明治大学大学院先端数理科学研究科
2017 年度
博士学位請求論文

New Folding and Gluing Robot for Automatic Assemble of 3D Shapes
三次元形状の自動組立用新規折畳みロボット

学位請求者　現象数理学専攻
フリアン　アンドレス　ロメロ　ヤノ

New Folding and Gluing Robot for Automatic Assemble of 3D Shapes
三次元形状の自動組立用新規折畳みロボット

A Dissertation
Submitted to the Graduate School of Advanced Mathematical Sciences
of Meiji University

by
Department of Advanced Mathematical Sciences, Meiji University
Julian Andres Romero Llano
フリアン　アンドレス　ロメロ　ヤノ

Supervisor: Professor Dr. Kokichi Sugihara
January 2018

Abstract

Origami, the ancient art of folding a flat-piece of flexible material such as paper into a three-dimensional shape, has attracted the attention of the scientific community because it can be used to create interesting-looking objects and mechanical structures with a vast number of applications. One of the most interesting applications of origami into science is in robotics. However, a proper solution in this matter has not been found, due to limitations in paper manipulation and the lack of possibilities to apply the robot systems into different applications. Some of these robots use complex mechanisms and require several sensors to work properly. These make it difficult for robot to be developed and its prices are increasing.

This study centers its attention in origami robotics, especially in folding-performing robots, and methodologies to create folding patterns for it. In this dissertation, an automatic folding robot is proposed and developed. This robot uses a crease pattern that combines simple folding patterns with small gluing areas to reduce the number of manipulations. Two methodologies for creasing pattern development are also proposed here. These methodologies are created to be applied into the proposed robot, and used to create vast number of three-dimensional shapes.

Chapter 1. Introduction

This chapter describes the background of this thesis. The purpose and research contents of this thesis are described. Different types of objects have been created inspired in origami such as boxes for protecting goods safely, satellite antennas able to unfold in space to provide the system with solar energy. The sector of science that studies the paper folding behavior, its properties and applications is called “Origami engineering”. There are two types of objects in which origami-goods could be divided: Objects or products that take inspiration from origami and objects that use origami properties to work. Among the second category, is the “origami robotics”. In origami robotics, two on-going researches have been the center of attention in the recent years: the first is called “Self-Folding Robots”, and the other one is called “Origami performing robots (a.k.a Folding robots)”. Self-folding robots consider each facet as rigid structures joined by hinges (folding lines). Each hinge is considered as an independent actuated mechanism. On the other hand, “folding robots” are devices designed to fold any paper-like material into a three-dimensional shape.

This work centers its attention in the folding robot category. Thus, several works related with this type of robots are studied. Balkcom et.al provide a fundamental basis for robotic paper manipulation. However, they used a robot that was specially designed for simple folds. Moreover, the generalization to more complex folds, such as squash and petal folds that required multiple manipulations simultaneously was not possible. Yao and Dai focus on the dexterous manipulation of origami cartons using robotic fingers based on the Interactive Configuration Space (ICS). However, their control strategy is very limited to a specific folding. Kihara and Yokokohji, developed a folding robot able to fold a "tadpole" pattern. Their method synthesized a desired trajectory and sensory feedback control based on direct teaching data from a person. Although canonical correlations between forces and velocities of human trajectories are used to correct the motion deviation from nominal trajectories, complex direct teaching (e.g. many repetitions) and no implicit manipulations are included to avoid fluctuations of the paper.

Chapter 2. Kansei applied into Origami Engineering

The main idea of this chapter is to quantify the appreciations of persons to different geometrical shapes in origami. Origami is an art, and an important part in analyzing the final shape is the attractiveness. The idea here is to use our previous works in kansei engineering to create a quantitative model of perceptions. We compare this model, with a quantitative model that analyzes the quality of the final shape using the expected area and the resulting area to obtain a deformation rate value. This could help us to understand which type of figures are more visually attractive to most people, and what characterize this attractiveness. These results are then applied to the shapes created with the proposed robot to evaluate the attractiveness and the quality of the results. The results shows a common attraction to rounded shapes and a mix of dark and light colors.

Chapter 3. Origami-performing robot design

Previous works in origami-performing robots have shown that the required manipulations in paper-folding could be a challenge. The robots with these works use complex robot systems, or adjust standard robot manipulators that exist in the market to fold paper sheets. However, these existing robot devices have complex mechanisms and are expensive. To reduce the number of manipulations in the folding process, the use of simple folding procedure combined with gluing areas is proposed in this chapter. The use of LEGO MINDSTORMS NXT®is also proposed to reduce the price of the system, and gives us an easy way to analyze and test different concepts (e.g. kinematic and dynamic models, control scheme, and applicability) during the development process. To represent the functional behavior of the proposed robot, the kinematic model, dynamic model, and the control scheme have been analyzed. In this chapter, the trajectories calculations are exposed, this includes, the kinematic model, and some trajectory considerations to implement the proposed robot into mass-production procedures. Then, several algorithms related with the paper properties such as: spring-back and stacking effects, are included into the trajectories generation algorithms to improve the accuracy of the movements.

Chapter 4. Control scheme applied into the proposed robot

It is extremely difficult to introduce automation in robots to handle shape-changing objects such as sheets of paper. There are two main problems associated with paper handling. One of them is controlling deformation of objects that has infinite degrees of freedom using a finite number of manipulated variables. The other problem is how to handle fragile material such as paper without exerting excessive stress. In this chapter, a new method that uses feedback error learning (FEL) to control the robot is proposed. Adaptive learning algorithms are implemented and compared using several artificial neural networks (NN) to perform precise and smooth manipulations of paper sheets. Several feed-forward controllers based on artificial neural networks are tested and compared with other types of classic controllers. Adaptive algorithms are evaluated, to improve the convergence of the tested NN and ensure stability of the controlled system. These results show a clear superiority in the performance of systems using Holographic Neural networks (HNN). Such superiority is due to the simplicity of the HNN and the small of data required for convergence.

Chapter 5. Methodology for designing crease patterns for the proposed robot

The methodology applied to create crease patterns with the proposed robot is based on shapes of surface of revolution (a.k.a rotational sweep). The origami pattern developed with previous software, is intended to be assembled by hand, and has folds that are very difficult to execute with a robot due to handling problems. In this new methodology, a modified version of the surface of revolution (SOR), created by Jun Mitani in 2009 is proposed. This new methodology, uses a combination of simple folding patterns with gluing areas to reduce the number of manipulations required by the robot to execute a 3D shape. Some modifications are made to add the gluing areas into the folding process, and some extra additions are included to allow the final shape to be opened correctly. Several examples are exposed and assembled using the proposed robot to demonstrate the applicability of this methodology.

Chapter 6. Improved methodology for designing complex 3D shapes by the proposed robot

The methodology explained in chapter 5, can be used to create 3D shapes that are based on SOR, able to be built by the proposed robot. Though this methodology expands the applicability of the proposed robot, there are figures that are not based on SOR, because they have irregular shapes. Several software and methodologies have been made to create crease patterns of irregular shapes. However, the resulting 2D crease pattern for these complex shapes is even more complicated to build than the SOR counterparts, having inside folds that required multiple manipulations at the same time, making extremely difficult to assemble these 3D shapes automatically using a robot. To solve irregular 3D shapes using a slightly modified version of our already existing robot, a novel methodology extracted from the previous SOR methodology is proposed and explained in this chapter. This new methodology can be applied to figures with star-shaped-projected polyhedrons. A star-shaped-projected polyhedron is a shape that is composed and contains an infinite number of star-shaped-polygons. A star-shaped-polygon is an irregular 2D polygon that contains a point, from which the entire boundary of the polygon is reachable.

Chapter 7. Applications of the proposed methodologies and robot

In this chapter, different applications of the proposed methodologies and robot into several areas are exposed. These applications include many things such as guard protection for sweet fruits in growths, medicine and architecture, etc. The first application is on guard protection for sweet fruits in growths. The main idea in this application is to create a recycle-paper sag to cover the fruits during this period of ripening. This sag or cover can be created using a diverse number of shapes and sizes, depending on the size and shape of the corresponding fruit. The shape of the stamp for this type of sags has usually pseudo-cylindrical shape, with the final and starting edges unglued. The second application is on packaging, where using the proposed robot and methodology, interesting looking boxes or envelops can be assembled. The third application is in the creation of scaled-house models. In house design the option to observe the result through a scale model is essential to analyze and make decisions about the structure, and appearance of the building. This scale-model can help also to provide information to potential buyers by the house sellers about the house location and spatial comparisons (e.g. size of the living rooms, bathrooms, windows and doors). In this application, a software to create three-dimensional (3D) house-model using its elevation views (i.e. North, south, east, and west views) is created. The final application is on medicine, where the proposed methodology can be used to create shape-shifting mechanisms to execute several tasks. These tasks go from non-invasive surgeries to open-close mechanisms to deliver medicine directly to an affected area.

Chapter 8. Conclusions and future works

This chapter summarizes the contributions of the current work into the research area. Some general conclusions are exposed, and some future works are included.

There are several contributions in this dissertation:

- An automatic paper folding robot is designed. We show that using simple folds in combination with gluing areas, good-looking 3D shapes can be achieved with this robot.

- Simulations and experimental results with the proposed robot show a clear superiority of the learning algorithms developed with HNN.
- Proper modifications are made to the SOR methodology to include gluing areas into the crease pattern.
- Additions to the crease pattern from SOR are made to allow free movement of the flaps, reducing the degrees of freedom and the complexity in the folding process.
- A enhanced version of the SOR methodology is proposed to create crease patterns of irregular shapes that can be folded by a robot.
- Multiple of real-world applications are exposed to demonstrate the applicability of the robot and methodologies.

Acknowledgments

First, I will like to say thanks to God, that support me always. I will like to thanks to all people that somehow helps or contribute in the realization of this project. They have been several years to this project comes to be a reality and without your help will not been possible. I extend my grateful thanks to everyone in Hagiwara's Laboratory and Interlocus Inc. for their unconditional support. Special thanks to Dr. Ichiro Hagiwara, and Dr. Luis Diago for their advice and suggestions on these last years. To my university and laboratory colleges: Dr. Kokichi Sugihara, Mrs. Yang Yang, Dr. Eri Nakayama, Dr. Thai Thao, Mr. Bo Yu, Mrs. Yuko Adashi, and Prof. Chie Nara; without your support this project will not be possible. To my colleges on Interlocus Inc: Mr. Junichi Shinoda, Mrs. Mariko Hamada, Ms. Akiko Fujita, and Mrs. Yukiko Takewaki, among all other support personal, for their advice and knowledge. To the Japanese government to give me the opportunity to study in such beautiful and stunning country as Japan, this experience transform me for life. A sincerely thanks to my family, friends and professors in Colombia and Japan, specially to my mother, to always been attentive to my educational and personal growth. A sincere thanks to my father, for always been there when I need him, until its pass away in 2015. There were you are right now father, thank you very much, without you I will not been where am I right now. Finally, I will like to greet my wife for been with me, and make my life's days happier and amazing every day I pass with her. For all the people that contribute in my life, thank you sincerely.

Contents

Chapter 1

Introduction

Origami, the ancient art of folding a two-dimensional flat materials such as paper into three-dimensional (3D) objects. It has recently gained popularity among scientists and engineers because the technique can be used to create interesting-looking packages [14], shape-changing structures [21], or be applied in robotics [8, 75, 30, 20]. The sector of science that studies the paper folding behavior, its properties and applications is called "Origami engineering". The basis of "origami engineering" can be tracked to 2002, when professor Nojima [51], proposes the term that everyone uses now at days.

In this study, a paper folding-gluing robot using LEGO MINDSTORMS NXT ® technology has been designed to simplify the design process and test the effectiveness and applicability of this system in a mass-production procedure. To improve the robot system to be used for mass-production, the use of Feedback-Error learning controllers is proposed to increase the performance in the trajectory following and reduce the time to produce a unit. In order to simplify the folding process and reduce the number of manipulations, a novel methodology to produce three-dimensional (3D) objects to be assemble by the proposed robot based in surface of revolution (SOR) is proposed. Later, this methodology is improved to be applied into non-regular shapes, i.e. any type of closed shape. All considerations to transfer the SOR methodology into these type of shapes is explained in detail in final chapters of this thesis.

1.1 Origami Engineering

Different types of objects have been created inspired in origami; from boxes to keep goods safe, to satellite antennas able to unfold in space to provide the system with solar energy. There have been many attends that tried to explains what is origami engineering. The better way to explain what products are part of origami engineering is to divide them into two groups. The group of objects or products that take inspiration from origami(see Fig.1.1a), and the objects that use properties from origami to work (see Fig.1.1b). It is clear from Fig.1.1 that elements inspired by origami, are static objects such as: houses, sculptures or hallways, and not have any

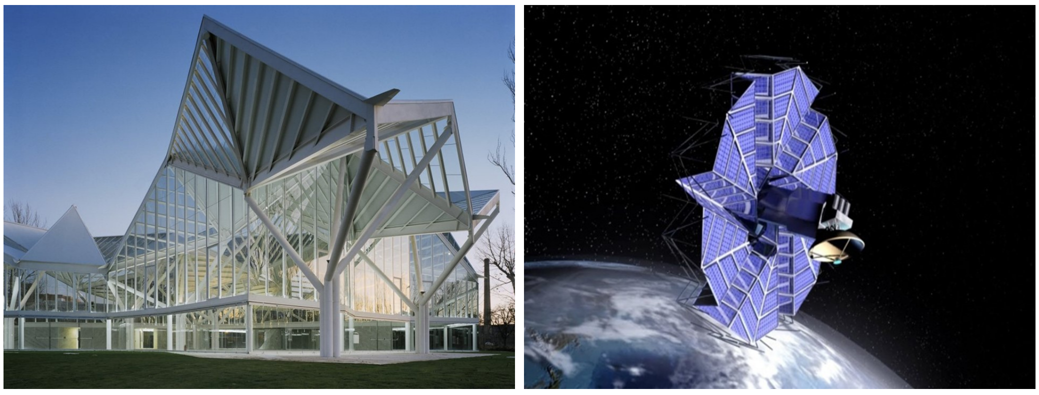

(a) (b)

Figure 1.1: Products created from origami. (a)Origami inspired architecture (Klein Bottle House, Rye, Victoria, Australia, Designed by McBride Charles Ryan). (b) Origami satellite antenna (NASA)[77]

.

mechanical device or shape-shifting behavior. On the other hand, elements that were created using origami principles, are mechanical devices, able to be moved, folded or shape-shifted. These particular objects use the basic principles of origami, such as flat foldability, or rigid foldability, in combination with other sciences like robotics, to create interesting and very useful products. Origami uses complex elements of geometry and mathematics in the folding process, that are explained in detail in following chapters.

1.2 Origami Robotics

One of the most interesting applications of origami engineering is origami robotics. In origami robotics, two particular on-going researches have been the center of attention in the recent years: the first is called “Self-Folding Robots” (see Fig.1.2a), and the other one is called “Origami performing robots (a.k.a. Folding robots)” (see Fig.1.2b). Self-folding robots consider each facet as rigid structures joined by hinges (folding lines). Each hinge is considered as an independent actuated mechanism. On the other hand, “folding robots” are devices designed to fold any paper-like material into a 3D shapes. This dissertation focuses on the “folding robots” category. Several folding robots have been developed in the past to automatically transform a 2D flat sheet of paper into a 3D object. However, it is extremely difficult to introduce automation in robots to properly handle flexible objects such as paper. The difficulty in handling flexible objects has been an on-going problem in the robotic community for over 19 years [48]. There are two main problems associated with paper handling. One of them is controlling deformation of objects that has infinite degrees of freedom using a finite number of manipulated variables. The other problem is how to handle fragile material such as paper without exerting excessive stress, i.e. how to handle it safely and reliably.

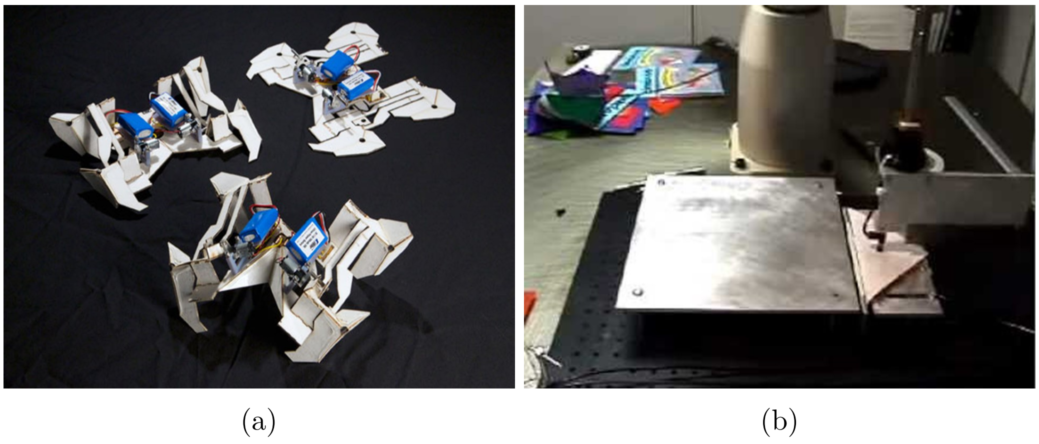


Figure 1.2: Origami robot types. (a) Self-folding robots [21]. (b) Origami-performing robots [8].

Balkcom et.al (see Fig.1.4a) provide a fundamental basis for robotic paper manipulation. However, they used a robot that was specially designed for simple folds. Moreover, the generalization to more complex folds, such as squash and petal folds that required multiple manipulations simultaneously was not possible. Yao and Dai (see Fig.1.4b), focuses on the dexterous manipulation of origami cartons using robotic fingers based on the Interactive Configuration Space (ICS). Furthermore, ICS includes a geometric configuration of control vectors associated with a spherical five-bar mechanism allocated in the four corners of the cartons. Trajectories for the manipulations using four robotic fingers are generated simultaneously using the ICS. However, their control strategy is very limited to an specific folding. Kihara and Yokokohji (see Fig.1.4c), developed a folding robot able to fold a ”tadpole” pattern. This pattern includes a squash fold in the folding sequence, that is consider to be a more complex type of folding, requiring more manipulations. Their method synthesized a desired trajectory and sensory feedback control based on direct teaching data from a person. Although canonical correlations between forces and velocities of human trajectories are used to correct the motion deviation from nominal trajectories, complex direct teaching (e.g. many repetitions) and no implicit manipulations are included in order to avoid fluctuations of the paper. Namiki et. al[49], use real-time modeling and visual tracking in their robot. The measured data of the paper is used to create the trajectories of the robot in real-time operation. The method from Namiki et. al[49] applies a visual recognition by camera and impedance control to produce softness in the trajectories of the fingers. The required kinematics adjustments are applied according to the data measured by the small sensors in the fingers. Despite this robot can produce some interesting results, the complexity in the systems and high cost make it not suitable to be used. Elbrechter et. al.(see Fig.1.4d), model the bending of a sheet of paper, and paper crease lines in order to monitor deformations. Tactile-based and vision-based closed loop controllers are introduced in an anthropomorphic robot to fold paper. Although high-speed manipulations of flexible objects can be achieved, this implementation still very complex

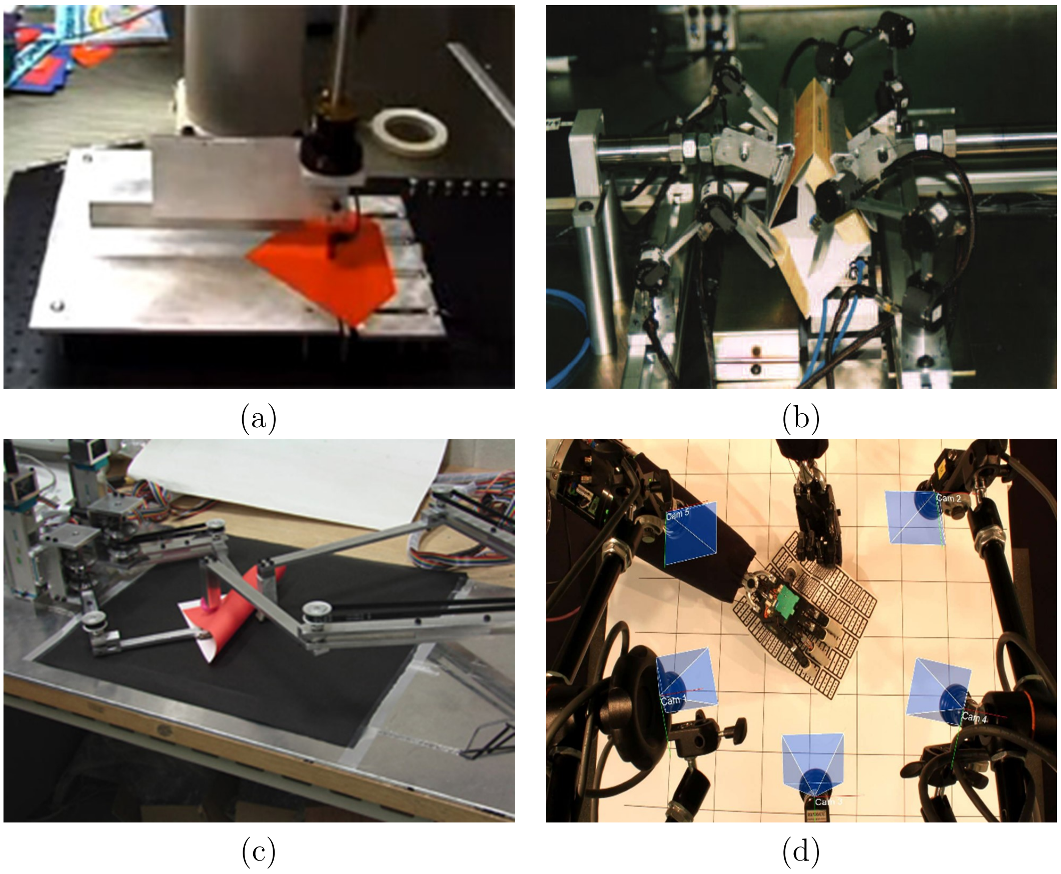


Figure 1.3: Origami-performing robots. (a)Balkcom and Mason, 2008 [8]. (b)Yao and Dai, 2007 [75]. (c)Kihara and Yokokohji, 2010 [30]. (d)Elbrechter et.al., 2012 [20].

to be implemented and very expensive.

1.3 Feedback-Error Learning

Based on forward and inverse models of the cerebellum, Kawato et.al. [29], proposed the Feedback-Error Learning (FEL) control model in 1987. This model comprised a fixed feedback controller (e.g. PID controller) that ensured the stability of the system, and an adaptive feed-forward controller that improves the control performance [28]. Ever since its introduction, many applications have been developed based on the model of Kawato. However, most of them use a classic proportional-integral-derivative (PID) controller in the feedback loop and in many cases, they can address practical issues, such as: how to select proper feedback gains to ensure FEL stability. As shown in previous researches [44], the use of a PID controller can generate non-damped, non-asymptotic responses. This is due to difficulties in a proper selection of feedback gains [47] and coupling effects in multiple-input-multiple-output (MIMO) systems, caused by movement sharing between links in the robot [2, 3, 59]. Owing to these problems, researchers have been improving the FEL algorithms using more sophisticated controllers such as linear quadratic regulators, or linear parameterization controllers [47, 3].

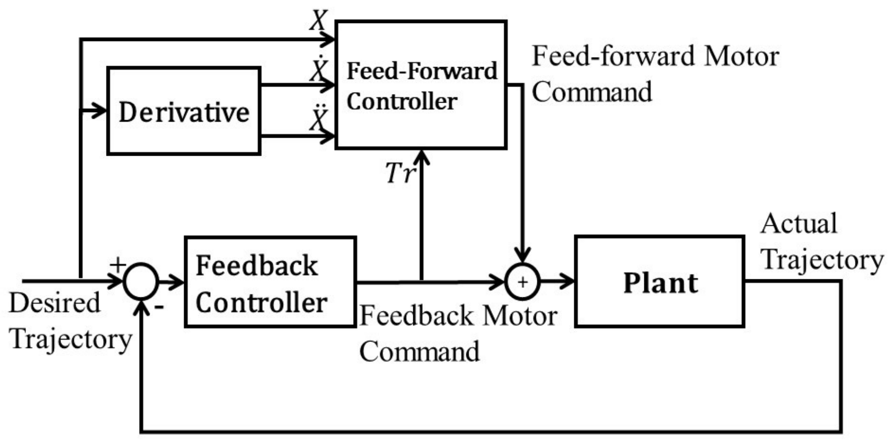


Figure 1.4: Classic Feedback-Error Learning control scheme

Nakanishi and Schaal [47] mathematically verified the stability and convergence of the FEL algorithm. The authors discussed the relationship between FEL and nonlinear adaptive control with adaptive feedback linearization. Moreover, they showed that FEL can be interpreted as a form of nonlinear adaptive control. From the practical point of view, the authors provided relations between feedback gains that guarantee the asymptotic stability of the closed loop dynamics in the frame of adaptive control for the single-input-single-output (SISO) case. Alali et al. [2, 3] have studied some generalizations of the FEL scheme for MIMO systems. Their work in [3] also covered an application for MIMO-FEL technique developed in [2] to a practical problem in terms of two-link manipulators. The basic idea in [3] is to achieve an approximated inverse of the plant model, using linear parameterization. Recently, Sugimoto's group [39, 64] has compared different architectures for the conventional schemes of FEL. First, they proposed the Virtual-FEL [39], which does not use actual feedback error signals, by modifying the two-degree-of-freedom structure in conventional schemes. Then, a local weighted regression is proposed to seek local models by a scheduler parameter, instead of searching time series by input/output data during the FEL [64]. However, the precision of the scheduler parameter is still not sufficient and a more exhaustive evaluation is still needed for general applications.

Instead of using above linear parameterizations of the machine, the proposed control method tries to focuses on the original works of Kawato [29, 28, 73], where the feedback motor command is used as an error signal for online training of a NN, which generates a feed-forward motor command to the plant. Several works using supervised learning models such as NNs, have been reported in the literature for learning the inverse model of a plant [59, 36, 46]. Using multi-layer NNs, the authors in [36] can fully utilize the learning capabilities regardless of whether or not the plant is linearly parameterized or not. Recently, FEL has been improved by a partially linear mapping algorithm called PaLM-tree for the highly accurate control of an active camera in tracking a moving target despite the unknown mechanical friction and relatively large closed-loop control latency [46]. Furthermore, PaLM-tree is based on a regression tree algorithm, which can represent a nonlinear mapping between arbitrary dimensional spaces. The work developed in Ruan et.al. in [59] is similar to one of the control schemes used in this thesis proposed robot. Ruan et al. have used an online back-propagation algorithm with a self-adaptive learning rate to realize the combination of learning and control within the FEL scheme. A self-adaptive learning rate algorithm similar to the one used in [59] is applied to the proposed robot and compared with other adaptive training algorithms.

1.4 Crease pattern design methodologies

Using a sheet of paper, different types of 3D shapes can be created. Origami is an old traditional art that has been in continuous change during history. So many computational programs to create crease patterns of complicated 3D shapes have been created through the resent years. Robert. J. Lang and his software “Tree Maker” [34] is one of the examples that can be found on internet, see Fig.1.5. The software “Tree Maker” creates the origami crease pattern based on the internal stick structure of the target figure (the tree). The target figure can have any type of complicated structure such as insects or any other type of animal. However, the final crease pattern created with this software can be extremely complicated to fold, making it very difficult to perform patterns created with this methodology using a robot.

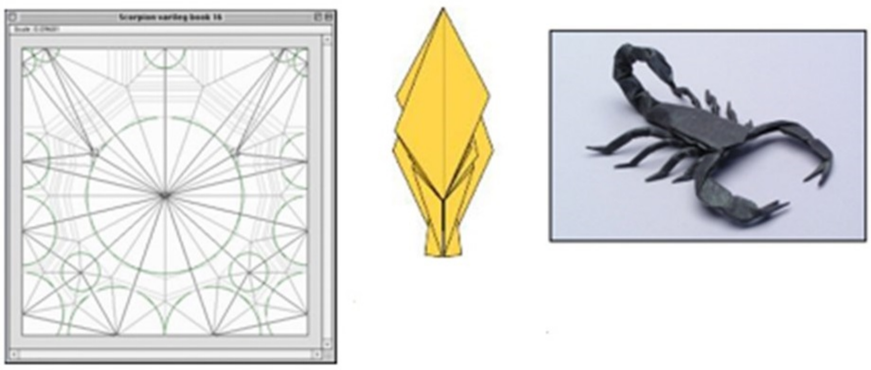


Figure 1.5: TreeMaker software by Robert J. Lang.

Another interesting program that can be found online is the “Origamizer” created by Tomohiro Tachi [67], see Fig.1.6. This program can generate crease patterns from an arbitrary polygonal model that possesses the topology of a disk. This program has a wide applicability as most of the 3D shapes can be modified to match the topology of a disk by adding some cut lines to the final crease pattern. Although "Origamizer" can be applied to create crease patterns of rotational based shapes as well, the final crease pattern could be over-complicated to fold.

