

Politecnico di Torino

Collegio di Ingegneria Informatica, del Cinema e Meccatronica.

Master Degree in Mechatronic Engineering



Thesis:

Sensor Integration and Controller Design for a Tensegrity - Modular Robot.

Supervisor from Politecnico di Torino:

Prof. Dr. Marcello Chiaberge

Supervisors from Ecole Polytechnique Federale de Lausanne:

Prof. Dr. Dario Floreano

Prof. Dr. Yusuke Ikemoto

Candidate

Dr. Jun Shintake

Luigi Campanaro

Davide Zappetti

April 2018

Table of contents

1	Introduction.....	6
1.1	Tensegrity structure principle	6
1.2	Modularity in robotics.....	7
1.3	Tensegrity modular robots	8
1.4	Objective	11
2	State of the art.....	12
2.1	State of the art: sensors	12
2.1.1	State of the art: sensors in modular robotics.....	12
2.1.1.1	Modular ATRON.....	12
2.1.1.2	PetRo	12
2.1.1.3	Cellular robotic system: CEBOT.....	13
2.1.1.4	Polypod.....	13
2.1.1.5	CONRO robot.....	14
2.1.1.6	YaMoR	15
2.1.1.7	Odin	15
2.1.2	State of art: sensors in tensegrity robotics	16
2.1.2.1	SUPERball.....	16
2.1.2.2	ReCTeR	17
2.2	State of art: control.....	18
2.2.1	State of art: control in modular robotics	18
2.2.1.1	Gait Table	18
2.2.1.2	Hormone controller.....	19
2.2.1.3	Sinusoidal controller.....	19
2.2.1.4	Central Pattern Generator	20
2.2.2	State of art: control in tensegrity robotics	21
2.2.2.1	NASA ReCTeR	21
2.2.2.2	Tensegrity quadruped	23
2.2.2.3	Spine-like tensegrity robot.....	23
2.2.2.4	Skelton approach to closed-loop control of tensegrity structures.....	25
2.2.2.5	ULTRA Spine robot	25
2.2.2.6	Neural Networks and tensegrities.....	26
3	Validation environment.....	27
4	Proposed sensor technologies.....	28
4.1	Single module approach.....	28
4.1.1	Definition of the variable measured by the sensor	28

4.1.2	Sensing deformation in literature	28
4.1.3	Conclusions	29
4.2	Multiple modules approach.....	30
4.2.1	Multiple modules approach definition.....	30
4.2.2	Advantages and disadvantages wrt single module approach.....	30
4.3	Candidate sensors technology	31
4.4	Conclusions.....	32
4.5	Sensor positioning.....	33
5	Hardware components realization	37
5.1	Capacitive strain sensors	37
5.1.1	Manufacturing process	37
5.1.2	Electrodes connection.....	39
5.2	Design of supports for tensegrity's struts.....	40
5.3	Sensor integration.....	42
5.3.1	Sylgard silicone	42
5.3.2	Solution 1: sensor made in Ecoflex 00-30.....	42
5.3.3	Solution 2: NinjaFlex net + sensor made in EcoFlex 00-30.....	43
5.4	Capacitive strain sensor test.....	44
5.4.1	LRC meter results.....	44
5.4.2	Resistive sensor characterization: hardware and results.....	47
5.4.2.1	Voltage divider	47
5.4.2.2	First order filter design	48
5.4.2.3	Modified servo motor for hysteresis analysis.....	49
5.4.2.4	Resistive sensor characterization: results	50
6	Proposed controller	52
6.1	Artificial Neural Network	52
6.2	Central Pattern Generator.....	54
7	Controller implementation and tuning.....	59
8	Conclusion	63
9	Future work.....	65
10	References.....	66
11	List of figures.....	71

1 Introduction

1.1 Tensegrity structure principle

The word tensegrity is the fusion between two terms: tensional and integrity. It has been coined by the architect Buckminster Fuller [1,2,7] to describe a structural principle based on discontinuous compression elements, the struts, suspended in a web of tensional elements, the cables [5,6,8] **Figure 1-1**.

The combination of these two components leads to free-standing structures which are characterized by a highly efficient use of materials [4] and in which each elements experience either pure compression or pure tension.

