.

2.1 Solution concept

All the gripping solutions presented in the previous chapter have their advantages and disadvantages, which make them suitable for specific applications. Payload, energy conservation, gripper autonomy, reliability, manoeuvrability, release time, safety and costs are some factors to consider on the development. Seeking to fulfill the aspects mentioned above, the work is focused on the development of a passive self-adaptive two-finger gripper using a holding force based on its inherent flexibility. While controlling the gripper bend lines by changing the boundary conditions and inertia, the main objective is to increase substantively the contact length between the object and the gripper fingers.

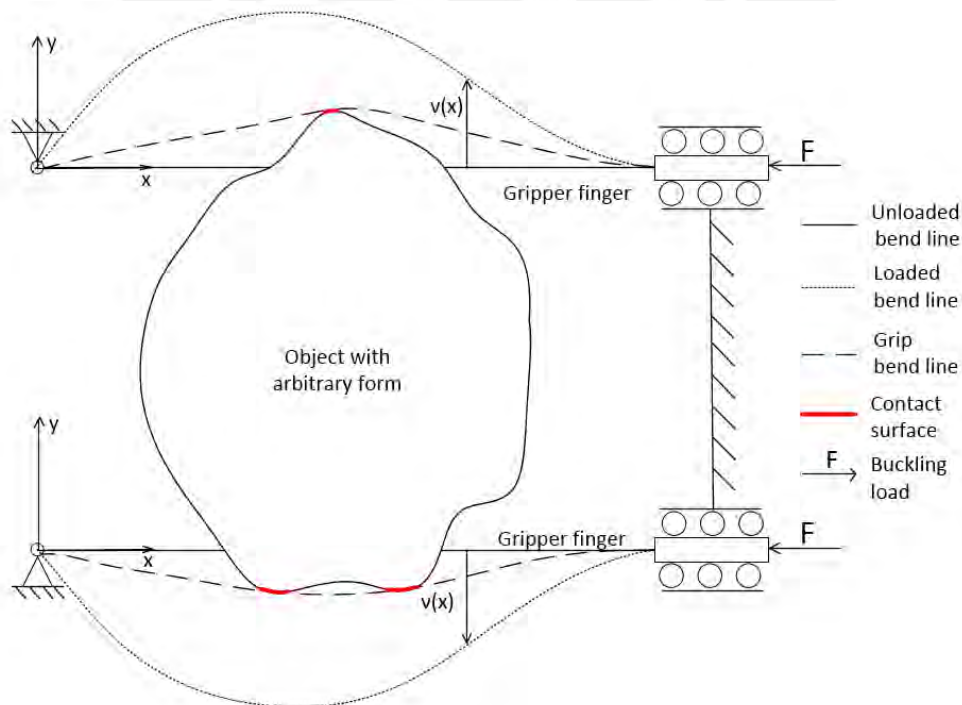


Fig. 2.1 Concept of an adaptable gripper with two compliant gripper fingers

The translatory movement of the gripper finger is generated by an actuator, which applies a buckling load F parallel to the x -coordinate. Different bend lines of the leaf spring-like gripper finger can be achieved. In order to set up a desired bend line, different materials and geometries can be implemented. These two factors are related to the elastic module and moment of inertia respectively. On the other hand, boundary conditions can also be modified by changing the height and inclination angle at the left end of the gripper finger. In this work, different boundary conditions and gripper finger elements are investigated searching for gripping optimization. Hereafter, two mechanical models are sketched as well as gripper fingers with variable stiffness along their length. Furthermore, due to the symmetry of the mechanism, just one finger of the compliant gripper is sketched in the models.

2.2 Mechanical models of the compliant gripper

The deformation results when a translatory displacement d parallel to the x -direction is applied. The magnitude of the deformation on the y -direction is represented by $v_0(x)$ and $v(x)$ under unloaded and loaded condition respectively.

Model 1: clamped-pinned

The first sketch in figure 2.2 is based on the clamped-pinned model. The boundary conditions to be varied are the following:

- a) $y(x=0) = y_0$ (variable)
- b) $M(x=0) = M_0 = 0$
- c) Stroke length = d (variable)

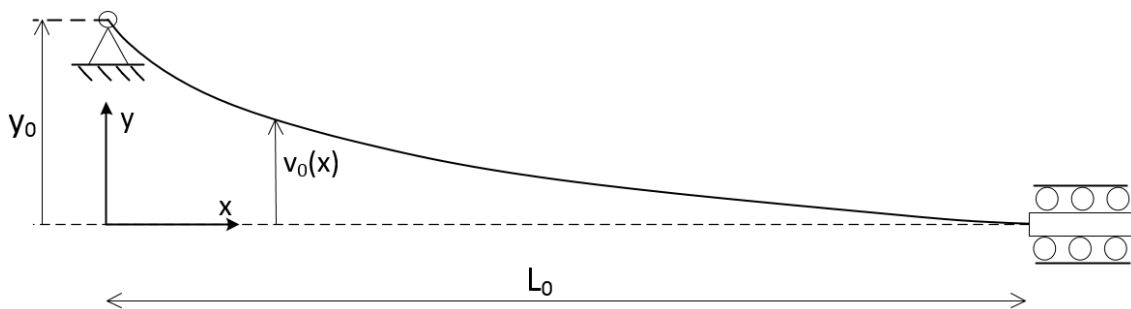


Fig. 2.2: Model 1 - Gripper finger in unloaded condition

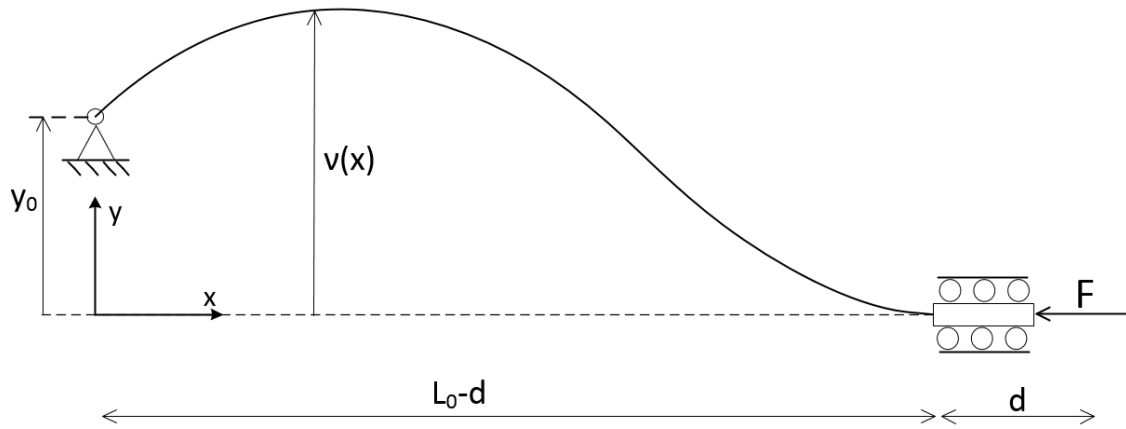


Fig. 2.3: Model 1 - Gripper finger in loaded condition

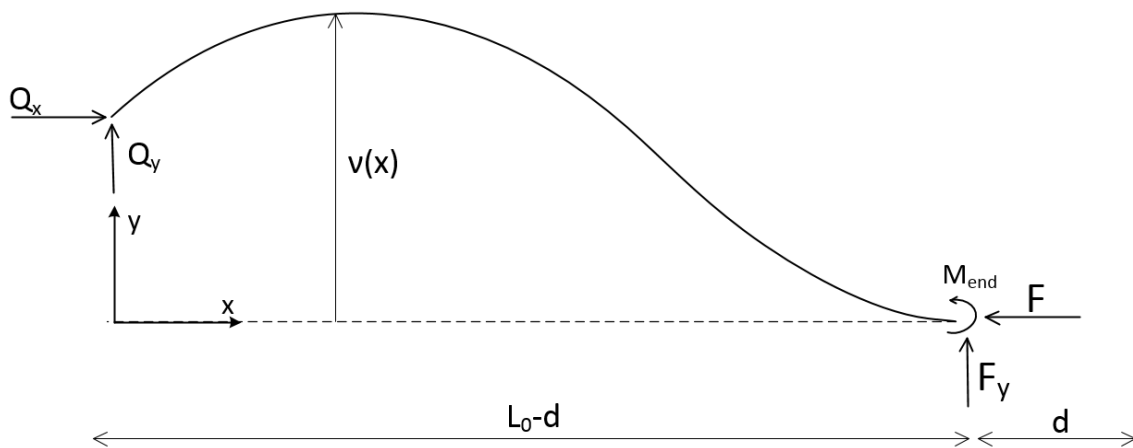


Fig. 2.4 Model 1 - Gripper finger - free body diagram

Model 2: clamped-clamped

The model 2 in figure 2.5 is based on the clamped-clamped model. The boundary conditions to be varied are the following:

- a) $y(x=0) = y_0$ (variable)
- b) $\phi(x=0) = \phi_0$ (variable)
- c) Stroke length = d (variable)

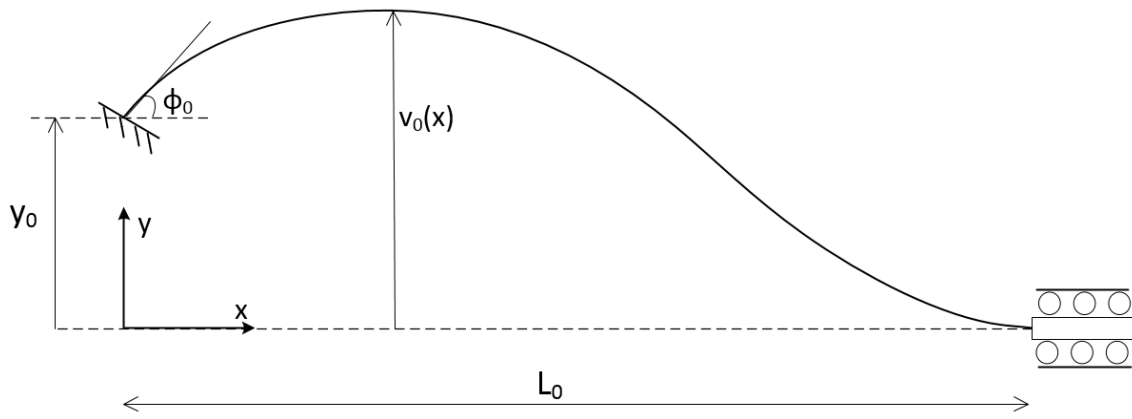


Fig. 2.5: Model 2 - Gripper finger in unloaded condition

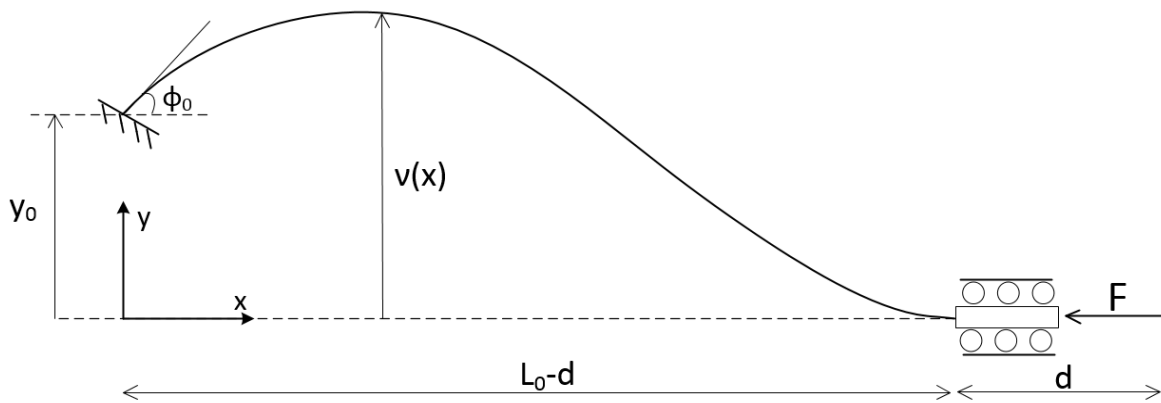


Fig. 2.6: Model 2 - Gripper finger in loaded condition

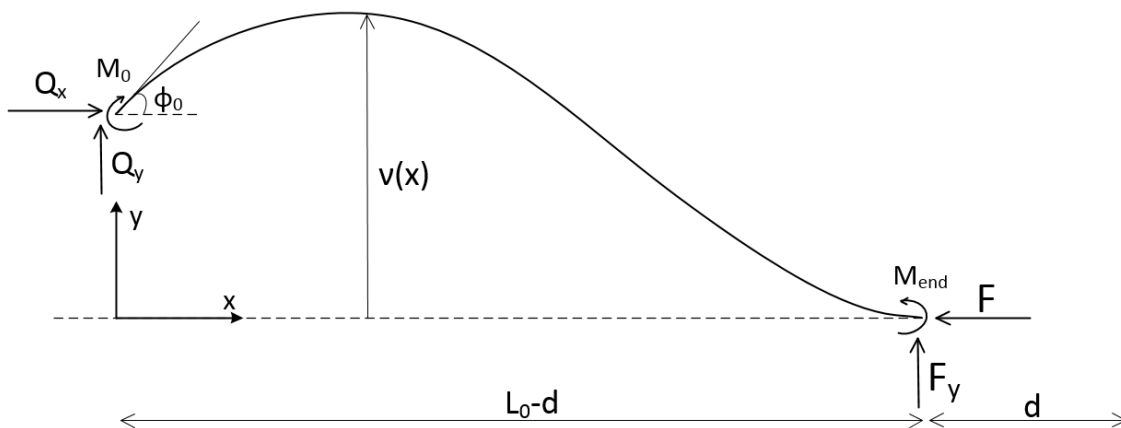


Fig. 2.7: Model 2 - Gripper finger - free body diagram

As mentioned before, in order to obtain diverse bend lines, different stiffness for the gripper finger will be investigated. These are sketched in figures 2.8, 2.9, 2.10 and 2.11.

Gripper finger elements

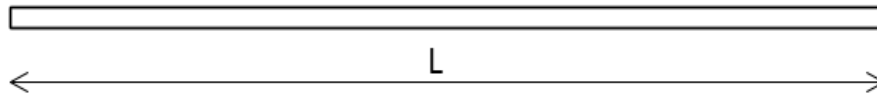


Fig. 2.8: Element A - Constant stiffness gripper finger

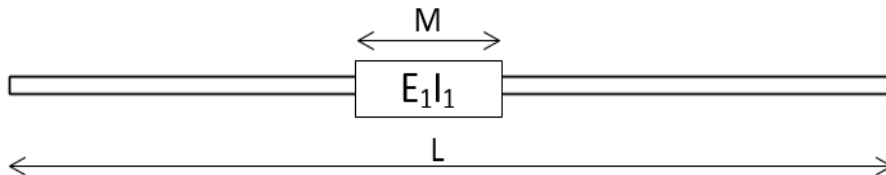


Fig. 2.9: Element B - Gripper finger with one enhanced segment

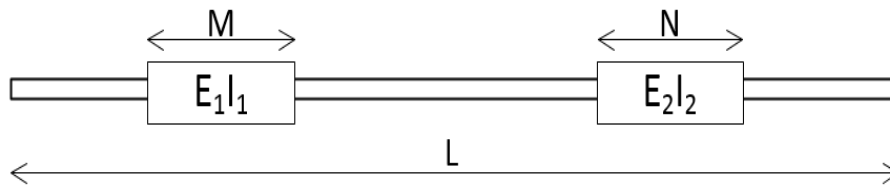


Fig. 2.10: Element C - Gripper finger with two enhanced segments

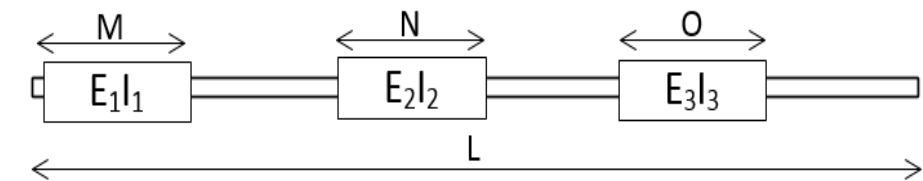


Fig. 2.11: Element D - Gripper finger with three enhanced segments

Depending on the shape of the object, the length of segments M, N and O are adjusted for obtaining an increased contact length between the gripper finger and the object.

On the one hand, as illustrated in figure 2.12, due to the constant stiffness along the element A, it is foreseeable that under load and grip conditions, the resulting shape (while employing the element A as a gripper finger) turns into a curved bend line, which is suitable for gripping round-elliptical objects. Moreover, the bend lines of the two-gripper fingers are symmetric with respect to the x-axis.

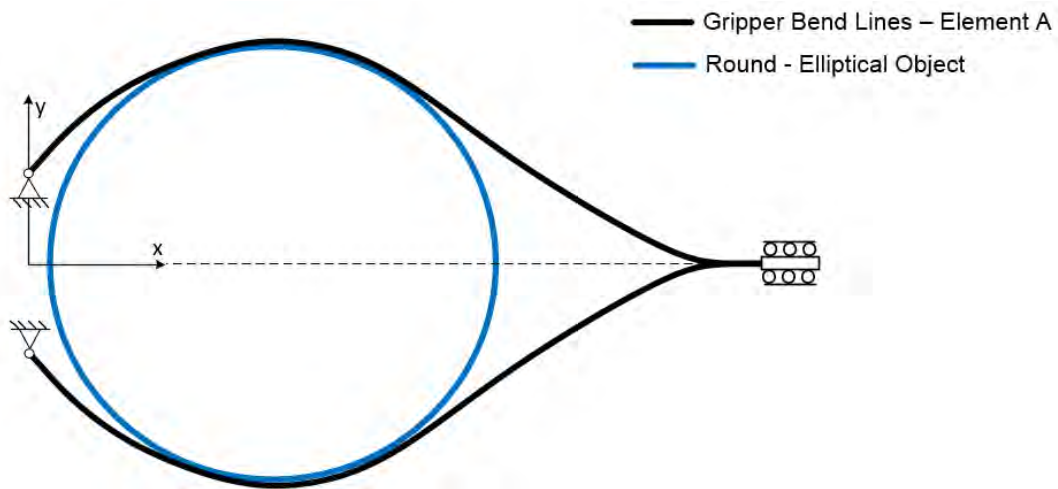


Fig. 2.12: Model 1 - Element A, grip condition – elliptical objects

For model 2, the resulting shape will also be a curved one, appropriate for round-elliptical objects because of the same reason explained in the previous paragraph. On the other hand, for square, rectangular and triangular object shapes, which own straight sides, some segments of the gripper finger should be reinforced by increasing their stiffness, so that under load conditions they remain straight as illustrated in figures 2.13, 2.14, 2.15, 2.16 and 2.17. For this reason, elements B, C and D are going to be considered in the analysis.

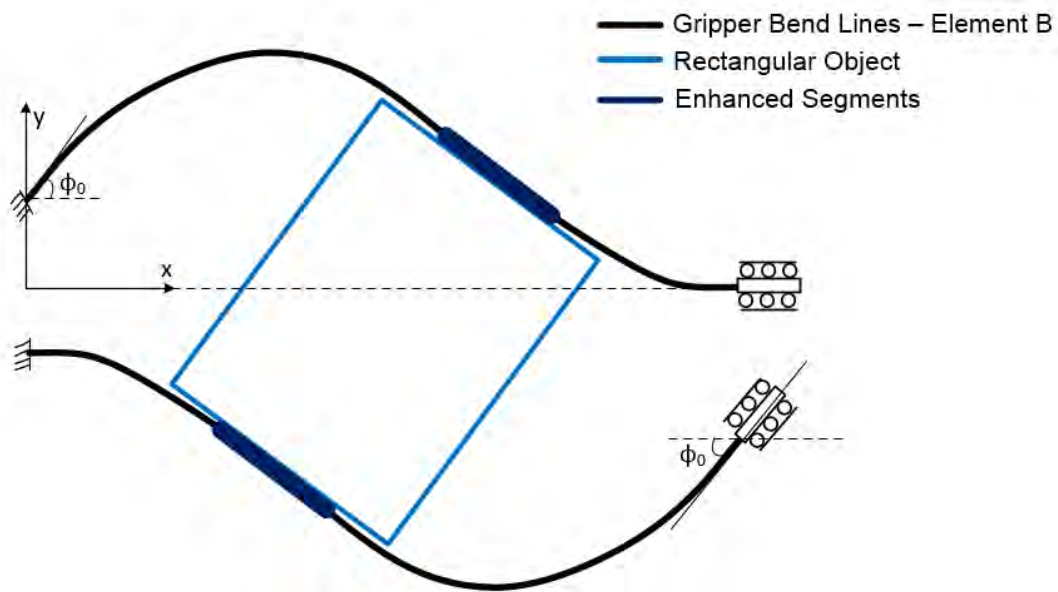


Fig. 2.13: Model 2 - Element B, grip condition – rectangular object

Note: In figure 2.13, the bend lines of the two gripper fingers are not symmetric. On the contrary, the boundary conditions are transposed.