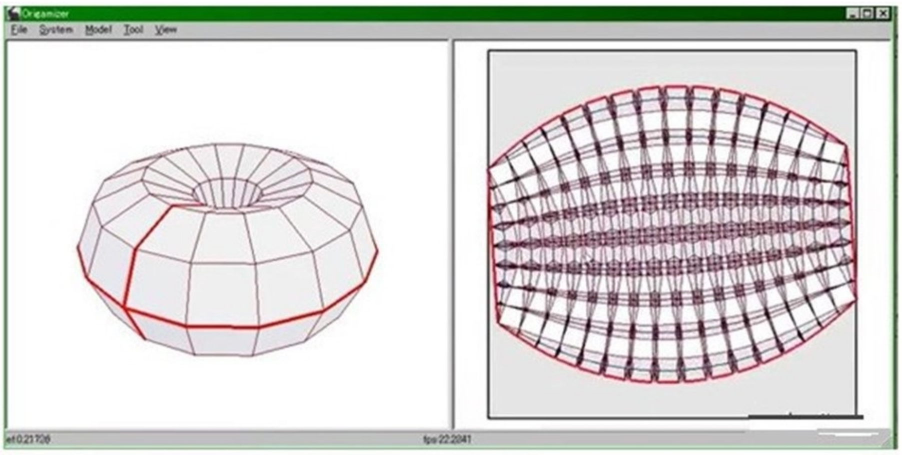


Figure 1.6: Origamizer software by Tomohiro Tachi.

Other interesting example that can be found in the internet is the software "Pepakura Design" [37], observed in Fig. 1.7, developed by Tama Software Ltd. The interesting point of this program is the capability to create paper-craft patterns of very interesting and complex shapes. However these patterns are divided in multiple modules an common paper-craft object, and the final results do not follow an standard shape. This make it very difficult to be implemented into a robot system procedure.

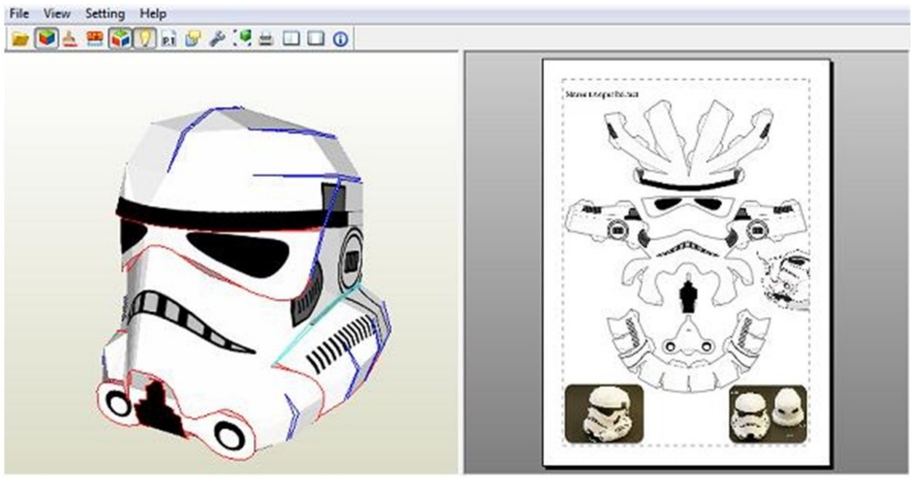


Figure 1.7: Pepakura Design by Tama Software Ltd.

The software "ORI-REVO" [43] designed by Jun Mitani, uses a rotational sweep to create the crease pattern from a 2D poly-line, see Fig.1.8. This program can be used to create a vast number of 3D shapes that can be used in different type of applications. The methodology used by ORI-REVO is explained in [42]. ORI-REVO can create crease patterns less complex than TreeMaker or Origamizer, but is limited to only figures using surface of revolution. Despite the crease pattern is not very complicated to fold by hand, proper adaptations of this methodology must be carried out in order to be implemented in the proposed robot.

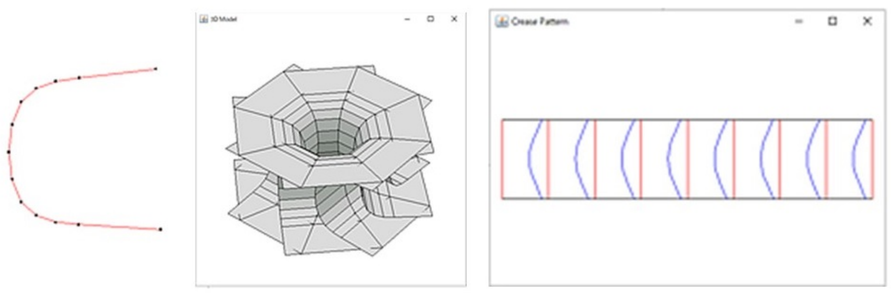


Figure 1.8: ORI-REVO software by Jun Mitani.

The origami pattern developed with ORI-REVO or any of the previously mentioned software were intended to be assembled by hand, and have folds that are very difficult to execute by a robot due to handling problems or excess of crease lines. Let see the example of a crease pattern created by ORI-REVO in Fig.1.9, the folds highlighted in orange circles represent folding lines allocated inside the paper. This means that these type of folding lines cannot be folded from an edge of the paper to the opposite one, requiring to perform multiple manipulations at the same time.

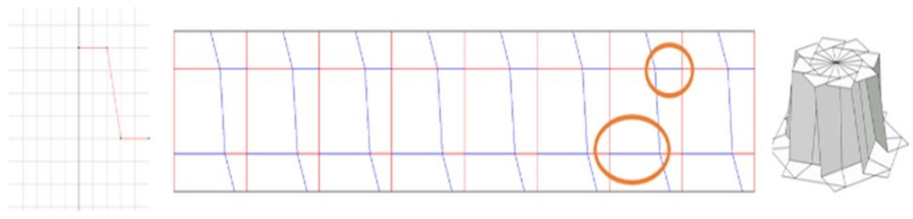


Figure 1.9: Example of an origami hat crease pattern developed with ORI-REVO. Blue lines: valley folds, red lines: mountain folds, orange circles: folds difficult to execute using robots.

1.5 Structure of the Thesis

The main focus of this dissertation is the design and development of a robot device able to fold a sheet of paper. Hence, many elements of control theory, robotics and origami engineering were combined in order to complete this goal. First, a summering of previous works related with origami engineering, origami robotics, control based in FEL, and origami crease pattern design software were exposed in chapter 1.

Chapter 2, expose some of our previous works in Kansei (feeling) engineering and the relationship with origami shapes. The previous works in kansei analysis was applied to evaluate the attractiveness of some origami shapes for different people. What characteristics make an origami good-looking and special from other shapes and structures.

Chapter 3, exposed the design and trajectory generation of our proposed origami-performing robot. Previous works related with this topic, center their attention in mimicking human behavior using complex manipulator with a great number of actuators, making the process complicated and expensive. The combination of simple-folding and gluing areas has been proposed here to reduce the number of manipulations in the folding process and making possible to use robots for automatic folding. This chapter, expose in detail the design procedure, the mechanism used in this robot. Also, some considerations for improving the accuracy of the robot's trajectory are introduced.

Chapter 4, explains the control scheme applied into the proposed robot. This control scheme is based in the cerebellar control model, a.k.a feedback-error learning. Adapting algorithms are applied into classic FEL control, for enhance trajectory following. Also, complex-value NN are proposed as a solution for FEL where smooth and precise movements are required.

Chapter 5 explains a proposed methodology to create a vast number of shapes using the proposed robot. This methodology is based in surface of revolution, to create simple folding patterns, that are easy to glue. Some modifications of previous works were executed to adjust this methodology into the proposed robot.

Chapter 6 explains an enhanced version of the previous methodology that can be applied to create 2D patterns of irregular shapes. The geometrical equations and rules are explained in detail, and several examples were exposed to demonstrate the effectiveness of this method.

Chapter 7 shows some applications of the proposed robot and methodologies into different field. This chapter shows the initial idea of the proposed robot, and the different applications that have surge from it.

Finally, chapter 8 summarize all conclusions and give some future works for both, the methodologies and the proposed robot.

Chapter 2

Kansei applied into Origami Engineering

The creation of artificial recognition systems of emotional states has attracted the attention of many researchers during last years due to the necessity of including preferences of costumers about a product into perceptual design elements [17]. Hence, many attends to quantify human's perceptions, known to be a qualitative measures into a quantified model has been a challenge during many years. Although it is not yet known the reasons why people have an specific attraction to some elements, colors, smells, or music, there have been many attends to quantified emotions and feelings. These techniques go from quantification of face's expressions to brain signal analysis. Three approaches have been designed in our previous works [16, 57, ?].

In our previous work [16], a new neuro-fuzzy method was proposed to investigate the characteristics of the facial images perceived as Iyashi by one hundred and fourteen subjects. Iyashi is a Japanese word used to describe a peculiar phenomenon that is mentally soothing, but is yet to be clearly defined. Using the proposed fuzzy-quantified Holographic Neural Netwoks (FQHNN), fuzzy rules to Iyashi expressions are extracted from a data-set of 20 images. The results show that FQHNN achieves $2 - 8\%$ increase in the prediction accuracy compared with other traditional neuro-fuzzy classifiers.

In the second previous work [56], the condition of the eyes of the person is used to validate the fuzzy rules extracted from the computational models. A simple and effective classifier is proposed to evaluate the closeness of the eyes during the evaluation of a small database of portraits. The experimental results show that closed-eyes can be detected only after the proposed shift of the normalized histogram is applied.

In our third previous work [57], a EEG device called NeuroSky' s B3 Band was evaluated as a minimally invasive method to measure brain signal behavior (Alpha, Beta) of a subject during a evaluation of a group of portraits showing facial expressions. Also different algorithms to reduce data dimensionality were evaluated in order to improve the level of prediction of a Holographic Neural Networks (HNN). Nine data dimensional reduction algorithms for brain signals are explored to improve the level of prediction of the HNN. The experimental results show that the percent of prediction with HNN can be increased in more than 10% if data reduction methods are applied to Beta brain waves.

The main idea of this chapter is to quantify the appreciations of persons to different geometrical shapes in origami. Origami is an art, and an important part in analyzing the final shape is the attractiveness that these shapes generate in persons. The idea here is to use our previous works in kansei engineering to create a quantitative model of perceptions. We compare this model, with a quantitative model that analyzes the quality of the final shape using the expected area and the resulting area to obtain a deformation rate value. This could help us to understand which type of figures are more visually attractive to most of the persons, and what characterize this attractiveness. These results are then applied to the shapes created with the robot proposed in this thesis to evaluate the attractiveness and the quality of the results.

2.1 Quantification of face expressions using FQHNN

Psychological researchers use several methods to investigate and classified emotions [11, 33, 41]. These procedures varies from imagery inductions to film clips and static pictures. One of the most widely used stimulus sets now at days, is the International Affective Picture System (IAPS) [33], a set of static images based on a dimensional model of emotion [60]. The image set contains various pictures depicting mutilations, snakes, insects, attack scenes, accidents, contamination, illness, loss, pollution, puppies, babies, and landscape scenes, among others. However, while many samples are desirable for estimating the response of a person more accurately (e.g. how much the person likes a product), in a real world situation, only a small number of samples needs to be obtained because of the efforts required for the persons to provide their responses from many samples. Hence as in our previous works [16] we use a small data-set of portraits to teach the machine to classify the facial images in the same way that people perceive them.

In previous works [31], the term “Iyashi expressions” was defined to study facial expressions related with the peculiar phenomenon of Iyashi. In Japan, “Iyashi” is a popular buzzword, referring to anything that is physically or mentally soothing [31, 40]. Iyashi goods (i.e. books, music, pictures, incense and aromas, bath salts, and plants) abound, offering to heal the physical and psychological stress of the workplace and of daily life in general. In society the expression Iyashi-kei is frequently used to describe laypersons that simply help people to relax. However, the word Iyashi (like other such as pleasure, well-being or satisfaction) does not have a consensus among people.

To gain a clear insight into the reasoning made by the nonlinear prediction models such as HNN in the classification of Iyashi expressions, the interpretability of the proposed fuzzy-quantified HNN (FQHNN) is improved by reducing the number of input parameters, creating membership functions (MFs) and extracting fuzzy rules from the responses provided by the subjects about a limited data-set of 20 facial

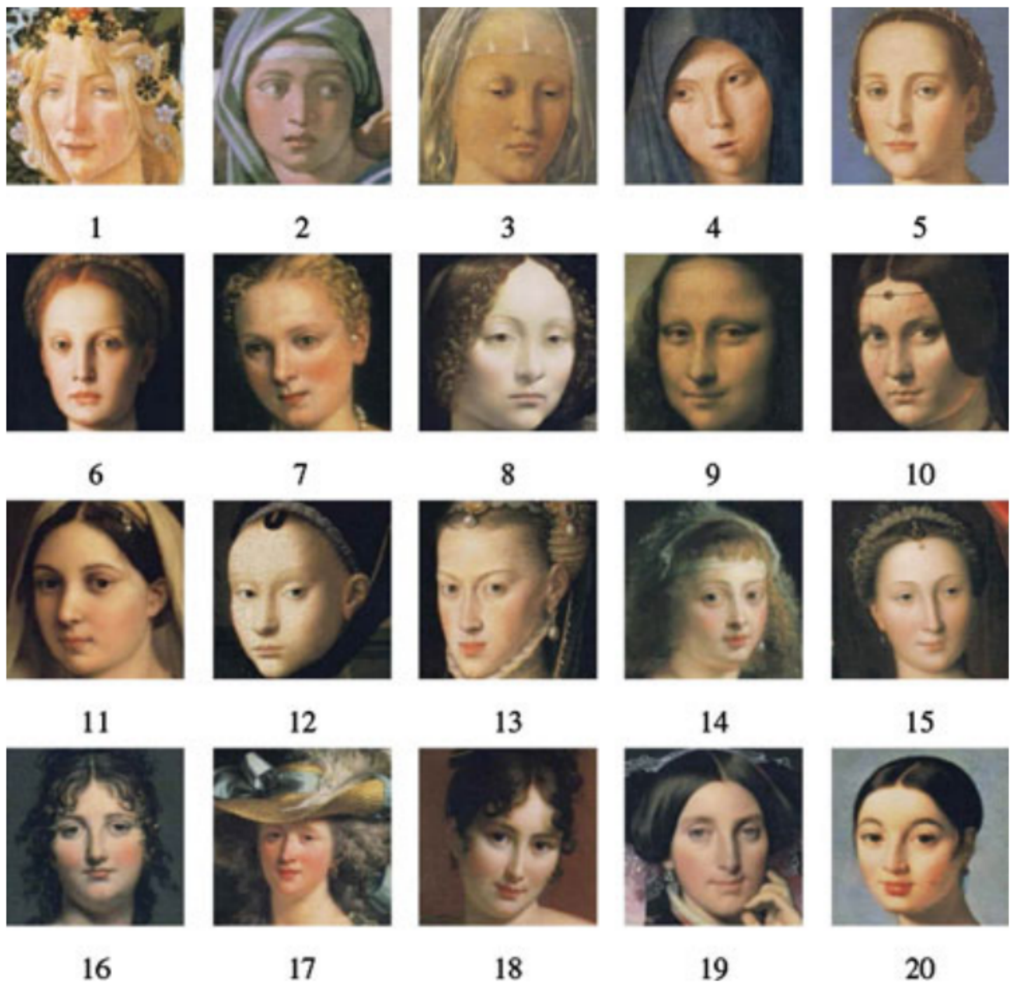


Figure 2.1: Collection of 20 special paintings. Each stimulus is rated on the scale ‘0’ -NO, ‘1’ -DON’ T KNOW, ‘2’ -YES to express whether or not they feel the emotion in question from observed portraits.

images. The experimental results in [16] show that the proposed FQHNN achieves $2 - 8\%$ increase in the prediction accuracy compared with traditional neuro-fuzzy classifiers while it extracts 35 fuzzy rules explaining what characteristics a facial image should have in order to be classified as Iyashi-stimulus for 87 subjects. Figure 2.1 shows the 20 images used to investigate the meaning of “Iyashi” in our previous research [16].

Above facial expressions are represented by feature vectors composed by twenty parameters considering the area of the right eye (p1), left eye (p2), nose (p3), mouth (p4), face (p5), right eyebrow (p6) and left eyebrow (p7). Distances between different elements of the face (p8-p23)are also considered (See details in [16]). Sixty-three MPEG feature points are used to compute above parameters and seventeen normalized features (x1,⋯, x17) used to find the relationship between induced emotions (e.g. Iyashi). Evaluations according to 114 subjects between 15 and 70 years old (102 Japanese and 12 non-Japanese, 47 females and 67 males) are shown in [16], together with the extraction of 35 rules describing the properties of the images. For this simple example, it should be noticed that each subject defines a different problem (i.e., different class distribution). Note that the naturally distinguishable classes in this paper are defined by the terms used in the context of the word Iyashi. As the meaning of Iyashi varies widely and is very individual, instead of focusing on the classification with a ground truth based on an average observer, we focus on developing an individual classifier and reporting the results of the classification accuracy and extracted rules for each subject. In [18], we presented a method to explore the properties of each image in Fig. 2.1, by transformation of the original images using radial basis functions (RBF). CSRBFs, firstly introduced by Wendland [71], are used for tuning facial parameters and mapping between facial images within opposite classes. In this way we can evaluate the validity of the rules extracted from the computational models.

2.2 Verification of Models of Personal Perception of Faces by Closed-eye Classifier using Histogram Correlation

The face is usually used to determinate the center of attention of someone in a specific object or frame of mind in a specific moment. One of the most common sources of information is the eyes behavior and its state. A specific state of the eyes could represent that the person is focusing in something or not. Also, in our previous work [56], we have proposed a simple and effective classifier in which the condition of the eyes of the subject is used to validate the fuzzy rules extracted from the computational models. The proposed classifier is based on the histogram shifting and the correlation between the shifted-histogram and a reference histogram computed by averaging the histograms of the open eyes in the BioID database [27]. The accuracy of the proposed classifier using the BioID dataset was 95.15% (3% higher than the best of the Convolutional NN-classifiers using LeNet-like architecture). For live videos obtained during the evaluation of the portrait images in unconstrained environments using our previously proposed system [18], the proposed correlation-based classifier [56] can obtain up to a 66.6% average accuracy in the extreme case of very dark lighting conditions. We are still conducting research to eliminate the influence of lighting conditions that produce shadows on the face during use of the proposed system in uncontrolled environments. Due to above illumination problems, we could not find a clear relationship between the state of the eyes and the evaluation of the portraits. In some cases the images assessed as 2 the user closed the eyes, likewise for some images evaluated as 0.

2.3 Adding Brain signal analysis into the classification process

As mentioned before there are different ways to analyze emotions in a more quantitative way using a computer [60, 68, 10, 53]. These studies show that humans can recognize emotions from voice and faces with 70-98% of accuracy and computer can classify facial expression with about 80-90%. However it has to be noted that these results were obtained on very controlled circumstances, and could differ in ordinary situations [68, 10]. Despite this, emotions are not always displayed on a facial expression. In psychology, an explicit separation is made between the physiological arousal, the behavioral expression (affect), and the conscious experience of an emotion (feeling). Facial expression and voice can be consciously adapted, and its interpretation is not objective. For this reason, research has been conducted to look for other physiological aspects that are more difficult to self-modify like the user’ s heart rate, skin conductance, and pupil dilation [68, 10, 53]. In the past decades, neuro-science researchers have tried to apply EEG data analysis techniques developed in electrical engineering and information theory, including time/frequency analysis and Independent Component Analysis (ICA) [45, 22]. For example, short-term changes in spectral properties of the ongoing EEG in specific frequency bands may be correlated with cognitive processes, e.g. expectancy of a target stimulus and with visual awareness. To date, the majority of Brain-Computer Interface (BCI) systems relies on EEG recordings. In our previous work [57], NeuroSky’ s B3 Band was evaluated as a minimally invasive method to measure brain signal behavior (Alpha, Beta) of a subject during an evaluation of the 20 portraits in Fig. 2.1. Also different algorithms to reduce data dimensionality were evaluated in order to improve the level of prediction of the HNN and the interpretability of the computational models.

2.4 Origami quality quantification method

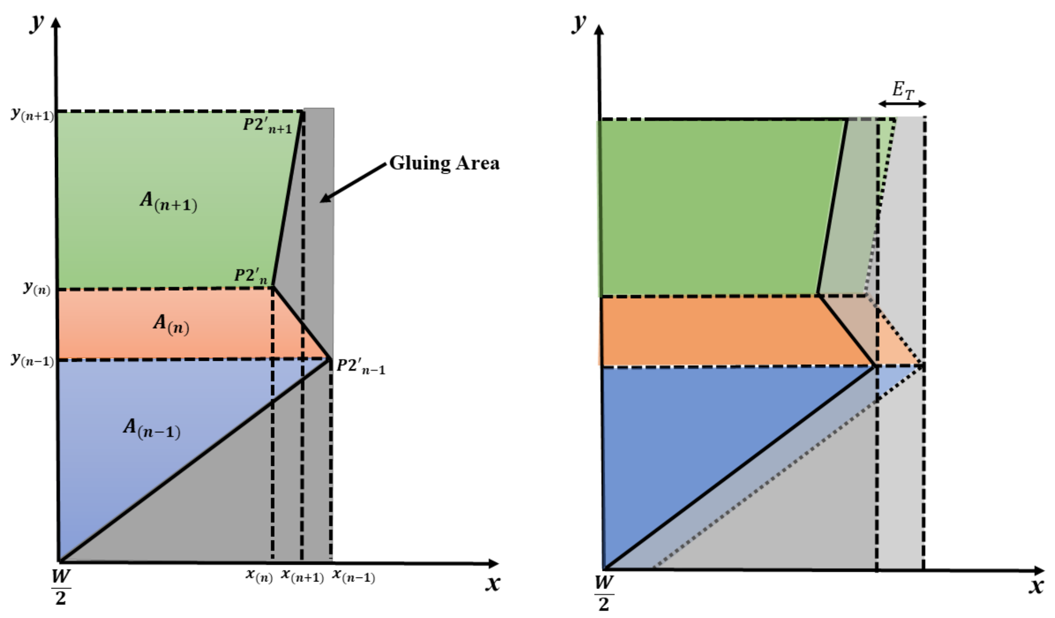
To evaluate the quality of the final 3D shape, we propose to calculate the external area of the 3D shape as a comparison value. To calculate the area of the surface of any 3D shape we use the integrative method using the information of the crease pattern from 5 and the following procedure.

To recreate the curve in Fig.2.2, different methods can be applied. For this case, a linear interpolation was used (See chapter 3.1.4). The area of each sub-segment of the curve can be obtained as follows:

$$A_{(n)} = \int_{y_{(n)}}^{y_{(n+1)}} m_{(n)} y + p_{(n)} dy. \tag{2.1}$$

$$m_{(n)} = \frac{x_{(n+1)} - x_{(n)}}{y_{(n+1)} - y_{(n)}}, \quad p_{(n)} = x_{(n)} + E_T. \tag{2.2}$$

where E_T is the sum of the errors from the first and the second stages (for the reference shapes $E_T = 0$). From these equations, the value of the area of a segment



(a) (b)

Figure 2.2: Quality measuring by integrative method. (a)Relationship between the crease pattern and the integrative method to calculate the expected area, (b) Generated error by displacement in x axis. $x_{(n)} = P2'_{(n,x)}$, $y_{(n)} = P2'_{(n,y)}$, and E_T is the total error, generated in both stages(folding, and folding-gluing).

can be estimated as $A_{seg} = 2\sum_{n=1}^{K} A_{(n)}$, K is the number of points in the profile.

2.5 Preliminary results applied into origami shapes

In our previous work [18], a color and shape recommendation system that uses an optimization approach to model the preferences of the customers when selecting colors and shapes from a product, produced by a Iterative Genetic Algorithm (IGA) was designed applying the previous ideas from emotional analysis proposed in [16] and [57]. The results show that, Although the generated shapes are not good enough for the subject, during 5 iterations, the user only evaluated 20 shapes of 100 shapes generated by the IGA as shown in Fig.2.3, reducing the amount of user's evaluations at 80% and some combinations of colors and shapes agreed with his preferences.

These results show also, that the users tend to like shapes with rounded edges, with a combination of colors between light and dark. Usually avoiding the sharped shapes like the cubes or pyramids. These same results can be translated into origami. However, generating curved surfaces using paper is difficult, due to the properties of the paper. Therefore, an approximation to curved surfaces should be applied into our models in order to generate, good-looking and appealing shapes. Apart of this, the use of proper color in the paper material is also very important. The preferences

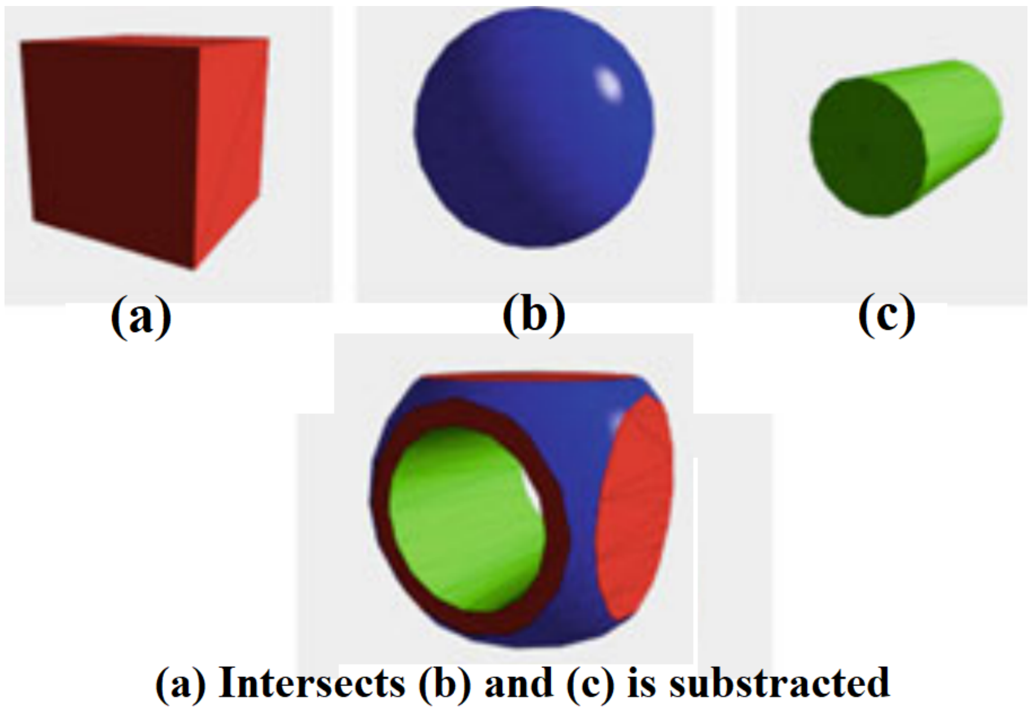


Figure 2.3: Example of one individual in the population obtained from the combination of three basic shapes. a Cube, b sphere, c cylinder.

of the user usually tend to a combination of dark and light colors. This idea was applied into the crease patterns created in chapters 5 and 6, where light colors were used in the base shape, and dark colors were used in the flap areas.

2.6 Summary

In this chapter, two method to generate a quantitative model of personal perceptions from origami shapes are introduced. The methodology introduced in [18], and enhanced in following papers [57] and [56] is used to generate a prediction system using FQHNN, in combination of eye-detection and brain signals. The system can predict with an accuracy of 60-70%, and rounded shapes were determinate as preferable figures among users. Another methodology, that analyzes the final shape using the difference between the real area and the expected area is also proposed. It was notice that, errors superior to increases or decreases about 10%, start to generate objects with distortion shapes. This has to be be avoided in order to obtain good quality in the shapes.

Chapter 3

Origami-performing robot design

In this chapter, the development of a folding-robot is described. Previous works in origami-performing robots [8, 75, 30, 49] have show that the required manipulations in paper-folding could be a challenge. The robots used on these works use complex robot systems, or adjust standard robot manipulators that exists in the market to fold paper sheets. However, these existing robot devices have complex mechanisms or are expensive. In order to reduce the number of manipulations in the folding process, the use of simple folding procedure combined with gluing areas is proposed in this dissertation. The use of LEGO MINDSTORMS NXT®is also proposed to reduce the price o the system, and give us a easy way to analyze and test different concepts during the development process.

Recent studies demonstrate extended use of Lego MINDSTORMS NXT® for educational purposes [32]. Most of them have been used for control designs [6], signal processing [9] and manipulation [74]. The work in [74], details the development of a model for a mobile robot constructed from LEGO MINDSTORMS NXT ®.