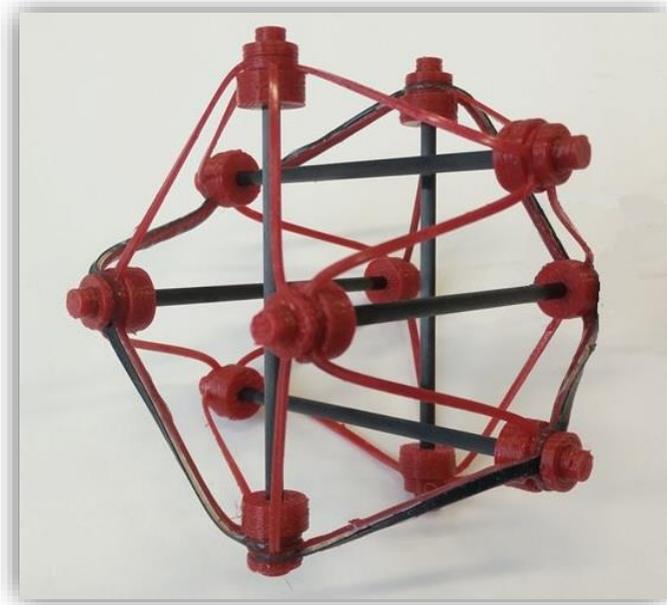


Figure 1-1: Tensegrity structure, cables in red and struts in black.

Tensegrity principle is not an artificial concept, in fact it has been discovered in many aspects of biological systems [9,10]. At different scales, they emerge to be heavily used in living organism: from the cytoskeleton of cells **Figure 1-2** to the skeletal structure **Figure 1-3**, that according to this theory is a net of muscles, tendons, etc. that surround and supports bones, without rigid connections between them [3].

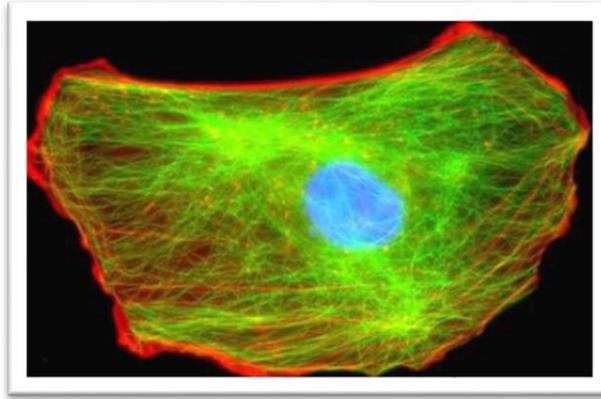


Figure 1-2: Cell picture obtained with fluorescence microscopy. In blue is the nucleus of the cell and at the bottom a sketch of the two main constituents of the cytoskeleton: microtubules and actin filaments [10].

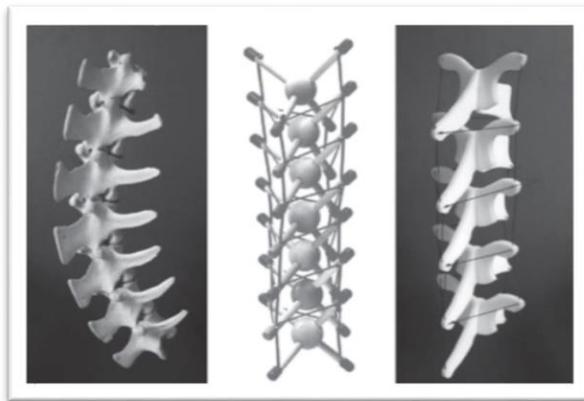


Figure 1-3: Tensegrity models of the spine show how vertebrae float without touching [2].

1.2 Modularity in robotics

Modular robotics **Figure 1-4** provide effective solutions to some applications for which available prototype cannot be adapted, for example due to changes in the environment in which the robot should work.

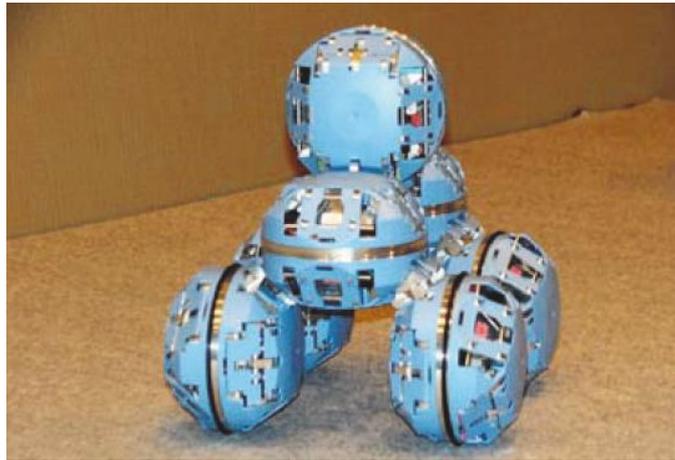


Figure 1-4: ATRON robot.

In addition, the maintenance and repair of classic robots requires external specialized personnel and time. While in case of modular robots, it is very fast and more economic to change just the broken modules and train workers to do that is much easier [11,12].