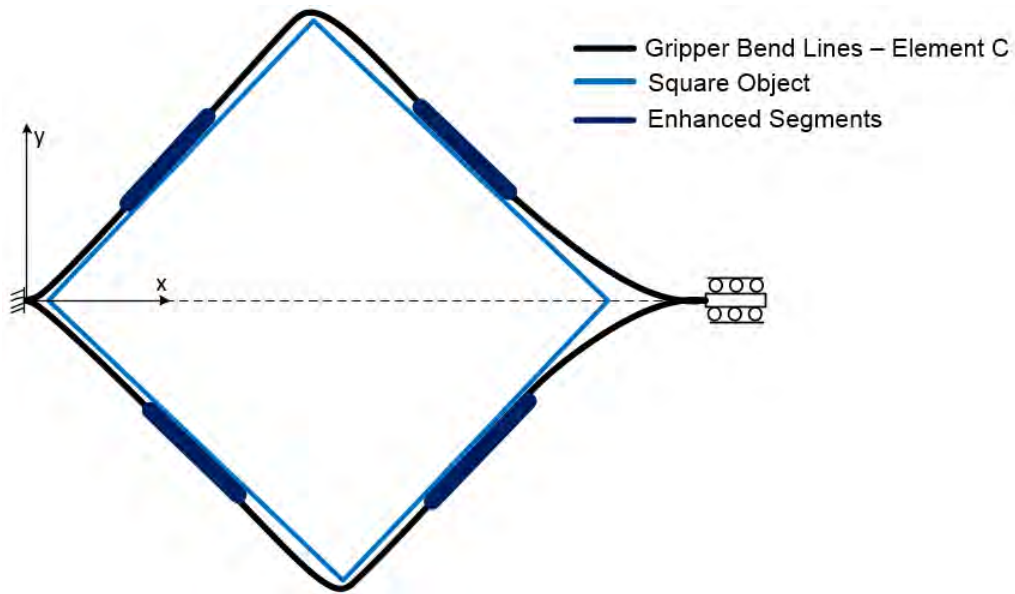


Fig. 2.14: Model 2 - Element C, grip condition – square object

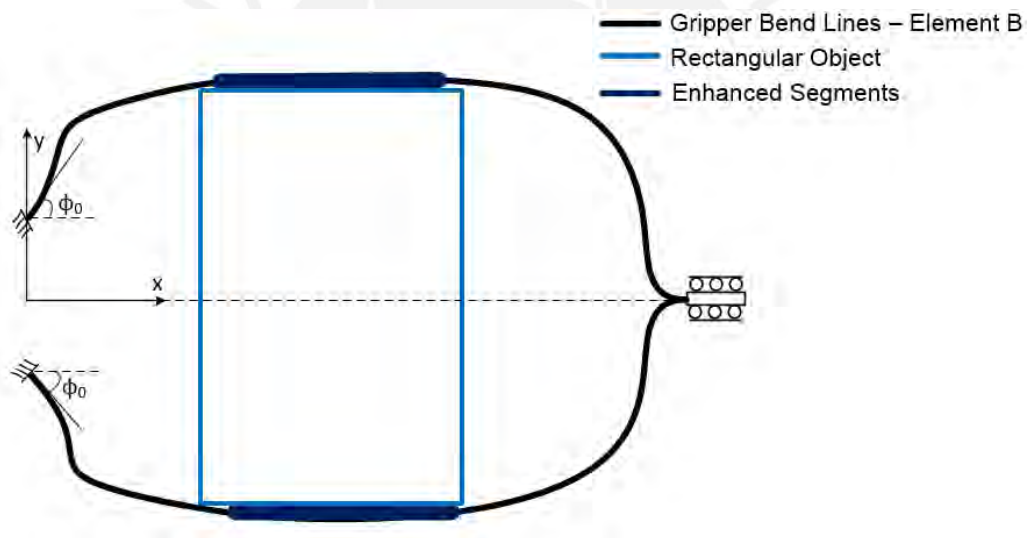


Fig. 2.15: Model 2 - Element B, grip condition – rectangular object

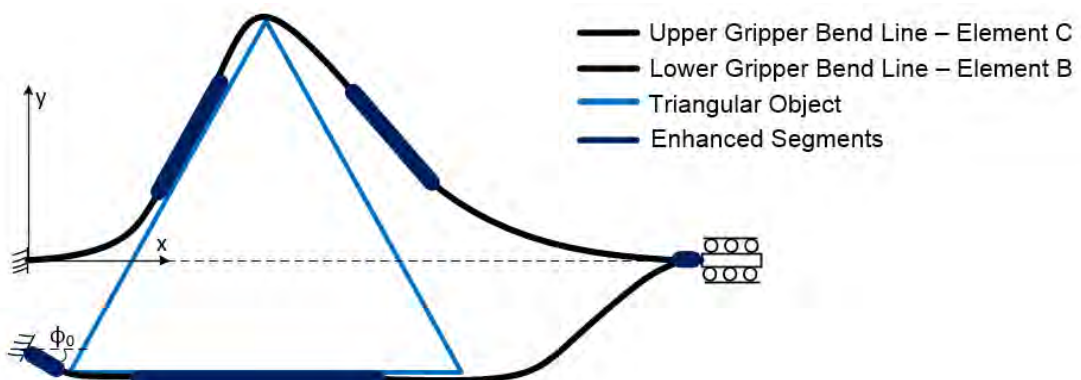


Fig. 2.16: Model 2 - Elements B and C, grip condition – triangular object

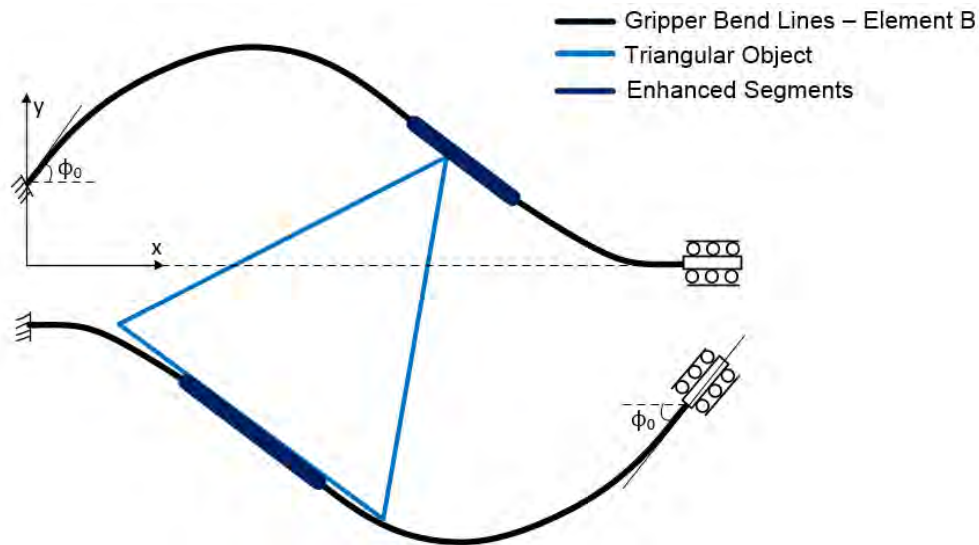


Fig. 2.17: Model 2 - Element B, grip condition – triangular object

As depicted in figure 2.16, the compliant gripper can be also comprised of two different elements. Coming back to the main goal of this work, which is the optimization of the gripping process, as illustrated for square and rectangular objects in figures 2.13, 2.14, 2.15, different contact lengths can be achieved depending on how the object is gripped. In this work, the goal will be to increase significantly the contact length. The variables to control are the boundary conditions (angle ϕ_0 and height y_0 at the left side of the fingers), the stiffness of the gripper fingers by using different types of elements (A, B, C, D) and the stroke lengths.

Models can be linear or non-linear, which leads to the task to solve linear or non-linear differential equations respectively. For the formers, small deformations on the element and linear material relationships through the Hooke's law are considered. For the latters, non-linear equations are generated by accounting large strains and/or non-linearity response deformation due to material properties. In this work, the gripper bend lines are calculated using a non-linear theory implemented in matlab. This theory considers large deformations along the length of the gripper fingers; nevertheless, the material properties are assumed to be linear.

Note: The shown bend lines in figures 2.12 to 2.17 do not result from a calculation but they only represent qualitative shapes. Additionally, in figure 2.17, the bend lines of the two-gripper fingers are not symmetric. On the contrary, the boundary conditions are transposed

2.3 Non-linear deformation theory

This theory is valid for thin rod-shaped elements under external loads. Moreover, a rod-shaped element can contain a hollow cavity inside where a pressurized fluid flows through it, generating bending motion on the element. For the model's design, large strains due to external load application are considered as well as linear material response. The assumptions for the non-linear theory are presented hereafter:

- a) It is considered a static problem
- b) Rod cross-sectional dimensions are much smaller than its length and curvature radius.
- c) The material of the rod complies with Hooke's Law
- d) Bernoulli hypothesis is valid, meaning that cross-sectional areas remain flat
- e) The principle of Saint-Venant is valid
- f) Rods have cavities, the deformation is due to internal pressure and other external loads
- g) The neutral fibre may be dictated by the location of an embedded stripe or thread, with one of the introduced coordinate axis extending in a cross-section through its center of gravity and the point of the neutral fibre being a principal axis of inertia.

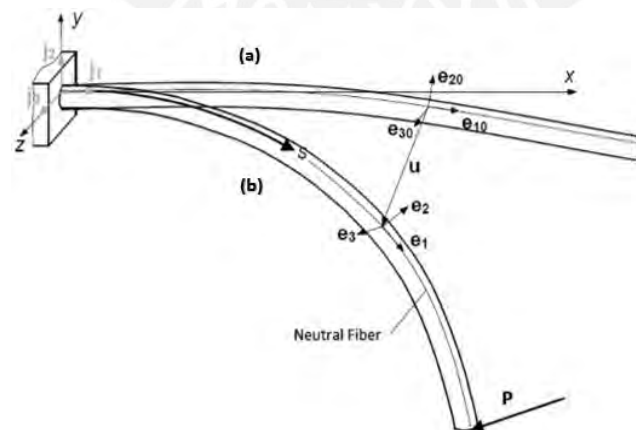


Fig. 2.18: Rod under large deformation \vec{u} ; (a) Unloaded condition; (b) Loaded condition [35]

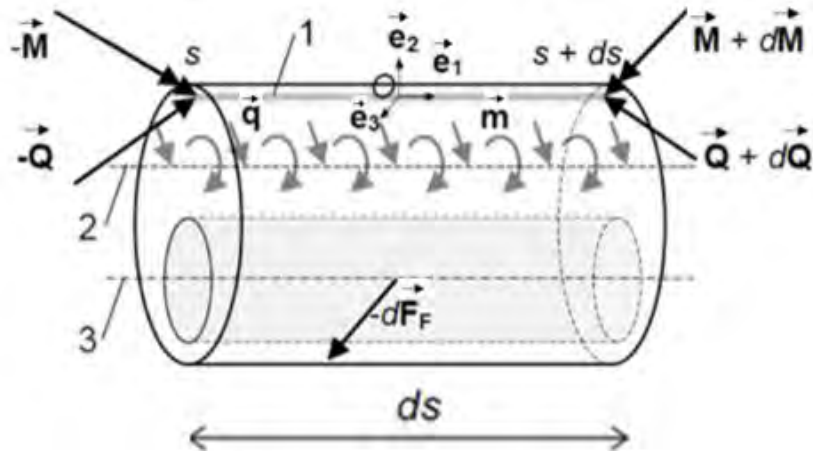


Fig. 2.19: Differential element of the rod under external loads [35]

In figure 2.19 a differential element is introduced where, 1: embedded string (neutral fibre), 2: center of mass of the rod, 3: central line of gravity of the hollow cavity, \vec{Q} : vector force, \vec{M} : moment vector, $d\vec{F}_F$: force applied on the rod due to the inside pressurized fluid, \vec{q} : force per unit length, \vec{m} : moment per unit length. The origin and demonstration of the equations hereafter can be reviewed in the following literature [35]. The equations are derived from equilibrium conditions on a differential element of length ds , which is loaded under the application of external forces and momentums. Furthermore, torsional and bending moments are expressed in terms of the curvature radius and the material properties as follows:

- $\vec{M}_1 = [GI_1(\kappa_1 - \kappa_{10})] \vec{e}_1$: Twisting effect generated by torsion momentum
- $\vec{M}_2 = [EI_2(\kappa_2 - \kappa_{20})] \vec{e}_2$: Bending effect respect to \vec{e}_2 axis
- $\vec{M}_3 = [EI_3(\kappa_3 - \kappa_{30})] \vec{e}_3$: Bending effect respect to \vec{e}_3 axis

The term κ_1 describes the torsion effect on the neutral fibre; κ_2 and κ_3 are bending projections of the neutral fiber on the \vec{e}_2 , \vec{e}_3 directions. The development of the model is performed in two dimensions; therefore, the following parameters are not considered: $Q_z, M_x, M_y, q_z, m_x, m_y, \kappa_1, \kappa_2, \theta_1, \theta_2, u_z$. Finally, the equations to be used for describing the model are written in non-dimensional terms.

$$\frac{d\tilde{Q}_x}{d\tilde{s}} + \tilde{q}_x = 0 \quad (1)$$

$$\frac{d\tilde{Q}_y}{d\tilde{s}} + \tilde{q}_y = 0 \quad (2)$$

$$\frac{d\tilde{M}_z}{d\tilde{s}} + \tilde{Q}_y \cos\theta_3 - \tilde{Q}_x \sin\theta_3 + \tilde{m}_z = 0 \quad (3)$$

$$\frac{d\theta_3}{d\tilde{s}} - \tilde{\kappa}_3 = 0 \quad (4)$$

$$\frac{d\tilde{u}_x}{d\tilde{s}} - \cos\theta_3 + \cos\theta_{30} = 0 \quad (5)$$

$$\frac{d\tilde{u}_y}{d\tilde{s}} - \sin\theta_3 + \sin\theta_{30} = 0 \quad (6)$$

$$\tilde{M}_z = [\tilde{E}I_z(\tilde{\kappa}_3 - \tilde{\kappa}_{30})] \quad (7)$$

where:

$$\tilde{s} = \frac{s}{L} ; \quad \tilde{L} = \frac{L}{L} = 1 \quad \tilde{\kappa}_3 = \kappa_3 L \quad \tilde{M}_z = \frac{M_z L}{EI_z} \quad \tilde{q}_i = \frac{q_i L^3}{EI_z}$$

$$\tilde{F} = \frac{FL^2}{EI_z} ; \quad \tilde{E}I_z = \frac{EI_z}{EI_z} = 1 \quad \tilde{Q}_i = \frac{Q_i L^2}{EI_z} \quad \tilde{u}_i = \frac{u_i}{L} \quad i = x, y$$

2.4 Least squares method

The least squares method is a procedure to determine the best fit line or curve to a given data [36]. The method find the best fit of the form:

$$y_i = \beta_1 + \beta_2 f(x_{2i}) + \dots + \beta_k f(x_{ki}) + \varepsilon_i \quad (i = 1, \dots, n) \quad (8)$$

where

β_k : Column vector of unknown parameters ($k \times 1$)

$f(x_{ki})$: Explanatory variables

ε_i : Column vector of unobserved disturbances ($n \times 1$)

It is not necessary for the functions $f(x_{ki})$ to be linear but that y_i is a linear combination of these functions. The method of least squares is a standard approach in regression analysis to approximate the solution of overdetermined systems, i.e. sets of equations in which there are more equations than unknowns. Least squares means that the overall solution minimizes the sum of the squares of the residuals made in the results of every single equation. Expressing the equation (8) in matrix notation:

$$\vec{y}_i = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}; [X] = \begin{bmatrix} 1 & f(x_{21}) & \dots & f(x_{k1}) \\ \vdots & \vdots & & \vdots \\ 1 & f(x_{2n}) & \dots & f(x_{kn}) \end{bmatrix}; \vec{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}; \vec{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

$$\vec{y} = [X]\vec{\beta} + \vec{\varepsilon} \quad (9)$$

If \vec{b} is a $k \times 1$ vector of estimates of β , then the estimated model is

$$\vec{y} = [X]\vec{b} + \vec{e} \quad (10)$$

where

\vec{e} : Column vector of residuals ($n \times 1$), orthogonal projection of \vec{y}

$[X]$: $n \times k$ Matrix of explanatory variables

A geometric interpretation can be given to the least squares method. Using the expression(10), the residuals, which are the orthogonal components of \vec{y} , can be written as:

$$\vec{e} = \vec{y} - [X]\vec{b} = \vec{y} - [X] ([X]^T [X])^{-1} [X]^T \vec{y} = [M]\vec{y} \quad (11)$$

where

$$[M]: [I]-[X]([X]^T[X])^{-1}[X]^T$$

The matrix $[M]$ is symmetric ($[M]^T=[M]$) and idempotent ($[M]^2=[M]$) also has the property $[M][X]=0$. From equation (11), $[X]^T \vec{e} = 0$. The parallel component of \vec{y} can be expressed as:

$$\vec{y}_p = [X]\vec{b} = [H]\vec{y} \quad (12)$$

where

$$[H]: [X]([X]^T[X])^{-1}X^T$$

\vec{y}_p : Parallel component of \vec{y}

From equations (10) and (11):

$$\vec{y} = [H]\vec{y} + [M]\vec{y} = \vec{y}_p + \vec{e} \quad (13)$$

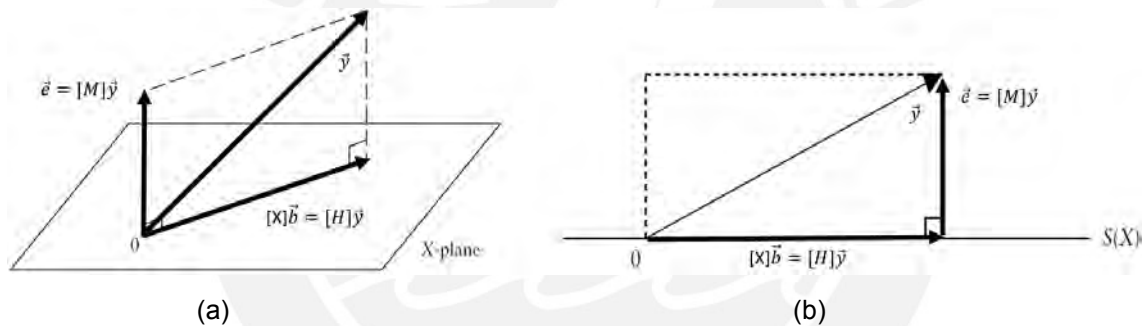


Fig. 2.20: (a) Three-dimensional representation; (b) Two-dimensional representation [36]

In figure 2.20, three - and two-dimensional geometric impression of least squares is presented, the vector of observations on the dependent variable \vec{y} is projected onto the plane of the independent variables \vec{x} . Let $S(X)$ be the space spanned by the columns of $[X]$ (that is, the set of all $n \times 1$ vectors that can be written as $[X]\vec{a}$ for some $k \times 1$ vector \vec{a}) and let $S^\perp(X)$ be the space orthogonal to $S(X)$ (that is, the set of all $n \times 1$ vectors \vec{z} with the property that $[X]^T \vec{z} = 0$). The matrix $[H]$ projects onto $S(X)$ and the matrix $[M]$ projects onto $S^\perp(X)$. In $\vec{y} = \vec{y}_p + \vec{e}$, the vector \vec{y} is projected into two orthogonal components, with $\vec{y}_p \in S(X)$ according to equation (12) and $\vec{e} \in S^\perp(X)$ according to (11).

The best column vector \vec{b} minimizes the total error $e(x)$.

$$e(x) = \|[X]\vec{b} - \vec{y}\|^2 \quad (14)$$

Using calculus for performing the derivatives $\frac{\partial e}{\partial b_i} = 0$, the following final expression of the least squares method is obtained:

$$[X]^T [X] \vec{b} = [X]^T \vec{y} \quad (15)$$

In general, n data points are being fitted by k parameters b_1, b_2, \dots, b_k . The matrix $[X]$ has k columns and $k < n$. The derivatives of $\|[X]\vec{b} - \vec{y}\|^2$ give the n equations $[X]^T [X] \vec{b} = [X]^T \vec{y}$.

Summarizing:

- 1) The least squares solution $[X]\vec{b}$ minimizes the error $e(x) = \|[X]\vec{b} - \vec{y}\|^2$. This is the sum of squares of the errors in the n equations.
- 2) The column vector containing the best coefficients \vec{b} come from the normal equations $[X]^T [X] \vec{b} = [X]^T \vec{y}$.
- 3) The solving of the n equations $[X]^T [X] \vec{b} = [X]^T \vec{y}$ gives the least squares solution, the combination with smallest MSE (mean square error).

Chapter 3

Goal description and process flow diagrams

3.1 Goal description

Two possible procedures (direct and indirect) for finding a gripping solution can be performed. On the one hand, in the direct procedure, the design parameters of the compliant gripper should be calculated for every single object size, meaning that the calculation process is not automatized. On the other hand, the indirect procedure find first the object that best fits a gripper geometry. Then, by changing the geometry of the compliant gripper and object comparison, the gripping solution can be found for different object sizes. Using the power of computation for performing the required iterations, enables the possibility to have an automatized process, saving money and time in the design process. In the indirect procedure, an optimized object from now on called object 1, is compared to a desired object 2 to which the gripper should adapt. If the dimensions of the object 1 and 2 match with each other, then, the compliant gripper geometry is selected as the gripping solution. In this thesis, the indirect approach has been developed.

The scope of this work with respect to round-elliptical objects consists in the implementation of an algorithm that calculates an optimized object for a compliant gripper under different boundary conditions. The development of the algorithm *Elliptical Object Fitting* is a previous step in order to reach the final goal for the indirect procedure explained in the first paragraph. As data output, the algorithm returns an optimized object (significantly increased contact length) for a compliant gripper geometry. This algorithm is comprised of five modules. In module 0, the gripper bend lines are calculated by solving the differential equations (1)-(7) of the non-linear deformation theory. Because only a part of the length of the gripper finger is in contact with the object, a range filtering is settled in module 1. The refinement of the range depends on the boundary conditions and is performed to retrieve the best suitable range for the next process. Immediately, this refined range is used as data input in module 2, where a second algorithm called *ellipse_fit* is employed. In this second algorithm, the least squares method is implemented. In the first iteration process, the round-elliptical object might intersect the bend lines of the gripper finger, meaning that it is not completely fitted; thus, in modules 3 and 4, further data processing is performed so that for the next least squares iteration, a better fit is achieved. All the modules described

before run under a closed loop until a specific exit-loop condition is fulfilled. The exit-loop conditions are settled according to the boundary conditions, stroke lengths and logical values, which are generated during the data processing. Finally, after the fulfillment of an exit-loop condition, the loop is broken and the final solution is obtained.

3.2 Process flow diagram: round-elliptical objects

Before starting with the detailed description for each module of the algorithm *Elliptical Objects Fitting*, the process flow diagram 3.1 illustrates an overview of the processes in order to reach the final goal described in section 3.1. In addition, the scope of the thesis is settled, as well as the processes involved in it.

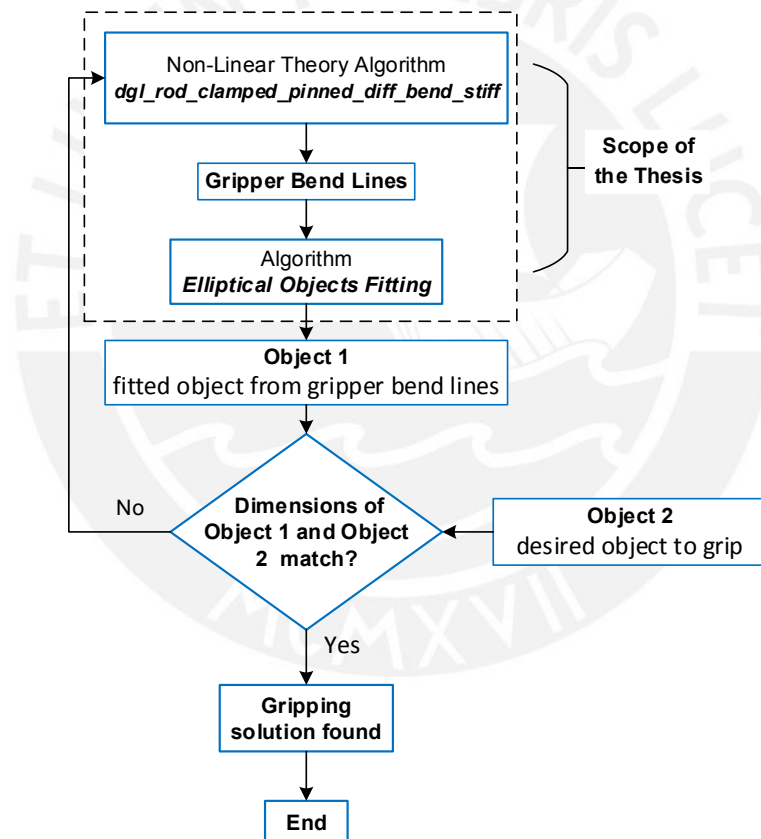


Diagram 3.1: Process flow diagram, final goal overview

Hereafter, the modules involved in the algorithm *Elliptical Objects Fitting* are displayed in diagram 3.2. The total number of modules is four besides the appropriate exit-loop conditions, which controls the continuity of the loop.