Figure 3.2 shows the proposed robot from this dissertation. First, to create a 3D shape using this robot, the desired crease pattern has to be designed using a design methodology based on a surface or revolution (SOR) technique [42] explained later in chapter 5. After the crease pattern is created, the robot proceed to fold the paper sheet. Mechanical information such as widths between folds and the shape of the gluing stamp are extracted from the pattern design process. The folding process of the proposed robot is explained in chapter 3.1, and is divided into two stages: the first one (F), consists in a pre-folding process that creates the crease lines of the folding pattern. The second stage (G), performs two processes at the same time, a gluing and a folding process, and it generates an accordion shape that adopts different forms after opened.

To represent the functional behavior of the proposed robot, the kinematic model, dynamic model, and the control scheme have been calculated (To see the dynamic calculations refer to chapter 4). In this chapter, the trajectories calculations are exposed, this includes, the kinematic model, and some trajectory considerations to implement the proposed robot into mass-production procedures. Then, several algorithms related with the paper properties such as: spring-back and stacking effects, were included into the trajectories generation algorithms to improve the accuracy of the movements. Finally, several results were expose and some conclusions were summarized.

3.1 Trajectory generation

3.1.1 Extraction of the required manipulations

In order to extract the required manipulations in a regular folding process, a procedure executed by a person to create a folding-gluing object is analyzed. This object was designed with the methodology exposed in chapter 5 and has an spherical shape. It was observed, that the person requires guide-lines to facilitates the folding of the paper and applying the glue properly . Figure 3.1 shows the procedure executed by hand for both, folding and gluing processes.

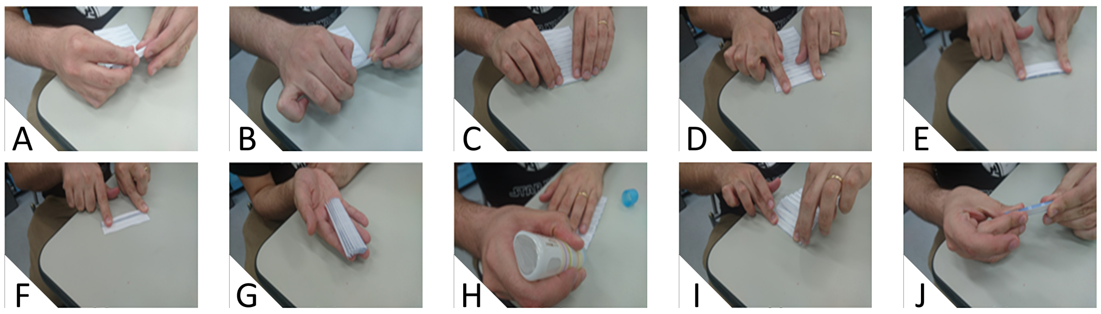


Figure 3.1: Handmade procedure to create a folding-gluing sphere.

To create a 3D object by hand several procedures have to be accomplished. It can be observed in Fig.3.1:A and 3.1:B that a person usually performed a smooth folding, and then executes a refinement of the same folding by passing the edge of the finger over the previously scratched crease line. Then, this person repeats these two processes multiple times and turns the paper 180 degrees to intercalate between valley and mountain folds (Fig.3.1:C-G). In the gluing process (Fig.3.1:H-J), applying the glue to the paper could be difficult to be executed by hand, depending on the size or shape of the pattern's gluing area. This gluing process can be made in different ways. Despite this, it was notice that the paper, after been pasted, has to be hold for certain time(like in Fig.3.1:J), because the glue takes some time to get dry. Taking in account all the information obtained from Fig.3.1, an extraction of requirements and possible solutions was generated and exposed in Table.3.1 .

3.1.2 Robotic manipulations

The proposed robot has two separately stages. In the first stage, called folding stage(Fig.3.2:F) the robot performs scratches of the desired crease lines alterning between valley and mountain folds. After the paper is folded in this folding stage,

Table 3.1: Requirements and possible solutions.

Requirements	**Possible Solutions**
The paper has to be folded in both directions, valley and mountain fold directions.	•Turn the paper 180° using a hand mechanism or manipulator. • Use two mechanisms, one for valley folds and another for mountain folds.
The glue has to be applied as precisely as possible.	• Use a stamp with the desired shape.
The paper has to be hold after glued.	• Use a mechanism to hold the paper.

enters to the folding-gluing stage (Fig.3.2:G). To create the trajectories of this stage, information from previous folding step is used. Using the information of the output widths from the folding stage, the spring-back effect can be estimated and used to generate the trajectory of the folding-gluing stage. This value is calculated using the mechanical properties of the paper (i.e. paper type, and thickness), and the real values of the widths obtained from the advancing sensor(Fig.3.2:F4) in the folding stage. Both trajectories are loaded into the corresponding NXT Brain-Bricks (Fig.3.2:F5 and G7), the information form the first brick is transmitted into the second brick through Bluetooth communication.

To folding a 3D shape using the proposed robot, the following steps have to been done (see Fig.3.3):

Step-A : The machine starts with a sheet of paper inside the tray of the robot. Then the pulling roller (Fig.3.2:F1) pushes the paper forward until the edge of the paper is in the desired position.

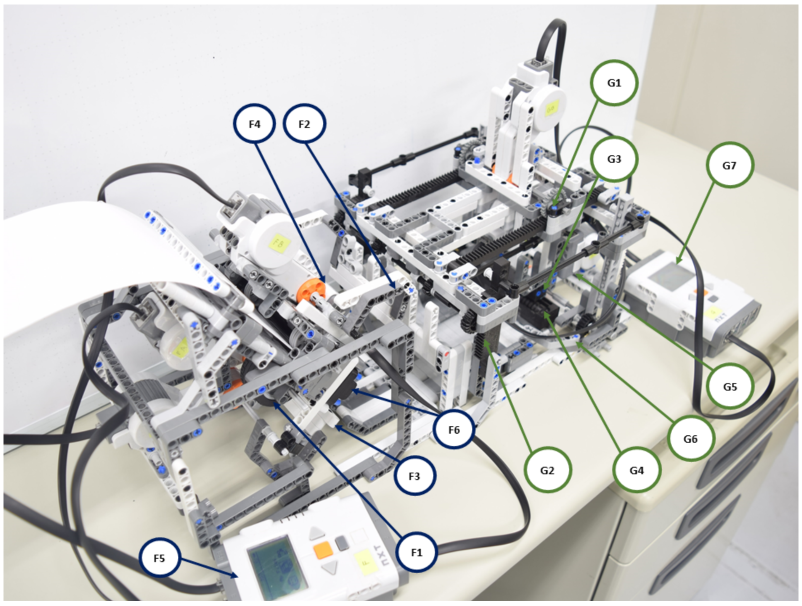
Step-B : The inferior handle (Fig.3.2F3) rotates and presses the paper against the folding base generating a valley crease fold. The turning angle in the folding base is set to 45 degrees in order to generate sharp crease lines without cutting the paper.

Step-C : The inferior handle is released and returned to its initial position, letting the paper to move forward. Note that the spring-back effect in the paper keeps the paper folded with an angle of more than 90 degrees.

Step-D : The pulling roller continues pushing the paper forward until the edge of the paper is in the desired position.

Step-E : The superior handle (Fig.3.2:F2) rotates and presses the paper against the folding base generating a mountain fold.

Step-F : The superior handle is released and returned to its initial position, letting the paper to move forward. The steps A-F are repeated $N-1$ times. At the end, the pulling roller pushes the paper forward to allocate the paper in the second stage.



F1: Pulling roller r(NXT Servomotor)	G1: Rack-pinion forward-backward system(NXT Servomotor)
F2: Superior folding handle(NXT Servomotor)	G2: Rack-Pinion up-down system(NXT Servomotor)
F3: Inferior folding handle(NXT Servomotor)	G3: Pressing device and stamp location
F4: Advancing sensor(Tachometer)	G4: Glue-distribution system
F5: Folding Lego NXT Brain-Brick	G5: Gluing sensor(Tachometer)
F6: Folding Base	G6: Holding fingers(NXT Servomotor)
	G7: Gluing NXT Brain-Brick

Figure 3.2: Proposed robot system (F) Folding step (G) folding-gluing step.

Step-G : In the second stage, the paper starts with one edge against the bottom wall located in the glue-distribution system (Fig.3.2:G4). In Fig.3.3H-N, the glue-distribution system is represented by a rectangular shape covered with a green semi-circled roller. The forward-backward system (Fig.3.2:G1) is represented with an "L-shape" and the two holding fingers (Fig.3.2:G6) are represented with a "T-shape".

Step-H : The forward-backward system ("L-shape") is moved backward using four rack-pinion mechanisms, forcing the gluing stamp (Fig.3.2:G3) to pass over the roller of the glue-distribution system (Fig.3.2:G4) to impregnate the stamp with glue. The forward-backward system stops at a distance computed from the combination of two factors: the spring-back effect in the paper and the accumulation of the paper forming the accordion shape after the paper is folded and glued.

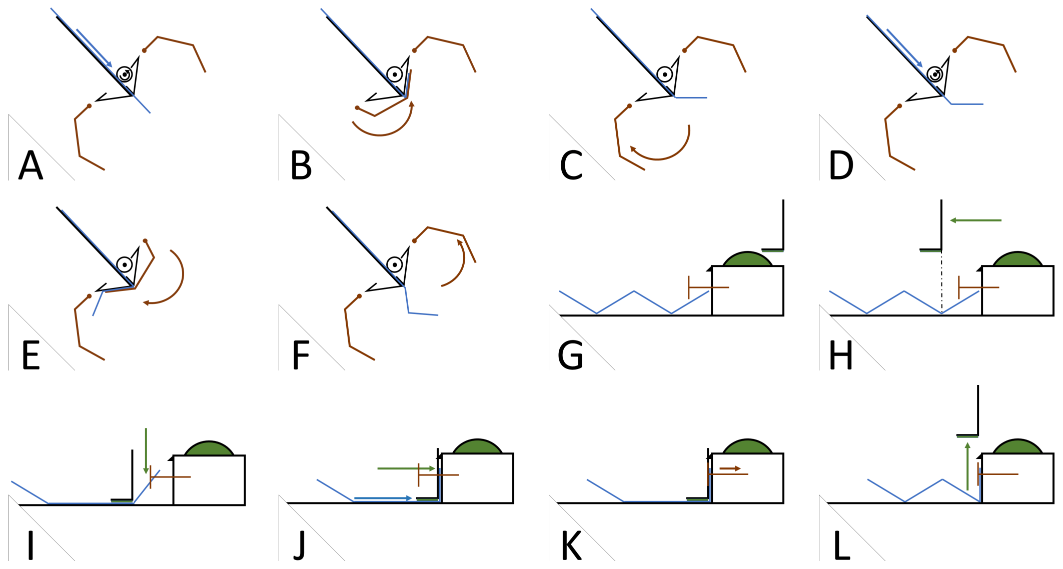


Figure 3.3: Automatic folding-gluing procedure (See the details of each step A-L in the text)

Step-I : Then, the stamp is pressed against the paper using other four rack-pinion mechanisms (Fig.3.2:G2), impregnating the paper with glue.

Step-J : While the paper is held to the bottom base, the pressing device is moved forward until it reaches the bottom wall.

Step-K : Then the paper is fixed to the wall using the holding fingers (Fig.3.2:G6).

Step-L : The pressing device (Fig.3.2:G3) is released from the paper and returned to its initial position.

Finally, process from G to L is repeated $N - 1$ times until the paper adopts an accordion shape. The geometry of the final shape is deformed mainly in steps I and K. In step I, the stopping position of the forward-backward system must be accurately calculated to avoid pushing the stamp on top of a previously folded line. In step J, previous glued and folded segments of the paper are accumulated to the wall and should be considered to accurately calculate the displacement of the forward-backward system.

3.1.3 Kinematic model of the robot

One of the most important parts to understand a robot behavior is to know its kinematic properties. The kinematics study the movement of a robot inside a reference frame. The kinematics is an analytic description of the movement of the links of the robot in time, and particularly studies the relationship between the position and the orientation of the final point of the robot and the coordinates that this point takes with time. There are two different kinematic models to be understood, the forward kinematic model (FKM) and the inverse kinematic model (IKM).

The FKM obtains the space coordinates and orientation of the final point of the robot, using the longitude of the links and angles of rotation of these links of the robot. To perform this task a method called Denavit-Hatemberg parameters method, is usually used to solve this problem due to a reduction in computations. On the other hand, the IKM, obtains the angles of each link of the robot using the initial and final Cartesian position of the final point of the robot. To solve the IKM, the trigonometric method (also known as geometric method) is recommended. It has to be notice that IKM could have multiple solutions, depending on the number of links in the robot [62].

3.1.4 Trajectory interpolation

Is recommended in a trajectory following, to perform smooth movements on each preset point to reduce the effects of mechanical noises (e.g. gear backslash, or abrupt overshoots). To accomplish this, a third grade interpolator (a.k.a cubic interpolator) was used [62]. Having two points in space $\theta_{(t_0)}$, $\theta_{(t_f)}$, using the following equations and dividing the trajectory between these two points in a number of samples we have:

$$\theta_{(t)} = a_0 + a_1 t + a_2 t^2 + a_3 t^3. \tag{3.1}$$

$$\dot{\theta}_{(t)} = a_1 + 2a_2 t + 3a_3 t^2. \tag{3.2}$$

where a_0, a_1, a_2, and a_3 must be found to solve the interpolator between boundaries. We should notice that the values of $\theta_{(t)}$ and $\dot{\theta}_{(t)}$ are known for both times t_0 and t_f.

$$\theta_{(t_0)} = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3. \tag{3.3}$$

$$\dot{\theta}_{(t_0)} = a_1 + 2a_2 t_0 + 3a_3 t_0^2. \tag{3.4}$$

$$\theta_{(t_f)} = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3. \tag{3.5}$$

$$\dot{\theta}_{(t_f)} = a_1 + 2a_2 t_f + 3a_3 t_f^2. \tag{3.6}$$

Equations (3.3)-(3.6) can be represented in a matrix ways as:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \\ \theta_f \\ \dot{\theta}_f \end{bmatrix}. \tag{3.7}$$

The simple example to solve (3.7) is consider the initial and final velocities are equal to zero. Suppose we have $t_0 = 0$, and $t_f = 1$ *sec*, with $\dot{\theta}_0 = 0$, and $\dot{\theta}_f = 0$. Replacing these values in equation (3.7) we obtain:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ 0 \\ \theta_f \\ 0 \end{bmatrix}. \tag{3.8}$$

This the equivalent to have the the following four equations:

$$a_0 = \theta_0. \tag{3.9}$$

$$a_1 = 0. \tag{3.10}$$

$$a_2 + a_3 = \theta_f - \theta_0. \tag{3.11}$$

$$2a_2 + 3a_3 = 0. \tag{3.12}$$

that later can be solved to obtain the following equations:

$$a_2 = 3\left(\theta_f - \theta_0\right). \tag{3.13}$$

$$a_3 = -2\left(\theta_f - \theta_0\right). \tag{3.14}$$

Then, replacing all values of a_0 to a_3 into equations (3.1) and (3.2), the representative polynomial function of the cubic interpolator is obtained as:

$$\theta_{(t)} = q_0 + 3\left(\theta_f - \theta_0\right)t^2 - 2\left(\theta_f - \theta_0\right)t^3. \tag{3.15}$$

$$\dot{\theta}_{(t)} = 6\left(\theta_f - \theta_0\right)t - 6\left(\theta_f - \theta_0\right)t^2. \tag{3.16}$$

$$\ddot{\theta}_{(t)} = 6\left(\theta_f - \theta_0\right) - 12\left(\theta_f - \theta_0\right)t. \tag{3.17}$$

3.2 Considerations in the trajectory to consider for mass-production

The paper dynamics are taken in account to enhance the robot trajectory, and feedback-error learning based control was used to improve the accuracy of the actuators and increase the performance of the robot for mass-production processes.

3.2.1 Spring-back

Paper-like materials are composed mainly by cellulose fibers, and considered to be orthotropic materials, this means that their mechanical properties are defined depending on the orientation of the sheet [61]. Due to the manufacturing process of a orthotropic material, their mechanical properties are defined in machine (MD), cross (CD), and thickness (ZD) directions, see Fig.3.4. The calculations in this paper were done considering only the CD properties, in order to consider the sheet of paper as a isotropic material. However, the proper selection of the direction still an on-going research by the moment.

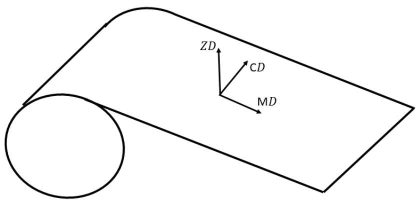


Figure 3.4: Paper fabrication orthotropic directions.

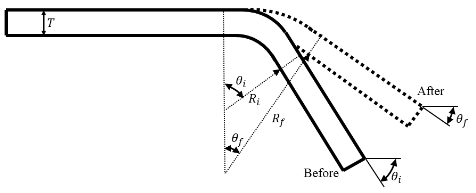


Figure 3.5: Spring-back effect in pure bending.

Spring-back can be described as a deviation of the sheet after the bending force is unloaded. All work piece materials have a finite modulus of elasticity, so each will undergo a certain elastic recovery upon unloading. This angle of recovery is known as spring-back, and can be used to calculate deformations in flexible materials such as paper. The spring-back effect is characterized using the spring-back factor as [50]:

$$K_s = \frac{\theta_f}{\theta_i}. \tag{3.18}$$

where θ_i denotes the bend angle prior to the removal of forces from the sheet, and θ_f is the bend angle after elastic recovery. A spring-back factor of unity means that the deformed sheet is perfectly plastic, where no elastic recovery has occurred. On the other hand, $K_s = 0$ corresponds to complete elastic recovery.

In a pure bending scenario (see Fig. 3.5), where the the neutral axis is located through the middle of the sheet, the spring-back is calculated as:

$$\frac{R_i}{R_f} = 4\left(\frac{R_i\sigma_y}{ET}\right)^3 - 3\left(\frac{R_i\sigma_y}{ET}\right) + 1. \tag{3.19}$$

where Ri and Rf are the initial and final bend radii, T is the sheet thickness, E is the Young's modulus, and σ_y is the yield strength. Equation(3.18) can be related to k_s, using the bend allowance, which is the arc length of the bend along the neutral axis, and calculated as:

$$Bend\ allowance = \left(R_i + \frac{T}{2}\right)\theta_i = \left(R_f + \frac{T}{2}\right)\theta_f. \tag{3.20}$$

arranging Eq.(3.20):

$$k_s = \frac{\theta_f}{\theta_i} = \frac{2\,(R_i/T) + 1}{2\,(R_f/T) + 1}. \tag{3.21}$$

Using R_i(in this paper is the radii of the edge's curvature of the folding base) and Eq.(3.18), R_f can be obtained. Then, using θ_i (in this paper is equal to $3\pi/4$) and Eq.(3.22), the value of θ_f can be estimated. The value of θ_f represent the spring-back recovery angle, that can serve us to calculate the recoil distance D_r of the pre-folded crease pattern after exits the folding stage as follows:

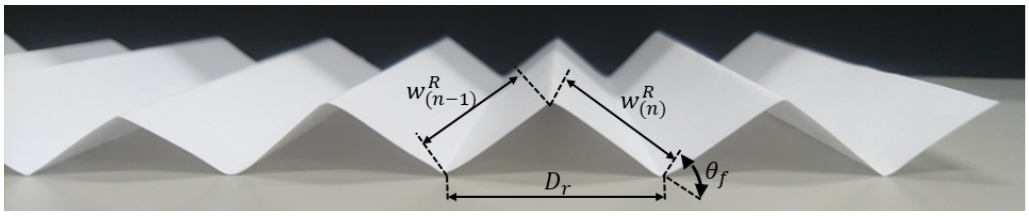


Figure 3.6: Effects of the spring-back in the crease pattern

$$D_r = \sqrt{(w^R_{(n-1)})^2 + (w^R_{(n)})^2 - 2w^R_{(n-1)}w^R_{(n)}\cos(\pi - \theta_f)}. \tag{3.22}$$

where $w^R_{(n-1)}$ and $w^R_{(n)}$ are the real width distances between folds of the previous and actual crease lines respectively. This real width are acquired from the advancing sensor in the first stage (i.e. The advancing sensor, Fig. 3.2:F4). This distance is computed from every step of the second stage and it let us know the exact position to stop the forward-backward system in Fig.3.3, step H.

3.2.2 Stacking effect

The staking effect occurs when the paper is folded multiple times, generating a pile with the double thickness every fold. Due to the particular design of the folding pattern (see chapter 5), a stacking effect is generated between steps I-K of the folding process (see Fig.3.3:G-L). This accumulation of thickness in the bottom wall has to be taken in account to modified the trajectories of the robot as in Fig.3.7. If this accumulation is not take in account, the robot could get stuck in certain moment of the folding process due to the pressure generated between the folding device, the paper and the bottom wall. The modifications to the global trajectory is represented as follows:

$$D_{s(n)} = \begin{cases} 0; & n = 0, \\ D_{s(n-1)} + 2T; & n > 0. \end{cases} \tag{3.23}$$

All the distances obtained in equations (3.22) and (3.23) are converted into angles to be introduced into the calculations of the trajectory. Having all trajectory considerations (i.e Spring-back distance, and stacking effect distance), the trajectory of the gluing finger from the gluing step can be represented as follows:

$$\theta_{(n)} = \begin{cases} \frac{D_r}{2} + D_i; & n = 0, \\ D_r + D_{s(n)} + D_i; & n > 0. \end{cases} \tag{3.24}$$

where D_i represent the width of the glue-distribution system.

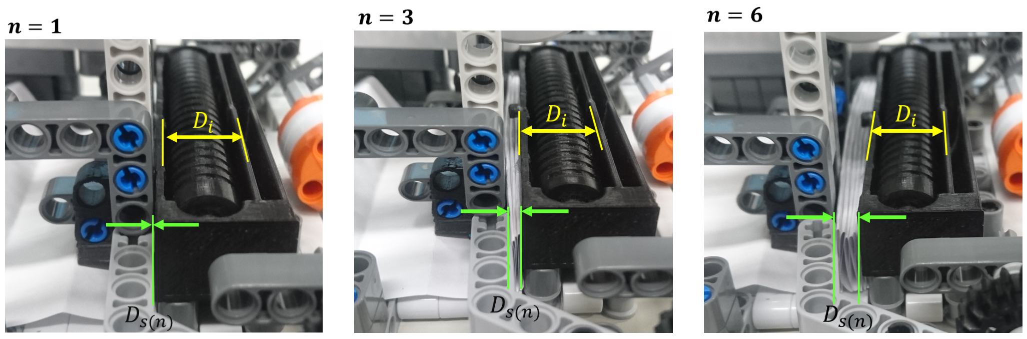


Figure 3.7: Stacking effect in the folding-gluing stage. From left to right: at $n = 1$, at $n = 3$, and at $n = 6$.

3.3 Trajectory control based on feedback-error learning

In chapter 4 a deep explanation of the control scheme selected in this paper is explained. The control scheme used in this chapter is based on feedback-error learning control (FEL). The advantage of FEL over traditional feedback control lies in the improved trajectory following, and impulse response. The use of FEL increases the performance of the system, which reduces the responsive time without losing precision. Many improvements to the original FEL since its creation have been made, from changing the feedback controller, to the use of alternatives for the feed-forward controller. In this work, a FEL controllers was applied into the robot's trajectory control. This FEL controller uses a traditional proportional-integrative-derivative(PID) as feedback controller, and an adaptive feed-forward controller based in holographic neural networks (HNN) created by [65].

In cases when a NN is used as Feed-forward controller, is recommended to use NN where a small number of computations have to be done. This means that NNs with a large number of layers or neurons have to be avoided. The HNN are a very special NN which, due to its properties, require an small amount of data to converge, performing few computations, therefore making it appropriate for FEL.

3.4 Creating a 3D shape with the proposed robot

The experiments carried out in this chapter have the following steps: First, the pattern is created using the SOR methodology by specifying a target profile and rotate it by a number of N segments (see chapter 5). The results of the applied SOR methodology are the segment width and the stamp shape to be allocated in the gluing stamp location(Fig.3.2:G3). This information serve us to generate the trajectories of the folding stage of the robot(Fig.3.2:F). After the paper is pre-folded in the folding stage enters to the folding-gluing stage (Fig.3.2:G). To create the trajectories of this stage, information from the folding step is used. Using the information of the

Table 3.2: Experiment parameters

Parameter	**Symbol**	**Value**	**Unit**
Thickness	T	0.98×10^{-3}	m
Young's Modulus of paper CD	E	4.94×10^{9}	Pa
Yield strength of paper CD	σ_y	$5.03e \times 10^{7}$	Pa
Width	W	26×10^{-3}	m
Number of segments	N	10	–

output widths from the folding stage, the spring-back effect can be estimated and used to generate the trajectory of the folding-gluing stage. This value is calculated using the mechanical properties of the paper (i.e. paper type, and thickness), and the real values of the widths obtained from the advancing sensor(Fig.3.2:F4) in the folding stage. Both trajectories are loaded into the corresponding NXT Brain-Bricks (Fig.3.2:F5 and G7), the information form the first brick is transmitted into the second brick through Bluetooth communication. Each one of the parameters(i.e. FEL with HNN, spring-back effect, and stacking effect) are individually measured and compared with the expected result to observe their effects in the final 3D shape. The error varies depending on the gluing area. If the gluing area is wide, the error due to variations in the line alignment increases. All these experiments were carried out with white Bond paper sheets, and using a 10-side box, similar to the one in Fig.5.8.(b). For the experiments with HNN, the holographic memory was initially set with matrices of zeros ($H_{(t=0)} = 0$ and $G_{(t=0)} = 0$). The parameters of the crease pattern and paper for these experiments are exposed in Table3.2:

3.4.1 Effects of the paper dynamics in the robot trajectory

Here, the effects of the dynamics of the paper in the robot trajectory calculations are analyzed. For this experiments, the effects of each one of the parameters that affect the trajectory calculations, i.e. FEL with and without HNN, the spring-back and stacking effects, are analyzed. Using equations (2.1) and (2.2), the error reduction from each one of these parameter can be estimated by comparing the error between the expected area and the resulting area using equations (2.1) and (2.2). Two PIDs were used as feedback controllers, one for the first stage and other for the second stage, with: $P_1 = 55, I_1 = 75, D_1 = 0.1$, and $P_2 = 40, I_2 = 20, and\ D_2 = 0$. The sample time was set to $0.05s$. Initially, the error of the proposed machine using only the PID controller is calculated. Then, one by one, each parameter was added, to estimate the error reduction. The results of these experiments can be observed in Table.3.3.

The error percentage($E\%$) was calculated as: $E\% = \left|100\frac{Real\ Area}{Expected\ Area} - 100\right|$. Although the first three experiments were carried out without the stacking effect calculation, it is recommended to do it using this parameter to prevent unexpected

Table 3.3: Effects of the paper in the final shape

Trajectory	**E1**(m)	**E2**(m)	**Expected Area**(m^2)	**Real Area**(m^2)	**Error** %
PID	2.9096×10^{-7}	1.8970×10^{-5}	0.0260	0.0260	0.1778
PID+HNN	6.4258×10^{-8}	1.7414×10^{-5}	0.0260	0.0260	0.1614
PID+HNN+Spring back	6.4258×10^{-8}	1.2979×10^{-5}	0.0260	0.0260	0.1204
PID+HNN+Spring back+Stacking	6.4258×10^{-8}	1.2657×10^{-5}	0.0260	0.0260	0.1174

Table 3.4: Effects of increasing the shape's creation speed ($E\%$)

	Speed multiplier					
Controller	**1.0X**	**1.2X**	**1.4X**	**1.6X**	**1.8X**	**2.0X**
PID+Spring back+Stacking	0.1192	0.1355	0.1908	0.2369	2.3397	2.3208
PID+HNN+Spring back+Stacking	0.1174	0.1185	0.1261	0.1405	2.1796	2.2011

stops in the robot's movement. We can observe that all the parameters reduce the error in the trajectories, been the spring-back effect calculation the parameter that reduces the error further.

3.4.2 Effects of increasing the speed of the shape's creation

Other objective in this dissertation is to use the proposed robot for mass-production procedures. Due to the increases in a shape's creation, the error increases to. Here, the use of FEL control was proposed to counter the increases in error due to the speed changes. Table.3.4 shows the results of increasing the speed in the shape's creation process, using the same shape and paper's type from the past experiment (see table. 3.2).

The error requirements vary depending on each application and 3D shape. However, as can be observed in table.3.4, the use of the HNN as FEL controller increases the performance of the machine, allowing to create more shapes in less time.

3.4.3 Resulting shapes build by the proposed robot

To demonstrate the applicability of the proposed robot, three different shapes, a pyramid, a hexagonal box, and a dome, were automatically folded. Their corresponding crease pattern, expected 3D shape, and final product are show in Fig.5.8.

The complexity reduction is possible due to adding gluing areas into the proposed crease pattern, that reduce movements in the robot by executing multiple tasks at the same time. A manipulation is considered as a combination of movements required to execute a task, e.g. a fold, a turn over, a rotation, or a squash [69]. Table.5.1 expose the number of manipulations in a group of crease patterns folded by hand and the proposed robot.