Furthermore, modular robots represent an intrinsic redundant and fault-tolerant architecture [26], ideal to work in un-structured environments.

1.3 Tensegrity modular robots

This research deals with a modular robot whose atomic components are tensegrity structures **Figure 1-5**, in particular, it involves icosahedron tensegrity structures **Figure 1-6**.

The module should be able to deform in every direction, it should involve the least number of struts and cables as possible to be easier manufactured and assembled and finally the internal volume should be left free to carry payloads.

The candidate that matched all these criteria is icosahedron: it has a spherical shape, the symmetry leads to similar mechanical properties along different directions, among the tensegrities with spherical shape it is the one with the lowest number of cables (24) and struts (6) and it has also a free inner volume not crossed by any bar [10].

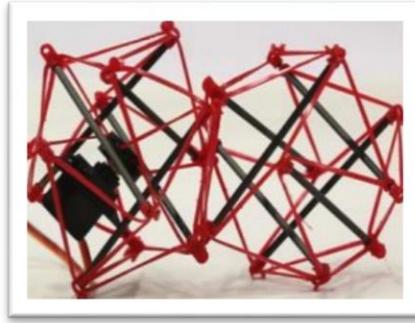


Figure 1-5: A couple of modules belonging to a tensegrity modular robot [10].

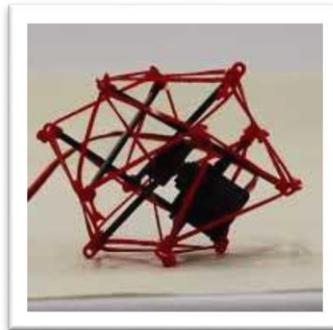


Figure 1-6: Icosahedron tensegrity structure [10]

The definition of module here is critical, in fact in tensegrity robotics literature with the word module sometimes they refer to a single strut that belongs to a tensegrity **Figure 1-7** [13]. For example, according to this idea, an icosahedron tensegrity is made by six different modules, maybe because usually at the bottoms of these struts there are motors, hence they are perceived as standalone entities.

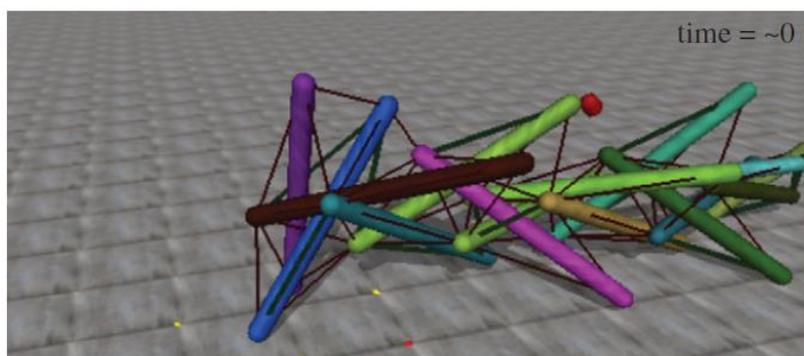


Figure 1-7: A complex and highly dynamically coupled 15-bar tensegrity structure [13].

To clarify this concept, a tensegrity spine has been considered **Figure 1-8**, it is made of rigid segments and cables. When segments are opportunely connected to cables and these ones are in tension the spine structure take shape. It is important to point out that each of this segment itself is not a tensegrity, it is just a rigid body and a robot as the one presented in [22] cannot be considered a modular tensegrity robot as it is considered here.

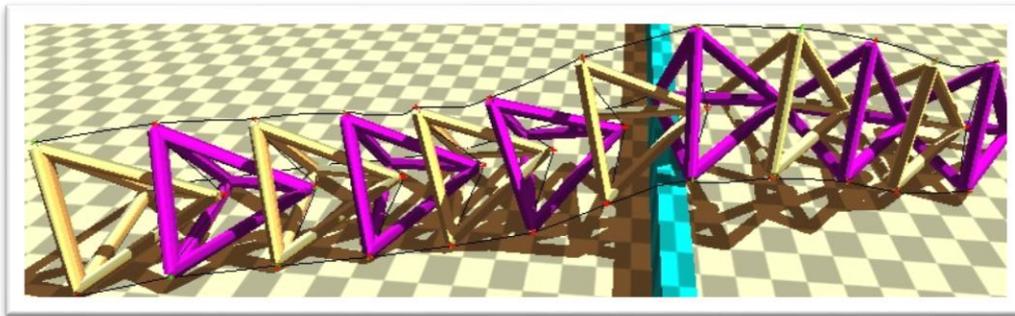


Figure 1-8: Spine-like tensegrity robot [22].