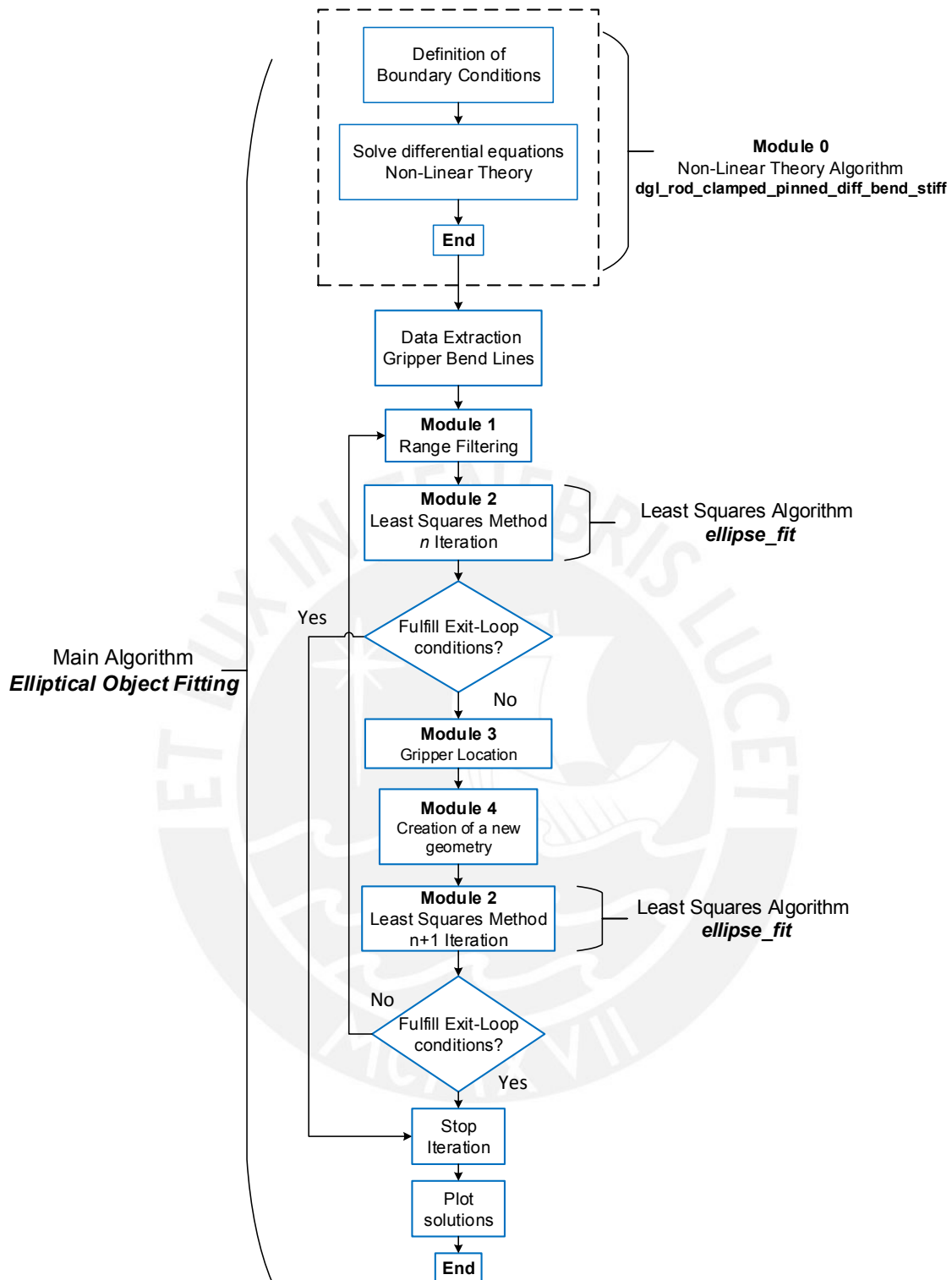


Diagram 3.2: Process flow diagram, algorithm *Elliptical Objects Fitting* – modules

Finally, the diagram 3.3 displays the arrangement of the modules, together with their inputs, outputs and the reset processes at the end of the loops. The reset of the variables avoids that these are biased due to the previous iteration.

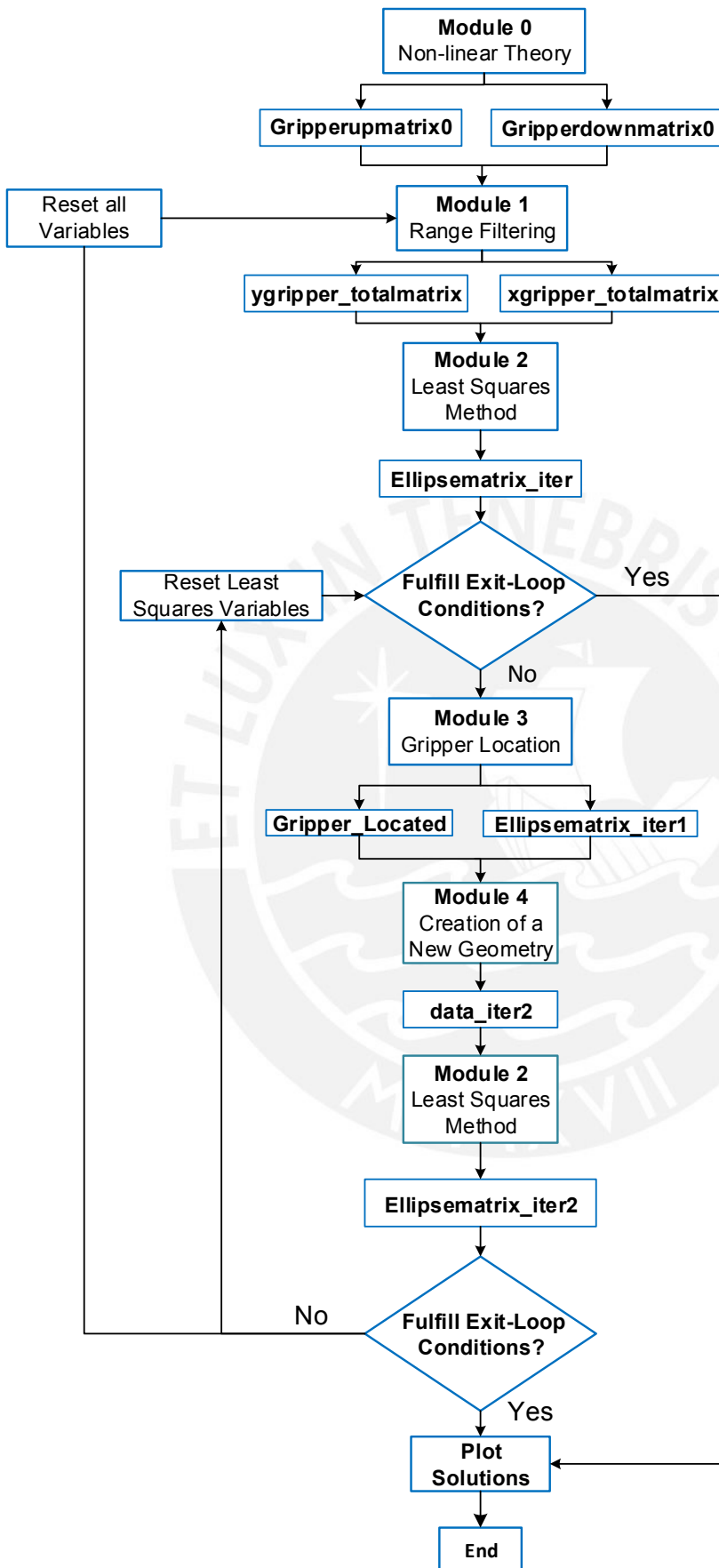


Diagram 3.3: Process flow diagram, algorithm *Elliptical Objects Fitting* - input and outputs

3.3 Process flow diagram: square-rectangular objects

The scope of this work with respect to square-rectangular objects consists in setting the ideas for the implementation of an algorithm that calculates the best-fitting object for a compliant gripper under different boundary conditions and stroke lengths. The concepts and theories for the algorithm for square-rectangular objects are similar to those of the round-elliptical ones. These are: data extraction of the gripper bend lines, range filtering, the least square method and suitable exit-loop conditions. It is foreseeable that depending on the type of the grip as shown in figures 2.15 and 2.13, different exit-loop conditions might be needed. In addition, the enhanced segments should be included in the logic of the algorithm. Further details regarding these issues are going to be addressed in chapter 5.

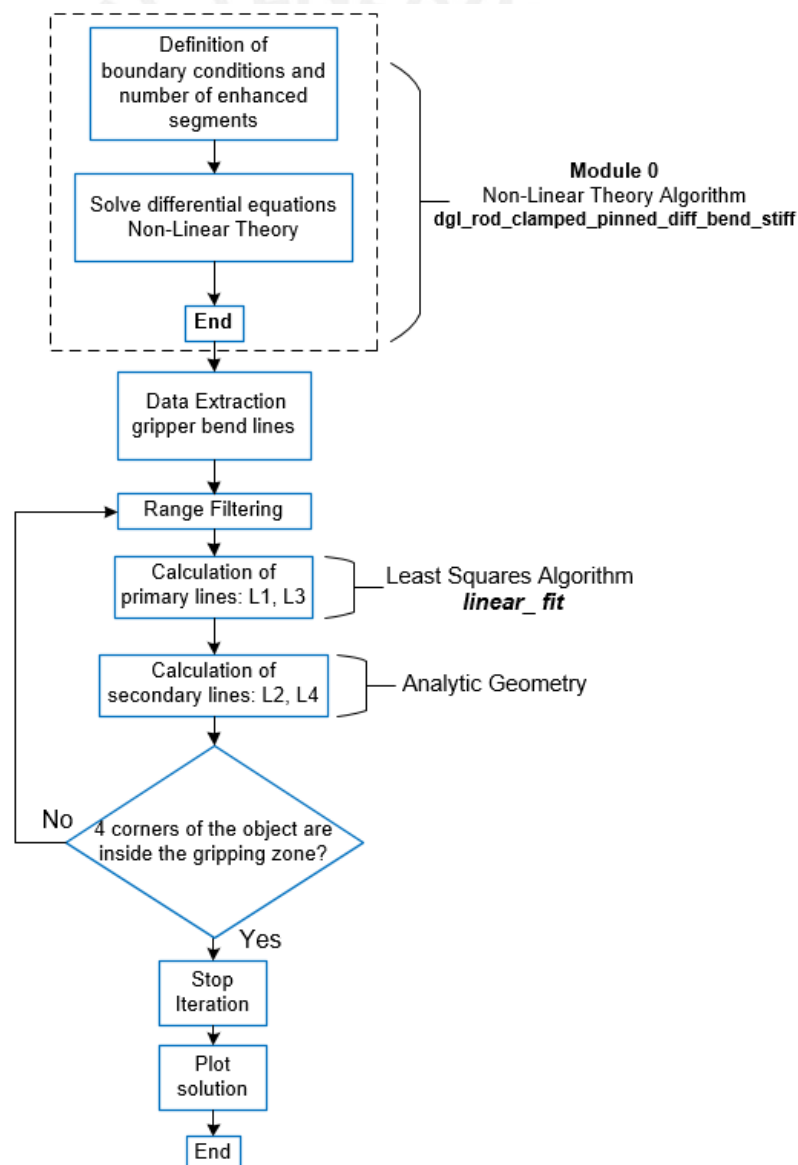


Diagram 3.4: Process flow diagram – ideas for an algorithm for square-rectangular objects

Chapter 4

Algorithm for fitting elliptical objects

In Chapter 4, a detailed description for each module of the algorithm *Elliptical Objects Fitting* is given. For further information of the variables involved in each module, a revision of the appendix is recommended. Moreover, when referencing a section of the algorithm, the following notation will be used: [L: A-B], where L is the abbreviation of the word "line" and A and B represent integer numbers greater than zero, which represent the number of the lines in the program.

4.1 Module 0: non-linear theory- differential equations

In order to get the data input for the least squares method analysis, first it is necessary to set the boundary conditions as well as the stroke length for the two-gripper finger model and solve the differential equations (2) – (8) of the non-linear theory presented in Chapter 2. For this purpose, the program *dgl_rod_clamped_pinned_diff_bend_stiff* is used. It employs as data input the stroke length, the initial values for the forces and momentums and the stiffness factor for the rod. As outgoing data, the points of the gripper bend lines under loaded condition are obtained. The appropriate managing of the data defines how well the least squares method approximation will be; therefore, before introducing the previously obtained points of the gripper bend lines to the least squares approach, a range should be selected.

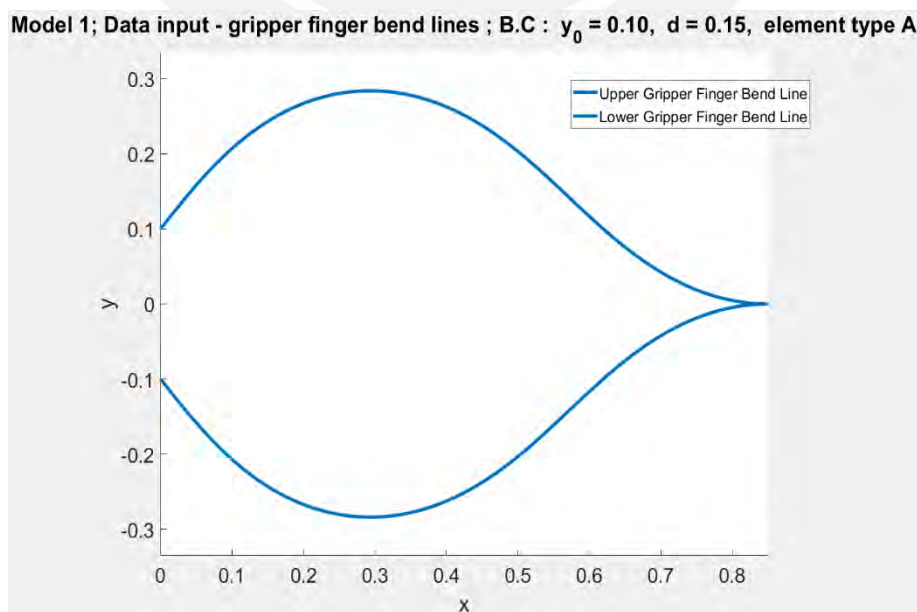


Fig. 4.1: Data input – plot of the gripper bend lines – Model 1

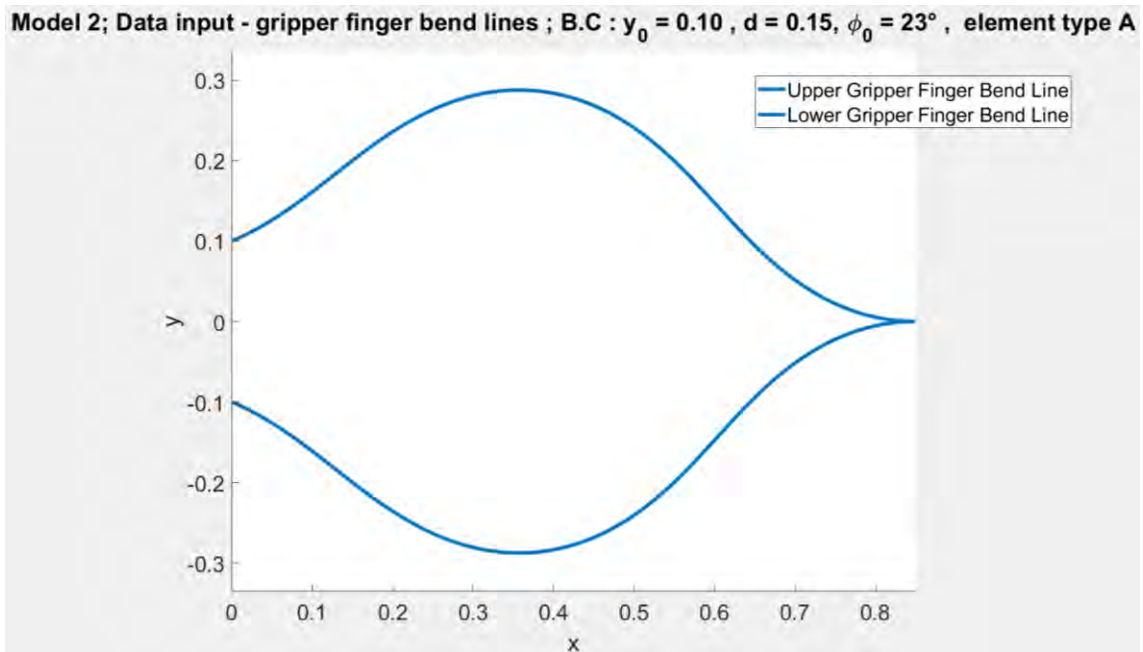


Fig. 4.2: Data input – plot of the gripper bend lines – Model 2

It is important to highlight that changing the boundary conditions and stroke lengths have a direct impact on the start and end positions of the elliptical shape in the gripper bend lines. For example, in figure 4.1, which corresponds to model 1, the elliptical shape is located in the range $[0,05\dots0,50]$. On the other hand, in figure 4.2, for model 2, the elliptical shape starts approximately between the values $[0,15\dots0,55]$. For this reason, it is crucial to define the appropriate range before starting with the least squares method.

4.2 Module 1: data input extraction and filtration of the range

After defining the boundary conditions and stroke length, all the points from the bend lines are stored in two variables, one for the upper gripper bend line and another for the lower gripper bend line. At this moment, the first filtration of the domain takes place. Depending on the boundary conditions previously established, the algorithm will use this information to start the first iteration of the least squares method with the best possible range where the elliptical shape appears. Hereafter, two examples containing different ranges are displayed.

Model 1; Data input - gripper finger bend lines ; B.C : $y_0 = 0.10$, $d = 0.15$, element type A

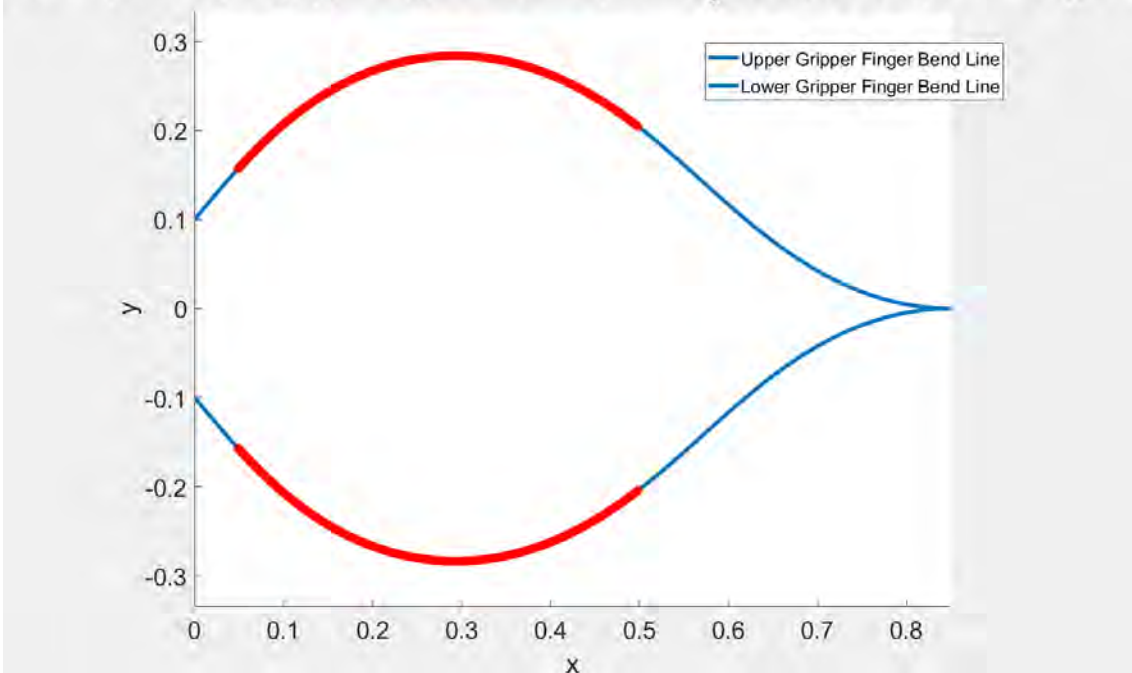


Fig. 4.3: Data input filtration – Model 1

Model 2; Data input - gripper finger bend lines ; B.C : $y_0 = 0.10$, $d = 0.15$, $\phi_0 = 23^\circ$, element type A

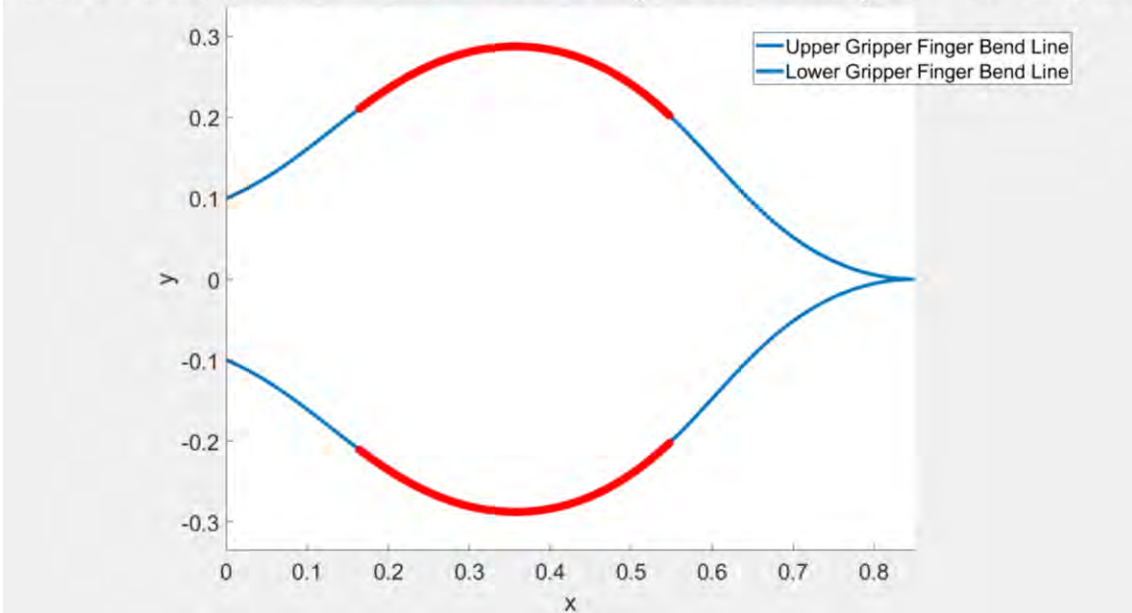


Fig. 4.4: Data input filtration – Model 2

After the filtration of the domain, the next step is to prepare the matrices for the least squares method. The matrices contain the information of the quadratic form of an ellipse described by the following expression:

$$ax^2 + 2bxy + cy^2 + 2d_e x + 2fy + g = 0 \quad (16)$$

The objective at this stage is to find out the best coefficients a, b, c, d_e, f and g so that the best first possible fit is attained using the filtered domain highlighted in red in figures 4.3 and 4.4.