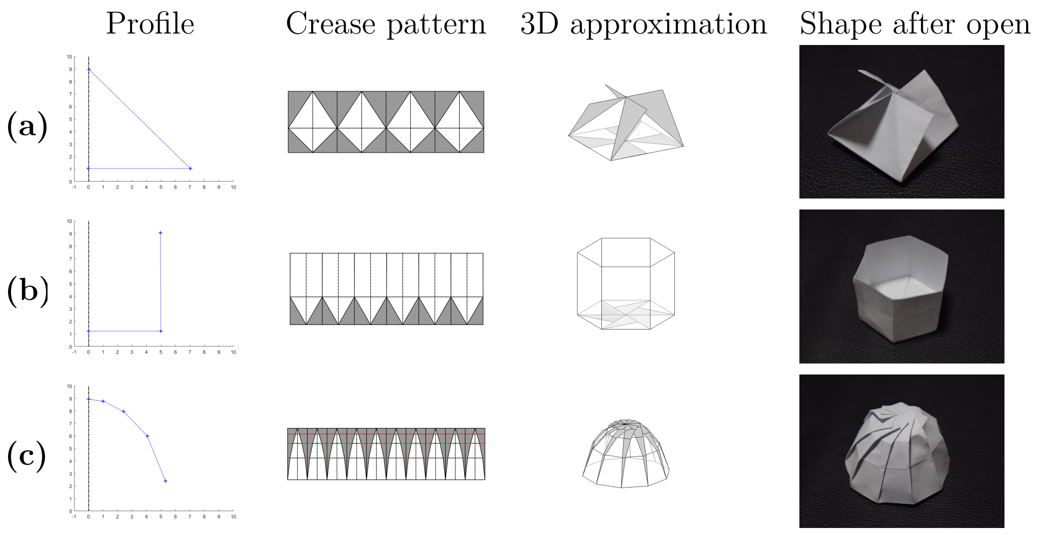


Figure 3.8: Result profiles, patterns and 3D shapes. (a) Pyramid ($N = 4, W = 14.1mm$), (b) Hexagonal box ($N = 6, W = 11.14mm$), (c) Dome($N = 10, W = 7.0mm$).

Table 3.5: Number of manipulations.

Pattern Name	Human's # manipulations	Robot's # manipulations	# of manipulations reduced
Pyramid (Fig.5.8(a)	40	16	24
Box (Fig.5.8(b)	36	18	18
Dome (Fig.5.8(c)	100	20	80

3.5 Summary

In this chapter, a folding-gluing robot able to build a 3D shape of a SOR-based crease pattern has been developed. Here, we show that using simple folds in combination with gluing areas, interesting 3D shapes can be achieve with the proposed robot (see Fig.5.8). The crease pattern used by this robot is based in a SOR methodology created in [42]. Several parameters of the crease pattern and paper type were used to calculate a series of trajectory enhancers, i.e. a FEL controller with HNN, and a spring-back and stacking effects calculations. The results shows that each one of these enhancers reduces the ME, produced by displacements or slips in the paper sheet. In order to analyze the performance of the robot for mass-production purposes, a FEL controller using a HNN was proposed. As can be observed in table.3.4, the HNN keeps the displacements controlled, even in twice the initial speed (2.0X). This permit us to reduce the production time because more shapes can be created in less time preserving the quality within a frame.

Chapter 4

Control scheme applied into the proposed robot

A paper-folding robot was developed in order to transform a flat sheet of paper into a 3D object, without human assistant. However, it is extremely difficult to introduce automation in robots to handle shape-changing objects such as sheets of paper. There are two main problems associated with paper handling. One of them is controlling deformation of objects that has infinite degrees of freedom using a finite number of manipulated variables. The other problem is how to handle fragile material such as paper without exerting excessive stress, i.e. how to handle it safely and reliably. Scientists have tried to solve these problems using control models based on the behavior of human brains. One of the best approaches comes from neuroscience. With the use of a control system known as cerebellar control model [73, 29], which is inspired by the human brain, precise and fast movements of a robotic arm can be performed. Kawato et.al [29, 28, 44], have proposed a cerebellar feedback-error learning model to solve control problems. However, few applications of folding robots using this control theory have been developed. In this chapter, the dynamic representation of the LEGO robot is exposed. A detailed explanation of the torque calculations using the Euler-Lagrange methodology is explained. Apart of the mechanical calculations, electric characteristics of the LEGO NXT MINSTORMS ® servomotor was included to be able to implement force-based control into the proposed robot. In this chapter, a new method that uses feedback error learning (FEL) to control the proposed robot is proposed. Adaptive learning algorithms are implemented and compared using several artificial neural networks to perform precise and smooth manipulations of paper sheets. Several feed-forward controllers based in artificial neural networks are tested and compared with other types of classic controllers. Adaptive algorithms were evaluated, to improve the convergence of the tested NNs and ensure stability of the controlled system.

4.1 Dynamic model of the proposed robot

The robot dynamic model of the robot is very important to understand the real behavior of the system, and also it is an indispensable tool to apply a control scheme into the robot and perform computational simulations. In almost all robotic applications, the equations of motion are described by non-linear differential equations. Because no-closed solution is available, the equations of motion are studied using a numerical method [7]. to properly understand the procedure to obtain the dynamics of a system, a practical example using a two-degree of freedom (DOF) manipulator is explained. For this example, the total potential and kinetic energies of the robot links are defined and used to form a Lagrangian, and then the Euler-Lagrange methodology is used to define the torques applied to each link [38]. Electric parameters from the LEGO MINDSTORMS NXT® servomotors are taken in account to perform better approach to a real behavior in the simulation process.

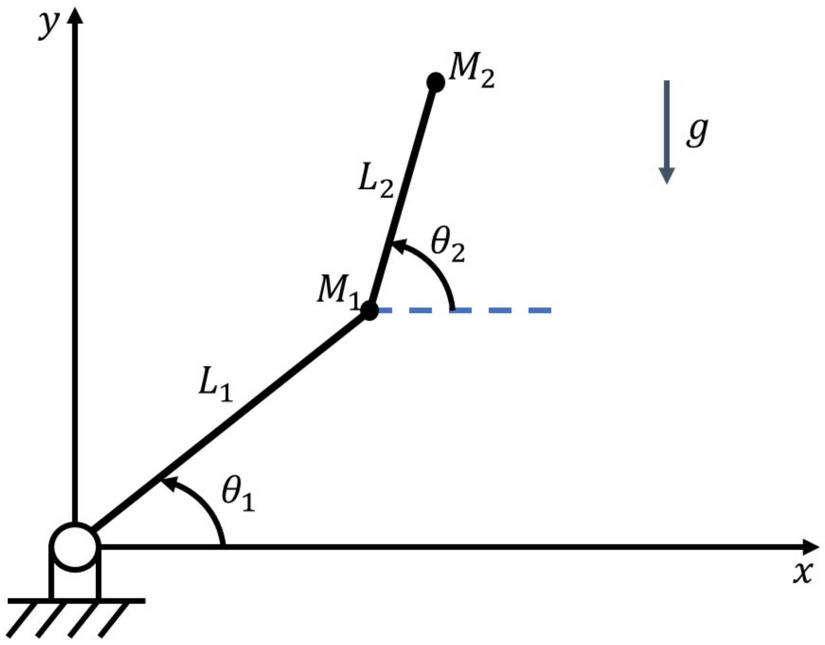


Figure 4.1: Graphic representation of a Two-DOF manipulator.

For the 2 DOF robot shown in Fig. 4.1, its current X-Y positions for each link can be obtained using the following geometrical equations:

$$x_1 = Lc_1 sin\theta_1; \quad y_1 = -Lc_1 cos\theta_1. \tag{4.1}$$

$$x_2 = L_1 sin\theta_1 + Lc_2 sin(\theta_1 + \theta_2); \quad y_2 = -L_1 cos\theta_1 - Lc_2 cos(\theta_1 + \theta_2). \tag{4.2}$$

where L_1 and L_2 are the links longitude, Lc_1 and Lc_2 are the distances from the join to each link´s mass center, and θ_1 and θ_2 are the turn angles of each link.

Using equations (4.1) and (4.2), the squared linear velocity for both links(v_1^2 and v_2^2) can be calculated using the following equations:

$$v_1^2 = \left(\tfrac{dx_1}{dt}\right)^2 + \left(\tfrac{dy_1}{dt}\right)^2; \quad v_2^2 = \left(\tfrac{dx_2}{dt}\right)^2 + \left(\tfrac{dy_2}{dt}\right)^2. \tag{4.3}$$

Using the Newton-Euler equations the kinematic energy can be calculated as follows:

$$KE_1 = \tfrac{1}{2}M_1v_1^2 + \tfrac{1}{2}I_1w_1^2; \quad KE_2 = \tfrac{1}{2}M_2v_2^2 + \tfrac{1}{2}I_2w_2^2. \tag{4.4}$$

$$KE_T = KE_1 + KE_2. \tag{4.5}$$

where M_1 and M_2 are the mass of each link, v_1^2 and v_2^2 are the square linear velocity calculated using (4.3), I_1 and I_2 are the moment of inertia of each link, and w_1^2 and w_2^2 are the angular velocities. In order to calculate the Lagrangian, the potential energy has to be calculated using the fallowing equation:

$$UE_1 = -M_1gy_1; \quad UE_2 = -M_2gy_2. \tag{4.6}$$

$$UE_T = UE_1 + UE_2. \tag{4.7}$$

where g is the gravitational acceleration. Using (4.5) and (4.7). the Lagrangian can be calculated using the following equation:

$$L = KE_T - UE_T. \tag{4.8}$$

With the Lagrangian, the required Torque of each link can be calculated using the following equations.

$$\tau_1 = \tfrac{d}{dt}\left(\tfrac{\partial L}{\partial w_1}\right) - \tfrac{\partial L}{\partial \theta_1}; \quad \tau_2 = \tfrac{d}{dt}\left(\tfrac{\partial L}{\partial w_2}\right) - \tfrac{\partial L}{\partial \theta_2}. \tag{4.9}$$

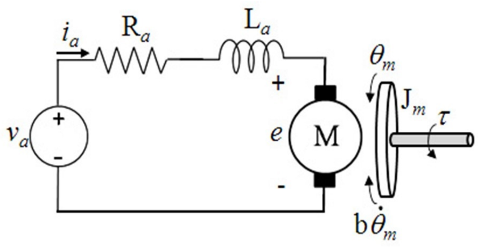
After applying (4.9), the resulting group of equations can be represented as the following general form:

$$\boldsymbol{\tau} = \boldsymbol{M}_{(\theta)}\ddot{\theta} + \boldsymbol{C}_{(\theta,\dot{\theta})}\dot{\theta} + \boldsymbol{G}_{(\theta)}. \tag{4.10}$$

where $\boldsymbol{M}_{(\theta)}$ is the system's inertial matrix, $\boldsymbol{C}_{(\theta,\dot{\theta})}\dot{\theta}$ represent the Coriolis forces matrix, and $\boldsymbol{G}_{(\theta)}$ is the gravity matrix. The vector $\boldsymbol{\tau}$ includes the required torque for each link. The input into the LEGO MINDSTORMS NXT® has to be given in volts, and not in torque. That is why the model in 4.10 has to be modified to include the electrical calculations from servomotors.

LEGO MINDSTORM NXT® provides servomotors (see Fig.4.2a) with a build-in rotational sensor incorporated. These servomotors can be resumed as a direct current (DC) motor model, presented in Fig.4.2b, where V_a is the applied DC voltage, R_a is the equivalent resistor, L_a is the equivalent inductance, i_a is the electric current, and e is the back electromotive force(EMF) voltage expressed by:

(a)



(b)

Figure 4.2: LEGO MINDSTORMS NXT® servomotors. (a) LEGO servomotor. (b) Electric schematic of a DC motor.

$$V_a = e + R_a i_a + L_a \frac{\partial i_a}{\partial t}. \tag{4.11}$$

The motor can supply a torque $\boldsymbol{\tau}$ and the load has a moment of inertia J_m, obtained from (4.10). The Current i_a is related with the developed torque τ_d by (4.12), and the back EMF voltage is related with angular speed $\dot{\theta}$ and the gear ratio n in (4.13), where k_b and k_t are motor constants that can be found by an experimental setup [24, 74].

$$\tau_d = k_t i_a. \tag{4.12}$$

$$e = n k_b \dot{\theta}. \tag{4.13}$$

Using (4.11), and if the inductance inside the circuit is negligible the current can be expressed as:

$$i_a = \frac{V_a - n k_b \dot{\theta}}{R_a}. \tag{4.14}$$

The final torque $\boldsymbol{\tau}$ that has to be applied into the load is computed by subtracting the friction force f_m from the developed torque τ_d:

$$\boldsymbol{\tau} = \tau_d - f_m. \tag{4.15}$$

Finally, using equations (4.12), (4.14) and (4.15), the equation that relates the applied torque with the DC voltage applied into the LEGO MINDSTORMS NXT® servomotors is obtained.

$$\boldsymbol{\tau} = \frac{n k_t}{R_a} V_a - \frac{n k_t k_b}{R_a} \dot{\theta} - f_m. \tag{4.16}$$

After the representative function of the system is created, i.e. equation where you can find the inputs and the outputs of the robot; the system has to be represented in a time domain form. There are several ways to represent functions in time domain [52], one way is to transform it into a differential equation, other way is to create the transfer or differential function of the system, and other way is to represented in the space of states. For this dissertation, the system is represented in space state form to be simulated using Matlab® with Simulink®.

Any system can be represented in space of states (SS) and have the form observed in Fig.4.3:

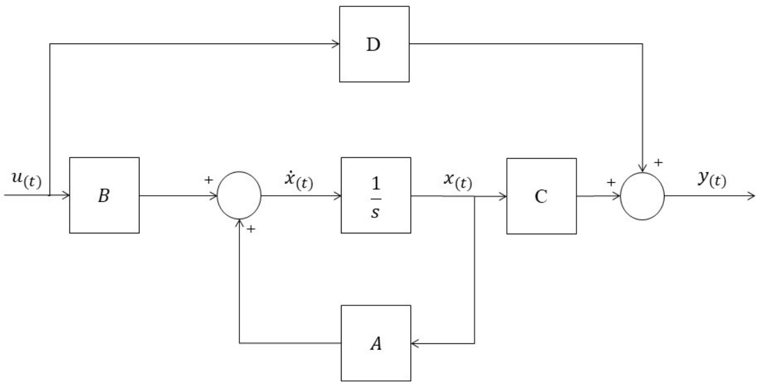


Figure 4.3: Graphic representation of SS system.

$$\dot{x}_{(t)} = \boldsymbol{A}x_{(t)} + \boldsymbol{B}u_{(t)}. \tag{4.17}$$

$$y_{(t)} = \boldsymbol{C}x_{(t)} + \boldsymbol{D}u_{(t)}. \tag{4.18}$$

where $\dot{x}_{(t)}$ is the highest degree derivative in the time t, $x_{(t)}$ is the vector of state variables (i.e. $\dot{\theta}$, θ), u is the vector of inputs (V_a), $y_{(t)}$ is the vector of outputs, and $\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{C}$, and $\boldsymbol{D}$ are matrices with the proper dimensions. Using the Tailor´s series approximation in a work point, a non-linear system represented by equalizing and sorting equations (4.10) and (4.16) can be linearized, and the matrices $\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{C}$, and $\boldsymbol{D}$ can be calculated as follow:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial \dot{x}_{(1)}}{\partial x_{(1)}} & \frac{\partial \dot{x}_{(1)}}{\partial x_{(2)}} & \cdots & \frac{\partial \dot{x}_{(1)}}{\partial x_{(n)}} \\ \frac{\partial \dot{x}_{(2)}}{\partial x_{(1)}} & \frac{\partial \dot{x}_{(2)}}{\partial x_{(2)}} & \cdots & \frac{\partial \dot{x}_{(2)}}{\partial x_{(n)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \dot{x}_{(n)}}{\partial x_{(1)}} & \frac{\partial \dot{x}_{(n)}}{\partial x_{(2)}} & \cdots & \frac{\partial \dot{x}_{(n)}}{\partial x_{(n)}} \end{bmatrix}; \quad \boldsymbol{B} = \begin{bmatrix} \frac{\partial \dot{x}_{(1)}}{\partial u_{(1)}} & \frac{\partial \dot{x}_{(1)}}{\partial u_{(2)}} & \cdots & \frac{\partial \dot{x}_{(1)}}{\partial u_{(m)}} \\ \frac{\partial \dot{x}_{(2)}}{\partial u_{(1)}} & \frac{\partial \dot{x}_{(2)}}{\partial u_{(2)}} & \cdots & \frac{\partial \dot{x}_{(2)}}{\partial u_{(m)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \dot{x}_{(n)}}{\partial u_{(1)}} & \frac{\partial \dot{x}_{(n)}}{\partial u_{(2)}} & \cdots & \frac{\partial \dot{x}_{(n)}}{\partial u_{(m)}} \end{bmatrix}. \tag{4.19}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}; \quad \boldsymbol{D} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{4.20}$$

As can be observed in equation (4.20), the matrix $\boldsymbol{D}$ is composed by zeros, with a size of $m \times r$, where m is the number of inputs, and r is the number of outputs. This can be modified if the disturbance or noise caused by the input values is taken in account, however, for practical effects this matrix is set to zeros. After having the equations in SS form(i.e. with the form of equations (4.17) and (4.18)), the analysis of the stability, observability, and controlability of the system can be perform.

4.2 Analysis of the dynamic of the system

For a system with a form such as equations (4.17) and (4.18), we can say that the state $x_{(t)}$ is controllable in $t = t_0$, if there is a continuous input of data $u_{(t)}$, which moves the the states to $x_{(t_f)}$, in a interval of finite time $(t_f - t_0) \geq 0$. If any state $x_{(t_0)}$ of the system is controllable in any interval of time, the system is consider to be completely controllable.

For a system described by the functions (4.17) and (4.18), the system is considered completely controllable if the matrix of controllability $\boldsymbol{M}$, with size $n \times m$, has rank equal to n.

$$\boldsymbol{M} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}. \tag{4.21}$$

Despite this method can be applied directly to obtain the controllability of the system, sometimes is difficult to use it with high-grade systems. In cases where the matrix $\boldsymbol{M}$ is not square, the matrix $\boldsymbol{M}\boldsymbol{M}^T$ (with size of $n \times n$) can be calculated. If the matrix is not singular, the system is controllable.

There is other characteristic of the systems in SS representation, and its the observability. For a system described by the functions (4.17) and (4.18), it is considered fully observable, if the observability matrix $\boldsymbol{S}$, with size $n \times nr$, has rank equal to n;

$$\boldsymbol{S} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}. \tag{4.22}$$

The next step is to obtain the system' s stability. This can be checked using the eigenvalues of the $\boldsymbol{A}$ matrix. All the resulting eigenvalues have to be negative different from zero if the system is stable, otherwise the system is unstable.

4.3 Basis of Feedback-Error Learning controllers

Based on forward and inverse models of the cerebellum, Kawato and partners proposed the FEL control model in [29] and [44]. This model comprised a fixed feedback controller (e.g. PID controller), that ensured the stability of the system, and an adaptive feed-forward controller that improves the control performance, usually an artificial neural network (NN), see Fig.**??**. The advantage of FEL over traditional feedback control lies in the improved trajectory following, and impulse response. The use of FEL increases the performance of the system, which reduces the responsive time without losing precision. Many improvements to the original work of Kawato since its creation have been made, from changing the feedback controller, to the use of alternatives for the feed-forward controller. In this thesis, a FEL controllers was applied into the robot's trajectory control. This FEL controller uses a traditional proportional-integrative-derivative(PID) as feedback controller, and an adaptive feed-forward controller based in artificial neural networks.

The work developed by Ruan et al. [59], can be considered as the most updated version of a FEL controller. Their work in [59] used an online back-propagation algorithm with a self-adaptive learning rate, to realize the combination of learning and control within the FEL scheme. A self-adaptive learning rate algorithm similar to the one used by Ruan et. al. is applied to our proposed robot and compared with other adaptive training algorithms later in this chapter.

4.4 Feedback controller

For the feedback controller design, we focus on the original works from Kawato's group [29, 28, 44], were the feedback controller consist in a traditional proportional-integrative-derivative (PID) controller.

The PID controllers are the most commonly used controllers in the industry, due to their easy implementation and versatility. A PID controller consists on the clustering of three different control actions: A proportional action (P), an integrative action (I), and a derivative action (D). The equation is represented as follows:

$$u_{(t)} = k_p e_{(t)} + \frac{k_i}{T_i} \int_0^t e_{(\tau)} d\tau + k_d T_d \frac{de_{(t)}}{dt}. \tag{4.23}$$

where k_p, k_i, and k_d are the proportional, integral and derivative constant gains, T_i and T_d are the integral and derivative times (commonly set to 1), $e_{(t)}$ is the error (difference between the desired value and the current output of the plant); and $u_{(t)}$ is the control action, which is also called the motor command. One of the main problems of PID controllers lies on performing a proper tuning of the constant gains k_p, k_i, and k_d. There are many methods to perform a tuning of these constants, however, in this work, a classic Ziegler and Nichols tuning method was used [76].

4.5 Artificial Neural Networks

The Artificial neural networks (ANNs), are computing systems inspired by the biological neural networks present in animals in nature. Such systems learn to do specific tasks by considering a series of samples (a.k.a training samples). An ANN is based on a collection of connected units called artificial neurons (analogous to biological neurons in an animal brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have a state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends through the NN. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times.

In the neural network research field, back-propagation neural networks (BPNNs) are the most popular models [72, 54]. The efficient supervised training of BPNNs is a subject of considerable ongoing research and numerous algorithms have been proposed for this matter. A common training approach, is to minimize the network learning error, which is a measure of its performance, and is usually based on the difference between the actual output vector of the network and the desired output vector (a.k.a supervised learning). The rapid computation of a set of weights that minimizes this error is a rather difficult task since, in general, the number of network weights is high and the error function generates a complicated surface in the weight space, possessing multitudes of local minima and having broad flat regions adjoined to narrow steep ones that need to be searched to locate an "optimal" weight set.

Applications of supervised learning can be divided in two categories: stochastic (a.k.a on-line) and batch (a.k.a off-line) learning. Batch supervised learning is the classical approach in machine learning. In this type of learning, a set of examples is obtained and used in order to learn a good approximating function (i.e. train the network), before the network is used in the application. On the other hand, in on-line learning, data gathered during the normal operation of the system are used to continuously adapt the learned function (as in FEL controllers).

In this chapter, different types of NNs are analyzed and compared as feed-forward controllers for FEL. Several of these NNs use the stochastic back-propagation algorithm to train the networks. For these NNs, an adaptive learning rate algorithm is used to improve the accuracy of the FEL controller by reducing the convergence time.

4.5.1 Multiple Adaptive Linear Neurons (MADALINE)

The Multiple Adaptive Linear Neurons or MADALINE is a clustering of multiple Adaptive linear neurons, or ADALINE that constitute a basic adaptive NNs. ADALINE, is a single-input, single-output (SISO) neuron that uses the linear weighted sum of inputs (the net) to update weights. The ADALINE uses the Least Mean Squared (LMS) learning rule, also known as the Widrow-Hoff learning rule to train the network [63]. During training, weights, biases and outputs are adjusted at each step time based on new input and target vectors. The LMS algorithm is an example of supervised learning, in which the learning rule is provided with a set of examples of the desired network behavior. Having a vector of inputs $X(t)$, and another vector of desired outputs $T(t)$, in a sample time t. As each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output. The LMS algorithm adjusts the weights and biases of the ADALINE to minimize the mean squared error (MSE) [15]:

$$MSE = E(t) = \frac{1}{p}\sum_{t=1}^{p} e(t)^2 = \frac{1}{p}\sum_{t=1}^{p} \left(T(t) - Y(t)\right)^2 . \tag{4.24}$$

$$Y(t) = w_{(j,i)}(t)X(t) + b_{j,1}(t). \tag{4.25}$$

where $w_{(j,i)}(t)$ is the vector of weights, and $b_j(t)$ is the vector of biases, in the actual time t.

Training is performed online, i.e. the weights of the neural network are updated after presentation of each training example [19]. Classically, at iteration t a given weight $w_{(j,i)}(t)$ is updated by adding a $\Delta w_{(j,i)}(t)$ to it:

$$\Delta w_{(j,i)}(t) = -\alpha \cdot \frac{\partial E(t)}{\partial w_{(j,i)}(t)}. \tag{4.26}$$

where α is the global learning rate (for this example the learning rate is considered to be the same to all layers).

Solving equation (4.26), the following group of equations, that represent the update rule in a discrete time, are obtained:

$$w_{(j,i)}(t+1) = w_{(j,i)}(t) + 2\alpha \left(T(t) - Y(t)\right) X(t)^T. \tag{4.27}$$

$$b_{j,1}(t+1) = b_{j,1}(t) + 2\alpha \left(T(t) - Y(t)\right). \tag{4.28}$$

where $(\cdot)^T$ represents a transpose.

4.5.2 Multiple layered Perceptron (MLP)

The multiple-layered perceptron (MLP) NNs, use an adapted version of the algorithms used for MADALINE. Due to the new layers included in this architecture the back-propagation algorithm is applied to obtain the updated rule of the weights in the initial layers. Having vector of inputs $X(t)$, and another vector of desired outputs $T(t)$, in a sample time t. The error has to be minimized applying a gradient decedent algorithm on each layer. Another difference between MLP and MADALINE is that, MLP usually uses different type of optimization function. The most commonly used, and the one that we use for the following example, is the sigmoid function. Having a MLP with one input layer, one hidden layer (sigmoid) and one output layer (linear), using a on-line training, for the hidden layer we have:

$$Z(t) = \frac{1}{1 + e^{\left(-u_{h(1,k)}(t)\right)}}. \tag{4.29}$$

$$u_{h(k,1)}(t) = w_{h(k,i)}(t)X(t) + b_{h(k,1)}(t). \tag{4.30}$$

where $w_{h(k,i)}(t)$, $b_{h(k,1)}(t)$ are the weights and biases vectors of the hidden layer, i represent the number of neurons in the input layer, k is the number of neurons in the hidden layer, j is the number of neurons in the output layer, and $Z(t)$ is the output of the hidden layer. For the output layer we have that:

$$Y(t) = \frac{1}{1 + e^{\left(-u_{o(1,j)}(t)\right)}}. \tag{4.31}$$

$$u_{o(j,1)}(t) = w_{o(j,k)}(t)Z(t) + b_{o(j,1)}(t). \tag{4.32}$$

where $w_{o(j,k)}(t)$, and $b_{o(j,1)}(t)$ are the weights and biases vectors of the output layer, and $Y(t)$ is the output vector in the actual time t. Now applying the back-propagated error algorithms in order to update the weights and biases using gradient descendant we have for each layer:

$$\begin{aligned} -\Delta w_{o(j,k)}(t) &= -\alpha \cdot \frac{\partial E_o(t)}{\partial w_{o(j,k)}(t)} = -\alpha \cdot \frac{\partial E_o(t)}{\partial Y(t)} \cdot \frac{\partial Y(t)}{\partial u_{o(j,1)}(t)} \cdot \frac{\partial u_{o(j,1)}(t)}{\partial w_{o(j,k)}(t)} \\ &= -\alpha \left[-2(T(t) - Y(t)\right] \left[Y(t)(1 - Y(t))\right] \left[Z(t)^T\right]. \end{aligned} \tag{4.33}$$

$$\begin{aligned} -\Delta b_{o(j,1)}(t) &= -\alpha \cdot \frac{\partial E_o(t)}{\partial b_{o(j,1)}(t)} = -\alpha \cdot \frac{\partial E_o(t)}{\partial Y(t)} \cdot \frac{\partial Y(t)}{\partial u_{o(j,1)}(t)} \cdot \frac{\partial u_{o(j,1)}(t)}{\partial b_{o(j,1)}(t)} \\ &= -\alpha \left[-2(T(t) - Y(t)\right] \left[Y(t)(1 - Y(t))\right]. \end{aligned} \tag{4.34}$$

$$\begin{aligned} -\Delta w_{h(k,i)}(t) &= -\alpha \cdot \frac{\partial E_o(t)}{\partial w_{h(k,i)}(t)} = -\alpha \cdot \frac{\partial E_h(t)}{\partial Z(t)} \cdot \frac{\partial Z(t)}{\partial u_{h(k,1)}(t)} \cdot \frac{\partial u_{o(j,1)}(t)}{\partial w_{h(k,i)}(t)} \\ &= -\alpha \left[-2(T(t) - Y(t)\right] \left[Y(t)(1 - Y(t))\right] \frac{\partial u_{o(j,1)}(t)}{\partial w_{h(k,i)}(t)}. \end{aligned} \tag{4.35}$$

$$\begin{aligned} \frac{\partial u_{o(j,1)}(t)}{\partial w_{h(k,i)}(t)} &= \frac{\partial u_{o(j,1)}(t)}{\partial Z(t)} \cdot \frac{\partial Z(t)}{\partial u_{h(i,1)}(t)} \cdot \frac{\partial u_{h(i,1)}(t)}{\partial w_{h(k,i)}(t)} \\ &= [w_{o(j,k)}(t)][Z(t)(1 - Z(t))][X(t)^T]. \end{aligned} \tag{4.36}$$

$$\begin{aligned} -\Delta b_{h(k,1)}(t) &= -\alpha \cdot \frac{\partial E_o(t)}{\partial b_{h(k,1)}(t)} = -\alpha \cdot \frac{\partial E_h(t)}{\partial Z(t)} \cdot \frac{\partial Z(t)}{\partial u_{h(k,1)}(t)} \cdot \frac{\partial u_{o(j,1)}(t)}{\partial b_{h(k,1)}(t)} \\ &= -\alpha \left[-2(T(t) - Y(t)\right] \left[Y(t)(1 - Y(t))\right] \frac{\partial u_{o(j,1)}(t)}{\partial w_{h(k,i)}(t)}. \end{aligned} \tag{4.37}$$

$$\begin{aligned} \frac{\partial u_{o(j,1)}(t)}{\partial b_{h(k,1)}(t)} &= \frac{\partial u_{o(j,1)}(t)}{\partial Z(t)} \cdot \frac{\partial Z(t)}{\partial u_{h(i,1)}(t)} \cdot \frac{\partial u_{h(i,1)}(t)}{\partial b_{h(k,1)}(t)} \\ &= [w_{o(j,k)}(t)][Z(t)(1 - Z(t))]. \end{aligned} \tag{4.38}$$

A MLP with four layers (one input, two hidden, and one output) was used as feed-forward controller for FEL. The desired trajectories (one for each actuator), and their first and second derivatives were the inputs to the network. The first hidden layer has nine neurons, and a sigmoid transfer function was used. The second hidden layer also has nine neurons, and a sigmoid function was used as well. The output layer has 3 outputs (motor commands for each actuator), and uses a linear transfer function. The MLP architecture used in this thesis is shown in Fig. 4.4.