A modular tensegrity robot, indeed, is intended as a modular robot whose atomic components are tensegrity structures. When the modules are separated, each of them obeys to the kinematics, statics and dynamics laws of tensegrity structures.

Tensegrities are extremely lightweight structures able to passively adapt to external forces and redistribute loads effectively through the tension network [5,6,7,8,10]. Thus, increasing mechanical robustness and tolerance to forces [4,6], features that make them ideally suited for operation in dynamic environments **Figure 1-9** where contact forces cannot always be predicted [2,7]. Moreover, their intrinsic compliance allows them to work beside humans safely [7,8]. Tensegrities are also robust to failures, in fact they can degrade performance in the event of actuation or structural failure [6]. Another feature that makes tensegrity appealing for robotics is the deployability from compact configurations, represented in **Figure 1-10**.

Despite these useful properties, tensegrities have a highly non-linear dynamics that causes non-linear propagation of forces in the structure and complex oscillatory motion, due to their interaction with the environment and make them very complicated to control with traditional algorithms [5,6,13].



Figure 1-9: Super Ball Bot all-in-one landing and mobility platform based on tensegrity structures. All credits to NASA research center.

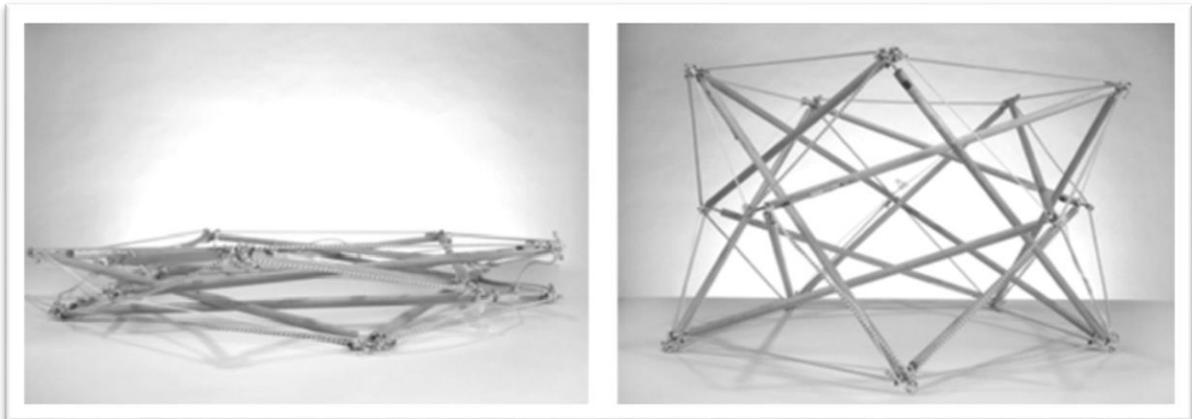


Figure 1-10: Tensegrity deployability example, from the compact configuration to the expanded configuration [25].

1.4 Objective

All the features seen make very appealing a robot made with tensegrity structures and to perform tasks with it, a controller is needed.

Hence, the objective of this research is to design a controller for tensegrity-modular robots and particular attention has been reserved to sensors integration, since it has been proven their role in improving the performance of tensegrity robots on rough terrains and in presence of external forces [2, 22].

2 State of the art

2.1 State of the art: sensors

2.1.1 State of the art: sensors in modular robotics

2.1.1.1 Modular ATRON

The ATRON **Figure 2-1** system consists of several fully self-contained modular robots, each having their own processing power, power supply, sensors and actuators. Each module can be connected up to eight other modules [33]. Every ATRON module is equipped both with internal and external sensors. For external sensing each module is equipped with a 2-axis accelerometer to detect tilt, consequently due to missing information related to the 3rd axis the module cannot detect if is upside down. Moreover, each module mounts infra-red sensors, that can be used as primitive distance sensor and for obstacle detection.

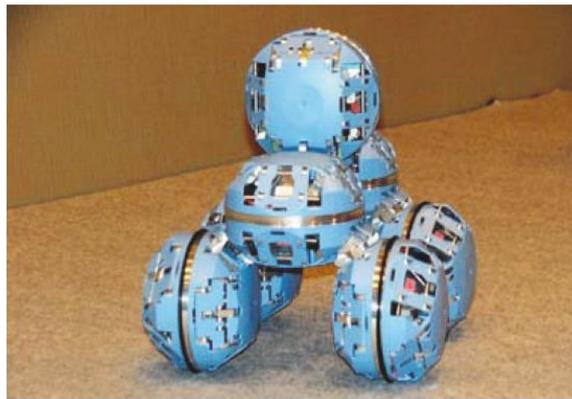


Figure 2-1: ATRON System [34].