4.3 Module 2 : function *ellipse_fit* - least squares method

4.3.1 Creation of the matrices using the filtered data input

In the algorithm *Elliptical Object Fitting*, the program of the least squares method is called by the function *ellipse_fit.m*. This function processes the data, which was filtered according to the boundary conditions so that the best possible fit is performed. The equation to solve is the following:

$$[X]^T [X] \vec{b} = [X]^T \vec{y} \quad (17)$$

In order to find the column vector \vec{b} , we have to build the matrix $[X]$ and the column vector \vec{y} . Dividing the expression (16) by the coefficient a and moving to the right side of the equation the term x^2 :

$$2bx_i y_i + cy_i^2 + 2d_e x_i + 2fy_i + g = -x_i^2 \quad (18)$$

Writing equation (18) in matrix notation for the i points:

$$\begin{bmatrix} x_1 y_1 & y_1^2 & 2x_1 & 2y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i y_i & y_i^2 & 2x_i & 2y_i & 1 \end{bmatrix} \begin{bmatrix} b \\ c \\ d_e \\ f \\ g \end{bmatrix} = \begin{bmatrix} -x_1^2 \\ \vdots \\ \vdots \\ \vdots \\ -x_i^2 \end{bmatrix} \quad (19)$$

Simplifying equation (19)

$$[X] \vec{b} = \vec{y} \quad (20)$$

where

$[X]$: $n \times 5$ matrix containing the functions $f(x_i, y_i)$

\vec{b} : $[b \quad c \quad d_e \quad f \quad g]^T$, least squares coefficients, note that $a = 1$

\vec{y} : column vector containing the function $f(x_i)$

4.3.2 Least squares iterations : ellipse parameters calculation

The matrix $[X]$ and the column vector \vec{y} have the n points $P(x_i, y_i)$ of the filtered data input evaluated in $f(x_i, y_i)$ and $f(x_i)$ respectively. Finally, the coefficients of the quadratic form of the ellipse, which are the components of the column vector \vec{b} are obtained by solving the following expression:

$$\vec{b} = ([X]^T [X])^{-1} [X]^T \vec{y} \quad (21)$$

The vector \vec{b} contains the coefficients of the quadratic ellipse equation, which best fits the data input. The next step consist on using these coefficients to figure out the necessary parameters, which characterise the ellipse obtained by the least squares method. The parameters to calculate are the following: center coordinates (x_0, y_0) , length of semi-major and -minor axes and angle of rotation. However, because of the gripper bend lines symmetry with respect to the (x,y) axes, also the ellipse axes are parallel to the cartesian coordinates (x,y) ; therefore, the rotation angle of the ellipse is zero. The following equations are extracted from literature [37].

$$\Delta = b^2 - ac \quad (22)$$

$$x_0 = \frac{(cd - bf)}{\Delta} \quad (23)$$

$$y_0 = \frac{(af - bd_e)}{\Delta} \quad (24)$$

$$\text{Semi-Axis 1} = \frac{\sqrt{2(af^2 + cd_e^2 + gb^2 - 2bdf - acg)}}{\sqrt{\Delta[\sqrt{(a-c)^2 + 4b^2} - (a+c)]}} \quad (25)$$

$$\text{Semi-Axis 2} = \frac{\sqrt{2(af^2 + cd_e^2 + gb^2 - 2bdf - acg)}}{\sqrt{\Delta[-\sqrt{(a-c)^2 + 4b^2} - (a+c)]}} \quad (26)$$

Depending on the stroke length d applied at the beginning of the calculation, the semi-minor and semi-major axes can get the values of equations (25) or (26) and viceversa, thus, an if-statement should be considered so that the algorithm recognises the correct minor and major axes.

4.3.3 Evaluation of the semi-major axis of the ellipse

All the operations described in sections 4.3.2 are compute in the algorithm *ellipse_fit*, which takes as data input the filtered data from section 4.2. An important step consist on the determination of the semi-major axis of the ellipse.

Model 1; Data input - gripper finger bend lines ; B.C : $y_0 = 0$, $d = 0.10$, $M_0 = 0$, element type A

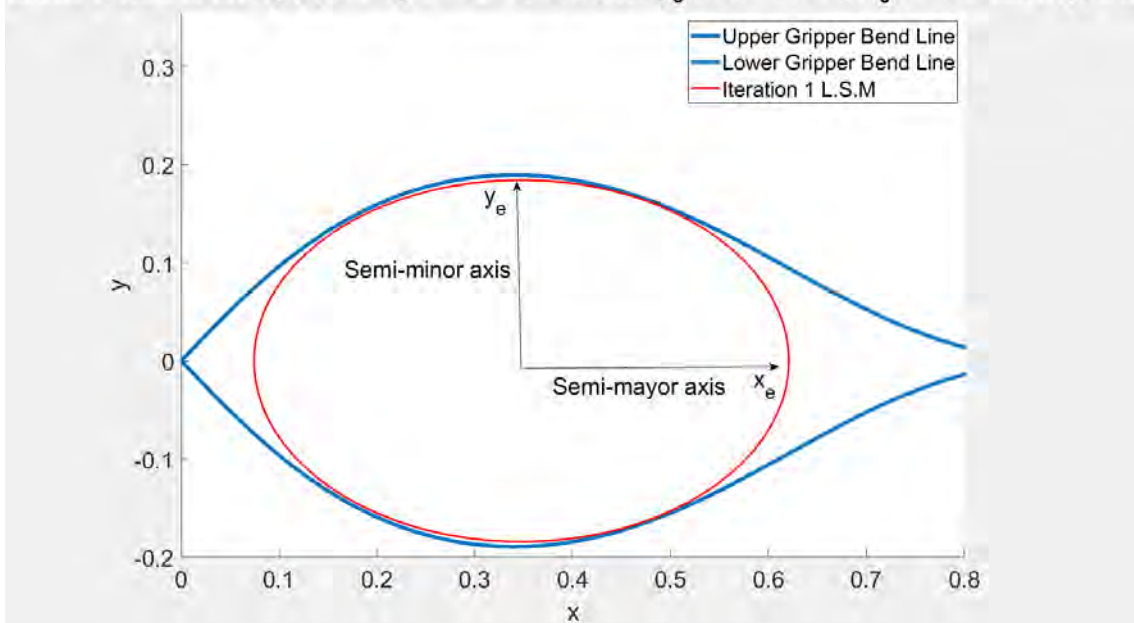


Fig. 4.5: Ellipse plot first least squares iteration, $d < 0.15$

Model 1; Data input - gripper finger bend lines ; B.C : $y_0 = 0$, $d = 0.25$, $M_0 = 0$, element type A

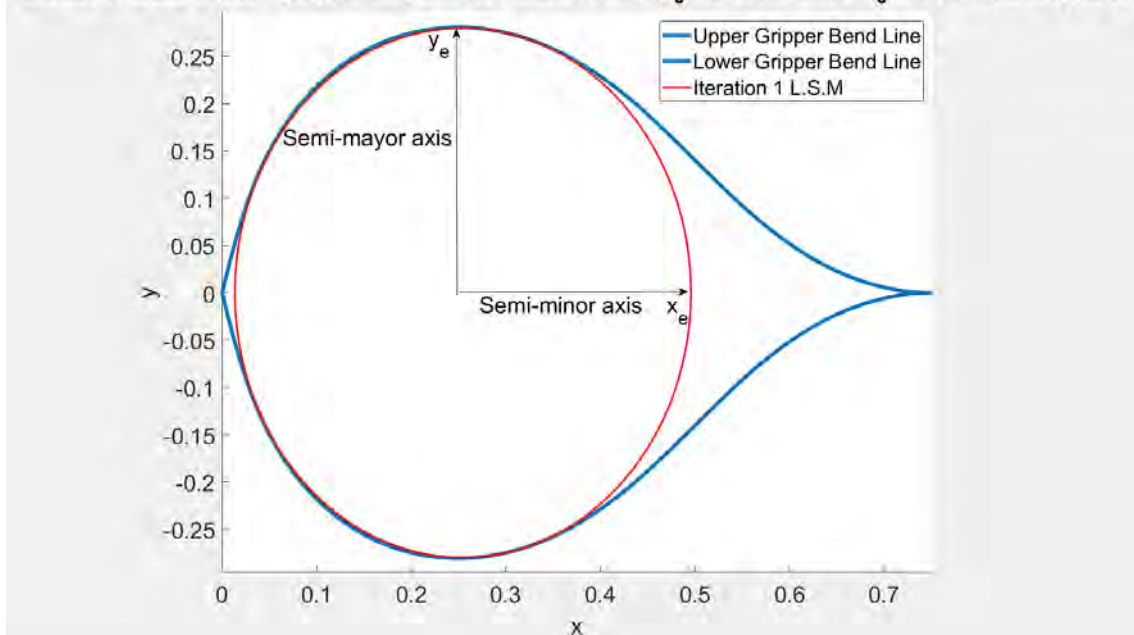


Fig. 4.6: Ellipse plot first least squares iteration, $d > 0.15$

On the one side, in figure 4.5 the semi-major axis is the axis x_e and the semi-minor lies in the axis y_e . On the other side, in figure 4.6 the semi-major axis is now the axis y_e , consequently the semi-minor lies on the axis x_e . After simulating models 1 and 2 under diverse boundary conditions and stroke lengths, it has been found that the value at which this change occurs is when the stroke length d is larger than 0.15. The program must recognise the change of the semi-major axis, therefore an *if*-statement using the stroke length d as defining factor is implemented before the plotting of the solutions. [L: 195-202]

Hereafter, two plots containing the solution of the first least squares iteration under the same boundary conditions and filtered data input are presented. On the one hand, in figure 4.7, the ellipse is plotted taking into consideration the evaluation of the semi-major axis of the ellipse. On the other hand, in figure 4.8, the filter is not considered; thus, the error is greater than in figure 4.7. Furthermore, even though the ellipse in figure 4.8 can be fitted through further least squares iterations into the gripper bend lines, the contact length will not be as large as that from the fit that can be obtained after the same number of iterations for the ellipse in figure 4.7. For these reasons, the evaluation of the semi-major axis of the ellipse is implemented in the algorithm.

Model 2; Data input - gripper finger bend lines ; B.C : $y_0 = 0$, $d = 0.10$, $\phi_0 = 23^\circ$, element type A

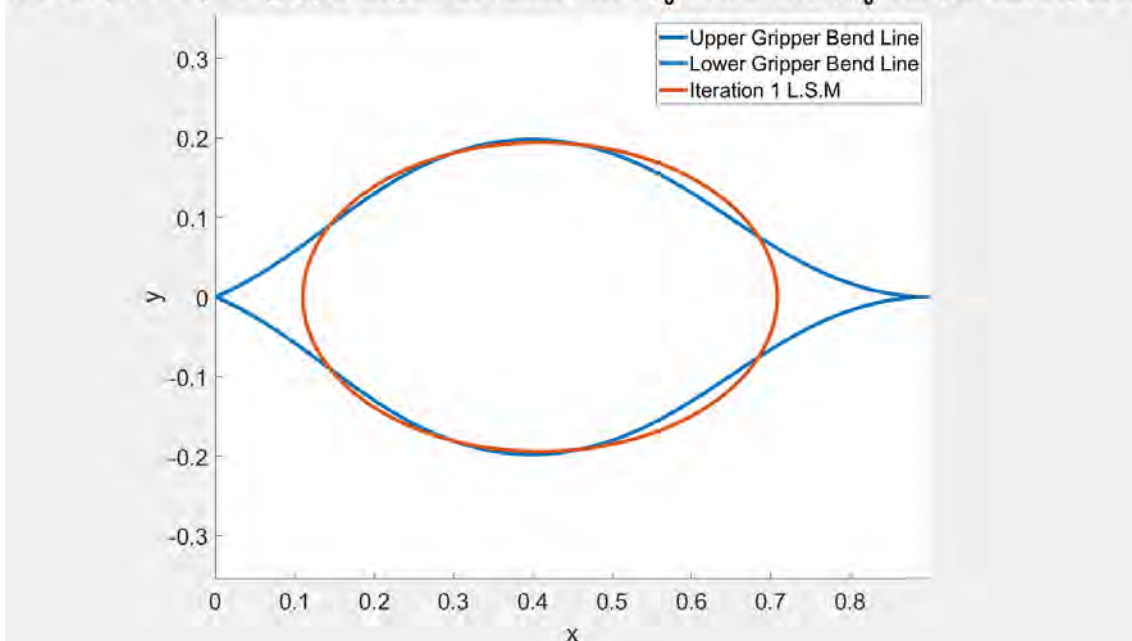


Fig. 4.7: Ellipse plot – first least squares iteration with filtering of the semi-major axis

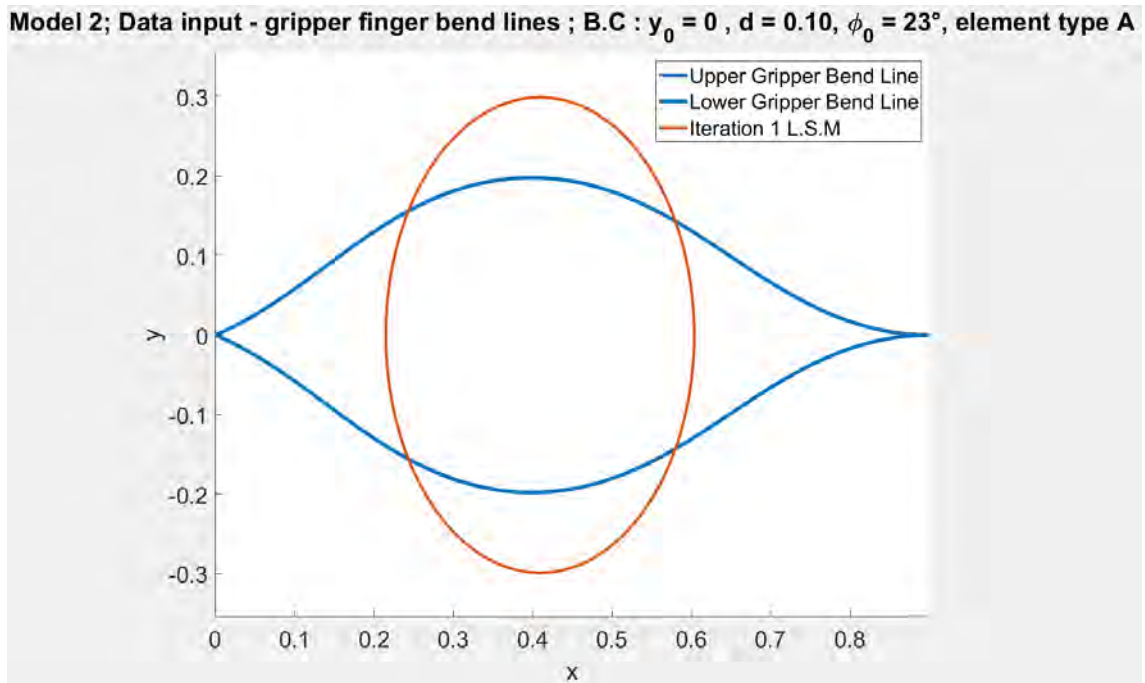


Fig. 4.8: Ellipse plot – first least squares iteration without filtering of the semi-major axis

4.3.4 New tasks to solve

As shown in figure 4.7, sometimes the resulting elliptical object from the first least squares iteration does not completely fulfill an appropriate fit. The reason is that even though a reduced range is used (depending on the boundary conditions), so that the best first approximation is retrieved, the least squares method minimizes the error between each given point and the ellipse. Furthermore, for the first least squares iteration, only a section of the curve is given as data input, consequently, one iteration is not sufficient.

By using the least squares approximation, it does not mean that every point from the data will belong to the resulting ellipse, but the sum of the square distance between each point $P(x,y)$ of the ellipse and the data input is minimized. By geometric fitting, the errors are defined by the orthogonal distances from the given points to the geometric feature to be fitted. A sketch regarding this issue is presented in figure 4.9. The nearest point (X'_i, Y'_i) satisfies the orthogonal contacting conditions and lies in the same quadrant of the standard position of the given point

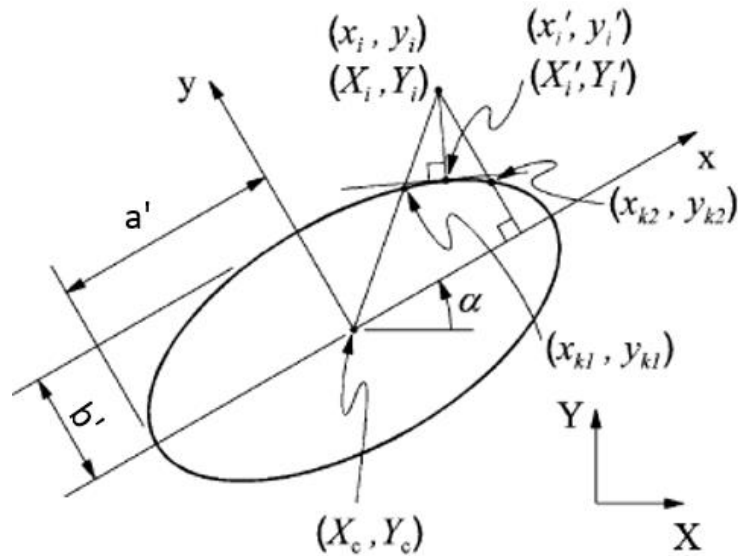


Fig. 4.9: Ellipse centered at (X_c, Y_c) , axis lengths a' , b' , angle α [38]

With this condition, some of the points may form part of the ellipse, others do not; thus, additional iterations using a new data input (from now on call *data_iter2*) must be carried out. The variable *data_iter2* consists of points $P(x_e, y_e)$ from the ellipse generated in iteration 1, and points $P(x_g, y_g)$ from the gripper bend lines. In order to define which data points are going to be taken, some previous tasks must be executed. Hereafter, an overview of them is listed.

- Task 1 - Location of the points $P_{g_start_located}$ and $P_{g_end_located}$: as displayed in figure 4.10, these points will be defined by the start $P_{el_start}(x_e, y_e)$ and end $P_{el_end}(x_e, y_e)$ points of the ellipse obtained in the first least squares approach.
- Task 2 - Alignment: because each curve (ellipse and gripper bend lines) have different step sizes for the x_i values, misalignment is foreseeable; therefore, a match of the x_i values for every point $P_i(x, y)$ of the ellipse and the located gripper bend lines (figure 4.10) is necessary so that the correct subtraction of the y_i components can be performed.
- Task 3 – Calculation of the variable *differenceY*: the difference between the y_i components will be defined by the following expression $difference = abs(y_g) - abs(y_e)$.

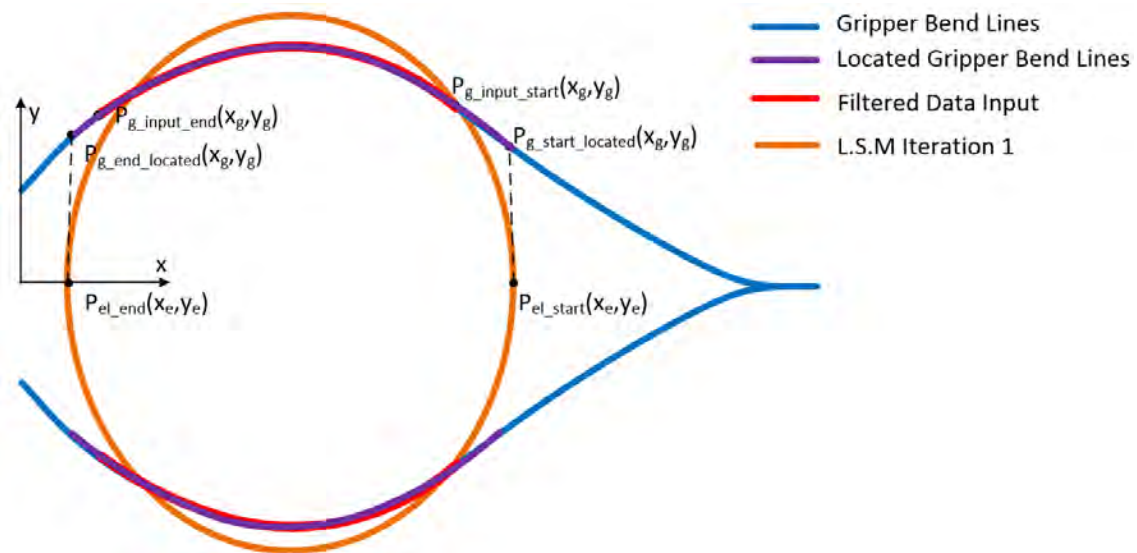


Fig. 4.10: Location process – finding the points $P_{g_start_located}$ and $P_{g_end_located}$

4.4 Module 3 : Task 1 – Location process

As shown in figure 4.7, after the first least squares iteration, the ellipse lies outside the gripper bend lines; which is practically not a valid solution; therefore, a second iteration is required. Before the new iteration takes place, the next step is to localise the points $P_{g_start_located}$ and $P_{g_end_located}$, which belong to the gripper bend lines. These points are necessary for defining the extreme x points for the new iteration range. The way of finding these two points is by matching the x coordinates of the points of the ellipse $P_{el_start}(x_e, y_e)$ and $P_{el_end}(x_e, y_e)$, which are generated from the least squares method, with those from the gripper $P_{g_start_located}$ and $P_{g_end_located}$ respectively. After finding $P_{g_start_located}$, $P_{g_end_located}$, it is guaranteed that the start and end points are almost the same for the gripper and for the ellipse under a tolerance of 10^{-4} . The range defined by both points for each gripper bend line is plotted in purple in figure 4.10. In this work, that is the meaning of location. Through this process, the range where the difference of the y_i components takes place is defined. The next concern is to match the values of the x_i components for every point in the purple range (figure 4.10), because each matrix (gripper and ellipse) have different step sizes; consequently the difference between the x_i values of both curves becomes larger with every step size. For overcoming this problem, a filter condition is implemented by introducing a new variable called *differenceX*.

4.5 Module 4: Creation of the new geometry

4.5.1 Task 2 – Alignment of x_i values

The next step consist in aligning the points $P_i(x,y)$ contained in the purple range (figure 4.11) for every step size Δx between the gripper and the ellipse. For example, if the element number tenth of each column vector x_{10} is extracted and compared, what would be found is that the value of x_{10} of the gripper does not coincide with that from the ellipse, thus an if-statement should be implemented. To do so, the variable *differenceX* is defined as the absolute difference of $x_{gi} - x_{ei}$, where x_{gi} is a generic point of the gripper bend line and x_{ei} from the ellipse. The variable *differenceX* is utilised in the following logic statement, which is implemented as a code in the algorithm:

“While the value of *differenceX* is bigger than 10^{-4} , then *differenceX* will be recalculated using the x_{gi+1} term until *differenceX* fulfill the established condition. When this condition is achieved, the loop will break and proceed with the calculation of *differenceY*”. [L: 238-261]

4.5.2 Task 3 - calculation of the variable *differenceY*

The variable *differenceY* is a row vector containing the difference of the y_i components for all the points $P_i(x,y)$ between the range defined by $P_{g_start_located}$ and $P_{g_end_located}$. The subtraction is made for every point $P_i(x, y)$ inside the range determined by $P_{g_start_located}$ and $P_{g_end_located}$.