4.5.3 Adaptive Learning rate

In this chapter, a comparison of the behavior of three NN architectures is executed. Two of the NNs (MADALINE, and MLP) use static learning rates, which represents the learning speed in a training procedure. However, each one of these NNs architectures is designed and trained in different ways, and some features such as the sample time, reference frequency, number of weights, and even the PID parameters could affect the response of the training. In this chapter, the calculation of a global learning rate based on a modified “bold driver” algorithm [46], was implemented.

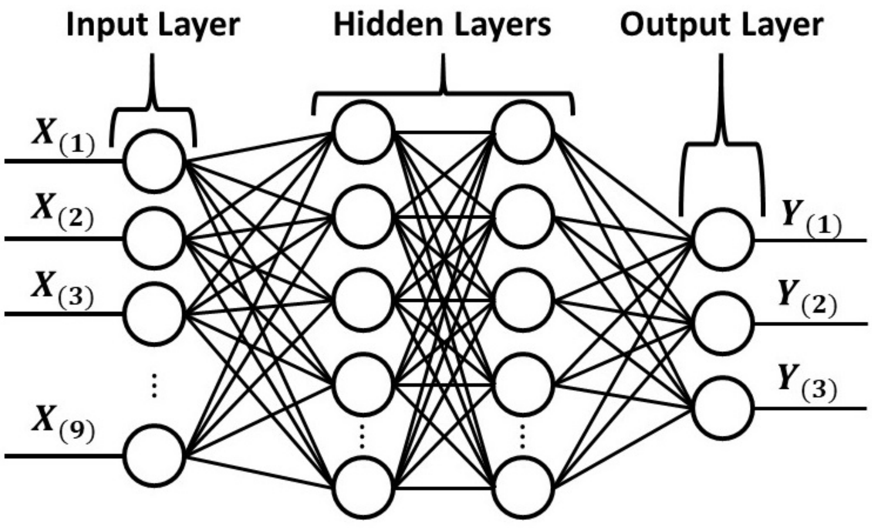


Figure 4.4: Multiple-layered perceptron used in this works.

However, the use of classic bold driver algorithm in on-line training such as FEL NNs, is not possible due to the interference between the present reference data and previous reference data, leading to saturation or low convergence. Consequently, a method able to support non-stationary (time-varying) data is required in this paper. Some of the best solutions to this problem came from variations in the on-line training methods proposed by Duffiner and Garcia [19]. The method used in this paper denoted as ALAP1, uses a common learning rate for all weights:

$$\alpha(t+1) = \alpha(t) + \gamma \left\langle \nabla E(t), \nabla E(t+1) \right\rangle . \tag{4.39}$$

where $\alpha(t = 0) = 10^{-7}$ for all weights, $\gamma = 0.01$ and $\langle \cdot, \cdot \rangle$ stands for the inner product in $\mathbb{R}^{n}$ [54].

4.5.4 Holographic Neural Networks

In cases when a NN is used as Feed-forward controller, is recommended to use NN where a small number of computations have to be done. This means that NNs with a large number of layers or neurons have to be avoided. The HNN are a very special NN which, due to its properties, require an small amount of data to converge, performing few computations, therefore making it appropriate for FEL.

The holographic method proceeds considerably beyond the neural paradigms. It can be used in divers applications like classifiers, or filters. The HNN are based on the fact that the input-output relationship is linearized according to a mapping of inputs into outputs at the complex plane. After such mapping, the HNN processes the information like a conventional real-valued NN except that all the parameters and variables are complex.

Suppose two real vectors X_i(a row vector with the desired trajectory X and its derivatives) and T_r (a row vector with all the feedback controller's motor commands). Each element of vector X_i and T_r is transformed into angles $\theta_k(k = 1, ..., l)$ and $\phi_j(j = 1, ..., m)$ by the transformation/mapping functions $f_{(X_i)}$ and $f_{(T_r)}$:

$$\theta_k = f_{(X_i)}, \quad \phi_j = f_{(T_r)}. \tag{4.40}$$

In this paper, the sigmoid function was used to obtain an ideal symmetrical distribution of the input data, which has normal distribution [65]. The transformation is carried out using the following equations:

$$\theta_k = \frac{2\pi}{\left(1+e^{\left(-AX_{i_k}^b\right)}\right)}, \quad \phi_j = \frac{2\pi}{\left(1+e^{\left(-AT_{r_j}^b\right)}\right)}. \tag{4.41}$$

where the parameters A and b are constant gains, commonly set to 3 and 1 respectively ([65]. Next, angles are mapped on the complex plane using the following functions:

$$s_k = \lambda_k e^{(i\theta_k)}, \quad r_j = \gamma_j e^{(i\phi_j)}. \tag{4.42}$$

where i is the imaginary unit of a complex number ($i^2 = -1$), and λ_k and γ_j are unitary vectors with size n and m respectively. Through equations (4.40) - (4.42), arrays X_i and T_r are mapped into the complex plane and denoted as stimulus $S_{(t)}$ and response $R_{(t)}$ respectively, t denotes the current sample time.

$$S_{(t)} = \begin{bmatrix} s_1 & s_2 & \cdots & s_l \end{bmatrix}, \quad R_{(t)} = \begin{bmatrix} r_1 & r_2 & \cdots & r_m \end{bmatrix}. \tag{4.43}$$

If the relationship between $S_{(t)}$ and $R_{(t)}$ is stored in a matrix $H_{(t)}$, called as holographic memory, an analogy to the weight matrix, and a matrix $G_{(t)}$, an analogy for the bias matrix in a real-valued NN, and have the following form:

$$H_{(t)} = \begin{bmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{l1} & \cdots & h_{lm} \end{bmatrix}, \quad G_{(t)} = \begin{bmatrix} g_1 & \cdots & g_m \end{bmatrix}. \tag{4.44}$$

then the difference between $R_{(t)}$ and the product $S_{(t)} \cdot H_{(t)}$ can be expressed by the error vector $\Delta_{(t)} = [e_1, \cdots, e_m]$ as:

$$\Delta_{(t)} = R_{(t)} - S_{(t)} \cdot H_{(t)}. \tag{4.45}$$

The elements of matrix $H_{(t)}$ are computed so that the norm of the error vector $\Delta_{(t)}$ is minimal. The output $V_{(t)}$ for a new vector $S_{(t)}$ is calculated by:

$$V_{(t)} = S_{(t)} \cdot H_{(t)} + G_{(t)}. \tag{4.46}$$

Finally, the values of $H_{(t+1)}$ and $G_{(t+1)}$, that represent the updated values of the holographic memory, are calculated using the input $S_{(t)}$, output$V_{(t)}$, and reference $R_{(t)}$, from the following equations:

$$H_{(t+1)} = H_{(t)} + \tfrac{1}{c}\left(R_{(t)} - V_{(t)}\right) S^{*}_{(t)}, \quad G_{(t+1)} = G_{(t)} + \tfrac{1}{c}\left(R_{(t)} - V_{(t)}\right). \qquad (4.47)$$

where c is a normalization parameter computed from vector $S_{(t)}(c = l)$. To get the real-valued output vector(FF Motor command of Fig.??), from the output $V_{(t)}$, it is necessary to reverse the mapping in Eq.(4.42):r_j.

4.6 Comparison of the performance of the NNs for FEL

Several NNs have been used in the literature to develop FEL controllers, however, no comparison of the proposed architectures has been reported for a real application. The experimental section focuses on the comparison of the above feed-forward controllers (i.e. NNs) for FEL, in order to create a sphere pattern designed with the methodology explained in detail in chapter 5.

4.6.1 Experimental setup

Three types of NNs architectures (MADALINE, MLP, and HNN) were compared for FEL feed-forward controllers. In MADALINE, and MLP the learning rates are adapted online using the algorithm explained in chapter 4.5.3. The influence of the adaptive learning rate on this NNs is analyzed in the following section by place it and remove it from the NNs. This make us analyze the contribution and possible improvements that this adaptive algorithm incorporates into the feed-forward controller's performance. In order to evaluate the quality of the final shape of the sphere, we calculate the final area and compare the result value with the one we obtained with the proposed robot. The numerical difference gives the overall error in the origami object, to calculate this error the methodology explained in chapter 5 was used.

Two PIDs where used as feedback controllers, one for the first stage and other for the second stage, with: $P_1 = 55, I_1 = 75, D_1 = 0.1$, and $P_2 = 40, I_2 = 20, and\ D_2 = 0$. The sample time was set to $0.05s$. All initial weights and biases in all NNs were set to matrices of zeros with their corresponding size. For MADALINE nine neuros were set to the processing layer. For MLP two hidden layers were used, with nine neurons each one. For the HNN, also nine complex neurons were set into the processing layer. The static learning rates in MADALINE and MLP were set to $\alpha = 0.01$ in both cases.

4.6.2 Influences of applying a Adaptive learning rate into a FEL controller

The main objective of using an adaptive learning rate (ALR) in a NN training process is to prevent the overfitting or slow convergence. This ALR reduces the mean error in online training such as FEL, thus increasing the response time significantly [59]. In Table 4.1, the Mean Error ($ME = \sqrt{MSE}$) and the Error Percentage ($Error\%$) between the target trajectory and the output trajectory of the system, using all NNs, are presented.

Table 4.1: ME performances of NNs in both stages

	MADALINE	MADALINE + ALR	MLP	MLP + ALR	HNN
ME Folding Stage	0.009320	0.011854	0.013918	0.01437	0.006699
ME Gluing Stage	0.003732	0.017001	0.015964	0.017002	0.010035

In Table 4.1, it can be observed that in both cases where ALR algorithms are applied the ME is increased. This is contrary to the results obtained by Ruan et al. [59]. Moreover, this occurs due to the lower number of samples used in the adaptation process, when compared with [59, 4]. In Ruan et al. [59] the number of samples was not specified but, in Almeida et al. [4] about 32000 samples were used. In above experiments, only 5000 samples were used (In Table 4.1). Furthermore, HNN obtained the best results out of all NNs. These results show that HNN is an excellent option to be used in FEL controllers, where low number of samples are available (e.g., robots with sample time greater than 0.01s).

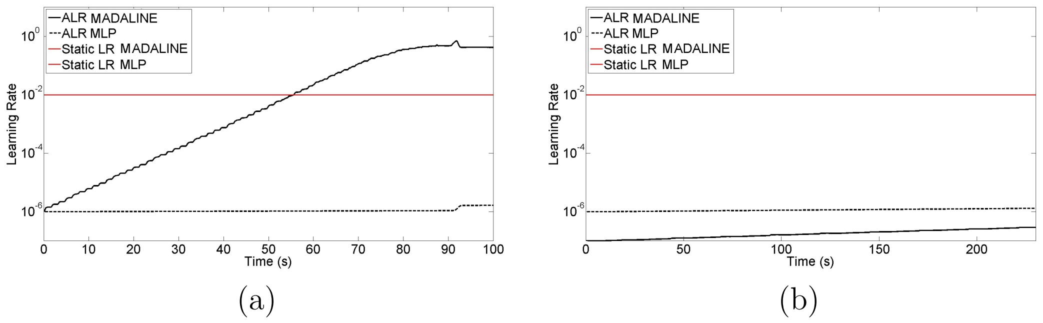


Figure 4.5: Adaptation of the learning rate during the training process. (a) Learning rate adaptation from the first stage. (b) Learning rate adaptation from the second stage.

As can be observed in Fig. 4.5, the learning rates in both NNs change constantly in time. Furthermore, in some cases like that in Fig. 4.5b, the ALR values never approach to the static learning rate. This means that, as in this case, the static learning rate could be far from the optimum value, and this type of algorithms improve the performance of the NNs despite its reduction in reach a local minima.

4.6.3 Performance of FEL controllers to changes in the reference frequency

The following experiment analyzes the performance of all FEL controllers, with changes in the trajectory frequencies of the reference. First, we present the error performance analysis using a normal frequency . We then increase the frequency by a factor of 1.2 times, then by a factor of 1.4 times, and so on, until it is 2.2 times the regular frequency value. The following tables 4.2 and 4.3 show the error performances between the desired and real folding and gluing locations, for all controllers in both machines due to the increases frequencies of the reference.

Table 4.2: Error performances by velocity: Folding step. Bold numbers represent the best performance in a velocity multiplier.

	MADALINE	MADALINE + ALR	MLP	MLP + ALR	HNN
1.0X	0.009320	0.011854	0.013918	0.014370	**0.006699**
1.2X	0.015507	0.018149	0.020085	0.020230	**0.010322**
1.4X	0.024051	0.025616	0.026178	0.026015	**0.016150**
1.6X	0.028900	0.029510	0.029592	0.029274	**0.018084**
1.8X	0.047250	0.042042	0.042344	0.041658	**0.027830**
2.0X	0.047250	0.042042	0.042344	0.041658	**0.027830**
2.2X	0.049915	0.076115	0.077433	0.075749	**0.032350**

Table 4.3: Error performances by velocity: Gluing step. Bold numbers represent the best performance in a velocity multiplier.

	MADALINE	MADALINE + ALR	MLP	MLP + ALR	HNN
1.0X	**0.003732**	0.017001	0.015964	0.017002	0.010035
1.2X	**0.011353**	0.027548	0.027446	0.027581	0.014119
1.4X	0.024033	0.039975	0.041080	0.040004	**0.021735**
1.6X	0.030532	0.045678	0.047364	0.045687	**0.018227**
1.8X	0.039930	0.052560	0.055007	0.052561	**0.028580**
2.0X	0.050587	0.060451	0.063772	0.060453	**0.044940**
2.2X	0.062023	0.101009	0.108111	0.100988	**0.041024**

It can be observed that in both stages, the mean error increases due to the increase in the frequency velocity. This occurs due to a reduction in the establish time (i.e., time required by the response signal to reach a steady state) and a reduction in the working points. In the folding step the HNN has better accuracy for all velocities. On the other hand, in the gluing machine, MADALINE has the best results with velocities of 1.0X and 1.2X. Here, 1.0X means 330s to complete a pattern. If the velocity is multiplied for a factor of 1.4X or higher, the performance of HNN is the better.

4.6.4 Relationship between the ME by velocities and the Origami final shape

Using the information in tables 4.2 and 4.3, and the procedure explained in chapter 3, the error between the desired and real areas can be calculated. There is no precise error value that can indicate whether or not an origami figure could be considered as high quality. Consequently, we assume that error values over 5% are rejected. Using the threshold at this specific value, the maximum number of patterns per hour using each of the NNs can be estimated.

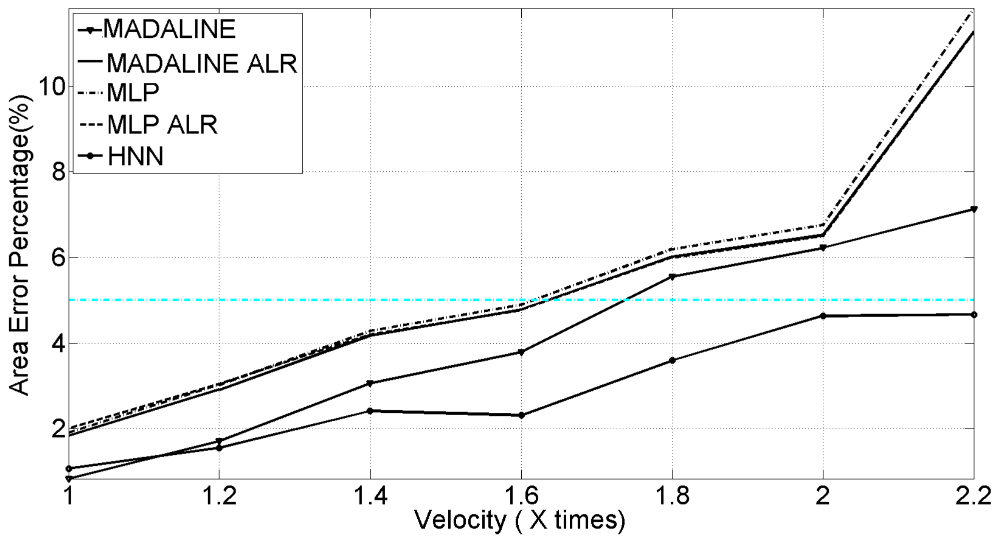


Figure 4.6: Final origami error % performance. Cyan dashed line represents the threshold.

If we calculate the number of patterns per hour with a normal speed ($1.0X$), we can deduce that it takes both machines $330s$ to complete a pattern. This means that both machines can make $10.91 patterns/h$, despite the NN architecture used in the process. An approximated of the maximum number of $Patterns/h$ that all FEL controllers can achieve without passing the 5% error threshold can be calculated using the information in Fig. 4.6.

Table 4.4: Max multiplier and $Patterns/h$ reachable for all FEL Controllers

	MADALINE	MADALINE + ALR	MLP	MLP + ALR	HNN
Max multiplier	1.738	1.634	1.616	1.638	2.287
Max $Patt/h$	18.96	17.82	17.63	17.87	24.95

As can observed in Table 4.4, the two best results were obtained using HNN, and MADALINE without the ALR algorithm. As explained before, both NNs architectures are very simple, owing to each having only a single layer. This reduces the number of computations in weight updating, resulting in faster convergence. However, the special properties of the complex domain of NNs such as HNN show a clear superiority in convergence time, when compared with the other NNs architectures.

4.7 Discussions

The FEL controller has been studied in numerous researches over the past 30 years. This is due to its simplicity and good performance for improving trajectory following. At the same time, during those 30 years new NNs architectures have also been in development, thus making it possible to improve the performance of FEL controllers. Ruan et al. [59], improve the FEL performance by including an adaptive learning rate algorithm created in Plagianakos et al. [54], and adding a momentum parameter into the training algorithm. This revealed that by including an ALR algorithm, the convergence is reached faster than using a regular back-propagation training algorithm. However, in this paper, we found an interesting relationship between the velocities of the desired trajectories and the learning rate adaptation process. In some velocities, the ME is reduced, as can be observed in Table 4.2 with velocities between 1.6X - 2.0X. However, there are other cases where the ME is increased, thus making the ALR unsuitable for FEL applications. Ruan et. al did not consider the influence of the changes in the velocities for the ALR. We show the performance of two types of NNs that have not been analyzed in past researches. These NNs are the MADALINE and the HNN. The MADALINE is more commonly used for noise canceling in audio processing, and is one of the easiest NNs to be implemented [63, 15]. Despite its simple structure, very good performance results have been obtained for FEL control. The reduced number of operations and weight update computations makes MADALINE to converge twice as fast as MLP (see Table 4.1). On the other hand, the HNN has also been proposed in this work as a better solution for the FEL control problem. The HNN is more difficult to implement than MADALINE, due to the real-complex conversion. However, it improves the convergence time in comparison with all the other NNs (see Tables 4.2 and 4.3), thus making it the most suitable NNs architecture to be implemented in FEL control. Also, the ALR is automatically computed from the input and output mapping functions in HNN.

4.8 Summary

This chapter deals with the application of FEL into the proposed robot. In order to learn a feed-forward controller three types of NNs are examined and a method to adapt the learning rate was introduced. As it was shown in the experimental results in chapter 4.6, all NNs architectures reduced the time and mean error in the trajectory following. Simulations and experimental results with the proposed robot show a clear superiority of the learning algorithms developed with HNN. The results of these experiments show that there is a close relationship between the number of layer in the NNs and the convergence time. So, it is recommended to use HNN, to obtain a better performance by the FEL controller, due to the small number of computations performed in the training process and fast convergence time. As it was reported in the literature the balloon/sphere model is one of the shapes with high difficulty for robots. In this experiments, we could overcome the difficulties of the sphere model thanks to the proposed surface of revolution-based methodology, used to create the pattern and the combination of folding and gluing added in the proposed machine which reduced the number of paper manipulations, as was shown in Table 5.1 from chapter 3.4.3.

Chapter 5

Methodology for designing crease patterns for the proposed robot

The methodology applied to create crease patterns with the proposed robot is based in shapes of surface of revolution (a.k.a rotational sweep). In [42], a software called ORI-REVO, able to create 3D paper structures from a 2D poly-line, was described. This program can be used to create a vast number of 3D shapes that can be used in different type of applications. The methodology used in ORI-REVO is explained in [42]. ORI-REVO can be used to create crease patterns less complex than other very known software such as: Tree Maker or Origamizer, but is limited to figures using only surface of revolution. Despite the crease pattern is not very complicated to fold by hand, proper adaptations of this methodology must be carried out in order to be implemented into the proposed robot. To understand better the adaptations executed to this method let' s observe one crease pattern created using ORI-REVO in Fig. 5.1.

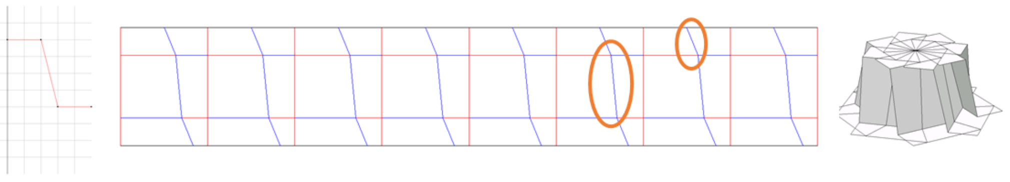
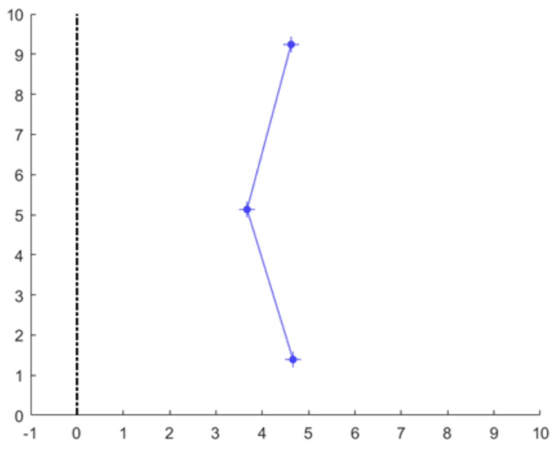


Figure 5.1: Example of an origami hat crease pattern developed with ORI-REVO. Blue lines: valley folds, red lines: mountain folds, orange circles: folds difficult to execute using robots.

The origami pattern developed with ORI-REVO or any of the previously mentioned software (e.g. Tree-Maker, or Origamizer), were intended to be assembled by hand, and have folds that are very difficult to execute with a robot due to handling problems. For example, the folds highlighted in orange circles in Fig. 5.1 are folding lines allocated inside the paper, and do not overstep from one edge to the other in the crease pattern. These type of lines required multiple manipulations during folding process, making them not suitable for automation. In our proposed methodology, a modified version of the SOR on [42] is proposed. This new methodology, uses a combination of simple folding patterns with gluing areas. In our previous works, there are times where we refer to this mixture between folding and gluing as “Norigami”. The word Norigami comes from the mixture between three words in Japanese language: “*nori*” that means glue, “*ori*” that means fold, and “*kami/gami*” that means paper. As can be deduced from its name, in norigami, ***simple origami*** crease patterns are used in combination with ***small glued segments***, to create 3D shape figures using paper or similar flexible materials. The main idea in norigami is to reduce the complexity in the folding process by reducing the number of manipulations required to build a 3D-paper shape. This is possible thanks to combining manipulations within the folding process and fusing some crease lines in nodes of the crease pattern, changing the kinematics in the final 3D shape.

5.1 Adaptations of the surface of revolution methodology for automation

On this chapter, a modified version of the method in [42] was applied to create patterns able to be fold by the proposed robot [58]. Having a profile as in Fig.5.2a, the resulting model consist on a set of trapezoids created from rotating the profile $2\pi/Nrad$ with resulting flaps that represent the areas where the glue is applied (see Fig5.2b).



(a)

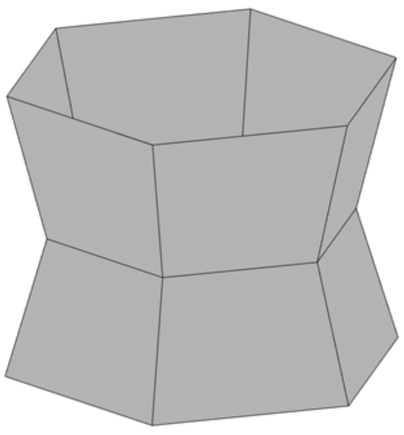

(b)

Figure 5.2: Example of a shape with two stages, (a) profile, (b) solid 3D structure after rotational sweep, $N = 6$

In the cases where there is only one stage in the profile (see Fig.5.3a and 5.3b), the flaps can rotate freely because there is no connection within another flap. However, adding new stages like in Fig.5.3c and 5.3d creates new connections within flaps.

This problem can be solved if the angle of rotation of the flaps γ is equal to zero. On the other hand, this only works if the flaps are placed outside the paper' s shape as in Fig.5.3d, and it is not possible to place the flaps inside the structure. We propose as solution to this problem to perform cut lines in specified sectors of the crease pattern. The proper location of these cut lines are explained later in this chapter.