2.1.1.2 PetRo

PetRo **Figure 2-2** is a tetrapod structured robot that can form 3D shapes. Each leg is connected to wheels to provide mobility, these also have a central role in connecting different modules. The IR sensors, on the connector faces, are useful for alignment during docking procedure [34].

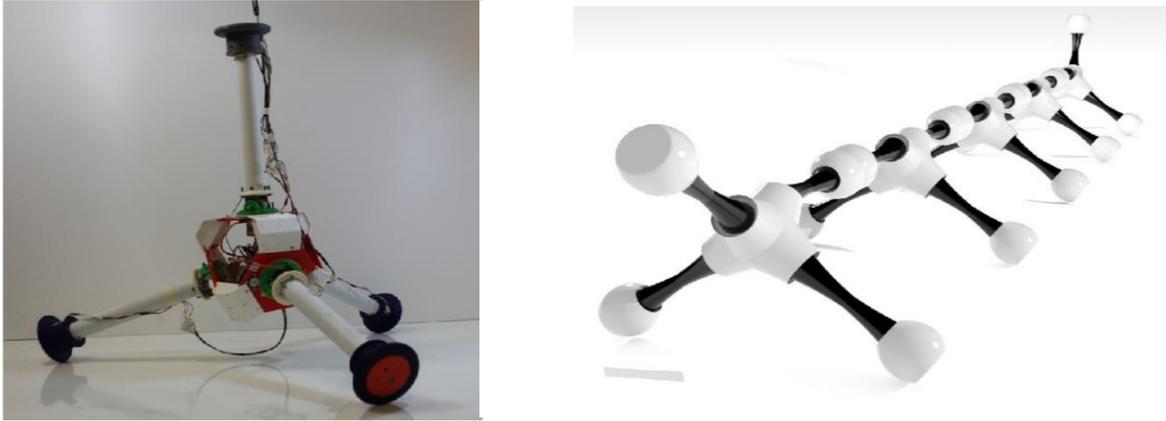


Figure 2-2: PetRo prototype on the left, caterpillar configuration on the right [35].

2.1.1.3 Cellular robotic system: CEBOT

The cellular robotics system (CEBOT) **Figure 2-3** is born to solve the adaptability problem of traditional robot in an industrial environment, in fact due to its cellular structure it is applicable to different environment and tasks. IR sensor have been mounted on board for docking and undocking operation [36].

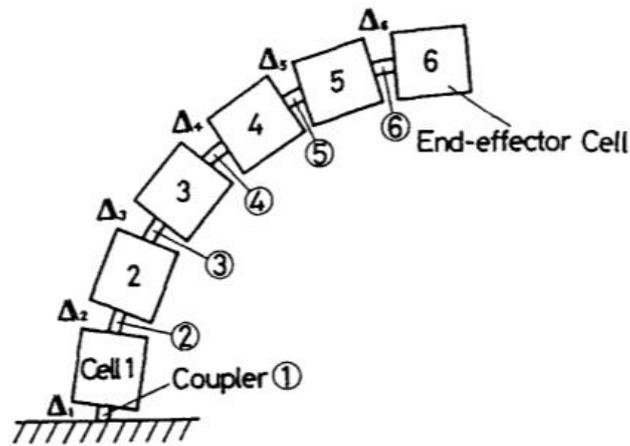


Figure 2-3: CEBOT robot [36].

2.1.1.4 Polypod

Polypod **Figure 2-4** is a modular robot developed at Palo Alto research center, it consists of two types of modules, namely segments and nodes. The nodes are cubic rigid module that presents a single connector for batteries on each face [34], while segments provide two DOF, being able to extend, contract and rotate left and right. Potentiometers are used as angular sensors and IR as proximity sensors [37].