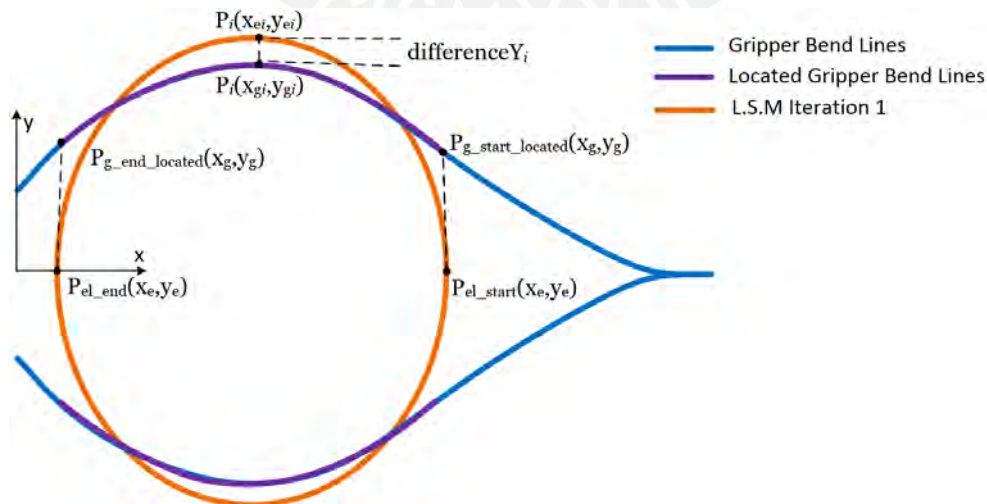


Fig. 4.11: Calculation of the row vector $differenceY = abs(y_g) - abs(y_e)$

The contours of the gripper bend lines in purple (figure 4.11) are used as boundaries. Any point of the ellipse generated by the least squares first iteration lying outside of the gripper bend lines is replaced by the correspondent point of the contour at the same x_i value. The other points, which are inside the blue contours, remain unaffected. The result is a close curved shape, which is utilized for the second least squares iteration.

4.5.3 Selection of the new y_i component for the new geometry

In order to select the desired new value of y_i , the following logical statement has been implemented:

“When $differenceY > 0$, then the new value of y_i will be the y_{ei} (ellipse), on any other case the new value of y_i will be y_{gi} (gripper).” [L: 265-272]

This logical statement sets the components of the variable $data_iter2$. Hereafter, the geometrical representation of this variable is plotted in red colour:

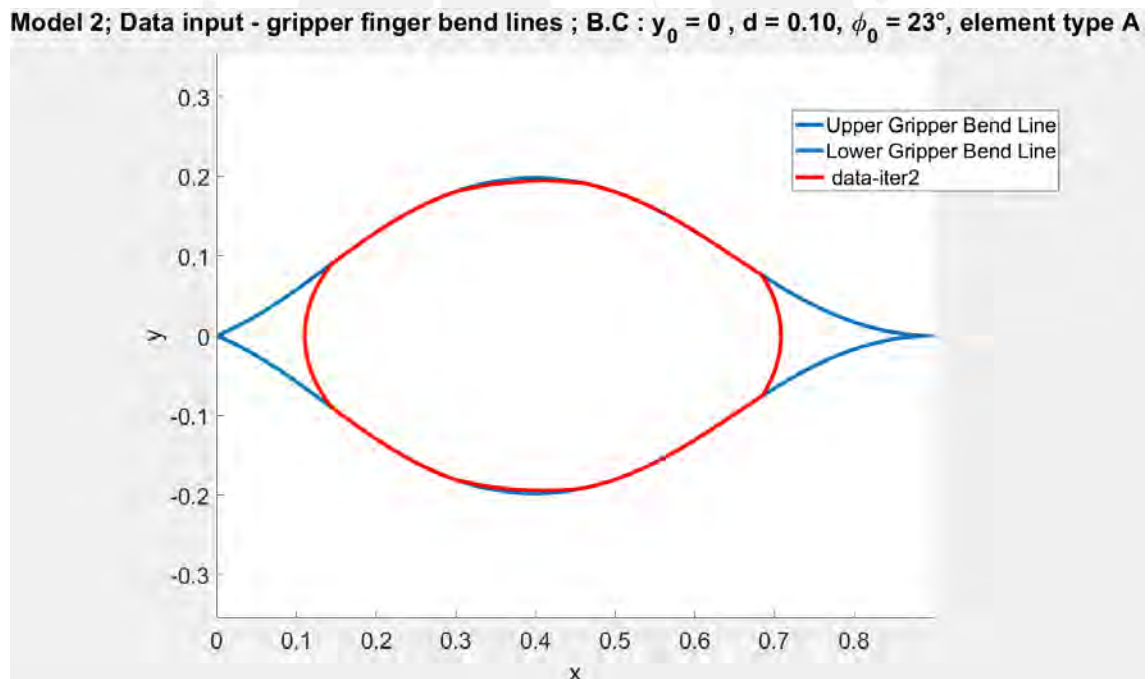


Fig. 4.12: Geometrical representation of $data_iter2$ for the next least squares iteration

4.5.4 Variable *data_iter2* and evaluation of the ellipse major – axis

After having filtered the y_i values for each x_i , all the new data is stored in a 2×2 matrix called *data_iter2*, whose values are introduced into the *ellipse_fit* algorithm. In contrast to the first least squares iteration, where just the upper and lower contour were known, for the second least squares iteration a close curved shape is available. As displayed in figure 4.12, the fact that this curved shape has more points distributed closing a perimeter, will result in one-step closer to fit the ellipse into the gripper bend lines. As well as in the first least squares iteration, the program must recognise the semi-major axis; therefore, an *if*-statement, which depends on the value of the stroke length d is implemented before the plotting.

Until now, all the processes described represent one single iteration of the algorithm *Elliptical Objects Fitting*. As displayed in figure 4.13, most of the times a second least squares iteration is also not enough to fulfill the fit of the ellipse into the gripper bend lines; therefore, the number of iterations should be increased, consequently, all the steps performed before should be done again until an appropriate exit-loop condition, depending on the case, is achieved. These exit-loop conditions should be settled so that at the iteration i , after m least squares approximations, the ellipse fits into the gripper bend lines with an optimized contact length.

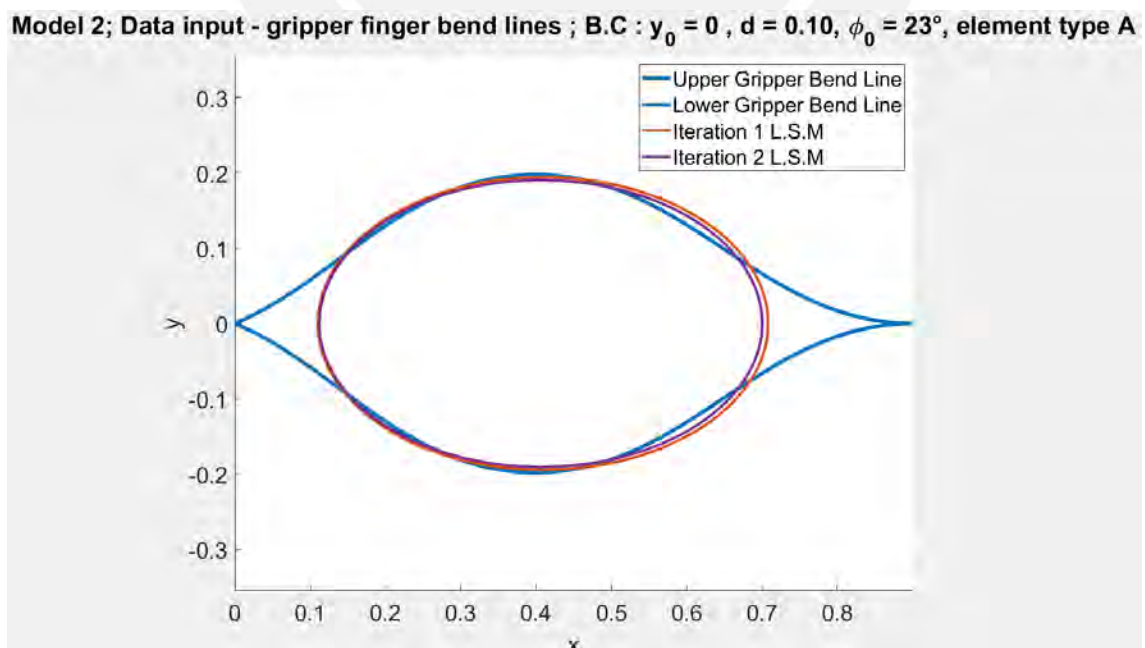


Fig. 4.13: Plots of the elliptical objects for the first and second least squares iterations

4.6 Exit-loop conditions

The exit-loop conditions have a direct relationship to the boundary conditions stroke length d and M_0 . By reading both, the program filters a huge group of elliptical objects, whose range is the most suitable for the fit. The information of the boundary conditions, stroke length d , plus the values *Logical_differenzY* and *sum_logicaldifferenzY*, which count and sum the number of points fulfilling the condition $abs(differenzY) \leq 10^{-4}$ respectively, are enough to define an exit-loop condition. For the filtered group of solutions, a minimum value of the variable *sum_logicaldifferenzY* is required so that the least squares fit is efficient. These minimum values were settled by testing the algorithm under different boundary conditions and stroke lengths. Hereafter, an example of an exit-loop condition is presented:

Exit-loop condition 1

```
if stroke length  $d \geq 0.25$  &&  $M_0 \leq 10^{-4}$  && sum_logicaldifferenzY  $\geq 35$ 
```

In this case, for the exit-loop condition 1: “If the stroke length d is greater than 0.25 and the moment at the left end of the gripper finger is 0 (model 1) and if the elements fulfilling the condition $abs(differenzY) \leq 10^{-4}$ are equal or greater as 35, then the loop is broken and the solution is plotted (fitted ellipse)”.

In addition, inside each condition, there is a flag, which will take the value of one if the logical values in one exit-loop condition are true at the same time. Consequently, the buckle of the algorithm will be broken conceding the plot of the solution. In any other case, the least squares variables and/or the variables for the initialization of a new range are restarted. It is important to remember that the algorithm can fulfill one of the exit-loop conditions by two different methods. One method is by keeping the range constant and repeating n times the least squares procedure creating a new geometry for each iteration [L:187-305]. The other one consist in the variation of the initial range in the algorithm *Elliptical Objects Fitting* and use this new range to start the following n least squares iteration until one of the exit-loop conditions is fulfilled.[L:67-179]

4.7 Resetting the variables

When the conditions in one iteration with n least squares approximations are not yet accomplished by either of the two methods described before, some variables of the process containing data must be reset. By doing so, the new values of the variables will not be biased by those calculated in the previous iterations.

4.7.1 Resetting least squares method iteration variables

These variables are written in lines 339-342 and are the following:

- `xgripper_totalmatrix = x_new_inputdata1`
- `ygripper_totalmatrix = y_new_inputdata1`
- `x_new_inputdata1 = 0`
- `y_new_up_inputdata = 0`

4.7.2 Resetting range variables

The variables to reset are written in lines 349-353 and are the following:

- `gripperup_new_input_x = zeros (1,counter+1)`
- `gripperup_new_input_y = zeros (1,counter+1)`
- `gripperdown_new_input_x = zeros (1,counter+1)`
- `gripperdown_new_input_y = zeros (1,counter+1)`
- `counter = 0`

In this particular restart of the variables, the matrices are being filled up with zeros until the position `counter+1`. All the values saved in the previous iteration, up to the position `counter+1` are overwritten with zeros so that in the next iteration, when varying the range, they are withdrawn. Finally, the values, which were not overwritten with zeros are overwritten with the correspondent range values in the current iteration. The result is a filtered range, without zeros. The range values are saved in the following variables:

- `gripperup_new_input_y1`
- `gripperup_new_input_x1`
- `gripperdown_new_input_y1`
- `gripperdown_new_input_x1`

4.8 Plotting the solutions

After the fulfillment of one of the exit-loop conditions, which comprises one or more boundary conditions plus logical variables, the plot of the solution is obtained. In the algorithm, the final solutions are saved in the following variables: *Ellipsematrix_iter* and *Ellipsematrix_iter2*.

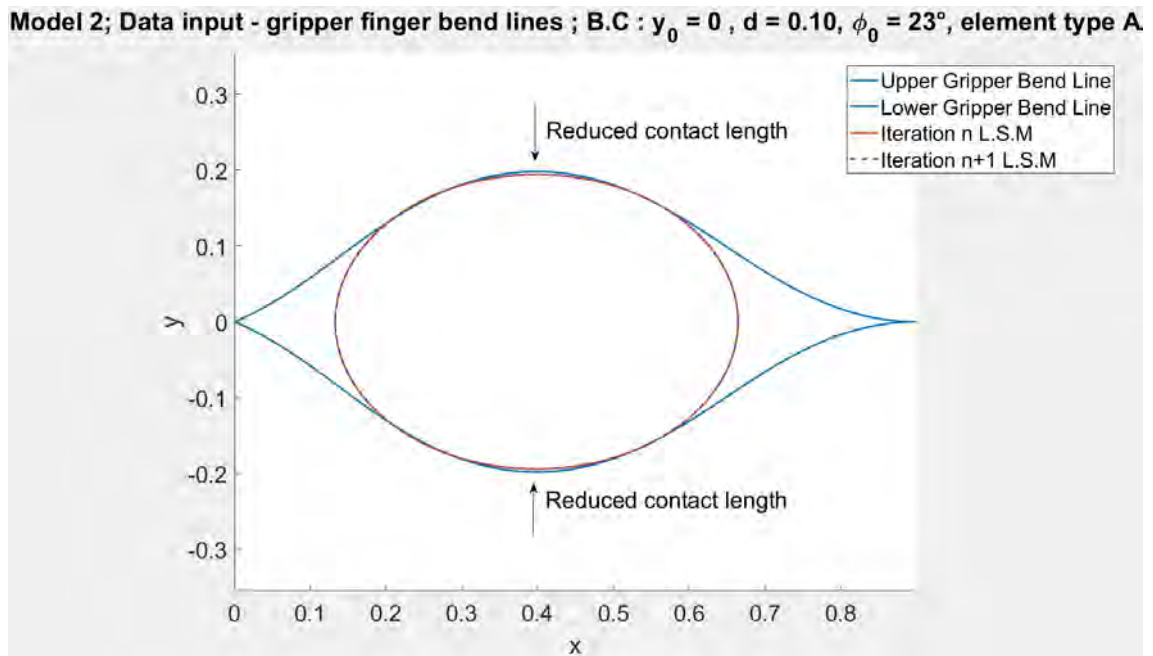


Fig. 4.14: Iteration $i = 1:882$, $n = 5$

In figure 4.14, for each i value, ten least squares iterations were performed. Note that the least squares iterations n and $n+1$ coincide; therefore, one overlap the other. If the number of least squares iterations is reduced, then more iterations, which vary the range as a function of the variable i are performed. For this particular case, it is displayed in figure 4.15 that a larger contact length was obtained for $i = 1:1733$, when $n = 1$. Therefore, it is not recommendable to set a high value for n because the contact length may be diminished by the number of

least squares iteration. In section 4.9, a case where the contact length reduction is more perceptible will be introduced and explained.

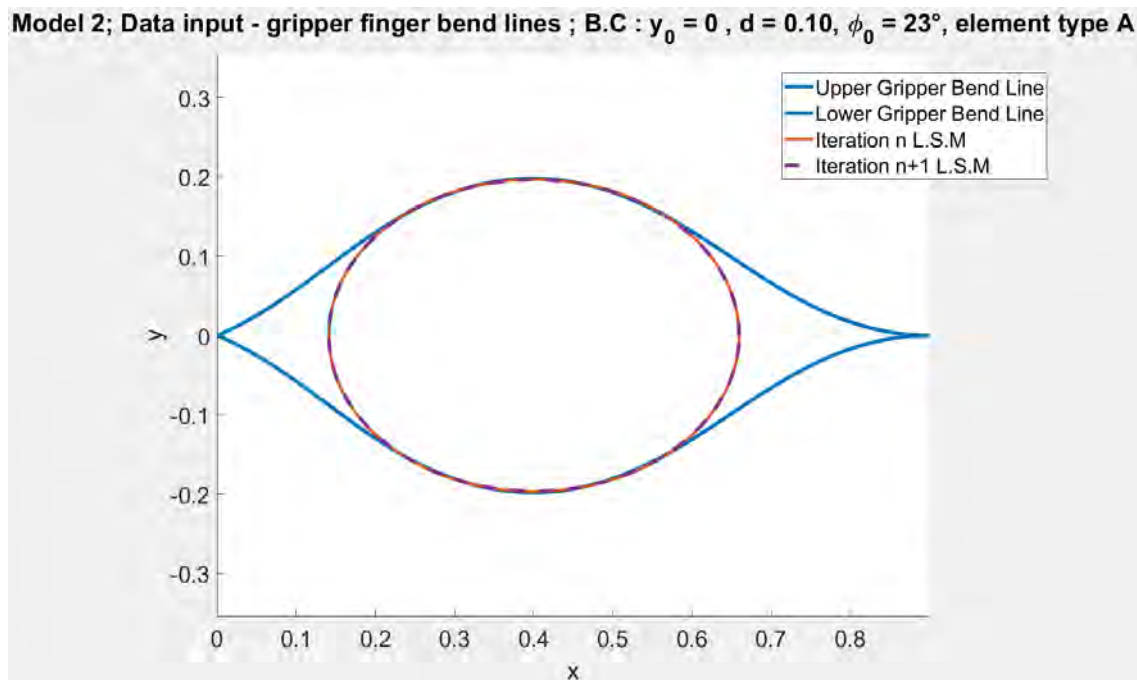


Fig. 4.15: Iteration $i = 1:1733$, $n = 1$

Even though in most cases, a value of $n = 5$ is enough, there are special ones which may require values of n up to 10, sometimes 20 in order to obtain a perfect fit. For these special cases, a value of $n = 5$ does not mean that the ellipse will be completely outside of the gripper bend lines, but that some short segments may not be yet inside. Despite this, an efficient but not perfect fit is obtained.

4.9 Operation modes

Even though there are three different operation modes, which can be employed to fit the ellipse into the gripper bend lines, they do not work separately and rely on each other for a full automatized and efficient fit. Moreover, the three of them can be integrated to work simultaneously by setting properly the counters i and n . On the one hand, the counter i in a for-statement written as an integer number greater than zero, defines a range $[ii, jj]$. On the other hand, i written as a range in a for - statement defines the number of iterations as well as a range variation per loop in the algorithm. Regarding the variable n , which is also an integer number greater than zero, a relationship with the number of least squares iterations for each i value is established. The relationship is the following: when

$n = m$, then $2m$ least squares iterations are performed. After defining the main variables to control the core algorithm, it is important to emphasize that data filtering is an important issue to take care about; therefore, the data input is previously filtered from the main matrices, which contain all the information of the gripper bend lines: *Gripperdownmatrix0* and *Gripperupmatrix0*. The filtration is done by analyzing the boundary conditions and the stroke length d , so that the major group of the possible elliptical shapes is captured. Afterwards, the ranges are separated in two, one for the upper gripper bend line and another for the lower gripper bend line. Both ranges are a function of the variable i . Hereafter, the description of the three operation modes is described.

4.9.1 Operation mode 1: range manipulation

The procedure to operate the program under this mode is the following:

- Define boundary conditions and stroke length d
- Assign a value to the counter i , which defines a certain range $[i, j]$ on each gripper bend line. The algorithm according to the given boundary conditions will filter this range.
- *Elliptical Objects Fitting* algorithm performs two least squares iterations
- Visual analysis of the obtained fit by the user is required.
- The value of i can be incremented or diminished depending on the previous results. Large variations of i are recommended to obtain a significant change on the final fit $[i + 200, i + 400, i + k]$. The value of i can be varied from one up to the number of elements of the matrix *Gripperupmatrix0*.

For each iteration, the range remains constant and two least squares iterations are carried out. It is important to highlight that a bigger number of i does not lead to a better solution, but leads to a variation of the filtered range. This range is employed as data input for the first least squares iteration. The main program will calculate the best possible fit with the filtered range, which is a function of i and of the boundary conditions. Under this operation mode, the exit-loop conditions

do not have any role because just one iteration takes place in the algorithm. Advantages with respect to this operation mode are:

- In average, the computing time lasts approximately 1 second.
- Even though sometimes the object may not be completely fit, fast corrections can be done after visual inspection by changing the value of the counter i as recommended.

As shown in figure 4.5 and 4.6, regarding model 1, for some stroke lengths d , it is faster and easier to retrieve the elliptical object that best fits into the gripper bend lines even after a first least squares iteration. The fact that the least squares method is very sensible to the variation of the data input opens the possibility to operate the algorithm by manipulating the range. Nonetheless, in figure 4.7 regarding model 2, where the angle ϕ_0 is one of the parameters to define as a boundary condition, even a second iteration of the least squares is not enough to accomplish a fit inside the gripper bend lines; therefore, further data processing is required.

4.9.2 Operation mode 2: range constant – n least squares iterations

In this approach, the filtered data input remains constant and just a section of the program, which is in charge of the number of iterations for the least squares method works until a final exit-loop condition is obtained, or until the number of least squares iterations are completed [L:187-338]. Further least squares iterations end up in a better fit but this method has its limitation, which appears when the same geometry is obtained as a feed back to the next least squares iteration. As a result, the ellipse lies a bit outside of the gripper bend lines. Another disadvantage is that a reduced contact length can be generated. Therefore, increasing the number of iterations of the least squares method should not be the only parameter to rely on. The procedure to operate the program under this mode is the following:

- Define boundary conditions and stroke length d .
- Assign a value to the counter i , which defines a certain range $[i, j]$ on each gripper bend line. The algorithm, according to the given boundary conditions, filters the range.

- The algorithm performs $2n$ least squares iterations for the given range, where n is an integer number greater than zero.
- Visual analysis of the obtained fit by the user.
- Increment the number of least squares iteration if required.[L:187]

Even though a bigger value for n will require more computing time, under this operation mode it is not significant. In addition, for any n value, $2n$ least squares iterations are performed. It is important to highlight that if one of the exit-loop conditions is fulfilled, the least squares iterations are stopped. Advantages with respect to this operation mode are:

- The calculation time lasts just 1-2 second, depending on the value of n .
- A fit inside the gripper bend lines is always possible.

Even though it is foreseen that the higher the number of least squares iterations, the better is the fit into the gripper bend lines, a reduced contact length might be obtained. For this reason, the number of least squares iterations should not be excessively high, but conservative. Hereafter, an example is presented where the contact length is reduced due to the elevated number of least squares iterations.

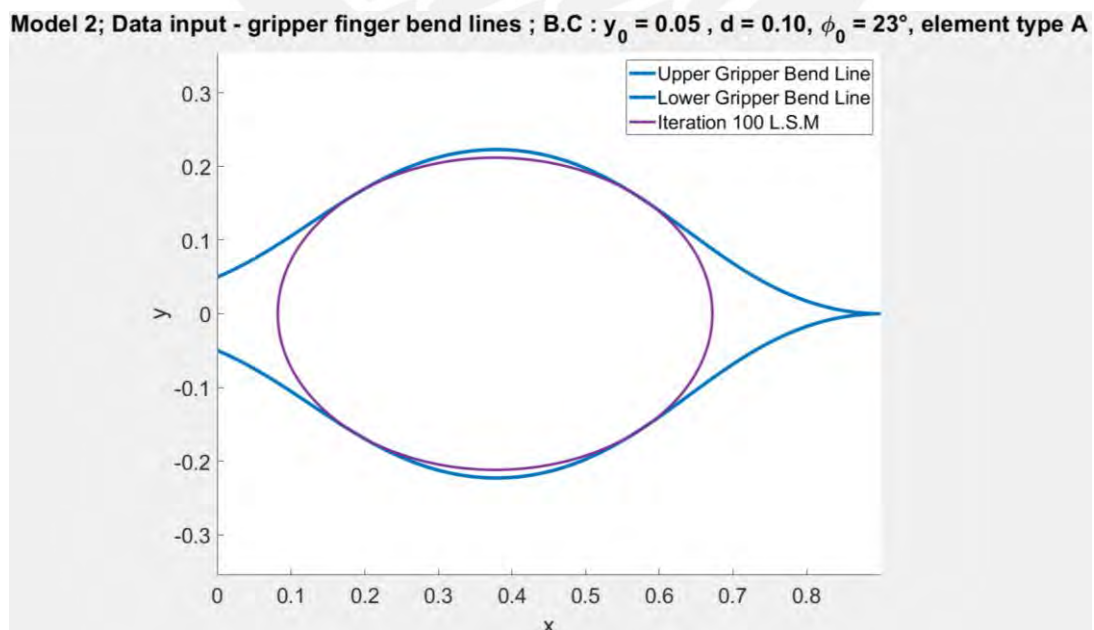


Fig. 4.16: Operation mode 2 – range constant, $i = 1$, $n = 50$

For each value of n , $2n$ least squares iterations are performed. Therefore, the purple plot in figure 4.16 represents one hundred least squares iteration. Even though the elliptical object fits properly into the gripper bend lines, the contact length is reduced.