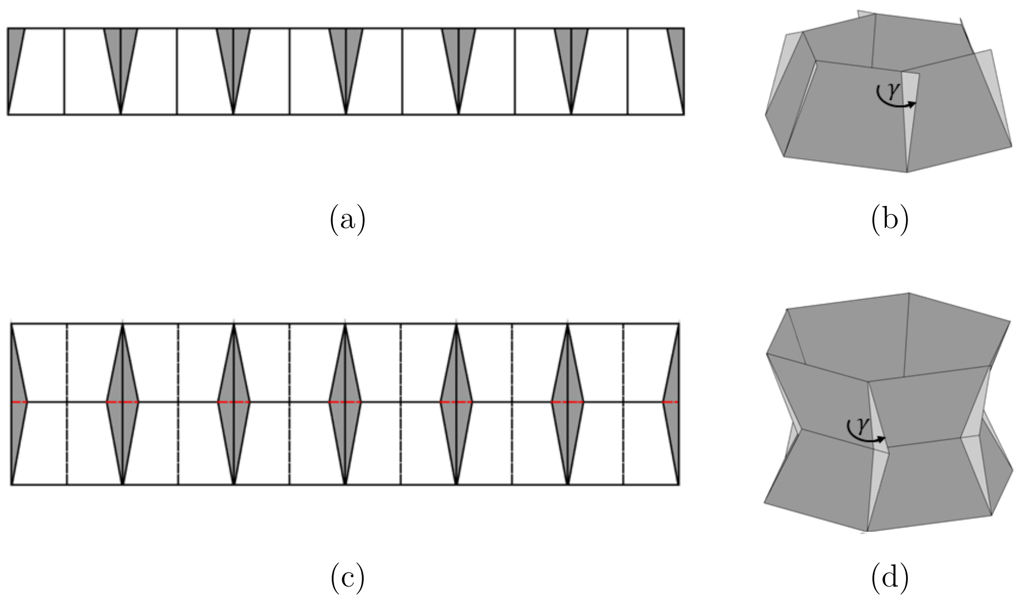


Figure 5.3: Example' s crease pattern, (a) crease pattern one stage, (b) folded shape with flaps one stage, (c) crease pattern multiple stages, (d) folded shape with flaps multiple stages

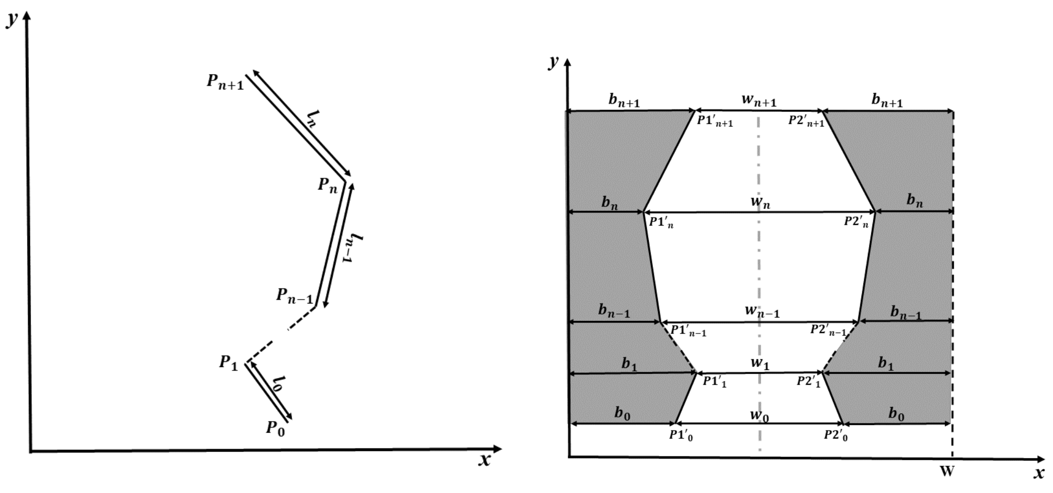
In Fig. 5.4a, $P_{(n,x)}$, and $P_{(n,y)}$ denote the x and y coordinates respectively of the vertex P_n in the input profile. For each point P_n in the input profile, two points ($P1'_n$ and $P2'_n$) are created in the crease pattern. In Fig. 5.4b, $P'_{(n,x)}$, and $P'_{(n,y)}$ denote the x, and y coordinates respectively of the vertex P'_n in the crease pattern. The new points $P1'_n$ and $P2'_n$ have the same y coordinate but different x coordinates. Using the information in Fig.5.4a, the values of Fig.5.4b can be calculated as follows:

$$w_n = 2P_{(n,x)} \sin \tfrac{\pi}{N}, \quad W = max(w_n). \tag{5.1}$$

$$b_n = \frac{W - w_n}{2}. \tag{5.2}$$

$$l_n = ||P_n - P_{(n-1)}||. \tag{5.3}$$

$$P1'_{(n,x)} = \tfrac{W-w_n}{2}, \quad P2'_{(n,x)} = \tfrac{W+w_n}{2}. \tag{5.4}$$



(a) (b)

Figure 5.4: Input profile and crease points defining a sequence of segments of K vertices ($n = 1 \cdots K$). (a) input profile, (b) segment of the crease pattern. The dashed gray line is a middle valley fold. The gray area is the gluing area and the white area is the area in the surface of the 3D object (non-glued area)

$$P1'_{(n,y)} = P2'_{(n,y)} = P'_{(n,y)} = \begin{cases} 0, & if(n=0), \\ P'_{(n-1,y)} + \sqrt{l_n^2 - \left(\frac{w_n - w_{n-1}}{2}\right)^2}, & if(n>0). \end{cases} \tag{5.5}$$

where w_n and b_n represent the width of the non-glued and glued areas respectively of each segment at point P_n of the profile, and W represents the size of each segment in the crease pattern. Note that the N segments forming the crease pattern are separated by mountain folds placed at distances multiples of W. The robot folds the paper sheet after the crease pattern is designed. The desired width between folds ($W/2$) and the shape of the gluing stamp are extracted from the designed pattern. Above information is stored in the NXT bricks (See Fig.3.2 parts F5 and G7) together with the program for the folding-gluing procedure.

After the crease pattern is concluded, the next step is to generate the 3D model using the information previously calculated. Apart of the values calculated using Equations (5.1) - (5.5), the angle between each one of the segments l_n, denoted in this paper as θ_n can be calculated using the following equations:

$$\theta_n = \tan^{-1}\left(\frac{U_{(n,x)}V_{(n,z)} - U_{(n,z)}V_{(n,x)}}{U_{(n,x)}V_{(n,x)} + U_{(n,z)}V_{(n,z)}}\right) \tag{5.6}$$

$$U_n = \begin{cases} [\ P_{(n,x)} \quad 0]\ , & if(n=0), \\ [\ P_{(n,x)} - P_{(n-1,x)} \quad P_{(n,z)} - P_{(n-1,z)}]\ , & if(n>0). \end{cases} \tag{5.7}$$

$$V_n = \left[P_{(n+1,x)} - P_{(n,x)} \quad P_{(n+1,z)} - P_{(n,z)}\right] \tag{5.8}$$

The value of θ_n can be used to define the position of the flap. Using all the information obtained from equations (5.1) - (5.8) the values can be applied into geometrical transformation matrices and obtain the location of all points in the 3D model and their flaps. One of the main differences between our proposed method and [42] is that our crease pattern could have cutting lines (see the red dashed lines in Fig. 5.3c). The value of $\theta_n(\theta_n > \pi)$ also allows the condition for adding cutting lines for interior flaps. If $\theta_n < \pi$ the paper could be flat-folded but it is complicated for the robot, so we decide to cut all flaps. These new lines make it possible to rotate the flaps an angle γ into any value, except multiples of π. The value of γ can be used for planning/simulation of the folding motion in current self-folding robots like in [21]. This new freedom movement in the flaps permit us avoid some problems that in [42] were impossible to solve; e.g. the flaps can be allocated even inside the structure. Despite there are some angles where the flaps crash within the structure or other flaps, the user has more freedom in the creation of 3D shapes.

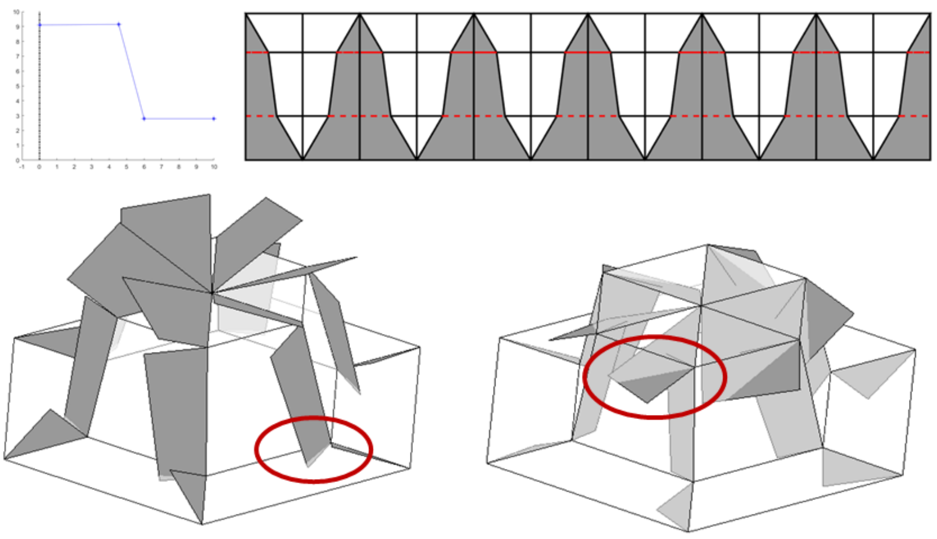


Figure 5.5: Norigami hat crease pattern and 3D shapes made by regular surface of revolution.

5.1.1 Patterns with independent gluing segments

As mentioned before, even performing cuts into the crease pattern, there are some cases when the flaps crash between the main structure or another flap. To create the norigami pattern that never has flap penetration, we propose to assume each line segment in the profile's poly-line as an independent line as shown in Fig. 5.6. The resulting crease patterns decrease the gluing area in comparison with the ones in Fig. 5.5.

These gluing segments have the minimum gluing area, and the resulting flaps can be rotated in any angle (except multiples of π,) and never collide within other segments or flap. However, this division also generates 3 different gluing areas

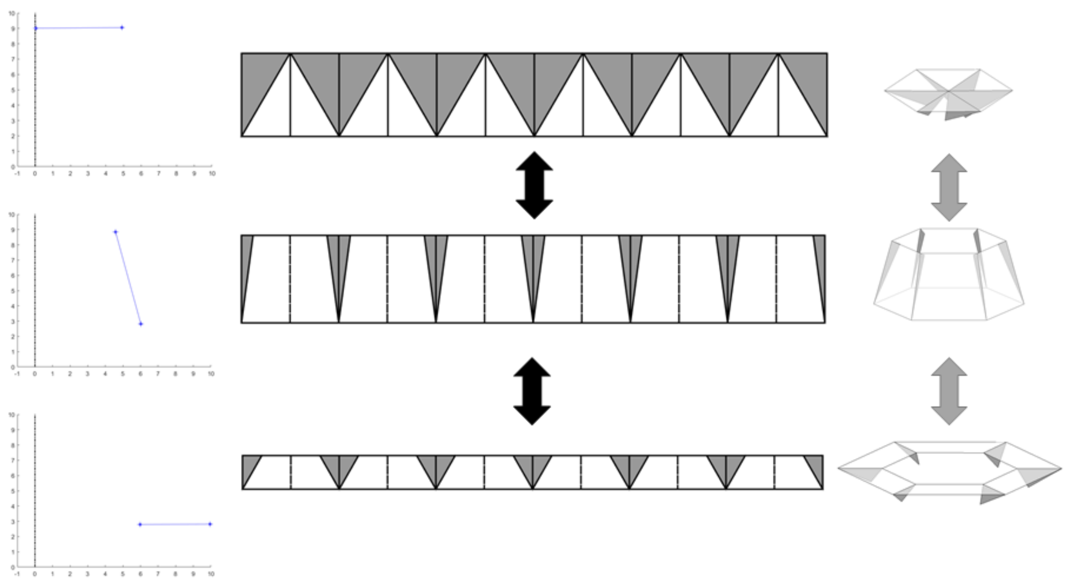


Figure 5.6: Norigami hat crease pattern and 3D shapes made by regular surface of revolution.

instead of 1, making the assembly process longer than before, if it is performed by hand.

We proposed to use the crease patterns in Fig.5.6 in combination with the crease pattern in Fig.5.5 and removing all the remaining paper material by cutting (see Fig. 5.7). We called this new crease pattern as "independent gluing pattern", and can be selected in our software to be applied totally or partially depending on the requirements. Furthermore, the crease patterns created with this methodology (with or without independent gluing pattern) can be now automatically folded and glued by the proposed robot from Fig. 3.2.

5.2 Results

To demonstrate the applicability of the proposed robot, three different shapes: a hexagonal box, and a pod, and a hyperbolioid were automatically folded and glued by the proposed robot. Their corresponding crease pattern, expected 3D shape, and final product are show in Fig.5.8. All of these shapes were made using $N = 6$;

The complexity reduction is possible due to adding gluing areas into the proposed crease pattern, that reduce movements in the robot by executing multiple tasks at the same time. We consider a manipulation as a combination of movements required to execute a task, e.g. a fold, a turn over, a rotation, or a squash [69]. Table.5.1 expose the number of manipulations in a group of crease patterns folded by hand and the proposed robot.

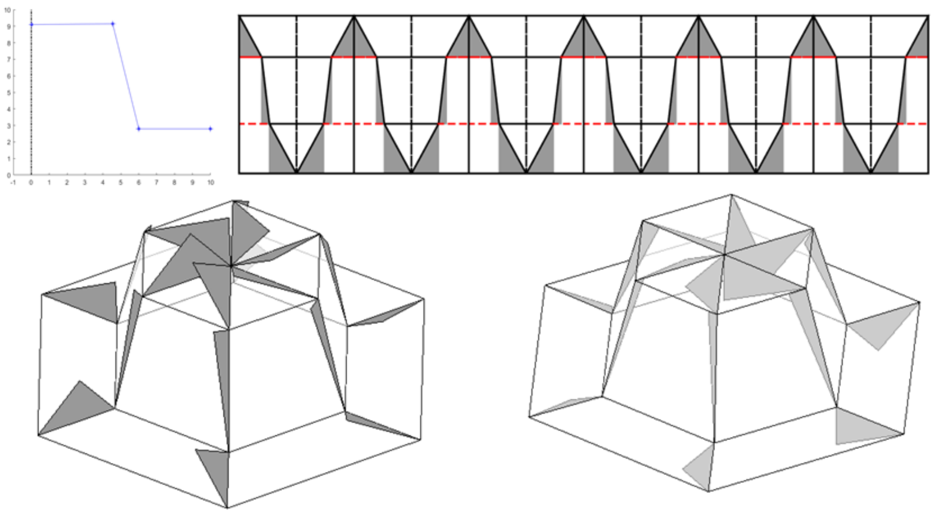


Figure 5.7: Norigami hat crease pattern and 3D shapes made by independent gluing pattern.

Table 5.1: Number of manipulations.

Pattern Name	# manipulations Origami crease pattern	# manipulations Norigami crease pattern	# of manipulations reduced
Box (Fig.5.8a)	31	31	0
Pod (Fig.5.8b)	67	31	36
Hyperboloid (Fig. 5.8c)	37	31	6
Hat (Fig.5.5)	61	31	30

5.3 Discussions

As can be observed in Fig.??, several 3D shapes can be created and successfully folded and assembled by the proposed robot using the proposed methodology. Due to the flexible properties of the paper, the final 3D shape has to be opened by hand adopting the desired shape. However, this process of opening the shape is difficult to execute in some cases, e.g. shapes that are completely closed, or figures with changes in the rotational direction of the angle θ like the one in Fig.5.5. As can be observed in the area surrounded by the red circles in Fig.5.5, there are some cases were the flaps can generate penetration in the paper. According to [42], to avoid this problem all coordinates of the y axis (i.e. the axis of rotation) of the vertices of the input profile have to be monotonically increasing or decreasing, i.e. $P_{(n-1,y)} \leq P_{(n,y)}(1 \leq n < N)$ *or* $P_{(n-1,y)} \geq P_{(n,y)}(1 \leq n < N)$. However, in our proposed method, new cutting lines can be included in the crease pattern reducing the area of the flaps, making possible to rotate them freely without collision (except with angles γ multiples of π). In table 5.1 we can observe the required manipulation

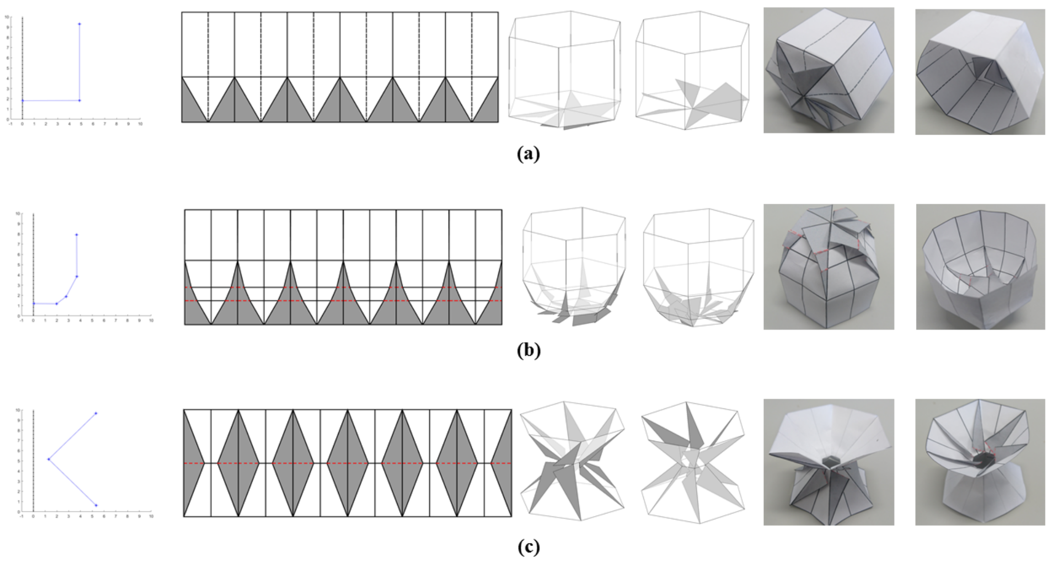


Figure 5.8: Crease pattern examples and folded and opened figure. (a) Hexagonal box, (b) Hexagonal pod, (c) Hexagonal hyperboloid approximation.

to accomplish several 3D shapes using origami and the proposed robot. It can be observed that in the box example the number of manipulation in both cases is the same, this is because the crease pattern only has one stage with changes in the x axis in its profile. This means that the complexity in manipulations is exponentially increasing by the number of stages plus one, if there is changes in the x axis in any of the stages in the profile, i.e. $Num.\ of\ Manipulations = (3K + 1)N + 1;\ if(P_{(n-1,x)} \neq P_{(n,x)}\ in\ all\ sub-segments)$ or $Num.\ of\ Manipulations = (3K - 1)N + 1;\ if(P_{(n-1,x)} = P_{(n,x)}\ in\ a\ sub-segment)$, K is the number of sub-segments.

5.4 Summary

In this chapter, a methodology to generate a crease patterns based on rotational surfaces has been proposed to construct several 3D paper models and then build by the proposed robot. The crease patterns made with this methodology can be assembled by hand as well. Due to some limitations in the resulting flap locations, some new cutting lines have been added to the original origami pattern to permit them to move freely, even inside the 3D paper model. Furthermore, a new proposed crease pattern called "independent gluing segment pattern" has been proposed as a solution for very complex shapes with not monotonically increasing or decreasing in the angle θ_n. Using the independent gluing segment pattern, previously problems such as overlapping between the flaps and the paper-body, or the possibility of introducing the flaps inside the paper' s structure have been resolved. Several shapes have been exposed to validate this method using a PC software developed in Matlab®. The results show that the number of manipulations were reduced, specially in crease patterns with several stages or number of segments (see table **??**).

Chapter 6

Improved methodology for designing complex 3D shapes by the proposed robot

The methodology explained in the previous chapter 5, can be used to create 3D shapes that are based on surface of revolutions (SOR), able to be build by the proposed robot. Despite this methodology expands the applicability of the proposed robot, there are figures that are not based in SOR, because they have irregular shapes. Software like: PepaKura Designer, Tree-Maker, or Origamizer can be used to create these type of irregular figures as we can observe in Fig. 6.1.

However, the resulting 2D crease pattern for these complex shapes is even more complicated to build than the SOR counterparts, having inside folds that required multiple manipulations at the same time, making extremely difficult to assemble these 3D shapes automatically using a robot. To solve irregular 3D shapes using a slightly modified version of our already existing robot, a novel methodology extracted from the previous SOR methodology is proposed in this thesis and explained in this chapter. This new methodology can be applied to figures with star-shaped-projected polyhedrons. A star-shaped-projected polyhedron is a shape that is composed and contains a infinite number of star-shaped-polygons. An example of these type of polygons is shown in Fig. 6.2.

A star-shaped-polygon is a irregular 2D polygon that contains a point Z_n, from which the entire boundary of the polygon is reachable, i.e. all vertex $P_{0,n}, \cdots, P_{k-1,n}, P_{k,n}, P_{k+1,n}, \cdots, P_{K,n}$, where K is the number of points in a height n [55].

The art-gallery theorem states that $\frac{K}{3}$ points are necessary to cover an K-gon, and in particular, this shows that polygons for which $3 \leq K \leq 5$ are necessarily star-shaped [1, 5], . There are many ways to calculate the center Z_n. One of the most efficient algorithms is the one proposed by Lee and Preparata [35]. However, in the tests performed in this chapter, we consider the points Z_n to be previously known.

As previously said, this new methodology translates and performs modification into the SOR methodology explained in chapter 5. One difference that can be notice

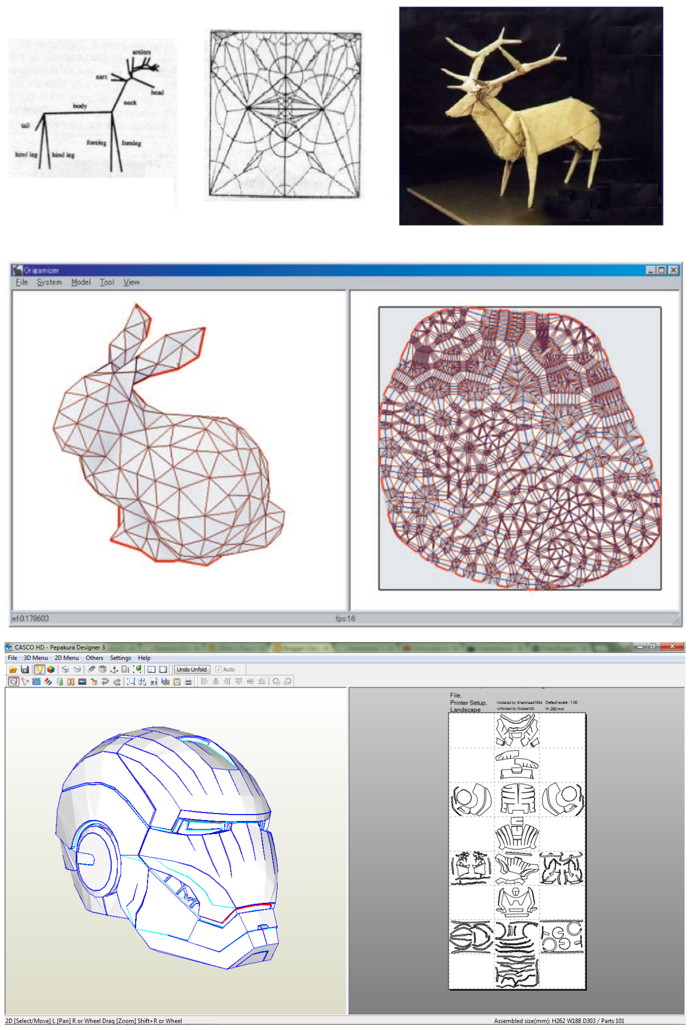


Figure 6.1: Examples of irregular shapes made by existing software. First row: TreeMaker. Second row: Origamizer. Third row: Pepakura Designer

when irregular forms are analyzed is that they have two or more profiles, instead of the single profile of SOR counterpart. Therefore, a novel procedure has to be develop to convert 3D irregular shapes, that have multiple profiles into a 2D crease patterns. In this chapter, a procedure to obtain the profiles is proposed and exposed. Then, some modifications to the profiles are performed, to be able to translate the spatial information into the 2D crease pattern. Finally the procedure to build the crease pattern and considerations for automation are given. Finally some examples are shown at the end of this chapter.

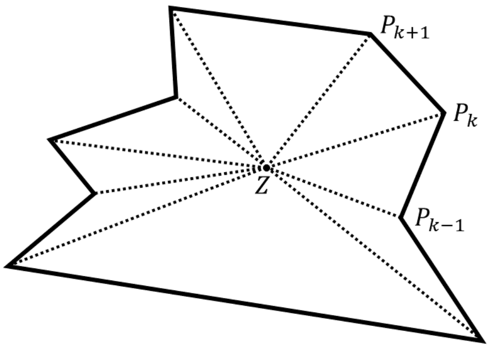
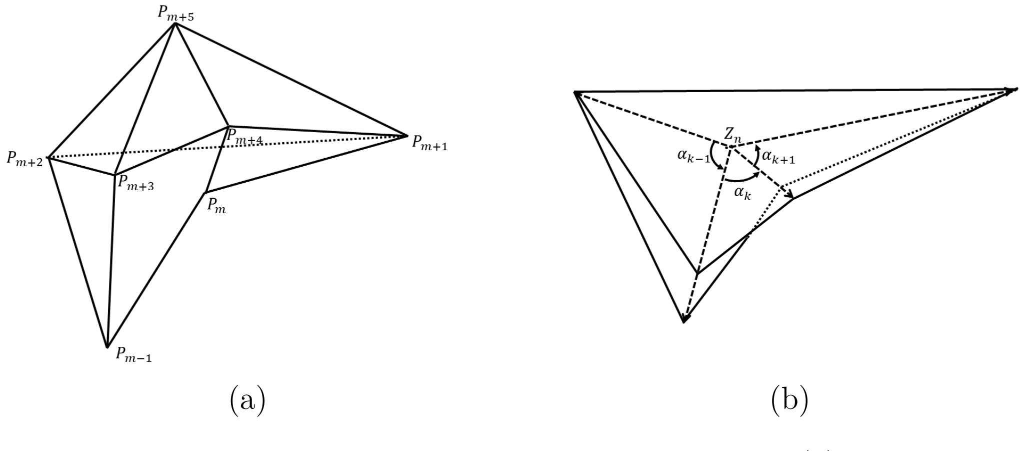


Figure 6.2: Example of a star-shaped polygon.

6.1 Obtaining the profiles from a 3D model

The first step in our methodology consist in obtaining and arranging all the possible profiles in a 3D model. These profiles required to have the same number of points, and all these points have to be at the same height, one within the other. Using the information of the vertices from the 3D model, the heights ($P_{k,n,z} = Z_{n,z}$) and rotation angles (α_k) can be obtained using the following algorithm:



(a) (b)

Figure 6.3: Example of a star-shaped projected polyhedron. (a) Isometric view of the polyhedron. (b) Top view and projections of the planes in 2D.

In these profiles the maximum height Z_{max} and minimum height Z_{min} have to be the same for all profiles. Also, the sum of all the angles in α_k have to be equal to 2π. After all the heights and rotational have been extracted from the 3D model, the next step is to obtain the profiles. To obtain these profiles, the procedure for obtaining the intersection between a plane and the exterior of the 3D shape was used. The cut planes are created using the information of the heights and angles

Table 6.1: Algorithm to obtain the heights and rotation angles from a 3D model

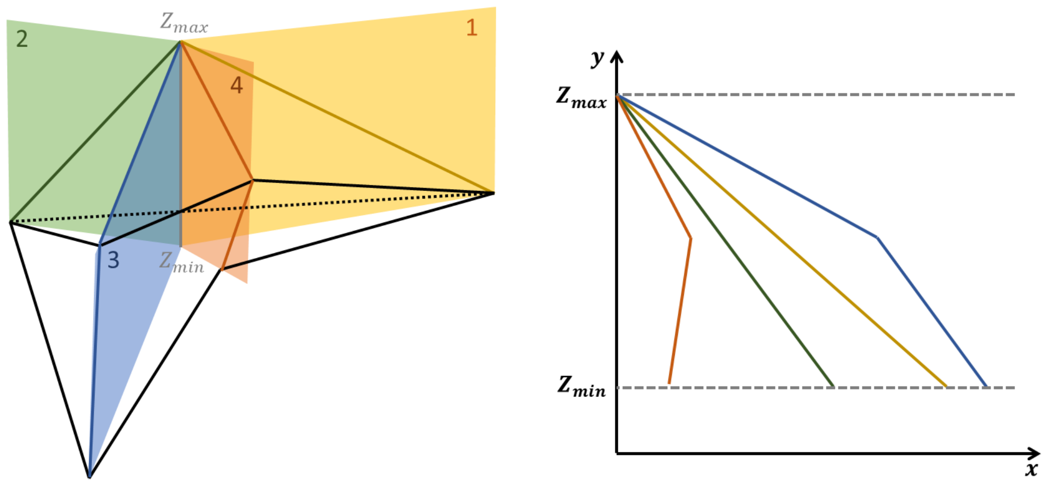
```
Having a 3D model as Fig. 6.3a.
  For each point $P_m$ in the 3D model:
    if ($Z_{n,z}$ and $\alpha_k$ are empty ):
      $Z_{n,z} = P_{m,z}$;

      $\alpha_k = tan^{-1} \frac{P_{m,y} - Z_{n,y}}{P_{m,x} - Z_{n,x}}$;
      if ($\alpha_k < 0$): $\alpha_k = \alpha_k + 2\pi$;
      if ($\alpha_k \geq 2\pi$): $\alpha_k = \alpha_k - 2\pi$;
      $n++$;
      $k++$;
    else:
      if ($P_{m,z} \notin Z_z$):
        $Z_{n,z} = P_{m,z}$;
        $n++$;

      $\beta = tan^{-1} \frac{P_{m,y} - Z_{n,y}}{P_{m,x} - Z_{n,x}}$;
      if ($\beta < 0$): $\beta = \beta + 2\pi$;
      if ($\beta \geq 2\pi$): $\beta = \beta - 2\pi$;
      if ($\beta \notin \alpha$):
        $\alpha_k = \beta$;
        $k++$;
```

previously calculated. The resulting cut planes and profiles from the example in Fig. 6.3, can be observed in Fig. 6.5.

It can be observed from Fig. 6.5b, that not all the poly-lines have the same number of points. The profile number 1 (yellow) and 2 (green) have two points. On the other hand, the profile number 3 (blue) and 4 (orange) have three points, all them with a equal height. In this chapter, an algorithm to add missing points into the poly-lines is proposed. First, the algorithm eliminates any redundant points using the algorithm in Table. 6.2. Then the algorithm in Table. 6.3 uses the information of the missing heights to add missing points into the poly-lines that required it. After the profiles are completed, they are arrange in a matrix of $n \times k$, where n is the number of points on each profile, and k is the number of profiles.