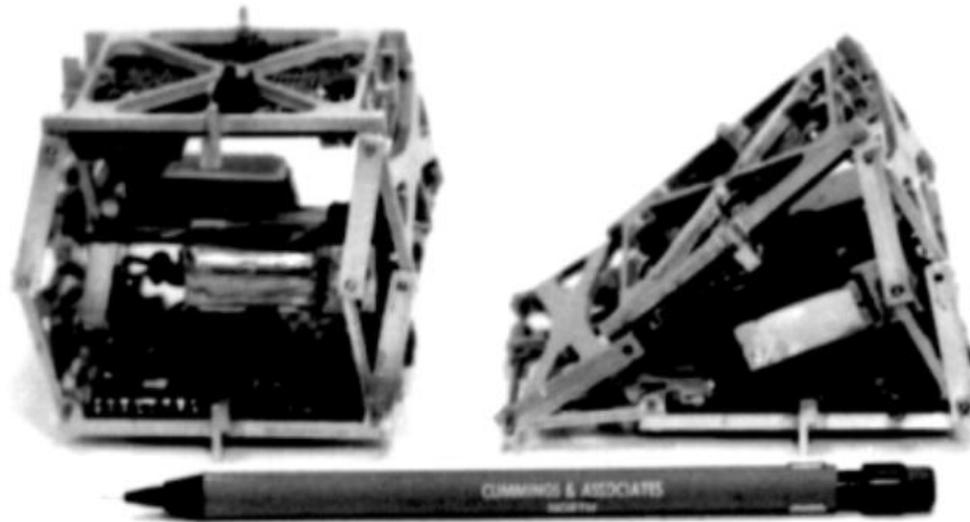


Figure 2-4: Two polypod's segments, one expanded on the left and one angled on the right [37].

2.1.1.5 CONRO robot

CONRO **Figure 2-5** robot is composed by modules that have two degrees of freedom, DOF1 dedicated to pitch and DOF2 dedicated to yaw [27], actuated through servomotors. IR transmitters are used during docking, while analog position sensors are used to check the actuator position.

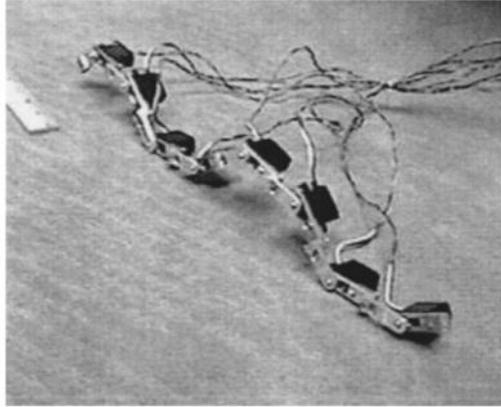


Figure 2-5: CONRO robot snake configuration [38].

2.1.1.6 YaMoR

YaMoR **Figure 2-6** is robot designed to explore adaptive locomotion control, in fact modules can be assembled in different types and shapes. Each module contains its own controllers and sensors; hence it can also be used to explore distributed control [39]. The sensors that have been embedded in the modules are a 3-axis accelerometer and IR proximity sensors [29].

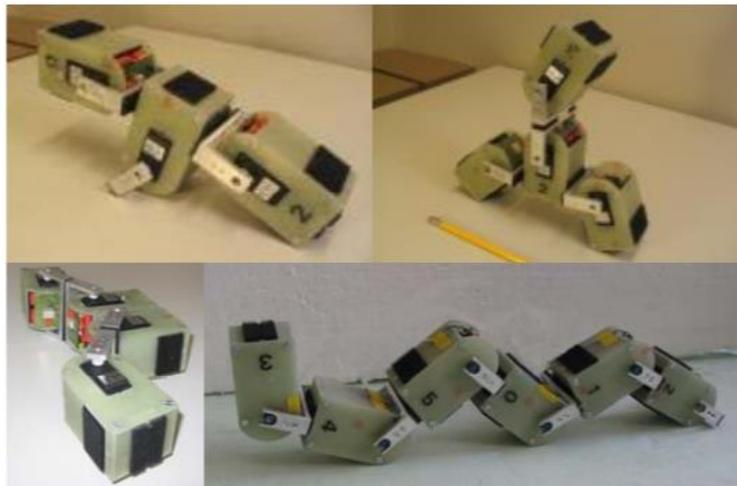


Figure 2-6: Different configurations of YaMoR modules [39].

2.1.1.7 Odin

Odin robot **Figure 2-7** has a deformable structure and depending on its configuration it can passively deform and adapt to external forces. Through actuators the robot can actively change the distribution of internal forces and consequently, the shape is the result of the internal and external forces applied to the system. Odin is made by four different types of modules: structure, actuated, sensor and power modules. To be able to read the absolute extension of the linear actuator, it is equipped with an optical sensor which detects the reflection of a grayscale gradient strip inside the align shell [40].