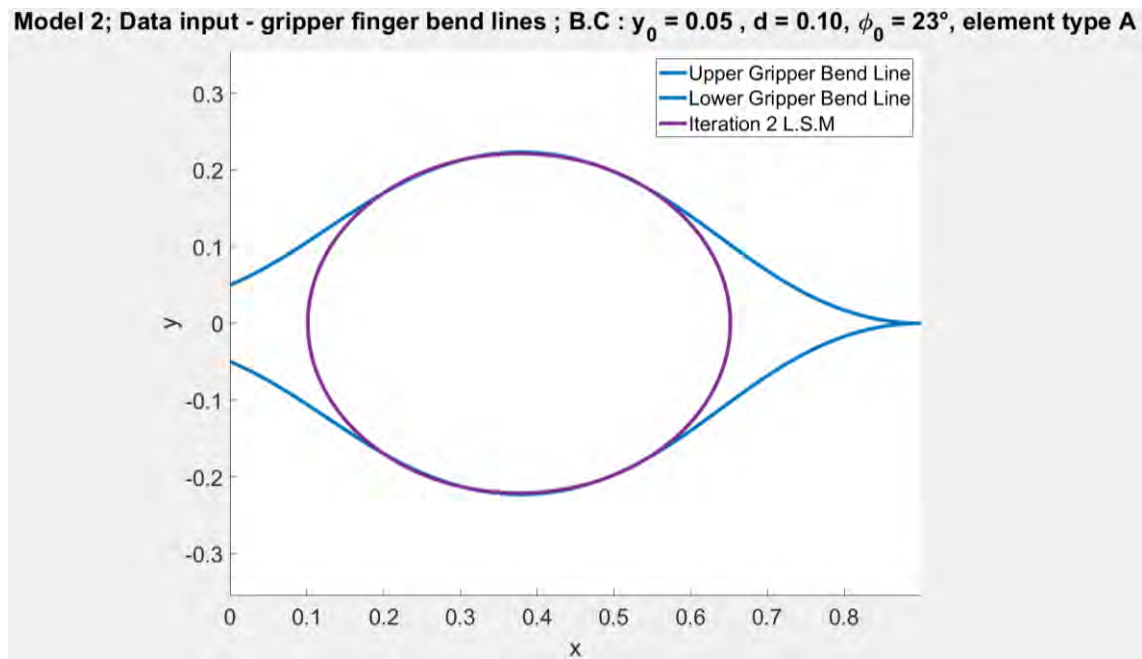


Fig. 4.17: Operation mode 3 – range variable-, $i = 1:992$ (992 iterations);, $n = 1$

The main weakness by working under this operation mode is that the final grip condition can be biased by the high number of least squares iterations; thus, range manipulation and the number of the least squares iterations should work together to obtain an optimized contact length between the round-elliptical object and the gripper bend lines. In figures 4.16 and 4.17, two solutions under the same boundary conditions are plotted. The main difference is that for the former, the range is constant and the number of least squares iteration is high (100 iterations). For the latter, a range variation is included and the number of least squares iterations is low (two iterations per each value of i). The difference between the contact lengths can be inferred from both figures. After comparing them, the longest contact length is obtained in figure 4.17, where the range varies as a function of i , if one of the exit-loop conditions has not been yet satisfied by applying two least squares iterations.

4.9.3 Operation mode 3: range manipulation and n least squares iterations

Under this approach, both mechanism work together to find out the best elliptical fit under diverse boundary conditions. The procedure to operate the program in this mode is the following:

- In order to set the number of iterations for the core algorithm, the variable i is varied inside a range whose start and end values are 1 and the length of the matrix *Gripperupmatrix0* respectively. This statement is written in matlab code as follows: `for i = 1:length(Gripperupmatrix0) [L: 67]` . For each particular value of i , a certain range $[i, j]$ is defined for the lower and upper gripper bend lines. If none of the exit-loop conditions is accomplished, the range varies as function of i per each iteration ($i, i + 1, i + 2 \dots i, \text{length}(\text{Gripperupmatrix0})$).
- Define the number of least squares iterations, which is a function of the variable n . Conservative numbers are recommended to obtain an optimized fit into the gripper bend lines. In most cases a value of $n = 5$ is high enough.
- One of the exit-loop conditions is achieved through the variation of the range, which is controlled by the value of i , or through the $2n$ least squares iterations for each value of i . It is inferred that, the higher the number of least squares iterations per i value (which defines a range), the lower value of i (number of iterations of the whole algorithm).

If a perfect fit is desired, assign the value of $i = \textit{Iteration}$ in the for-loop of line 67. *Iteration* is a variable that contains the iteration number at which the algorithm stopped itself automatically due to the fulfillment of one of the exit-loop conditions. For this particular value of i that defines a constant range $[k, l]$, the value of n can be incremented. Nevertheless, the user must discard if a biased result is obtained by incrementing the value of n . As explained before, this phenomenon may occur for high values of n . While performing these steps, operation mode 2 is being used.

Chapter 5

Investigations of square-rectangular and triangular objects

5.1 Enhanced segments

For square, rectangular and triangular objects, the gripper bend lines have been investigated implementing the element D as gripper finger because of its stability and manoeuvrability. This element (grripper finger) has segments whose stiffness has been enhanced, because it is required that they remain straight for achieving an appropriate fit (optimized contact length), if possible, by holding each face of the object. From the non-linear theory presented in chapter 2, replacing equation (7) in (4) and adding the reinforcement factor:

$$\frac{d\theta_3}{ds} = \frac{M_z}{EI_z 0.5(1 + \text{factor}(1) - 1)(\text{sign}(s - b_i) - \text{sign}(s - c_i))} \quad (27)$$

where

$b_i: [b_1 \ b_2 \dots b_n]$

$c_i: [c_1 \ c_2 \dots c_n]$

factor: $[z_1 \ z_2 \dots z_n]$

$f(x): \text{sign}(x)$

The equation (27) has been implemented in matlab code. The function $\text{sign}(x)$ returns the following values: -1, 0, and 1, when $x < 0$, $x = 0$, $x > 0$ respectively. Additionally, the length and number of the reinforced segments are determined by the vectors b and c as follows:

$$\text{Reinforced segment length} = c_n - b_n \quad (28)$$

where

n : number of reinforced segments

In this work, gripper fingers owning three enhanced segments have been investigated for finding suitable bend lines for square-rectangular and triangular objects; nevertheless, it is also possible to increment the number of reinforced segments by adding more components in the row vectors b and c . Finally, the magnitude of the enhancement is determined by the components of the vector

factor. Each segment is z_n times enhanced, where z_n is a real value equal or greater than one. Hereafter, gripper geometries for gripping square-rectangular objects are presented.

5.2 Simulations: square-rectangular objects

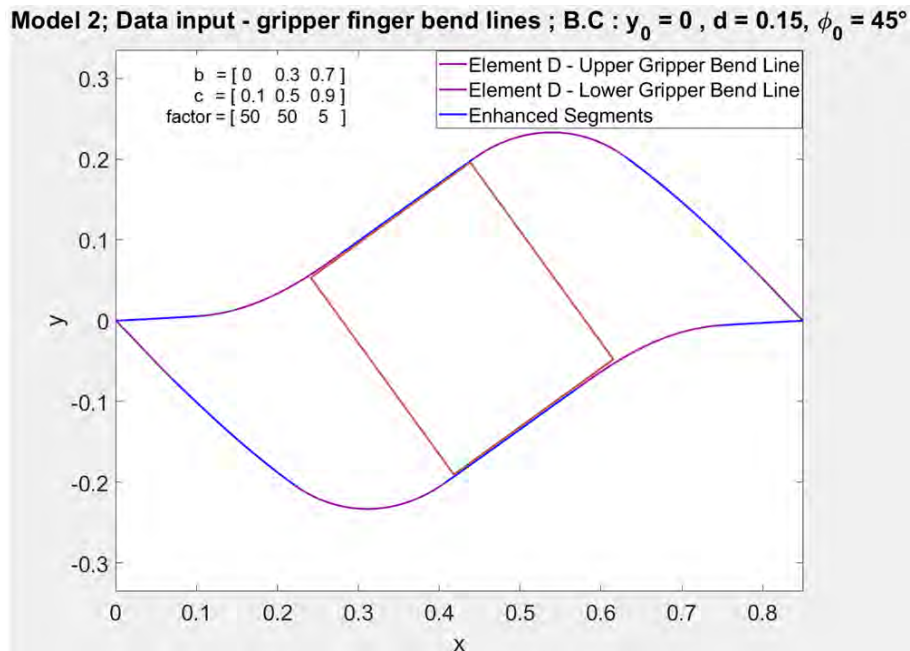


Fig. 5.1: Solution 1 for square and rectangular objects – inverted b.c

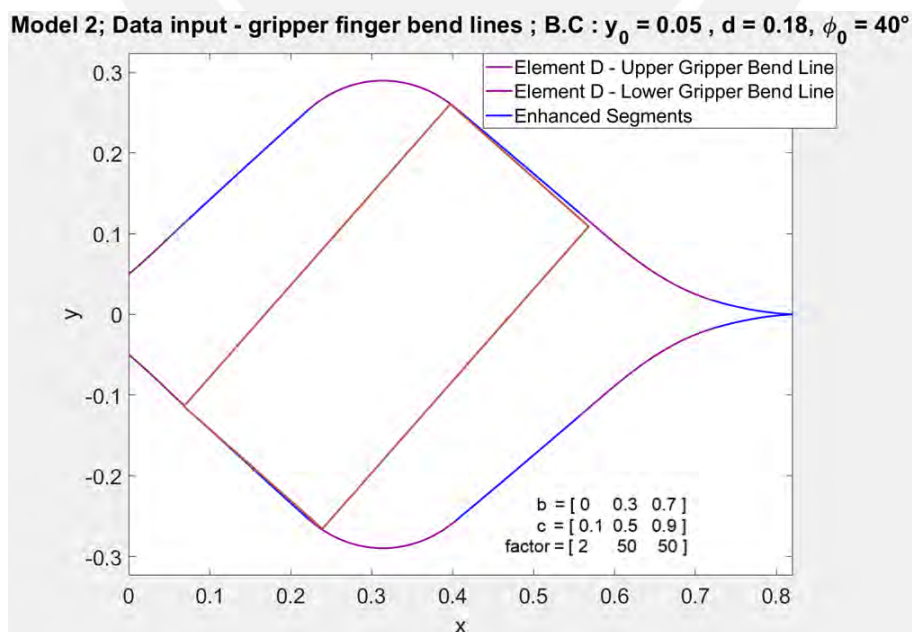


Fig. 5.2: Solution 2 for square and rectangular objects – symmetric b.c

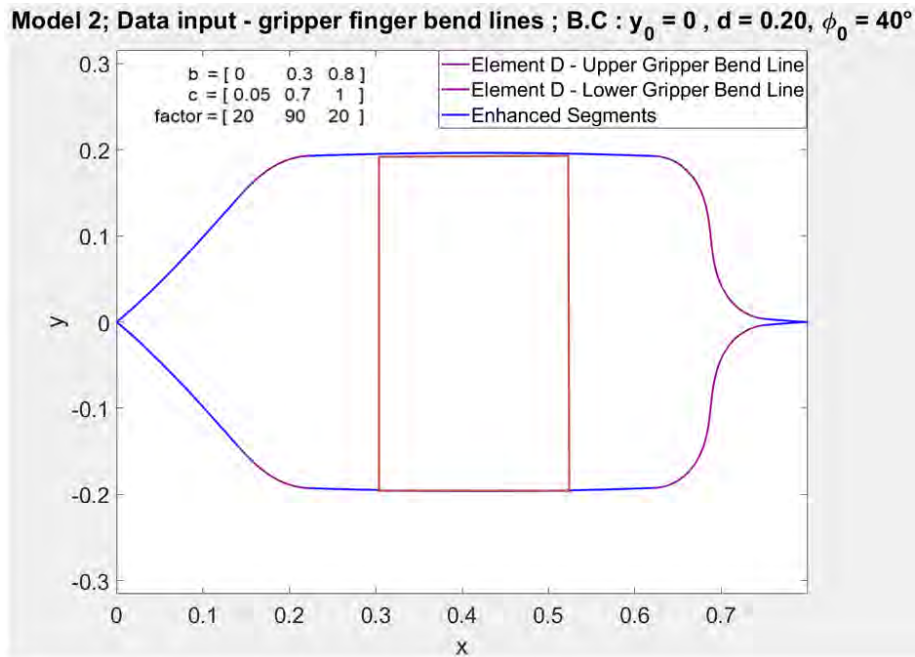


Fig. 5.3: Solution 3 for square and rectangular objects – parallel grip

On the one hand, the gripper bend lines in figures 5.1, 5.2, 5.3 are calculated using the algorithm *dgl_rod_clamped_pinned_diff_bend_stiff*. On the other hand, the square, rectangular objects are not automatically obtained. These objects are included as an example to show how the grip would look like. A possible solution to generate the square-rectangular object that fits the enhanced gripper bend lines consists in the implementation of an algorithm containing the least squares method, data filtering and appropriate exit-loop conditions.

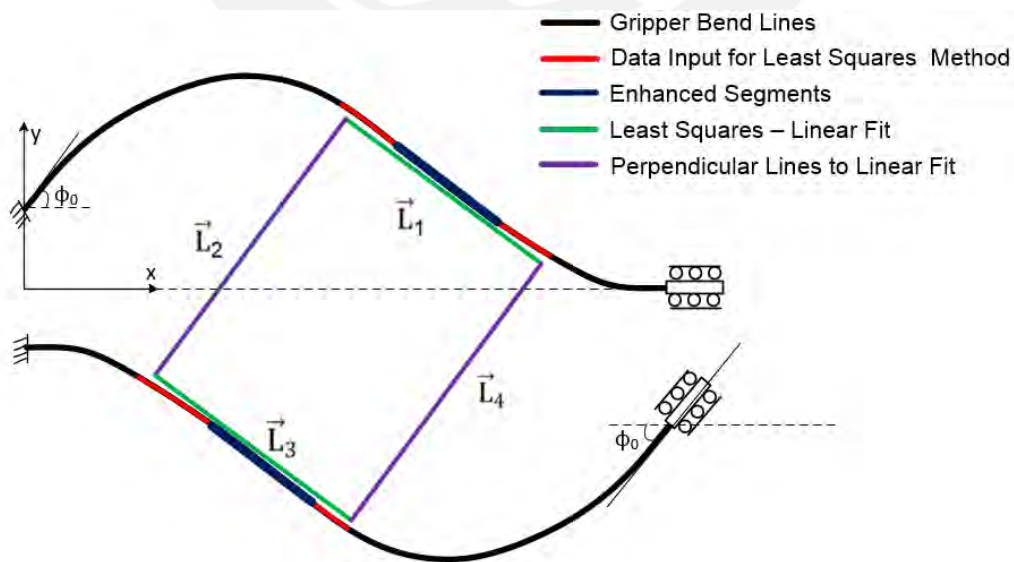


Fig. 5.4: Solution concept - rectangular objects algorithm

Hereafter, an overview of the ideas for the implementation of an algorithm for finding an optimized contact length between square–rectangular objects and the gripper bend lines is described.

5.3 Algorithm overview for fitting square–rectangular objects

The first step consists in the calculation of the primary lines \vec{L}_1 and \vec{L}_3 using as data input the segments in red or the enhanced segments in blue as displayed in figure 5.4.

$$mx_i + b = y_i \quad (29)$$

Writing equation (29) in matrix notation for the i points:

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_i & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix} \quad (30)$$

Simplifying equation (30)

$$[X]\vec{b} = \vec{y} \quad (31)$$

where

$[X]$: $n \times 2$ matrix containing the function $f(x_i)$

\vec{b} : $[m \quad b]^T$, least squares coefficients

\vec{y} : column vector containing the function $f(y_i)$

For example, in figure 5.1, the ranges $[0,3\dots0,45]$ and $[0,4\dots0,55]$ in the x-axis contain all the i points of the enhanced segments in the middle of the gripper finger. These ranges could be used to generate the primary lines \vec{L}_1 and \vec{L}_3 . Furthermore, it is also possible to obtain \vec{L}_1 and \vec{L}_3 by taking a larger range as the one highlighted in red in figure 5.4. On the one hand, applying the least squares method by using only the points of the enhanced segment as data input will turn into an appropriate fit into the gripper bend lines because the slopes of \vec{L}_1 and \vec{L}_3 would be parallel to the enhanced segment of the gripper. Moreover, all the corners will end up inside the gripping zone. Nevertheless, the disadvantage is that the size of the object will be smaller in comparison to the one that could be obtained if the range highlighted in red is taken for the generation of \vec{L}_1 and \vec{L}_3 . On the other hand, taking the points highlighted in red in figure 5.4 as data input also presents two main disadvantages. The former is that the lines \vec{L}_1 and \vec{L}_3

might not be parallel to the straight segments of the gripper, reducing the contact length. The latter is that some of the corners of the square object could end up out of the gripping zone; thus, an iterative process should be implemented until all the corners fit in it. As a conclusion, the range for the least squares linear approach can be defined as a function of the range of the enhanced segments, which is located in the gripping zone (middle zone) for each gripper finger or from a longer range as the one highlighted in red in figure 5.5. Afterwards, the slope of the secondary lines \vec{L}_2 and \vec{L}_4 are obtained because they are orthogonal to \vec{L}_1 and \vec{L}_3 as illustrated in figure 5.5. In addition, \vec{L}_2 and \vec{L}_4 pass through the points at both ends of lines \vec{L}_1 and \vec{L}_3 . With this information, the lines \vec{L}_2 and \vec{L}_4 are defined, together with the square–rectangular object.

Two different approaches are recommended to be investigated for further data processing of the primary lines \vec{L}_1 and \vec{L}_3 . In the first case, if the range taken for the creation of the primary lines \vec{L}_1 and \vec{L}_3 is longer than the range of the enhanced segments, the length of the resulting primary lines \vec{L}_1 and \vec{L}_3 should be reduced if any of the four corners is out of the gripper bend lines.

In the second case, if the range taken for the creation of the primary lines \vec{L}_1 and \vec{L}_3 is equal to the one of the enhanced segments, the length of the resulting primary lines \vec{L}_1 and \vec{L}_3 should be increased until any of the four corners is about to have contact with the gripper bend lines.

In addition, as illustrated in the simulations in figures 5.1, 5.2 and 5.3, different manners of gripping the object can be performed; consequently, additional conditions might be required for each specific case. Hereafter, common conditions for all the cases are mentioned.

Regarding the slopes m_{L1} and m_{L3} , which are retrieved from the least squares method, should have approximately the same value within 0.1% tolerance as the slope of the enhanced segment m_{es} . In addition, the slopes m_{L1} and m_{L3} are intended to be equal for optimization of the contact length as displayed in figure 5.5; nevertheless, for some solutions this is not possible. Fulfilling this condition sets the minimum contact length as the length of the enhanced segment. As mentioned before, it is required that the edges defined by the points $P_{L1}(x_{end}, y_{end}), P_{L1}(x_0, y_0), P_{L3}(x_{end}, y_{end}), P_{L3}(x_0, y_0)$ do not enter in contact with the gripper bend lines. The reason is that it will produce an undesired deformation on the gripper bend lines, reducing the contact length. Therefore, the end and start

points of the primary lines \vec{L}_1, \vec{L}_3 should be inside the gripping zone without coming into contact with the gripper bend lines as illustrated in figure 5.5. The corners of the square–rectangular object in figure 5.5 are represented as red points.

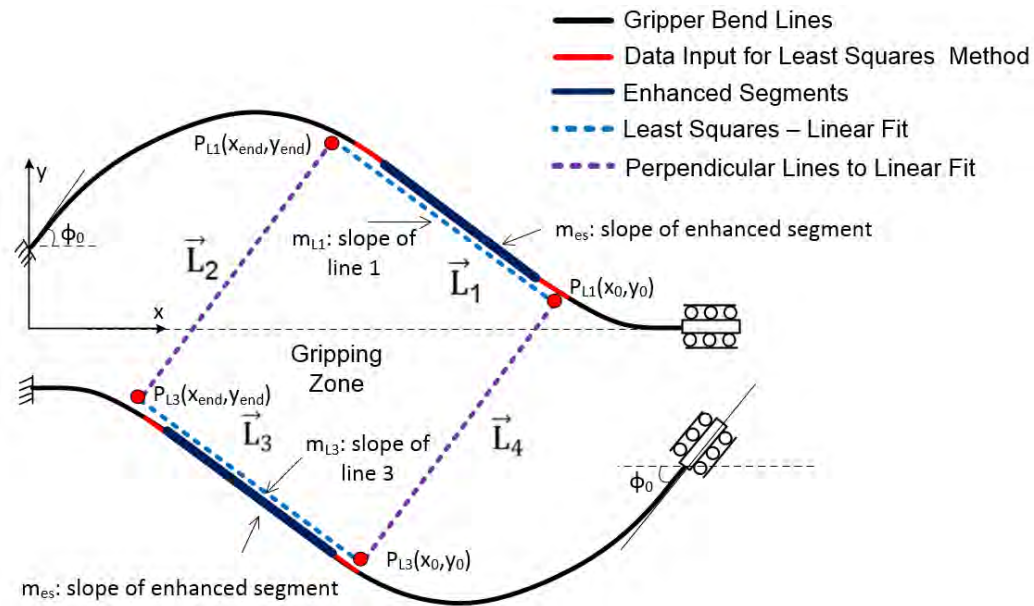


Fig. 5.5: Generation of the lines \vec{L}_1 and \vec{L}_3 by using a larger range (red)

If the data input highlighted in red in figure 5.5 is used for the creation of the lines \vec{L}_1 and \vec{L}_3 , then their length should be defined by the maximum possible distance in which the points $P_{L1}(x_{end}, y_{end}), P_{L1}(x_0, y_0), P_{L3}(x_{end}, y_{end}), P_{L3}(x_0, y_0)$ end up in the gripping zone. A tolerance for the distance between the corners (points above mentioned) and the gripper bend lines (upper and lower) should be implemented so that when one of the them reaches the tolerance, the perpendicular lines \vec{L}_2, \vec{L}_4 are created. The next step is to check if all the corners of the rectangle are inside the gripper bend lines.