6.2 Creating the 2D crease pattern from the profiles

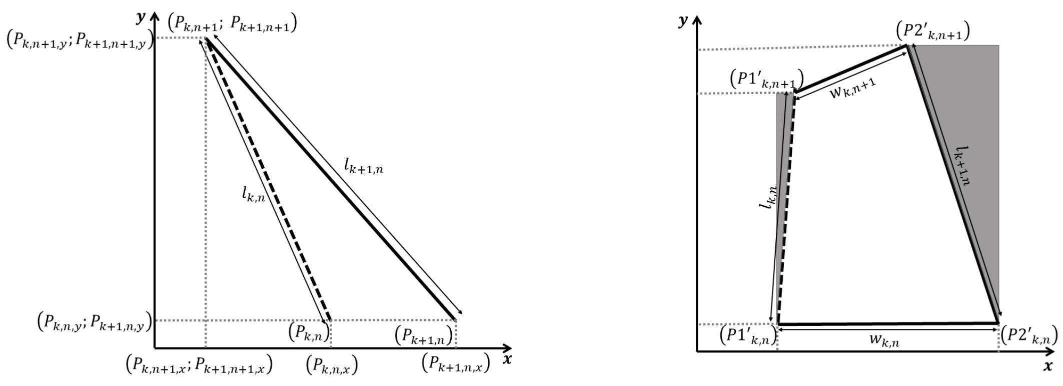
After the information of the profiles is stored in the previously mentioned matrix, and the angle vector α_k from Table 6.1, the next step is to use that information to create the 2D crease pattern. For the following example in Fig. 6.5, we are gonna to assume a pair of different profiles, with the same number of points, and the number



(a) (b)

Figure 6.4: Resulting profiles after extraction by plane cuts. (a) Isometric view of the polyhedron with the corresponding plane cuts. (b) Resulting profiles. Yellow: profile 1, green: profile 2, blue: profile 3, and orange: profile 4.

of segments is $K = 6$. This means that each profile is repeated 3 times intercalating between them, and the angle α_k between profiles is constant and equal to $pi/3$.



(a) (b)

Figure 6.5: Example to build a general crease pattern with two or more profiles. (a) Profiles. (b) Crease pattern equivalents. Dashed line: first profile, continuous line: second profile, gray areas: gluing areas.

As an starting point, the distances l_n and the widths $w_{k,n}$ between all points from a single pair of profiles have to be calculated as following:

$$l_{k,n} = ||P_{k,n+1} - P_{k,n}||. \tag{6.1}$$

$$w_{k,n} = \sqrt{P^2_{k,n,x} + P^2_{k+1,n,x} - 2P_{k,n,x}P_{k+1,n,x}cos\alpha_k}. \tag{6.2}$$

Table 6.2: Algorithm to eliminate redundancies in poly-lines

```
Variable to count the number of points per height: $m = 1$;
For each point $P_n$ on a profile $k$:
  if ($n == 1$):
    $P^f_n + = P_n$;
    $m + +$;
  else:
    if ($P_{n-1,z} \neq P_{n,z}$):
      $P^f_n + = P_n$;
      $m = 1$;
      $n + +$;
    else:
      $m + +$;
      if($m < 3$):
        $P^f_n + = P_n$;
        n++;
      else:
        $P^f_{n-1} + = P_n$;
Return: $P^f$;
```

After we calculate the widths and distances from equations (6.1) and (6.1), we perform an adaptation of the SOR methodology using the new information obtained previously to calculate the $P'_{k,n}$ from Fig. 6.5. In this new case, for each point in a pair of profiles $P_{k,n}; P_{k+1,n}$ we obtain a single point in the crease pattern, instead of the double point obtained in the previous SOR methodology.

$$W_k = max(w_{k,n}). \tag{6.3}$$

$$P1'_{k,n,x} = \tfrac{W_k - w_{k,n}}{2}; \quad P2'_{k,n,x} = \tfrac{W_k + w_{k,n}}{2}. \tag{6.4}$$

$$P1'_{k,n,y} = \begin{cases} 0, & if(n = 0); \\ P1'_{k,n-1,y} + \sqrt{l^2_{k,n} - \left(\frac{w_{k,n} - w_{k,n-1}}{2}\right)^2}, & if(n > 0). \end{cases} \tag{6.5}$$

$$P2'_{k,n,y} = \begin{cases} 0, & if(n = 0); \\ P2'_{k,n-1,y} + \sqrt{l^2_{k+1,n} - \left(\frac{w_{k+1,n} - w_{k+1,n-1}}{2}\right)^2}, & if(n > 0). \end{cases} \tag{6.6}$$

The resulting crease pattern made by using equations (6.1)-(6.6), generate an approximation of the 3D object after is assembled. The assembled shape has an error that depends on the desired shape angle differences. This produce some differences in the lengths of the 3D shape, made to guarantee symmetry in the gluing areas. This symmetry is necessary for execute the automation process using the proposed robot.

Table 6.3: Algorithm to add missing points into the poly-lines

```
Having a vector with all heights: $H_n$:
For each point $P_n$ on a profile $k$:
    Vector to save the new indexes: $i = [\ ]$;
    For each height $h$ in $H_n$:
        if($P_{n,z} \notin H_n$):
            if($h > P_{n,z}$):
                $L = P_n - P_{n+1}$
                $P^t_{n,x} = \frac{h - P_{n,z} L_x}{L_z} + P_{n,x}$;
                $P^t_{n,y} = \frac{h - P_{n,z} L_y}{L_z} + P_{n,y}$;
                $P^t_{n,z} = h$;
                $i+ = n$;
            else:
                $L = P_n - P_{n-1}$;
                $P^t_{n,x} = \frac{h - P_{n,z} L_x}{L_z} + P_{n-1,x}$;
                $P^t_{n,y} = \frac{h - P_{n,z} L_y}{L_z} + P_{n-1,y}$;
                $P^t_{n,z} = h$;
                $i+ = n - 1$;

    for each index $t$ in $i$:
        $P^f_n = P_{n:i} + P^t_i + P_{i:end}$;

    Return: $P^f_n$;
```

6.3 Results

For these experiments, three different shapes were used to analyze the proposed methodology. The first two shapes are irregular shapes that not require to add any points into their profiles. For the third example a irregular side hat was translated into a crease pattern. For this hat, both the redundant point elimination and the adding algorithms were used.

Figure 6.6 show three different examples of 3D shapes that can be made using the multiple-profiles methodology proposed in this thesis's chapter. The first shape is a star-shaped structure with two different profiles, for this case $K = 6$ and $N = 2$, so any point was need to be added or removed. Also, due to intercalating between profiles, the angle α_k is constant and has a value of π/K.

The shape in the second row is a pyramid-like structure with three different profiles. For this case $K = 3$ and $N = 2$, and the angle between each profile α_k varies between each profile. Equal to the first shape, any point has to be added or removed.

The last shape is a hat, similar to the one made in the previous chapter 5, but if it is analyzed carefully there are some differences. The main difference can be observed between the second and the third point in the poly-line; where the third

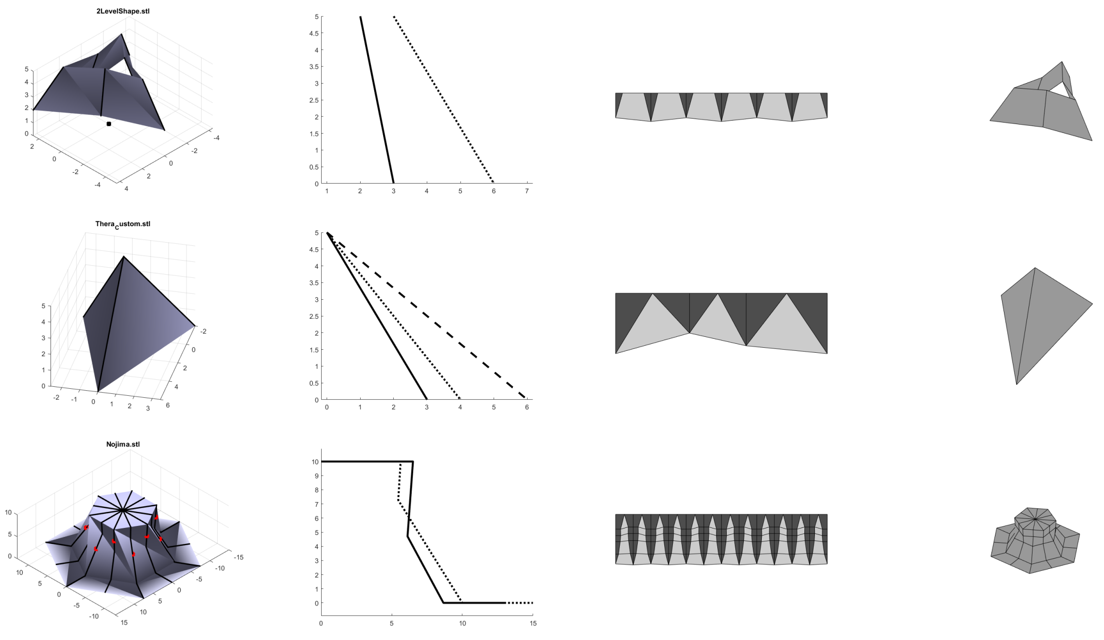
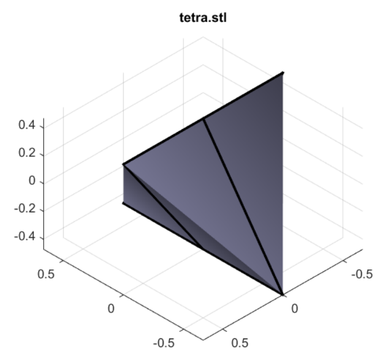


Figure 6.6: Experiments carried out with the multiple profiles methodology. From left to right: Original STL 3D model, extracted profiles, crease pattern, and the approximation.

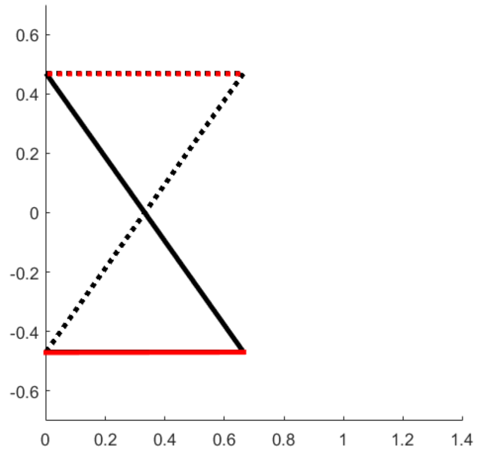
line was rotated π/Krad. This generates a turning effect that can be used later to compact the hat in a flat state (see Fig.6.8). This rotation generate a new profile, as can be observed in the second column of Fig. 6.6. For this case, points have to be removed and added to complete the poly-lines. The added points can be observed as red dots in the first column image of Fig 6.6. The number of sub-segments $K = 12$ and $N = 6$. The rotation angle is constant and has a value of $\alpha_k = \pi/K$. Despite the approximation is very close to the original figure, the transverse line, that generates the characteristic of compacting the hat is not present.

6.4 Discussions

The methodology proposed on this chapter focuses on the creation of more diverse 3D shapes, not limited to SOR. The main idea is to use this methodology in an automatic folding robot, doing as less modification as possible to the mechanisms that already exist. The results exposed in Fig. 6.6 show that several irregular-side shapes can be achieve by the proposed method. This technique is a enhanced version of our previous work, exposed in chapter 5. Although the shapes are not geometrically perfect, they are a very close approximations to the real objects. The rule of the gradually increases or decreases in the poly-lines points also applies into this methodology (see more info on chapter 5.3). However, in this case all the profiles have to achieve this rule. A good solution is to always apply the total independent gluing segment variation, explained in chapter 5.1.1. This allow us to freely allocate the generated flaps and used the minimal gluing area.



(a)



(b)

Figure 6.7: Lines in the profile to be avoided. (a) Isometric of a tetrahedron with edges perpendicular to the pivot axis. (b) Profiles of the tetrahedron. The red Lines are perpendicular to the pivot axis.

There is a special case within applying this methodology that has to be avoided, and its exposed in Fig. 6.7. In this case there is a edge of the 3D model that is perfectly perpendicular and over a radius line. The problem is caused at the moment of counting the points that compose a poly-line. Due to have multiple points in a single height, the reduction algorithm eliminates the equal points, and the adding point algorithm cannot found the border points that compose a line to create new points. To solve this problem it is recommended to rotate the 3D model to a more stable state. An stable state is where a face is perpendicular to the ground or parallel to the pivot axis.

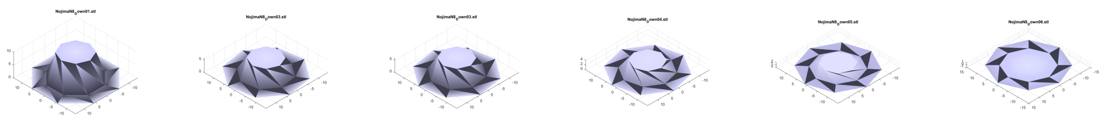


Figure 6.8: Closing the origami hat into a flat state.

The hat shape shown in Fig. 6.6 has a transverse crease line that allows the figure to be compacted into a flat state, as can be observed in Fig. 6.8. Due to the flaps generated in the gluing process, the open-close behavior is very difficult to execute by applying this methodology. However, making the traverse lines, and folding them by hand, could create figures that are flat-foldable partially.

6.5 Summary

On this chapter, a enhanced version of the methodology exposed in chapter 5 has been proposed and explained. Several examples have been shown to explain the methodology procedure. Proper adaptations of the SOR methodology have been made in order to be applied into star-shaped projected polyhedrons. As was explained, the resulting analysis generates multiple profiles that later are used to create the 2D crease pattern. Despite these crease patterns have irregular shapes, the gluing areas are symmetrical, but not congruent. This means that multiple stamps have to be used in the proposed machine, i.e. one for each profile.

The resulting 3D shapes are in the majority static and cannot be move as the model in Fig. 6.7. However, performing the proper crease lines by hand, and reducing the gluing area applying the independent gluing segment methodology, could help to create semi-flat foldable structures.

Chapter 7

Applications of the proposed methodologies and robot

In this chapter, we expose the different uses of the proposed methodologies and robot into several areas.

As was explained in chapter 1 and 2, there have been several attends to perform automatic folding of a sheet of paper using a robot. Some of these ideas have very interesting results. However, just a small among of them have been put into practice [75]. This is caused to the complex mechanism used in the robot, and the high-cost that some of these robot have [30, 20]. In this thesis, a simple folding robot was designed and explained in chapter 3-4. This robot was build using a low-cost Lego MINDSTORMS NXT ®, with some additions made by a 3D-printer.

To increase the number of patterns able to be automatically folded and glued by the proposed robot, a surface of revolution-based methodology was created in chapter 5. This methodology is a variation of an already crease pattern creation methodology created by Jun Mitani in [42]. This methodology increase the number of elements that could be made by the proposed robot significantly. However, irregular shapes cannot been made using this methodology. That is why a variation of this methodology was introduced in 6. This novel methodology performs adaptations to the SOR methodology to get crease patterns for multiple profiles shapes.

7.1 Application of the robot to build fruit covers for ripening

The first application of the proposed robot is a fruit cover for fruits in the growth process. In a growth process, specially when sweet fruits like mango or apples start to get ripen, it is important to cover them with a sag or a cover to prevent get invade by insects or any plague, and also helps in the growth process, see Fig. 7.1.

The main idea in this application is to create a recycle-paper sag to cover the fruits during this period of ripening. This sag or cover can be created using a diverse

Figure 7.1: Fruits covers in farming.

number of shapes and sizes, depending on the size and shape of the corresponding fruit. The shape of the stamp for this type of sags have usually pseudo-cylindrical shape, with the final and starting edges unglued (see Fig.7.2)

One of the positive points of using paper instead of a plastic or a polymer materials, is that the paper help in the ripening process, generating best-looking and sweeter fruits. The proposed machine is able to be modified to adapt the sags to the needed size and shape, without changing too many elements in the robot.

7.2 Application of the robot for packaging and mass-production products

Another application of the proposed robot, combined with the SOR methodology of chapter 5 is the creation of products made in paper-like materials. Due to the interchangeable mechanism of the proposed machine, and the control theory applied into it (see chapter 4), many of these products can be created in a mass-production way.

Using the proper colors and designs, which are visually attractive and interesting

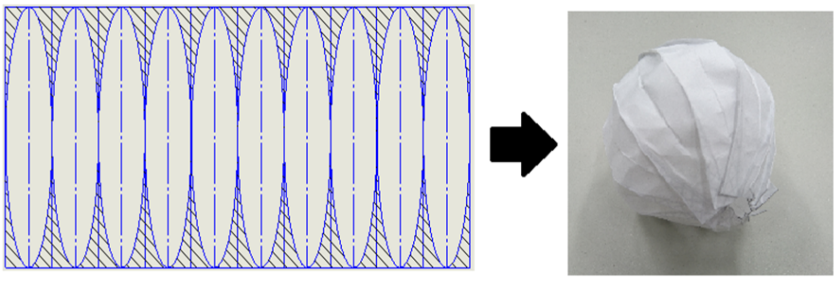
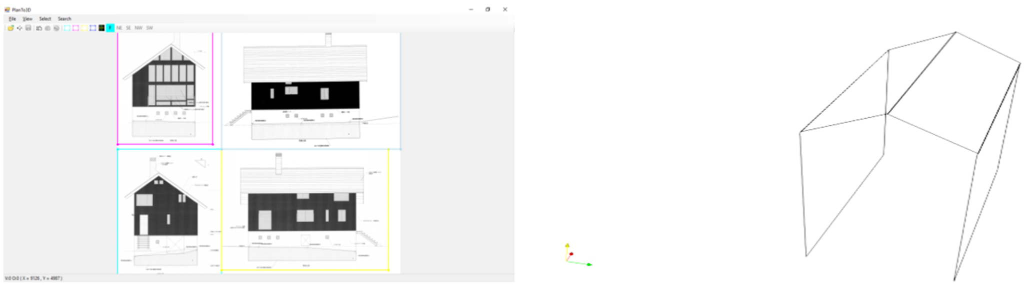


Figure 7.2: Cover for a mango or an apple-like fruits.

(see chapter 2), the costumers could be interested in products build by our proposed robot. An interesting application is in customized packaging, where sellers can create vast number of shapes among personalized boxes, envelops or bags, according to their costumers likes. Some examples of these type of shapes can be found in chapter 5, Fig. 5.8.

7.3 Application of the robot for scaled-house models in architecture

In house design the option to observe the final result through a scale model is essential to analyze and make decisions about the structure, and appearance of the building. This scale-model can help also to provide information to potential buyers by the house sellers about the house location and spatial comparisons (e.g. size of the living rooms, bathrooms, windows and doors). In this application, a software to create three-dimensional (3D) house-model using its elevation views (i.e. North, south, east, and west views) was created. In this software, the user is able to reconstruct the 3D shape of the external walls and ceilings by clicking into two of the elevation views (see Fig. 7.3(a)). The software reads the points introduced by the user and perform a correlation between these points to generate a 3D polygon (see Fig. 7.3(b)). The user can create any number of polygons until the whole structure is complete. Then, the software is able to generate a top view projection of the ceiling. This projected polygon is used as a polygon's template to search for a similar-shaped ceiling in a satellite image using genetic algorithms (GAs). After the best match is found, the scale of the satellite image is used to know the real scale of the 3D model and all its components. Then, the updated 3D model is used to create a crease pattern using the methodology from chapter 6. The model is assembled by hand and can be used in a mock-up scale model to represent a smaller version of the real house, in order to be used in building or selling processes.



(a) (b)

Figure 7.3: House 3D model creation software. (a) Elevation views. (b) 3D model

7.3.1 3D Model Construction from Elevation Views

A software able to create 3D model using information from elevation views was designed in C Sharp. In this software, the four elevation views can be introduced independently or selected from a unique images by surround them inside a rectangular areas as in figure 7.3(a). The origin point has to be introduced manually in all of the views. After the reference point in all views has been set the user is able to create 3D polygons using two adjacent elevation views (i.e. North-east, north-west, south-east, and south-west views). The user creates lines by left-clicking in any part of one of the views. The user can create any number of points, and when is satisfied with the polygon, he/she can finish the polygon creation by right-clicking anywhere. The polygon will be closed between the last and the first clicked points. Is very important to count the number of points and the order that these points were pressed, because the same corresponding points have to be selected in same order using the adjacent view in order to create the 3D polygon.

The software uses the information of the coordinates x and y from both views, given in number of pixels, to extrapolate each of the 3D points in the resulting model as follows.

$$P'_{(k,n,x)} = P^{NS}_{(k,n,x)}. \tag{7.1}$$

$$P'_{(k,n,y)} = P^{EW}_{(k,n,x)}. \tag{7.2}$$

$$P'_{(k,n,z)} = P^{NS}_{(k,n,y)}. \tag{7.3}$$

where $P^{NS}_{(k,n)}$ are the points clicked in the north or south view, $P^{EW}_{(k,n)}$ are the points clicked in the east or west views, and $P'_{(k,n)}$ is the resulting point n of a polygon k in 3D. As was explained before, all of the coordinates are given in number of pixels. This means that the resulting 3D model do not has a real size scale. That is why we propose to use the GA polygon matching procedure explained in the following section in order to know a scale that is more accurate with the real size of the building.

7.3.2 GA-based Polygon Matching

The main idea of this procedure is to find a particular template polygon in an image that is known to contain the same polygon inside. Template matching is a very used technique now on days due to its ease implementation and simplicity in order to find a particular image, known as template inside another bigger image [25]. However, template matching only works in cases when the searched template do not has rotations or scaling. In this paper, a template matching able to search for polygon templates inside images (i.e. captures from a satellite image) that is invariant to rotations or scaling is proposed.

Before the GA algorithm is applied, the searching image has to be pre-processed. There are a series of pre-processing algorithms used in this paper. The main goal of this processing is to increase the contrasts of the searching image, and apply an edge extractor to simplify the searching process. The pre-processing processes applied in this paper were: A histogram equalization to increase the contrast in the image, and a canny edge detector, for more extend explanation of the functionality of these procedures refer to [25].

After the pre-processing algorithms have been applied into the searching image, we proceed to search for the target polygon template using the following GAs (for more information about this topic refer to [23]). In order to make the template matching process invariant to rotations or scaling, the template polygon points $T_{(n)}$ have to be transform into a scaled and rotated points $T'_{(n)}$. This is carried out by multiply the point $T_{(n)}$ with a transformation matrix $R_{(\theta,s)}$ as following:

$$T'_{(n)} = R_{(\theta,s)} \cdot T_{(n)} = \begin{bmatrix} s\cos\theta & -\sin\theta & 0 \\ \sin\theta & s\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T_{(n,x)} \\ T_{(n,y)} \\ 1 \end{bmatrix}. \tag{7.4}$$

where s is the scaling factor and θ is the rotation angle. Both variables were used later as inputs of the fitness function in the GA procedure.

The fitness function of the GA uses the lines that can be created using the points of $T'_{(n)}$ and calculates the proximity of all edge points of the searching image to these lines using the distance of a point to a line segment. Having two points A and B, that are extremes of a segment of a line, and C that is a point outside the line, we have that:

$$u = \frac{(C_x - A_x)(B_x - A_x) + (C_y - A_y) * (B_y - A_y)}{(B_x - A_x)^2 + (B_y - A_y)^2} \tag{7.5}$$

$$d = \begin{cases} ||\overrightarrow{AC}|| & if(u < 0); \\ ||\overrightarrow{BC}|| & if(u > 1); \\ \frac{(B_x-A_x)*(C_y-A_y)-(B_y-A_y)*(C_x-A_x)}{||\vec{AB}||} & if(0 \le u \le 1). \end{cases} \tag{7.6}$$

The value d represent the continuous distance from a point to any of the lines of the template. However, this distance is given by a real-decimal number, and the distances of images has to be measured in number of pixels and represented using integer numbers. In this paper, the Bresenham algorithm[70] was used to count the number of pixels from the point to the line, in order to obtain an accurate result. Table 7.1 show the pseudo-code of the Bresenham algorithm for a line with border points (x_0, y_0) and (x_1, y_1).

Table 7.1: Bresenham algorithm pseudo-code

```
$plotLine(x_0, y_0, x_1, y_1)$
        $dx = x_1 - x_0$
        $dy = y_1 - y_0$
        $D = 2 * dy - dx$
        $y = y_0$

$for\ x\ from\ x_0\ to\ x_1$
        $plot(x, y)$
        $if D > 0$
              $y = y + 1$
              $D = D - 2 * dx$
        $end\ if$
$D = D + 2 * dy$
```

In table 7.1, D represent the integer distance of d from equation (7.6), and is the distance used for all calculations in the fitness function of the GA.

The GA in this application was designed to found the polygon $T'_{(n)}$ with the maximum number of pixel count within a random number of iterations n using the transformation matrix $R_{(\theta,s)}$ as element to maximize.

Table 7.2: GA features

Population Size :	600
Chromosome Length :	64
Iterations :	15
Selection Type :	*Ranking*
Crossover Rate :	0.9
Mutation Rate :	0.3

The GA designed in application, uses the information from table 7.2, and tries to maximize the response value from the fitness function exposed in table 7.3.

Figure 7.4 shows an example of a resulting matching using the proposed technique based in GA.

Table 7.3: Fitness function evaluation algorithm pseudo-code

```
$numPixelsReached = 0;$

$For\ a\ random\ pair\ of\ points (A\ and\ B)\ in\ the\ searching\ image:$
    $if\ ||\vec{AB}|| < 20:\ return\ 0;$
    $Reconstruct\ the\ template\ Polygon\ T_{(n)}\ in\ the\ searching\ image\ as\ T'_{(n)}\ using\ random\ values$
    $for\ each\ point\ C\ in\ the\ searching\ image:$
        $Calculate\ D\ for\ each\ C\ respect\ all\ lines\ in\ T'_{(n)};$
        $if\ D < 6:$
            $numPixelsReached = numPixelsReached + (int)\left(\frac{(6-D)}{6}\right)$
        $end\ if$

$Return\ (numPixelsReached)$
```

The crease pattern is made based in the methodology explained in chapter 6. A crease pattern of the model in Fig. 7.4 is shown in Fig. 7.5.

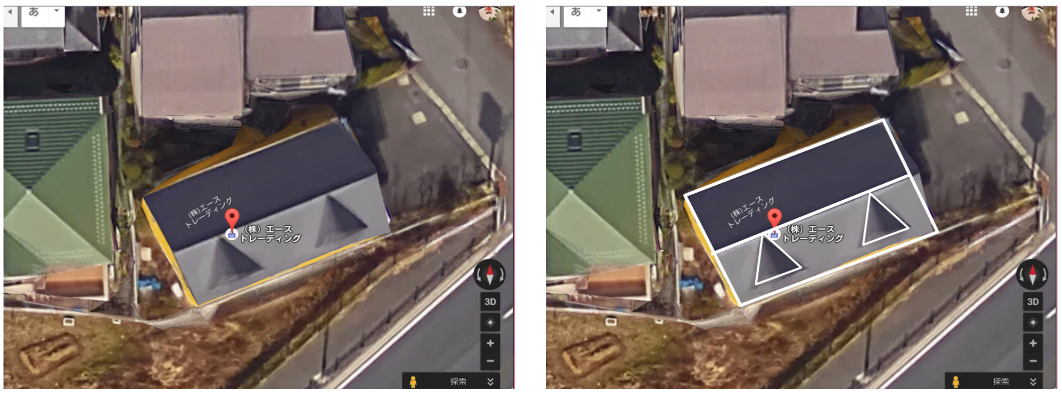
7.3.3 Using the information from the GA to build the crease pattern

Using the scale previously obtained and the methodology from chapter 6, the crease pattern, similar to the one in figure 7.5(a) is created. In order to be able to make the scale model the size of the model is resized to fit into a normal sheet of paper. The assembly process has to be made by hand and adopt a form similar to the one in figure 7.5(b). This application's model can be used later in a mock-up map to observe the final house appearance in construction or selling processes.

7.4 Application of the crease patterns in medicine

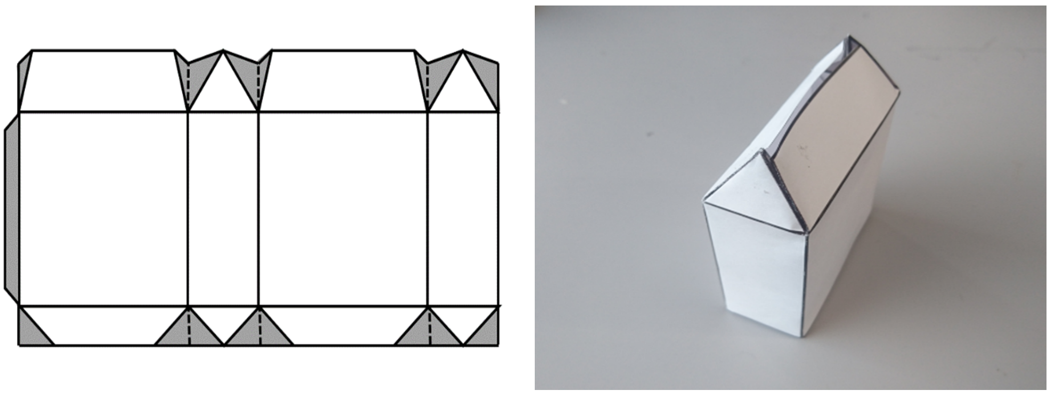
A very interesting application of the objects created with the proposed methodologies are objects with the property of been bi-stable [12]. This type of objects could be used as medical devices with a vast number of applications. The main problem in our shapes is to be able to open and close them from one state to the other.