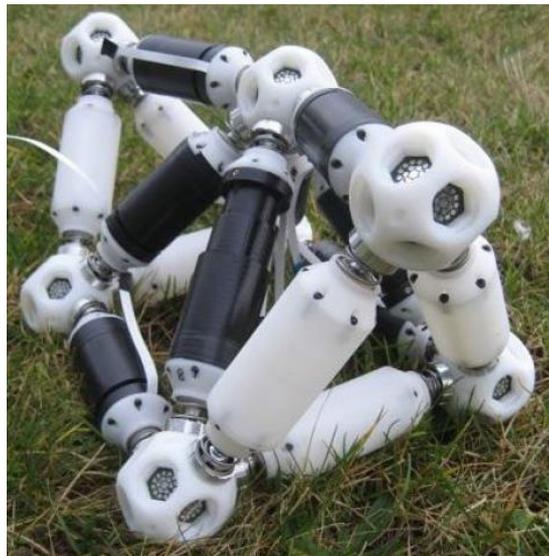


Figure 2-7: Odin robot with 21 modules: 8 telescoping link modules (black links), 6 rigid passive link modules (white links), and 7 joint modules.

2.1.2 State of art: sensors in tensegrity robotics

2.1.2.1 SUPERball

SUPERball is a tensegrity icosahedron (6 rods and 24 cables) robot built at NASA Ames research center, tensegrity is a very interesting class of robot due to their intrinsic properties, as lightness, compliancy, storability, etc. However, its state estimation is not a well known field [5]. The approach chosen by NASA's researchers is the implementation of an unscented Kalman filter fed with inertial measurements, ultra-wideband time of flight ranging measurements and actuator state information. Each rod in each end-caps mounts a

platform that includes an inertial measurement unit, a motor with encoders and a robot operating system. To track the position of the robot, an internal and external measurement system has been implemented. Using ranging modules, it is possible to compute, from the flight time of the packages sent and received, the distance between emitter and receiver. In this way the distance between the end-caps is obtained, while to measure the absolute coordinates wrt, a fixed reference frame (world), it is needed to install eight fixed beacons **Figure 2-8**.

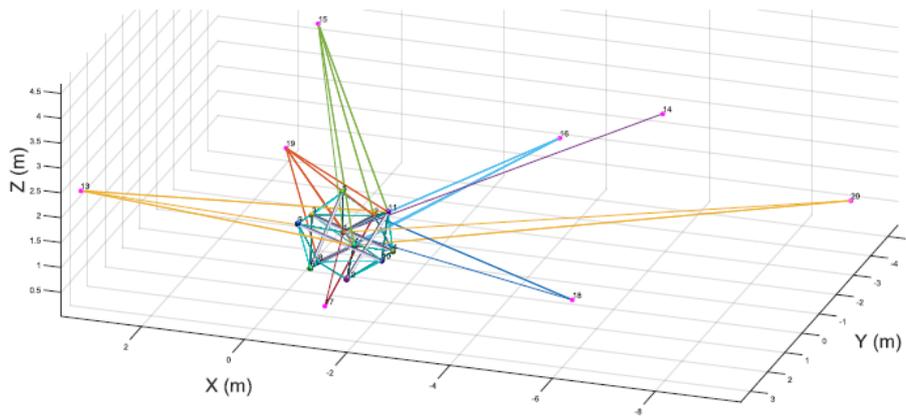


Figure 2-8: SUPERball and the 8 beacons installed to locate it wrt a fixed reference frame [5].

2.1.2.2 ReCTeR

Reservoir Compliant Tensegrity Robot (ReCTeR) is a highly compliant, lightweight (1.1 kg), underactuated tensegrity icosahedron robot. It is composed of 6 rods and 24 cables, only six of the latter are actuated. The properties seen (compliance, multi-path load distribution, nonlinear dynamics, etc.) make tensegrities ideal for physical interaction with the environment, but on the other hand present a significant challenge to traditional control approaches. To improve the performance of the controller researchers have added to the robot ground contact sensors in the end-caps [2].

2.2 State of art: control

This chapter offers an overview of the controllers that have been implemented in modular and tensegrity robotics.

2.2.1 State of art: control in modular robotics

2.2.1.1 Gait Table

One of the first locomotion control methods for reconfigurable robots is the gait table [30]. Its working principle is based on specifying for any time step the configuration of each DOF **Figure 2-9**. The control by gait table is easily implemented using a centralized controller. The decentralized version relies on synchronized timers in every module, however in case of timers' drift the robot starts moving in an unexpected way. These tables are easy to construct and to use, but when the configuration of the robot changes they need to be rewritten [26]. Furthermore, there is no communication among modules, hence they are in open-loop and consequently it seems to be difficult to integrate high level control actions [27].

Gait table has been applied to control CONRO robot, it is composed by modules that have two degrees of freedom, DOF1 dedicated to pitch and DOF2 dedicated to yaw [27]. When two or more modules are connected, it can accomplish different types of locomotion. In the pictures below is represented the caterpillar motion and the gait table used to implement it.