As displayed in figure 5.6, if one of the corners is out of the gripping zone, the length of \vec{L}_1 and \vec{L}_3 should be reduced and new perpendiculars \vec{L}_2 and \vec{L}_4 should be constructed. The exit-loop condition to implement would be:

“If all the corners are in the gripping zone, then stop the loop”. The other way around with more details, “for the values of x_{end} or x_0 , if y_{end} or y_0 are higher in absolute value than that of the upper and lower gripper bend lines for the same value of x , then the range for the next iteration should be reduced”.

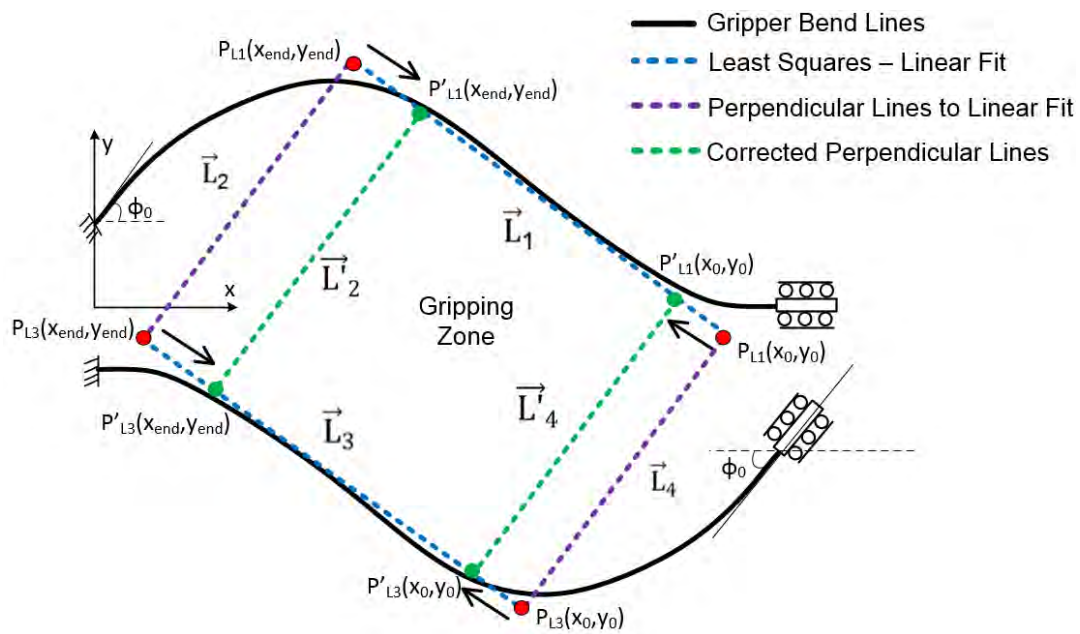


Fig. 5.6: Fitting the four corners of the rectangle inside the gripper bend lines

On the one hand, if an inversion of the boundary conditions is employed as displayed in figure 5.1, if one corner fits inside the gripper bend lines, the other ones will do as well. On the other hand, when using an arrangement as the one plotted in figure 5.2, the verification of each corner should be performed because depending on the enhancement and boundary conditions, the slopes m_{L1} and m_{L3} might not be parallel. The verification process should be done iteratively until all the corners are fitted. Finally, for solutions as the one presented in figure 5.3 (parallel grip), it is assumed that the object is previously aligned. Under this consideration, all the ideas described before are also valid and enough to obtain automatically an optimal square-rectangular object that fits into the gripper bend lines.

As a first approach, in order to get results and analyse the behaviour of the loops involved in the algorithm, the enhanced segments, which are employed for retrieving the slopes of \vec{L}_1 and \vec{L}_3 should be settled constant and should be also utilized to define the length of both lines. As illustrated in figure 5.7, it is foreseeable that the lines \vec{L}_1 and \vec{L}_3 are parallel to the enhanced segment.

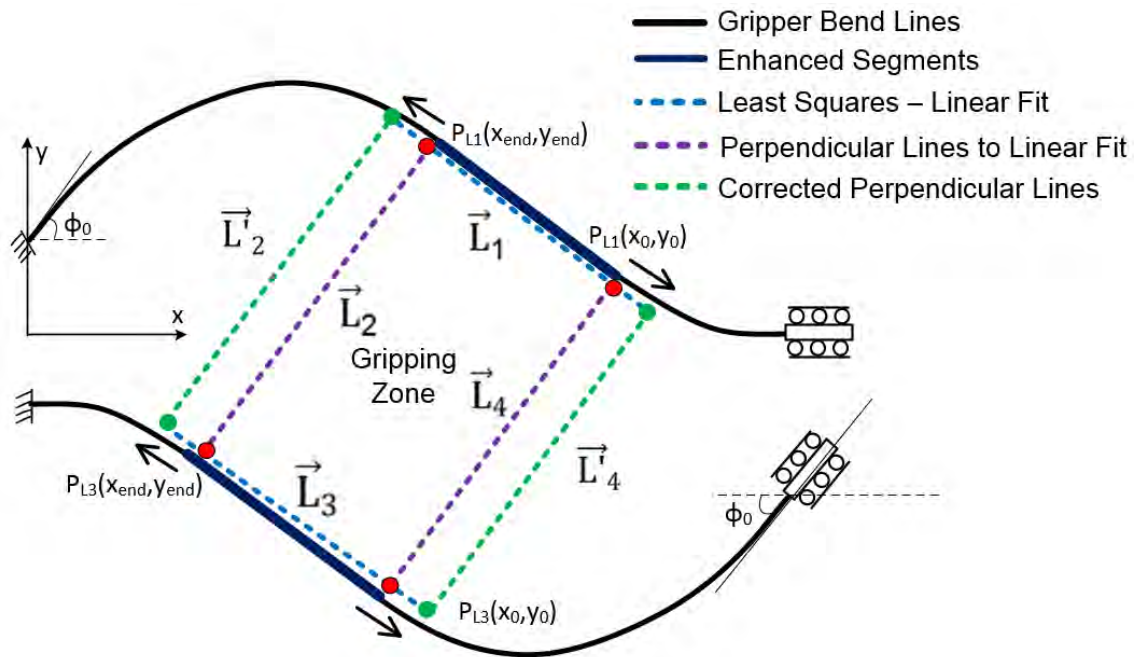


Fig. 5.7: Generation of the lines \vec{L}_1 and \vec{L}_3 by using the enhanced segments

As displayed in figure 5.7, if all of the corners are inside of the gripping zone, the length of \vec{L}_1 and \vec{L}_3 should be incremented and new perpendiculars \vec{L}_2 and \vec{L}_4 should be constructed. The issue to deal with consists in the calculation of the length of lines \vec{L}_1 and \vec{L}_3 . Similarly to the first case explained, an appropriate exit-loop condition should be settled. The exit-loop condition to implement would be:

For the x values x_{end} and x_0 of the rectangular object, “if y_{end} and y_0 are in absolute value, lower than the values of the y components of the upper and lower gripper bend lines for the same x values, then the range should be increased for the next iteration; otherwise, the loop should end”.

5.4 Algorithm overview for fitting triangular objects

Like-minded concepts can be applied for the case of triangular objects. On the one side, for the gripper configuration displayed in figure 5.8, it is desirable that one of the enhanced segments of the upper gripper bend line is parallel to \vec{L}_1 . On the other side, it is suitable that the lower gripper bend line remains straight for retrieving a high contact length with the base of the object defined by \vec{L}_2 .

The triangle in figure 5.8 is comprised of three lines \vec{L}_1 , \vec{L}_2 , \vec{L}_3 . Regarding the slope m_{L1} of \vec{L}_1 , it should be equal to m_{es1} . The slope m_{L2} of the line \vec{L}_2 is zero.

The compliant gripper under load condition presented in figure 5.8 employs two gripper fingers with different enhanced segments; therefore, the bend lines have a different appearance.

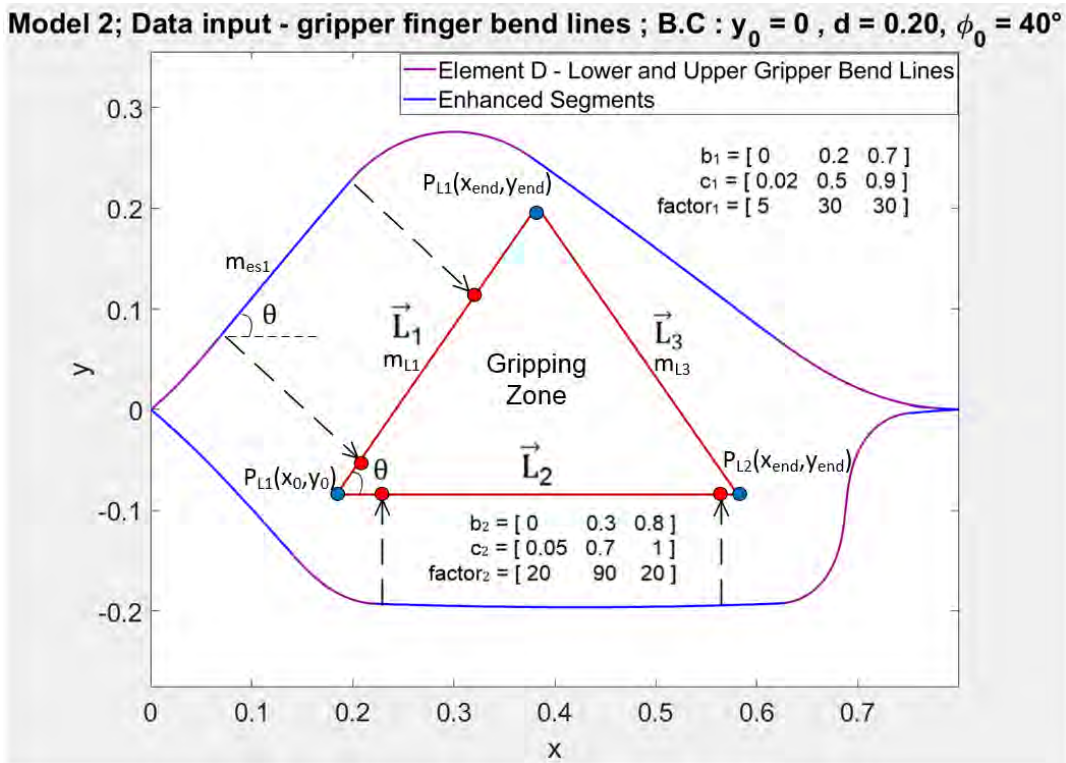


Fig. 5.8: Simulation with 2 different enhanced gripper fingers - Load condition

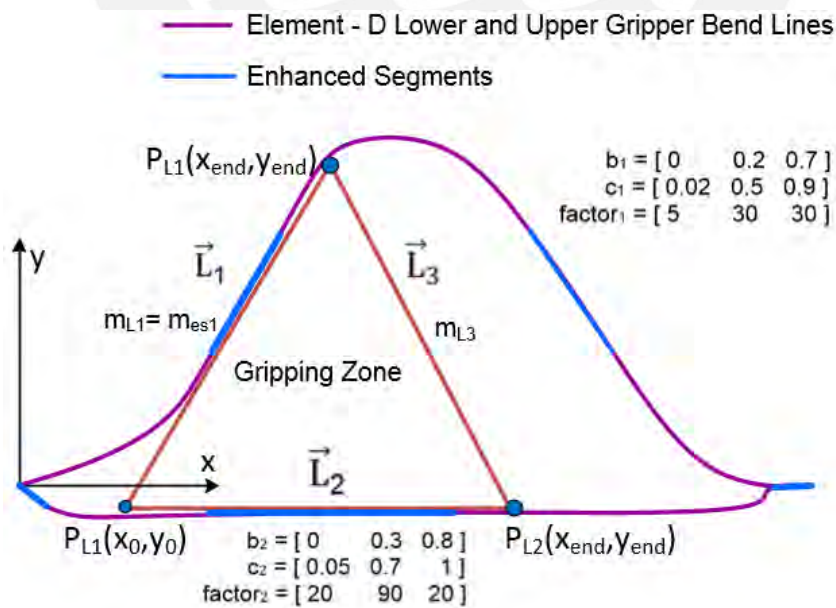


Fig. 5.9: Scheme of solution for triangular objects - Grip condition

The length of \vec{L}_1 and \vec{L}_2 is determined by the points of contact in the grip condition as outlined in figure 5.9. It is important to highlight that figure 5.9 is a scheme of how the grip would look like. Next issue to take care about is to avoid a rollover of the triangular object as displayed in figure 5.10.

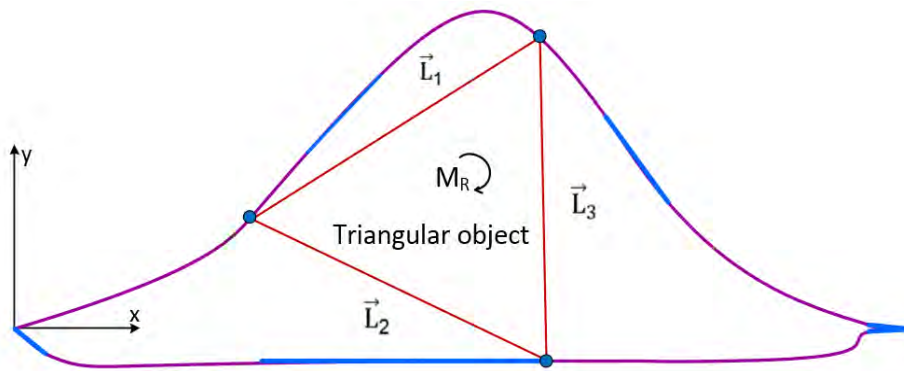


Fig. 5.10: Rollover due to forces of the gripper bend line on the triangular object

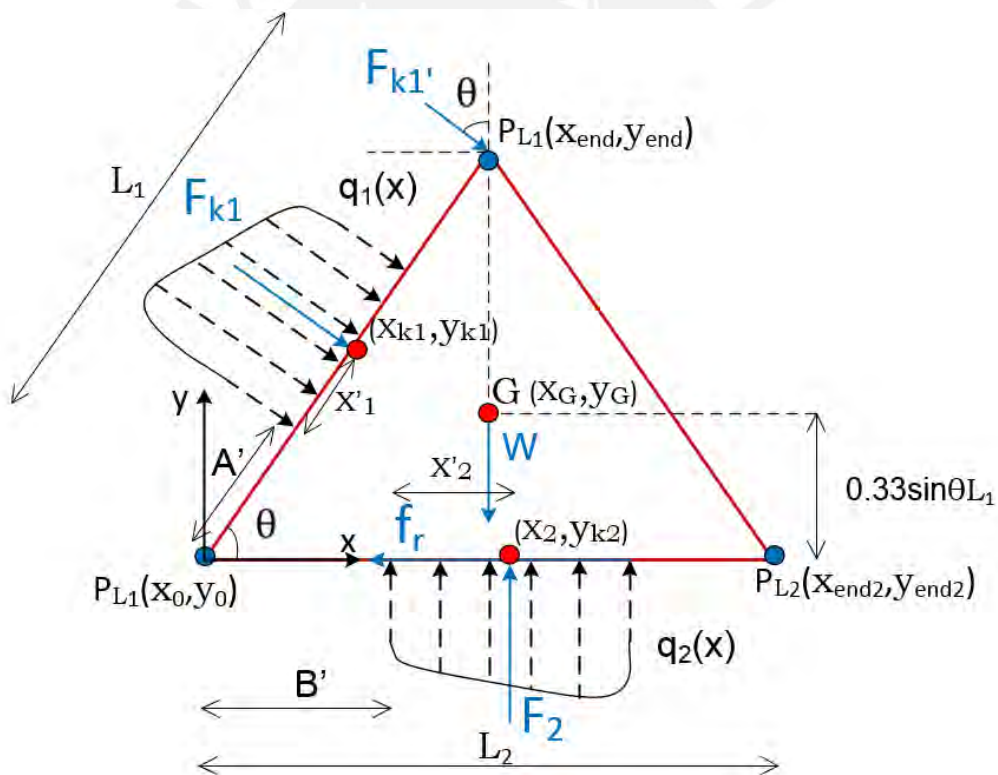


Fig. 5.11: Free body diagram – Generic triangular object

The forces F_{k1} and $F_{k1'}$ are a function of the deformation of the gripper bend lines at each point of contact with the triangle. A generic force F_{kn} at a distance s is sketched in figure 5.12. The force F_{kn} can be calculated through force and moment equivalent systems as well as F_{k1} and $F_{k1'}$.

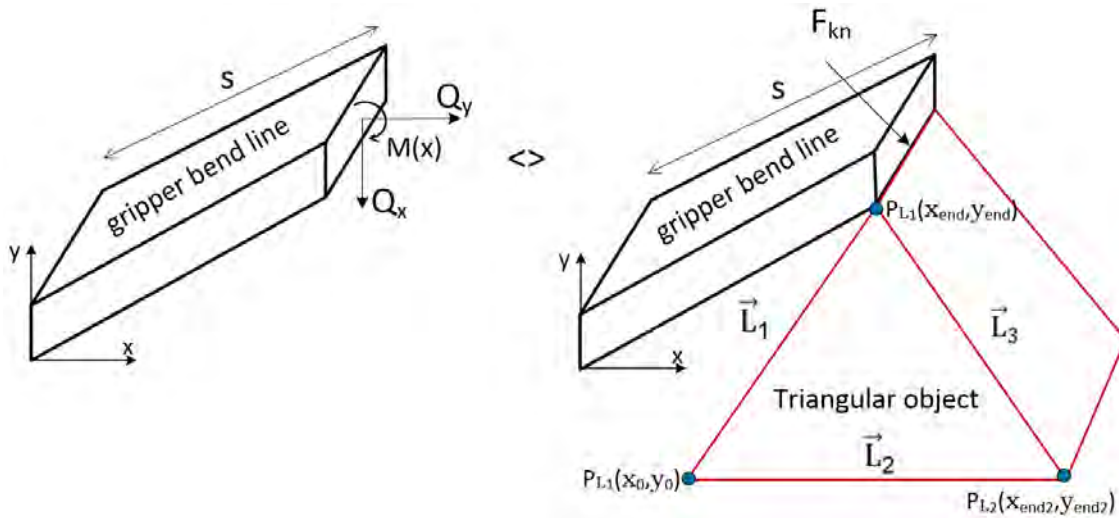


Fig. 5.12 Equivalent systems – Finding the forces F_{k1} and F_{k1}'

The program *dgl_rod_clamped_pinned_diff_bend_stiff* calculates the forces Q_x, Q_y and the moment $M(x)$ for every point along the gripper bend lines; therefore, F_{k1} and F_{k1}' can be obtained. Additionally, the distances A' and B' are defined when setting the length of the enhanced segments for each gripper bend line. Furthermore, the point G (center of gravity) is defined by the following equation:

$$G(x_G, y_G) = \left(\frac{1}{3}(x_0 + x_{end} + x_{end2}); \frac{1}{3}(y_0 + y_{end} + y_{end2}) \right) \quad (32)$$

The condition to avoid a rollover around the axis perpendicular to the center of gravity G and passes through it is defined by the equation of equilibrium of momentum as follows:

$$F_{k1x}(y_{k1} - y_G) - F_{k1y}(x_G - x_{k1}) + F_{k1'x}(y_{end} - y_G) - F_2(x_2 - x_G) + f_r \cdot 0.33 \sin\theta L_1 = 0 \quad (33)$$

The friction force f_r is equal to μF_{ry} , where μ is the friction coefficient between the surface of the triangular object and the surface of the gripper bend line and F_{ry} is the resultant force in y direction. The points of action (points in red colour) of the forces F_{k1} and F_2 are obtained by geometry relationships with the distances A' and B' and solving the following integral to find the distance x'_1 and x'_2 displayed in figure 5.11:

$$\dot{x}_i = \frac{\int x q_i(x) dx}{\int q_i(x) dx} \quad (34)$$

The functions $q_1(x)$ and $q_2(x)$ can be created from the data of the forces at every point of the gripper bend lines in contact with the triangular object. This verification process should be implemented for every obtained solution. During the gripping process for triangular objects, an eventual displacement of the object in the x-direction is acceptable as long as the non-rollover condition is satisfied.



Chapter 6

Conclusions and future work

6.1 Conclusions

The indirect approach for the calculation of a compliant gripper solution strikes the problem in the opposite direction. The solution process starts with the calculation of an optimized object using the least squares method and additional data handling.

Regarding round-elliptical objects, the algorithm *Elliptical Objects Fitting* fulfills the task of finding an optimized round-elliptical object, which fits the gripper bend lines with a significantly increased contact length. Until now, half of the work to find the optimized compliant gripper is done. The next step consists in the object comparison, meaning that if the dimensions of the objects 1 and 2 coincide, then the optimized compliant gripper is found, otherwise the geometry of the compliant gripper should be changed. As a conclusion, concerning round-elliptical object, the development of the algorithm *Elliptical Objects Fitting* is one step forward to obtain the optimized compliant gripper.

With respect to the square-rectangular objects, two main ideas for developing an algorithm to obtain a meaningful increment of the contact length using the least squares method are proposed. The main concept consists in handling each side of the square-rectangular object as lines. These lines are divided in two groups: primary lines \vec{L}_1, \vec{L}_3 , which are generated from the data input of the gripper bend lines applying the least squares method, and the secondary lines \vec{L}_2, \vec{L}_4 derived from the orthogonality of \vec{L}_1, \vec{L}_3 . Two possibilities are proposed, each having advantages and disadvantages. The main difficulty is to determine the maximum length of the primary lines \vec{L}_1, \vec{L}_3 in which all the corners of the square-rectangular object are inside the gripping zone. The details between each process are themes for a future work.

In reference to triangular objects, a parallel grip with three enhanced segments per gripper finger was analysed. Finding the lengths of \vec{L}_1 and \vec{L}_2 are themes for a future work as well as the additional non-rollover condition for avoiding the reduction of the contact length between the triangular object and the gripper bend lines. The non-roll over condition is described in equation (33).

Even though the ideas described in this work are based on the least squares method for retrieving the shapes of the objects, other approaches can be investigated. One of them is the artificial neural network, which has different field applications as for example image recognition, signal processing, speech production, etc. The neural network is a type of computer system architecture, which consists of data processing by neurons arranged in layers. The corresponding results are obtained through a learning process, which involves modifying the weights of the neurons that are responsible for the error. Neural networks learn by example, meaning that they can not be programmed to perform a specific task; thus, examples must be selected carefully; otherwise, useful time is wasted or even worse the network might work incorrectly. On the one hand, the advantages of the artificial neural network, which can be applied to fulfill the task of finding an optimized object that fits the gripper bend lines, are two. The former is that the network can work properly in case of incomplete information, for example, for round-elliptical objects, this case appears when performing the first least squares iteration, where just segments of the curve are available. The latter is that the network does not require knowledge of the algorithm solving the problem. On the other hand, the main challenges of this method in the compliant gripper design is to determine the structure of the artificial neural networks. Appropriate network structures are achieved through experience, trial and error, which means that a lot of data input should be introduced to the system for its learning process. Furthermore, because the network finds out by itself how to solve the problem, its operation can be unpredictable.

6.2 Future work

Elliptical objects:

As explained in section 3.1, the final goal is to find a compliant gripper, which grips the object 2 with an optimized contact length. After the implementation of the algorithm *Elliptical Objects Fitting*, which calculates the object 1, the next step consists in developing an algorithm, which compares the dimensions of the objects 1 and 2. After the comparison, if the dimensions of both objects are the same, then the gripping solution is obtained, else the boundary conditions should be changed and the calculation process should be restarted. The process previously described is matter of study and development. Hereafter, the work to do is highlighted in red in diagram 6.1.