A solution to open and close the 3D shapes created in chapter 5 and 6 using a simple mechanism that require very small or non-assistance from a human was designed. Although all previous crease patterns were considered to be made in a flexible material such as paper or plastic sheets, we can consider them to be made in a rigid material instead. This type of origami its called "rigid origami" and permit us to create a mechanism able to open and close the structure using a single actuation



(a) (b)

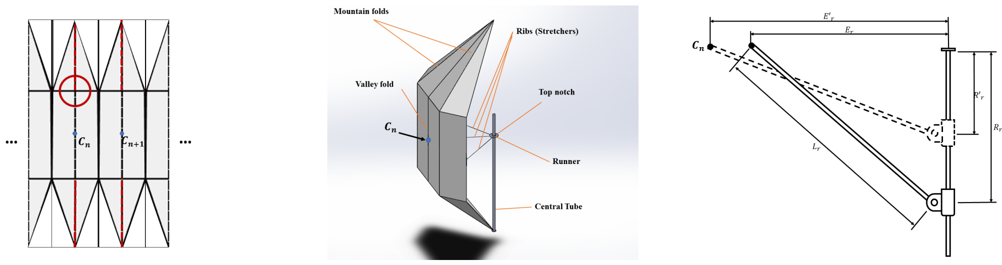
Figure 7.4: Polygon matching of the best ceiling in satellite image. (a) Original image. (b) Matching of the best ceiling



(a) (b)

Figure 7.5: Crease pattern example of a building (a) Crease pattern and gluing segments. (b) Paper shape after assembled by hand.

system (for more information please see the methodologies by [12] and [66]). In rigid Origami the crease pattern is considered to be composed by rigid plates linked by rotatory joints instead of crease lines. This joints allow us to transmit the movements of one panel into the others, connecting all independent movements into a single movement. The mechanism proposed in this paper is inspired by the mechanism used to open and close an umbrella. In this mechanism multiple external bars, usually flexible, are radially connected to one of the edges or top notch of a center bar (called the tube) using a rotatory axis that allows the external bars to rotate like a human arm. These external bars have another smallest bars connected to any place of the external bar using a rotatory axis as well, these bars are called ribs or stretchers. These ribs are linked to the center tube by the other extreme edge, and are able to both, rotate and slide within the center bar using a mechanism called the runner. The center tube and the ribs are considered to be rigid. These three bars form a rectangle where one of the sides changes depending on the runner's position, allowing to open and close the umbrella using a single actuation system (see Fig.7.6).



(a) (b) (c)

Figure 7.6: Open-Close Mechanism. (a) Segment of the norigami crease pattern, (b) Proposed system parts, (c) Proposed system design variables. C_n: point of connection between the external plate with the ribs. The red dashed lines in (a) represent cut lines

The shape in Fig.7.6 adopts the form of a spinning top when its opened, and a cylindrical asterisk when its closed. The external extreme of the Ribs are connected within the crease pattern in the middle point C_n of a vertical valley folds of the plates with biggest radius of each section. To be able to close the 3D shape between its open and close state a problem occurs in the vertex shown surrounded by the red circle in Fig.7.6a. In order to move all crease lines that converge into a single vertex, these vertexes require to accomplish both Maekawa's theorem and Kawasaki's theorem, see [26] for more information. In this particular vertex only Kawasaki's theorem is applied, however, Maekawa's Theorem, that says that the difference between mountain and valley folds converging into a vertex has to be equal to 2, do no occur. That is why we perform cuts in the crease lines marked in red dashed color in Fig.7.6a, to allow flat foldability in the 3D structure between the two required states. The runner's position shown in Fig.7.6c follows a linear direction, and extends the ribs reach following the formulation from equations (7.7)-(7.10):

$$R_r = \sqrt{L_r^2 - E_r^2} \tag{7.7}$$

$$L_r \geq \frac{W}{2} \tag{7.8}$$

where L_r is the longitude of the rips, and is a constant. E_r is the variable distance from the center to the exterior plate. The values E'_r and R'_r represent values between the device's extremes, and follow the following design constrains:

$$0 < E'_r \leq \frac{W}{2} \tag{7.9}$$

$$R'_r = \sqrt{L_r^2 - E'^2_r} \tag{7.10}$$

The devices made using this open-close mechanisms can be introduced through the veins or consumed through the mouth to later be open insede the body and used to perform surgeries or apply any medicine directly to the damage area.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

The conclusions remarks on this thesis were summarized in this section. First, in chapter 2 an analysis of the considerations of shapes made in origami were exposed. Previous works in the area of kansei (feeling) engineering and neuro science were exposed. The results show that costumers usually like non-sharped objects with a combination of dark and light colors. However, these results vary from person to person, and have to be analyzed in deep in a future, using the proposed FQHNN or any other machine learning technique.

In chapter 3 the proposed robot design was exposed. We show that using simple folds in combination with gluing areas, good-looking 3D shapes can be achieved with the proposed robot (see Fig.5.8). The crease patterns used by this robot were based in a SOR methodology created in [42] and explained properly in chapter 5. Several parameters of the crease pattern and paper type were used to calculate a series of trajectory enhancers, i.e. a FEL controller with HNN, and a spring-back and stacking effects calculations. The results show that each one of these enhancers reduces the mean error, produced by displacements or slips in the paper. In order to analyze the performance of the robot for mass-production purposes, a FEL controller using a HNN was proposed. As can be observed in table.3.4, the HNN keeps the displacements controlled, even in twice the initial speed (2.0X). This permit us to reduce the production time because more shapes can be created in less time. The proposed robot can be ease modified to be used to create vast number of paper shapes by changing the gluing stamp's shape. Was also demonstrated, that the number of manipulations was significantly reduced, specially with patterns with many sub-segments, as can be observed in Table. 5.1. The complexity reduction is possible due to adding gluing areas into the proposed crease pattern, that reduce movements in the robot by executing multiple tasks at the same time.

Chapter 4 deals with the application of FEL into the proposed robot. In order to learn a feed-forward controller three types of NNs were examined and a method to adapt the learning rate was introduced. As it was shown in the experimental results in chapter 4.6, all NNs architectures reduced the time and mean error in the trajectory following. Simulations and experimental results with the proposed robot show a clear superiority of the learning algorithms developed with HNN. The results of these experiments show that there is a close relationship between the number of layer in the NNs and the convergence time. So, it is recommended to use HNN, to obtain a better performance by the FEL controller, due to the small number of computations performed in the training process and fast convergence time. As it was reported in the literature, the balloon/sphere model is one of the shapes with high difficulty for robots. In these experiments, we could overcome the difficulties of the sphere model thanks to the proposed surface of revolution-based methodology, and the combination of folding and gluing added in the proposed machine which reduced the number of paper manipulations, as was shown in Table 5.1 from chapter 3.4.3.

Chapter 5 explains the methodology to generate crease patterns based on rotational surfaces to build several 3D paper models on the proposed robot. This methodology was based on a previous work made by Mitani [42], however, was modified in order to be used in the proposed robot. Using this methodology several 3D shapes were achieved by hand and using the previously developed robot. Due to some limitations in the resulting flap locations, some new cutting lines have been added to the original origami pattern to permit the resulting flaps move freely inside the 3D paper model. Furthermore, a new proposed crease pattern called "independent gluing segment pattern" was proposed as a solution for very complex shapes with not monotonically increasing or decreasing in the angle θ_n. Using the independent gluing segment pattern, previously problems such as overlapping between the flaps and the paper-body, and the possibility of introducing the flaps into the paper' s structure have been solved. Several shapes have been exposed to validate this method using a PC software developed in Matlab®.

In chapter 6, a enhanced version of the methodology exposed in chapter 5 was proposed and explained. Several examples have been shown to explain the methodology procedure. Proper adaptations of the SOR methodology have been made in order to be applied into star-shaped projected polyhedrons. As was explained, the resulting analysis generates multiple profiles that later are used to create the 2D crease pattern. Despite these crease patterns have irregular shapes, the gluing areas are symmetrical, but not congruent. This means that multiple stamps have to be used in the proposed robot, i.e. one for each profile. The resulting 3D shapes are in the majority static and cannot be move as the model in Fig. 6.7. However, performing the proper crease lines by hand, and reducing the gluing area applying the independent gluing segment methodology, could help to create semi-flat foldable structures.

Finally, chapter 7 shows several applications of the proposed methodologies and robot explain in the previous chapters. Among these applications we can found: cover sags for fruits in growths, packages, and objects for mass productions, medical devices, and scale-models of buildings. This shows that not only the proposed robot has applications, but also the methodologies can be used to create shape-shifting mechanisms, not only limited to paper materials. These objects can be use in house planning muck-ups and in surgeries that require non-invasive methods.

8.2 Future Works

8.2.1 Modification to be performed into the robot

The proposed robot was design using Lego MINDSTORMS NXT ®, limiting the applicability of the robot. Several design using different technologies have been in progress from the last year. These models were made using a CAD software. The machine shown in Fig. 8.1 occupies less space that the robot made by Lego. Also, can works with bigger sizes of sheets, from A4 to A2, works faster, and have multiple sensors to improve the trajectory following. The main reason to not been included in this thesis was that, proper experiments with the system were not carried out at the moment this document was write.

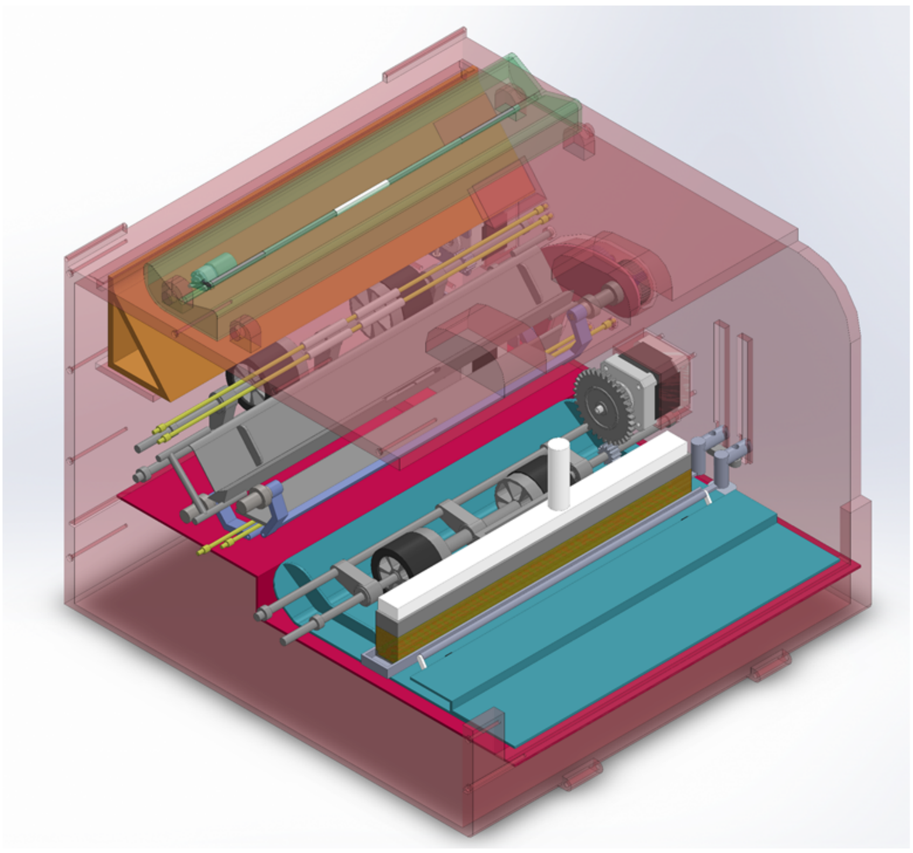


Figure 8.1: Future robot designed in Solid Works ®.

One of the bigger problems with the actual robot, is that the trajectory following is very difficult to execute, due to the properties of the paper. A measuring sensor, external to the motors is require to improve the accuracy of the robot. Also, due to work with glue, a controlled chamber is require in the robot to avoid the glue to be dried.

8.2.2 Considerations for the multiple profile methodology

The multiple profile methodology explained in chapter 6 was introduced in this dissertation. This methodology parts from the premise of having a star-shaped projected polyhedron. However, a vast number of shapes with irregular shapes usually do not have this type of form. As a solution to this problem, we propose to search for the skeleton of of the irregular shape [13], and use the resulting bones as independent pivot axis (see Fig. 8.2).

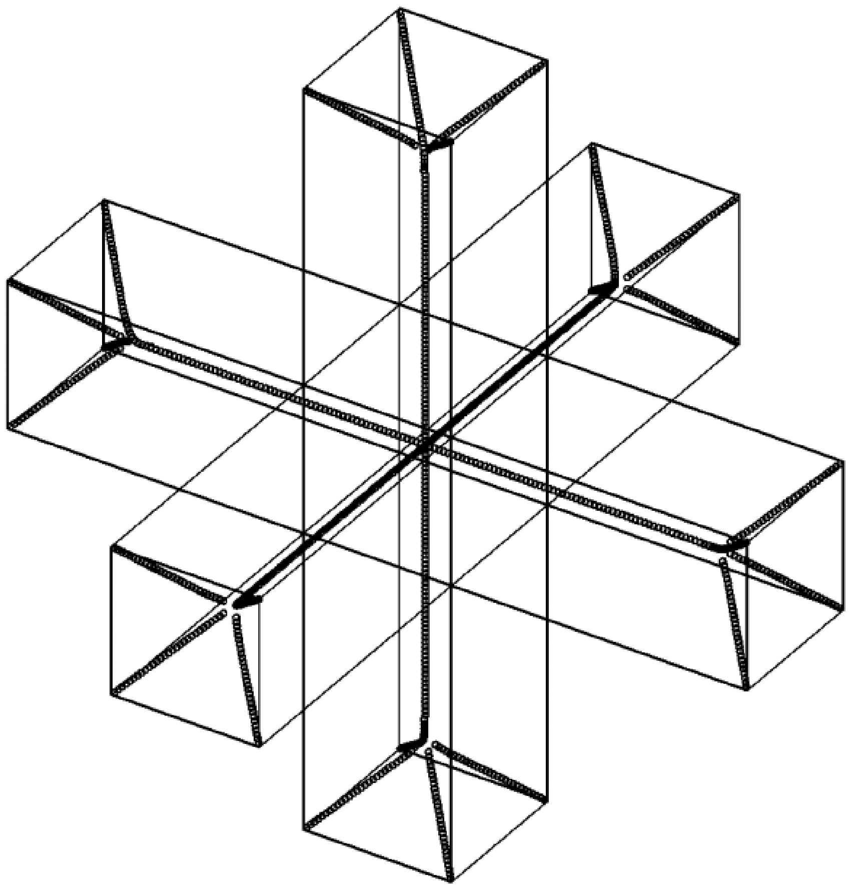


Figure 8.2: Irregular shape with multiple skeletons [13].

The result, will be a group of crease patterns that have to be assembled forming the final shape. If this could be done, any type of 3D structure can be automatically build using the proposed robot, reducing the time to make origami 3D shapes significantly.

Bibliography

[1] Alok Aggarwal. The art gallery theorem: its variations, applications and algorithmic aspects. 1984.

[2] Basel AlAli, Kentaro Hirata, and Kenji Sugimoto. Generalization of feedback error learning (fel) to strictly proper mimo systems. In American Control Conference, 2006, pages 6–pp. IEEE, 2006.

[3] Basel Alali, Kentaro Hirata, and Kenji Sugimoto. Can a two-link manipulator learn how to write? IFAC Proceedings Volumes, 41(2):4084–4089, 2008.

[4] Luís B Almeida, Thibault Langlois, José D Amaral, and Alexander Plakhov. Parameter adaptation in stochastic optimization. On-Line Learning in Neural Networks, Publications of the Newton Institute, pages 111–134, 1998.

[5] David Avis and Godfried T Toussaint. An efficient algorithm for decomposing a polygon into star-shaped polygons. Pattern Recognition, 13(6):395–398, 1981.

[6] Patrik Axelsson and Ylva Jung. Lego Segway project report. Linköping University Electronic Press, 2011.

[7] M Baccouch and Stephen Dodds. A two-link manipulator: simulation and control design. University of Nebraska at Omaha, 2012.

[8] Devin J. Balkcom and Matthew T. Mason. Robotic origami folding. The International Journal of Robotics Research, 27(5):613–627, 2008.

[9] Kristofer Berglund, Johan Ersvik, Svante Ekholm Lindahl, and Johan Söderman. Modelling and implementing synchronization of dual wheeled inverted pendulums with lego nxt. 2010.

[10] Danny Oude Bos et al. Eeg-based emotion recognition. The Influence of Visual and Auditory Stimuli, 56(3):1–17, 2006.

[11] Margaret M Bradley and Peter J Lang. Measuring emotion: the self-assessment manikin and the semantic differential. Journal of behavior therapy and experimental psychiatry, 25(1):49–59, 1994.

[12] Yan Chen, Rui Peng, and Zhong You. Origami of thick panels. Science, 349(6246):396–400, 2015.

[13] Jen-Hui Chuang, Chi-Hao Tsai, and Min-Chi Ko. Skeletonisation of three-dimensional object using generalized potential field. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1241–1251, 2000.

[14] Jian S Dai and Ferdinando Cannella. Stiffness characteristics of carton folds for packaging. Journal of mechanical design, 130(2):022305, 2008.

[15] Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. Neural network design. Martin Hagan, 2014.

[16] Luis Diago, Tetsuko Kitaoka, Ichiro Hagiwara, and Toshiki Kambayashi. Neuro-fuzzy quantification of personal perceptions of facial images based on a limited data set. IEEE transactions on neural networks, 22(12):2422–2434, 2011.

[17] Luis Diago, Julian Romero, Junichi Shinoda, Hiroe Abe, and Ichiro Hagiwara. A soft-computing approach for quantification of personal perceptions. In Advances in Affective and Pleasurable Design, pages 199–210. Springer, 2017.

[18] Luis Diago, Julian Romero, Junichi Shinoda, and Ichiro Hagiwara. Analyzing kansei from facial expressions by csrbf mapping. In KEER2014. Proceedings of the 5th Kanesi Engineering and Emotion Research; International Conference; Linköping; Sweden; June 11-13, number 100, pages 877–885. Linköping University Electronic Press, 2014.

[19] Stefan Duffner and Christophe Garcia. An online backpropagation algorithm with validation error-based adaptive learning rate. Artificial Neural Networks–ICANN 2007, pages 249–258, 2007.

[20] C. Elbrechter, R. Haschke, and H. Ritter. Folding paper with anthropomorphic robot hands using real-time physics-based modeling. In 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), pages 210–215, Nov 2012.

[21] S. Felton, M. Tolley, E. Demaine, D. Rus, and R. Wood. A method for building self-folding machines. Science, 345(6197):644–646, 2014.

[22] Takayuki Fujiwara, Takeshi Nishihara, Masafumi Tominaga, Kunihito Kato, Kazuhito Murakami, and H Koshirnizu. On the detection of feature points of 3d facial image and its application to 3d facial caricature. In 3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on, pages 490–496. IEEE, 1999.

[23] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley publishing company, Inc., 1989.

[24] José Gonçalves, José Lima, Paulo Malheiros, and Paulo Costa. Sensor and actuator stochastic modeling of the lego mindstorms nxt educational kit. In 10th Conference on Mobile Robots and Competitions, pages 11–16, 2010.

[25] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. Digital Image Processing Using MATLAB. Gatesmark Editorial, second edition, 2009.

[26] Thomas Hull. On the mathematics of flat origamis. Congressus numerantium, pages 215–224, 1994.

[27] BioID Inc. Bioid face database v1.2. `https://www.bioid.com/About/BioID-Face-Database`, 2011 (accessed May 14, 2015).

[28] Mitsuo Kawato. Feedback-error-learning neural network for supervised motor learning. Advanced neural computers, 6(3):365–372, 1990.

[29] Mitsuo Kawato, Kazunori Furukawa, and R Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. Biological cybernetics, 57(3):169–185, 1987.

[30] Yasuyuki Kihara and Yasuyoshi Yokokohji. Skill transfer from human to robot by direct teaching and task sharing -a case study with origami folding task-. IFAC Proceedings Volumes, 43(13):454 – 459, 2010. 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems.

[31] Tetsuko Kitaoka, Luis A Diago, Ichiro Hagiwara, Satoshi Kitazaki, and Shigeru Yamane. Definition, detection and generation of iyashi expressions. Journal of Computational Science and Technology, 2(4):413–422, 2008.

[32] Frank Klassner. A case study of lego mindstorms'™ suitability for artificial intelligence and robotics courses at the college level. In ACM SIGCSE Bulletin, volume 34, pages 8–12. ACM, 2002.

[33] Peter J Lang. International affective picture system (iaps): Affective ratings of pictures and instruction manual. Technical report, 2005.

[34] Robert J Lang. Treemaker v5.0. `http://www.langorigami.com/article/treemaker`, 2005 (accessed January 9, 2017).

[35] Der-Tsai Lee and Franco P Preparata. An optimal algorithm for finding the kernel of a polygon. Journal of the ACM (JACM), 26(3):415–421, 1979.

[36] Frank L Lewis, Aydin Yesildirek, and Kai Liu. Multilayer neural-net robot controller with guaranteed tracking performance. IEEE Transactions on Neural Networks, 7(2):388–399, 1996.

[37] Tama Software Ltd. Pepakura desing. `http://www.tamasoft.co.jp/pepakura-en/download/index.html`, 2010 (accessed November 04, 2017).

[38] HM Maldonado-Del Toro, R Silva-Ortigoza, ER Ramos-Silvestre, VM Hernández-Guzmán, and JC Rivera-Díaz. Modelado y simulación de un robot rígido de dos grados de libertad. Lat. Am. J. Phys. Educ. Vol, 5(1):321, 2011 (In Spanish).

[39] Yamato Matsubara, Makoto Noguchi, Atsushi Satoh, and Kenji Sugimoto. Virtual feedback error learning control. In SICE Annual Conference (SICE), 2011 Proceedings of, pages 1715–1720. IEEE, 2011.

[40] Takeshi Matsui. Institutionalization of consumer needs: The case of the" healing boom" in japan. 2008.

[41] Joseph A Mikels, Barbara L Fredrickson, Gregory R Larkin, Casey M Lindberg, Sam J Maglio, and Patricia A Reuter-Lorenz. Emotional category data on images from the international affective picture system. Behavior research methods, 37(4):626–630, 2005.

[42] Jun Mitani. A design method for 3d origami based on rotational sweep. Computer-Aided Design and Applications, 6(1):69–79, 2009.

[43] Jun Mitani. Ori-revo v1.01. `http://mitani.cs.tsukuba.ac.jp/ori_revo/#install`, 2011 (accessed January 2, 2017).

[44] Hiroyuki Miyamoto, Mitsuo Kawato, Tohru Setoyama, and Ryoji Suzuki. Feedback-error-learning neural network for trajectory control of a robotic manipulator. Neural networks, 1(3):251–265, 1988.

[45] Toshimitsu Musha, Yuniko Terasaki, Hasnine A Haque, and George A Ivamitsky. Feature extraction from eegs associated with emotions. Artificial Life and Robotics, 1(1):15–19, 1997.

[46] Takayuki Nakamura, Yoshio Sakata, Toshikazu Wada, and Haiyuan Wu. High-performance active camera head control using palm-tree. Journal of Robotics and Mechatronics, 21(6):720, 2009.

[47] Jun Nakanishi and Stefan Schaal. Feedback error learning and nonlinear adaptive control. Neural Networks, 17(10):1453–1465, 2004.

[48] Masaru Nakazawa. Special issue on handling of flexible object. Journal of Robotics and Mechatronics, 10(03):167–169, 1998.

[49] Akio Namiki, Koichi Hashimoto, and Masatoshi Ishikawa. A hierarchical control architecture for high-speed visual servoing. The International Journal of Robotics Research, 22(10-11):873–888, 2003.

[50] Paul S Nebosky, Steven R Schmid, and M-A Sellés. The springback characteristics of a porous tantalum sheet-metal. Journal of Manufacturing Science and Engineering, 133(6):061022, 2011.

[51] Taketoshi Nojima. Origami modeling of functional structures based on organic patterns. Master's thesis, Graduate School of Kyoto University, Kyoto, Japan, 2002.

[52] Katsuhiko Ogata and Yanjuan Yang. Modern control engineering. 1970.

[53] Timo Partala, Maria Jokiniemi, and Veikko Surakka. Pupillary responses to emotionally provocative stimuli. In Proceedings of the 2000 symposium on Eye tracking research & applications, pages 123–129. ACM, 2000.

[54] VP Plagianakos, GD Magoulas, and MN Vrahatis. Learning rate adaptation in stochastic gradient descent. In Advances in convex analysis and global optimization, pages 433–444. Springer, 2001.

[55] Franco P Preparata and Michael Ian Shamos. Introduction. In Computational Geometry, pages 1–35. Springer, 1985.

[56] J Romero, L Diago, J Shinoda, and I Hagiwara. Verification of models of personal perception of faces by closed-eye classifier using histogram correlation. In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pages V01AT02A056–V01AT02A056. American Society of Mechanical Engineers, 2015.

[57] Julian Romero, Luis Diago, Junichi Shinoda, and Ichiro Hagiwara. Comparison of data reduction methods for the analysis of iyashi expressions using brain signals. Journal of Advanced Simulation in Science and Engineering, 2(2):349–366, 2015.

[58] Julian A. Romero, Luis A. Diago, Chie Nara, Junichi Shinoda, and Ichiro Hagiwara. Norigami folding machines for complex 3d shapes. In ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, volume 5B: 40th Mechanisms and Robotics Conference, 2016.

[59] Xiaogang Ruan, Mingxiao Ding, Daoxiong Gong, and Junfei Qiao. On-line adaptive control for inverted pendulum balancing based on feedback-error-learning. Neurocomputing, 70(4):770–776, 2007.

[60] James A Russell and Lisa Feldman Barrett. Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. Journal of personality and social psychology, 76(5):805, 1999.

[61] Mikael Schill, Johan Tryding, Jesper Karlsson, Dynamore Nordic, and Tetra Pak. Simulation of forming of paperboard packaging using ls-dyna. 2015.

[62] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. Robot modeling and control, volume 3. Wiley New York, 2006.

[63] Samuel D Stearns. of aldapfive signal processing. 1985.

[64] Kenji Sugimoto, Akihiro Doi, and Tadashi Matsumoto. Feedforward learning control by scheduled locally weighted regression. IFAC Proceedings Volumes, 46(11):239–244, 2013.

[65] John G Sutherland. A holographic model of memory, learning and expression. International Journal of Neural Systems, 1(03):259–267, 1990.

[66] Tomohiro Tachi. Simulation of rigid origami. Origami, 4:175–187, 2009.

[67] Tomohiro Tachi. Origamizing polyhedral surfaces. IEEE transactions on visualization and computer graphics, 16(2):298–311, 2010.

[68] Kazuhiko Takahashi et al. Remarks on emotion recognition from bio-potential signals. In 2nd International conference on autonomous robots and agents, volume 3, pages 1148–1153, 2004.

[69] Kenta Tanaka, Yusuke Kamotani, and Yasuyoshi Yokokohji. Origami folding by a robotic hand. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pages 2540–2547. IEEE, 2007.

[70] Alan Watt. 3D Computer Graphics. Person Education Limited, third edition, 2000.

[71] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in computational Mathematics, 4(1):389–396, 1995.

[72] Paul John Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Doctoral Dissertation, Applied Mathematics, Harvard University, MA, 1974.

[73] Daniel M Wolpert, R Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. Trends in cognitive sciences, 2(9):338–347, 1998.

[74] Yorihisa Yamamoto. Nxtway-gs model-based design-control of self-balancing two-wheeled robot built with lego mindstorms nxt. Cybernet Systems Co., Ltd, 2008.

[75] Wei Yao and Jian Dai. Dexterous manipulation of origami cartons with robotic fingers based on the interactive configuration space. Journal of Mechanical Design, 130(2):022303–022303–8, 2007.

[76] John G Ziegler and Nathaniel B Nichols. Optimum settings for automatic controllers. trans. ASME, 64(11), 1942.

[77] Shannon A Zirbel, Robert J Lang, Mark W Thomson, Deborah A Sigel, Phillip E Walkemeyer, Brian P Trease, Spencer P Magleby, and Larry L Howell. Accommodating thickness in origami-based deployable arrays. Journal of Mechanical Design, 135(11):111005, 2013.