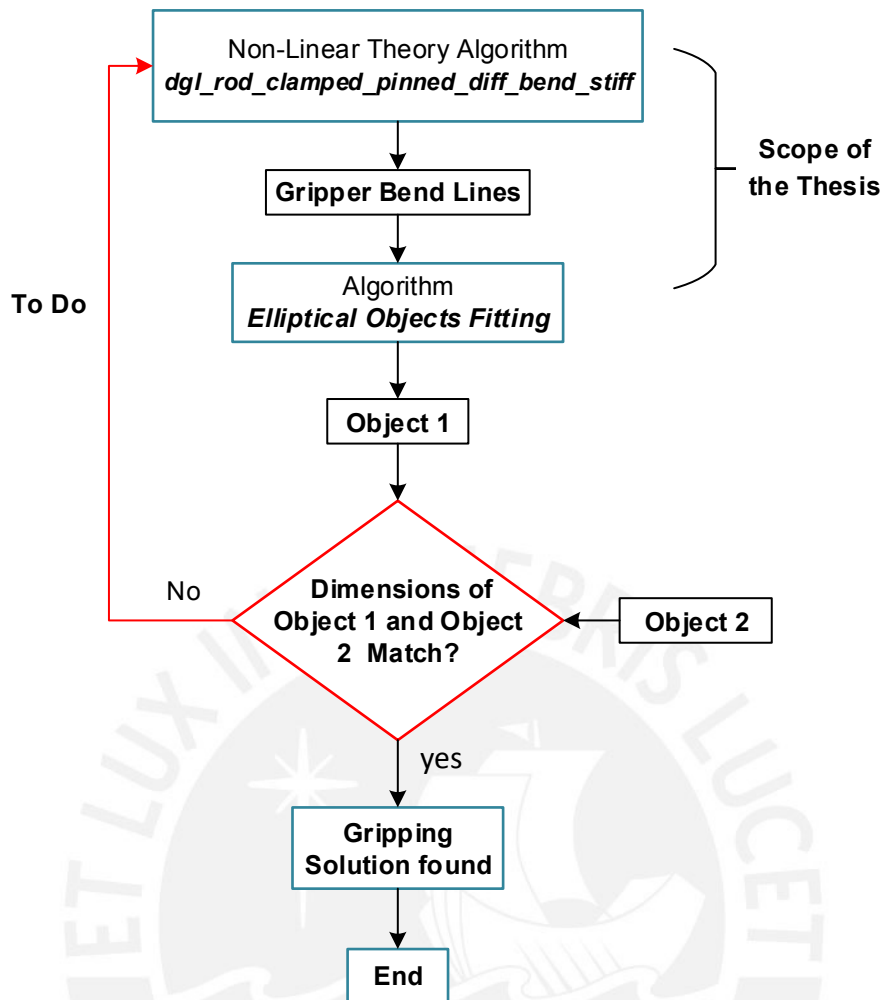


Diagram 6.1: Process flow diagram, future work highlighted in red

Square – rectangular objects:

For the square – rectangular objects, different manners of gripping were investigated using gripper fingers with enhanced segments. Even though the bend lines of the gripper fingers are generated by the algorithm *dgl_rod_clamped_pinned_diff_bend_stiff*, the square elements obtained in figures 5.1, 5.2, 5.3 are not calculated but inserted graphically; thus, an algorithm, which calculates automatically an optimized (significantly increased contact length) square-rectangular object should be developed. It is predictable that even though the main concepts for developing a new algorithm are pointed out, some steps in the middle might be missing. These steps are also matter of investigation for a future work.

Triangular objects:

The main tasks for a future work are the calculation of the length of the lines \vec{L}_1 , \vec{L}_2 and the implementation of the non-rollover condition.



References

- [1] G. J. Monkman, *Robot grippers*. Weinheim: Wiley-VCH, 2007.
- [2] K. Tai, A.-R. El-Sayed, M. Shahriari, M. Biglarbegan, and S. Mahmud, “State of the Art Robotic Grippers and Applications,” *Robotics*, vol. 5, no. 2, p. 11, 2016.
- [3] A. M. Gil Fuster, “Gripper Design and Development for a Modular Robot,” Technical University of Denmark, Lyngby, 2015.
- [4] L. Birglen, “From flapping wings to underactuated fingers and beyond: a broad look to self-adaptive mechanisms,” *Mech. Sci.*, vol. 1, no. 1, pp. 5–12, 2010.
- [5] L. Birglen and C. M. Gosselin, “Force Analysis of Connected Differential Mechanisms: Application to Grasping,” *The International Journal of Robotics Research*, vol. 25, no. 10, pp. 1033–1046, 2006.
- [6] D. Petković, N. D. Pavlović, S. Shamshirband, and N. Badrul Anuar, “Development of a new type of passively adaptive compliant gripper,” *Industrial Robot*, vol. 40, no. 6, pp. 610–623, 2013.
- [7] D. Petković and N. D. Pavlović, “Compliant multi-fingered passively adaptive robotic gripper,” *Multi Modelg in Mat & Struct*, vol. 9, no. 4, pp. 538–547, 2013.
- [8] M. P. Groover, *Industrial robotics: Technology, programming, and applications*, 2nd ed. New Delhi: McGraw-Hill, 2012, 1986.
- [9] D. T. Saha, S. Sanfui, R. Kabiraj, and S. Das, “Design and Implementation of a 4-Bar linkage Gripper,” *IOSRJMCE*, vol. 11, no. 5, pp. 61–66, 2014.
- [10] *Robotics and Machine Vision System*. [Online] Available: <https://www.slideshare.net/yaso4u/robotics-end-effector>. Accessed on: Sep. 02 2018.
- [11] Karokh Mohammed, “Design of a Gripper Tool for Robotic Picking and Placing,” Thesis, UPPSALA UNIVERSITET, 2010.
- [12] S. W. Rhythm, L. Terje, and Gareth Monkman, “Electromagnet Gripping in Iron Foundry Automation Part I: Principles and Framework,” *International Journal of Computer Science Issues*, vol. 8, <https://authorzilla.com/aDOzv/ijcsi-publication-2011-international-journal-of-computer-science.html>, 2011.
- [13] A. Yadav, “Effect of Temperature on Electric Current, Magnets and Electromagnet,” *Int J Adv Tech*, vol. 07, no. 04, 2016.
- [14] BYJU’S, *Differences Between Electromagnet and Permanent Magnet*. [Online] Available: <https://byjus.com/physics/difference-between-electromagnet-and-permanent-magnet/>. Accessed on: Jan. 15 2019.
- [15] A.K. Jaiswal and B. Kumar, “Vacuum Griper- An Important Material Handling Tool,” *International Journal of Science & Technology*, vol. 7, no. 1, <https://pdfs.semanticscholar.org/6b9a/807e44f478c0b7ed0a15cc1b222401601c06.pdf>, 2017.
- [16] Bridgestone, *Pneumatic Holders*. [Online] Available: <http://www.bridgestoneindustrial.eu/products/pneumatic-holders/>. Accessed on: Sep. 04 2018.
- [17] S. Schädle and E. Wolfgang, *Dexterous Manipulation Using Hierarchical Reinforcement Learning*. [Online] Available:

- https://www.festo.com/net/SupportPortal/Files/448583/LearningGripper_2013%20scientific%20pub%20Festo%20layout_en.pdf. Accessed on: Sep. 04 2018.
- [18] A. Rost and S. Schädle, *The SLS- generated Soft Robotic Hand-: An Integrated Approach using Additive Manufacturing and Reinforcement Learning*. [Online] Available: https://www.festo.com/net/SupportPortal/Files/448582/Soft%20Robotic%20Hand_ICMLA%202013%20scientific%20pub%20Festo%20layout_en.pdf. Accessed on: Sep. 04 2018.
- [19] W. Stoll, E. Wolfgang, and S. Schädle, *LearningGripper*. [Online] Available: https://www.festo.com/net/SupportPortal/Files/248131/54819_Broschuere_LearningGripper_en_130322_lo_L.pdf. Accessed on: Sep. 04 2018.
- [20] J. Guo *et al.*, “Toward Adaptive and Intelligent Electro adhesives for Robotic Material Handling,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 538–545, 2017.
- [21] M. Dadkhah, Z. Zhao, N. Wettels, and M. Spenko, *A Self-Aligning Gripper Using an Electrostatic/Gecko-Like Adhesive: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems : October 9-14, 2016, Daejeon Convention Center, Daejeon, Korea*. [Piscataway, NJ]: IEEE, 2016.
- [22] J. Shintake, S. Rosset, B. Schubert, D. Floreano, and H. Shea, “Versatile Soft Grippers with Intrinsic Electro adhesion Based on Multifunctional Polymer Actuators,” (eng), *Advanced materials (Deerfield Beach, Fla.)*, vol. 28, no. 2, pp. 231–238, 2016.
- [23] J. Shintake, V. Cacucciolo, D. Floreano, and H. Shea, “Soft Robotic Grippers,” (eng), *Advanced materials (Deerfield Beach, Fla.)*, e1707035, 2018.
- [24] W. Nachtigall, *Bionics by examples: 250 scenarios from classical to modern times*. Cham: Springer International Publishing, 2015.
- [25] W. Stoll, *MultiChoiceGripper*. [Online] Available: https://www.festo.com/net/SupportPortal/Files/333986/Festo_MultiChoiceGripper_en.pdf. Accessed on: Sep. 05 2018.
- [26] K. C. Galloway *et al.*, “Soft Robotic Grippers for Biological Sampling on Deep Reefs,” (eng), *Soft robotics*, vol. 3, no. 1, pp. 23–33, 2016.
- [27] W. Stoll and N. Gaißert, *FlexShapeGripper: Gripping modelled on a chameleon’s tongue*. [Online] Available: https://www.festo.com/PDF_Flip/corp/Festo_FlexShapeGripper/en/files/assets/common/downloads/Festo_FlexShapeGripper_en.pdf. Accessed on: Sep. 05 2018.
- [28] Lean Robotics, *Robotiq 2F-85 & 2F-140: Instruction Manual*. [Online] Available: https://assets.robotiq.com/production/support_documents/document/2-Finger_PDF_20180403.pdf. Accessed on: Sep. 05 2018.
- [29] Robotiq, *Robotiq 3-Finger Adaptive Robot Gripper Instruction Manual*. [Online] Available: https://assets.robotiq.com/production/support_documents/document/3-Finger_PDF_20180131.pdf?_ga=2.255237512.1212011490.1536166581-254720237.1534352439. Accessed on: Sep. 05 2018.
- [30] R. MacPherson *et al.*, *Dynamic Grasping in Space using Adhesive Grippers and Model Predictive Control*. [Online] Available: <http://asl.stanford.edu/wp-content/papercite-data/pdf/MacPherson.Pavone.ea.ICRA17.pdf>. Accessed on: Sep. 05 2018.
- [31] E. W. Hawkes, D. L. Christensen, A. K. Han, H. Jiang, and M. R. Cutkosky, “Grasping without squeezing: Shear adhesion gripper with fibrillar thin film,” in

- 2015 *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May. 2015 - May. 2015, pp. 2305–2312.
- [32] L. I. van Griethuijsen and B. A. Trimmer, “Kinematics of horizontal and vertical caterpillar crawling,” (eng), *The Journal of experimental biology*, vol. 212, no. Pt 10, pp. 1455–1462, 2009.
- [33] W. Crooks, S. Rozen-Levy, B. Trimmer, C. Rogers, and W. Messner, “Passive gripper inspired by *Manduca sexta* and the Fin Ray® Effect,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, 172988141772115, 2017.
- [34] W. Crooks, G. Vukasin, M. O’Sullivan, W. Messner, and C. Rogers, “Fin Ray® Effect Inspired Soft Robotic Gripper: From the RoboSoft Grand Challenge toward Optimization,” *Front. Robot. AI*, vol. 3, p. 465, 2016.
- [35] L. Zentner, *Nachgiebige Mechanismen*. München: Oldenbourg Wissenschaftsverlag, 2014.
- [36] R. Penney, *Linear algebra with applications: Ideas and applications*, 3rd ed. Hoboken, N.J.: Wiley, 2008.
- [37] E. Weisstein, *WolframMathworld*. [Online] Available: <http://mathworld.wolfram.com/Ellipse.html>. Accessed on: Dec. 25 2018.
- [38] R. Wolfgang, A. Sung Joon, and H.-J. Warnecke, “Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola,” 2000.



Appendix

Algorithm Variables

The variables are grouped per modules and type of variable (counters, flags). In addition, when referencing a section of the algorithm, the following notation will be used: [L: A-B], where L is the abbreviation of the word "line", A and B represent integer numbers greater than zero, which represent the number of the lines.

1.A Module 0: Non- linear deformation theory algorithm

- **Preload**

Stroke length, in text called d .

- **b and c**

Row vectors, which define enhanced segments in the gripper bend lines.

- **Factor**

Factor of reinforcement for the segments defined by row vectors b and c.

- **F_x**

Initial value of Q_x for solving the differential equations (2) – (8).

- **F_y**

Initial value of Q_y for solving the differential equations (2) – (8).

- **M**

Initial value of M_0 for solving the differential equations (2) – (8).

- **y**

$[Q_x Q_y M \emptyset x y]$ $m \times 6$ Matrix. Solution of the non - linear differential equations.

- **M0:**

Value of $M(x = 0) = M_0$

- **phi0**

Value of $\phi (x = 0) = \phi_0$

- **b_{max}**

Maximum deflection of the gripper bend lines.

- **x_{imax}**

x - Coordinate at which the maximum deflection **b_{max}** takes place.

1.B Module 1: Range Filtering Variables

This section is comprised of the variables, which are employed for filtering the range. The filtered range will be introduced into the algorithm *ellipse_fit*, which contains the least squares method. All the following variables are utilised for the data handling of both gripper bend lines.

- ***gripperup_new_input_y (jj)***

Row vector containing the filtered data of the y_{jj} components from the upper gripper bend line. This row vector contains also some zero values, which should be deleted.

- ***gripperup_new_input_y1***

Row vector containing the filtered data of the y_{jj} components from the upper gripper bend line. The zeros located in *gripperup_new_input_y (jj)* are deleted.

- ***gripperup_new_input_y2***

Column vector containing the filtered data of the y_{jj} components from the upper gripper bend line. The zeros located in *gripperup_new_input_y (jj)* are deleted.

- ***gripperup_new_input_x(jj)***

Row vector containing the filtered data of the x_{jj} components from the upper gripper bend line. This row vector contains also some zero values, which should be deleted.

- ***gripperup_new_input_x1***

Row vector containing the filtered data of the x_{jj} components from the upper gripper bend line. The zeros located in *gripperup_new_input_x (jj)* are deleted.

- ***gripperup_new_input_x2***

Column vector containing the filtered data of the x_{jj} components from the upper gripper bend line. The zeros located in *gripperup_new_input_x (jj)* are deleted.

- ***gripperdown_new_input_y(jj)***

Row vector containing the filtered data of the y_{jj} components from the lower gripper bend line. This row vector contains also some zero values, which should be deleted.

- ***gripperdown_new_input_y1***

Row vector containing the filtered data of the y_{jj} components from the lower gripper bend line. The zeros located in *gripperdown_new_input_y (jj)* are deleted.

- ***gripperdown_new_input_y2***

Column vector containing the filtered data of the y_{jj} components from the lower gripper bend line. The zeros located in *gripperdown_new_input_y (jj)* are deleted.

- ***gripperdown_new_input_x(jj)***

Row vector containing the filtered data of the x_{jj} components from the lower gripper bend line. This row vector contains also some zero values, which should be deleted.

- ***gripperdown_new_input_x1***

Row vector containing the filtered data of the x_{jj} components from the lower gripper bend line. The zeros located in *gripperdown_new_input_x (jj)* are deleted.

- ***gripperdown_new_input_x2***

Column vector containing the filtered data of the x_{jj} components from the lower gripper bend line. The zeros located in *gripperdown_new_input_x (jj)* are deleted.

- ***xgripper_totalmatrix***

Column vector containing the filtered data of the x_{jj} components from the upper and lower gripper bend lines.

- ***ygripper_totalmatrix***

Column vector containing the filtered data of the y_{jj} components from the upper and lower gripper bend lines.

1.C Module 2: Least Squares Method

This section is comprised of the variables involved with the function *ellipse_fit*. The function *ellipse_fit* contains the least squares method procedure.

Input, *n* least squares iteration

- ***Gripperupmatrix0***: Data- upper gripper bend line.
- ***Gripperdownmatrix0***: Data- lower gripper bend line.

Output, *n* least squares iteration

- ***b*₁₁**: Semi-Axis 1 of the ellipse.
- ***a*₁₁**: Semi-Axis 2 of the ellipse.
- ***x*₁₁**: x-Coordinate value of the center of the ellipse.
- ***y*₁₁**: y-Coordinate value of the center of the ellipse.
- ***phi*₁₁** : Inclination angle of the ellipse.
- ***x_{e_xy_iter_Is}***: Column vector - *x_e* values of the ellipse
- ***y_{e_xy_iter_Is}***: Column vector - *y_e* values of the ellipse
- ***x_{e_xy_iter_Is1}***: Resized column vector - *x_e* values of the ellipse.
- ***y_{e_xy_iter_Is1}***: Resized column vector - *y_e* values of the ellipse.

Input data, *n* + 1 Iteration

- ***data_iter2***: Column vector – filtered data input for iteration *n* + 1

Output, *n* + 1 least squares iteration

- ***b*₂₂**: Semi-Axis 1 of the ellipse.
- ***a*₂₂**: Semi-Axis 2 of the ellipse.
- ***x*₂₂**: x-Coordinate value of the center of the ellipse.
- ***y*₂₂**: y-Coordinate value of the center of the ellipse.
- ***phi*₂₂**: Inclination angle of the ellipse.

1.D Module 3: Location variables

This section is comprised of the variables, which are utilised for locating the gripper bend lines. In the program, the location process match up the start and end points of the ellipse and the gripper bend lines.

- ***Ellipsematrix_iter*** = [x_e xy_iter_ls y_e xy_iter_ls]

$m \times 2$ Matrix containing the x_e and y_e values of the first least squares iteration.

- ***Grippermatrix_located_logical_x***

Logical $m \times 1$ column vector. The values of the m elements are zero or one depending on the fulfillment of the filtering condition. The function *ismembertol* is used and the filtering condition is the following:

“if a x_e value of the Ellipse matches a x_g value of the gripper bend lines within a tolerance of 10^{-4} , then the value of m will be one, else the value of m will be zero”.

- ***idx_start***

Contains the index number of the x_g components from the matrix *Gripperupmatrix0*, which match the x_e values of the *Ellipsematrix_iter* within the tolerance of 10^{-4} .

- ***idx_start1***

Contains the index number of the x_g component from the matrix *Gripperupmatrix0*, which matches the start value x_e of the *Ellipsematrix_iter*.

- ***idx_end***

Contains the index number of the x_g component from the matrix *Gripperupmatrix0*, which matches the end value x_e of the *Ellipsematrix_iter*.

- ***Grippermatrix_located_x***

$m \times 1$ Column vector containing the x_g filtered values of the gripper bend lines after the process of location.

- ***Grippermatrix_located_y***

$m \times 1$ Column vector containing the y_g filtered values of the gripper bend lines after the process of location.

- ***Gripper_located = [Grippermatrix_located_x Grippermatrix_located_y]***

$m \times 2$ Column vector containing the x_g and y_g filtered values of the gripper bend lines after the process of location.

- ***Ellipse_iter1 = [x_e_xy_iter_ls1 y_e_xy_iter_ls1]***

$m \times 2$ Matrix containing the x_e and y_e values of the resized ellipse. The variable *Ellipse_iter* must be resized after the location process for further handling in the algorithm. The number of elements is the same as that from the matrix *Gripper_located*.

1.E Module 4: Creation of the new geometry

This section contains the variables, which filters the new input for the next iteration of the least squares method. In fact, the algorithm performs $2n$ least squares iteration and for each one the following variables are used for matching x components and then to apply a data refining of the y components for every point $P_i(x,y)$ in the range defined by *idx_start1* and *idx_end*.

- ***differenzx = Grippermatrix_located_x (ii,1) - Ellipse_iter1 (ij,1)***

Row vector containing the difference between $x_g(ii)$ and $x_e (ij)$. The difference is reset every single iteration in the loop and is used for matching the x values so that the correct difference of y values takes place. [L: 242-261]

- ***differenzy = Grippermatrix_located_x (ii,2) - Ellipse_iter1 (ij,2)***

Row vector containing the difference between $y_g(ii)$ and $y_e (ij)$. The difference is saved for every single iteration in the loop. [L: 265-272]

1.F Exit-Loop Variables

This section contains the variables to exit the loops and stop the iterations in the algorithm *Elliptical Objects Fitting*.

- ***Logical_differenzy***

It is a logical 1 x m row vector. The values of the m elements are zero or one depending on the fulfillment of the filtering condition. If the absolute value of *differenzy* is within a tolerance of 10^{-4} , then the value of m- element will be one, else the value of m- element will be zero.

- ***Sum_logicaldifferenzy***

It is an integer number greater than zero, which represents the sum of the components from the row vector *Logical_differenzy*. In order to break the loop, the minimum value of this variable has been adjusted depending on the boundary conditions.

1.G Flags

- ***flag***

Used for breaking the main loop (*i* counter loop) of the algorithm and plot the final solution.

1.H Loop Counters

This section contains the counters of the algorithm

- ***i***

On the one hand, the counter *i* as a unique number is an integer, greater than zero, which defines a range $[ii, jj]$, where *ii* and *jj* are real values, greater than zero. The range $[ii, jj]$ will be filtered before performing the first least squares iteration. On the other hand, the variable *i* as a range $[1: i]$ defines the number of iterations of the whole algorithm. For each new value of *i*, the range is varied. This variable is settled in a for-statement in line 67 and it is the main iteration counter of the algorithm.

- ***jj***

The counter *jj* is an integer number greater than zero, whose value varies in the following range: [1, length(Gripperupmatrix0)]. It is utilised to carry out the first data filtering and to fill up the filtered matrices of the gripper bend lines. The data is filtered according to the initial boundary conditions and the stroke length *d*. [L: 69-179]

- ***n***

The counter *n* defines a range comprised of integer numbers greater than zero, [1: *n*]. *n* defines the number of least squares iterations for each value of *i*. The algorithm will perform $2n$ least squares iterations for each value of *i*. This variable is setted in a for-statement, in line 187 and it is the core iteration counter of the least squares method section [L: 187-301].

- ***ii***

The counter *ii* is an integer number greater than zero, whose value varies in the following range: [1, length(Ellipse_iter1)]. It is employed to match the x_i values between the gripper bend lines and the ellipse generated in the *n* least squares iteration. [L:248- 261]

- ***ij***

The counter *ij* is an integer number greater than zero, whose value varies in the following range: [1, length(Ellipse_iter1)]. After filtering the x_i values, it is employed to fill up the row vector *differenzY*, which contains the difference between the y_i components of the located gripper matrix *Gripper_located* and the ellipse *Ellipse_iter1* obtained from the *n* least squares iteration. [L: 242- 272]

- ***iteration***

Counts the number of iterations for the algorithm *Ellipse Object Fitting*.

- ***counter***

Used for restarting the variables:

- ***gripperup_new_input_x , gripperup_new_input_y***
- ***gripperdown_new_input_x , gripperdown_new_input_y***