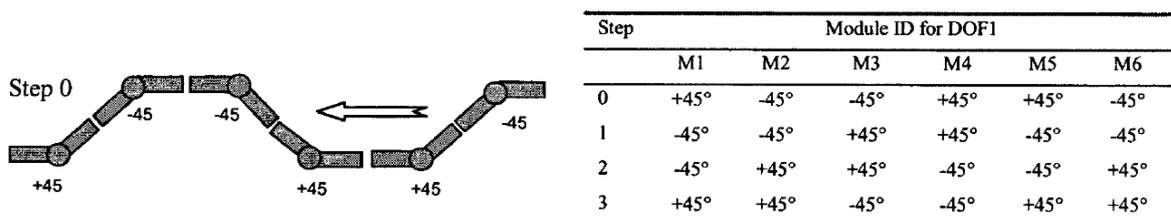


Figure 2-9: SUPERball and the 8 beacons installed to locate it wrt a fixed reference frame [5].

2.2.1.2 Hormone controller

Digital hormone is a bio-inspired controller developed to cope with the synchronization problem in decentralized gait tables. It is based on communication among modules, that can pass state information, the hormone, to the neighbors and in this way, they can share synchronized signals [30]. This communication system allows to cope with failures and changes in the initial configuration. Hormone controller result being a decentralized, homogeneous and scalable controller [26]. Usually the synchronization is in open-loop, in fact the initiator continues through the sequence of steps without knowing if the other modules have completed theirs. It can be made closed-loop feeding-back a hormone to the initiator to ensure that it does not continues to the next step if all the other modules have not completed the previous one. Nevertheless, this introduce a weakness in the system, because if a hormone is lost the whole robot stop moving, since the modules are waiting for that signal [30].

2.2.1.3 Sinusoidal controller

Phase shifted sinusoidal oscillators is an open loop controller with which it is possible to control worm like robots. This is the case of the robot implemented by Gonzalez-Gomez et al. [26], in which modules with only one DOF were connected. Also, Zhang et al. [26] has proposed caterpillars with shifted sinusoids, without any communication among modules. A similar approach has been considered in Zappetti et al. [10], where a modular robot made of three tensegrity structures **Figure 2-10** has been developed and controlled using phase shifted sinusoids.

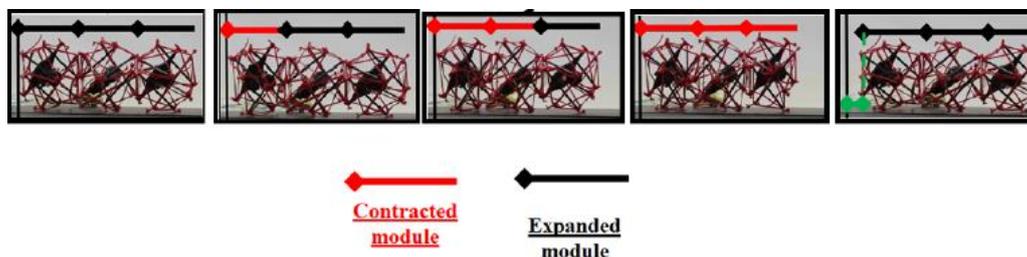


Figure 2-10: Tensegrity modular robot [10].

2.2.1.4 Central Pattern Generator

Central Pattern Generators (CPGs) are neural circuits present either in vertebrates and invertebrates, that have the ability to produce rhythmic outputs without the need of rhythmic inputs. They are called “central” because a sensory feedback from the peripherals of nervous system is not needed for generating the rhythmic behavior. In leaving creatures CGPs are involved in a lot of different activities: chewing, breathing, digesting and also locomotion.

Even if the feedback from sensory peripherals is not necessary, it accomplishes a very important role in shaping the rhythmic patterns, maintaining the movements coordinated [19]. Usually CPGs in robotics are built with two coupled differential equation to produce an oscillatory behavior. CPGs are decentralized control systems and their application in modular and reconfigurable robots has already been explored [19,20,21].

CPGs are neural networks and the results of the training procedure should be validated via hardware, because some interaction forces are not easily implementable in simulators and the results obtained there compared to the ones in the real environment could be different [19].

An example of CPG applied to modular robotics is Roombots **Figure 2-11**, which implement furniture that moves and self-reconfigures. The CPG model used can readily be implemented in a distributed modular robotic system, from a computational perspective it is cheap, it exhibits limit cycle behavior; temporary perturbations are rapidly forgotten, it is able to produce smooth trajectories even when control parameters are suddenly changed and it is robust against imperfect communication among modules [28].



Figure 2-11: Roombots modules.

Another example of Central Pattern Generator applied to modular robots is YaMoR **Figure 2-12**, it is a system of n coupled amplitude-controlled phase oscillators and in this case the learning session has been implemented on-line [29].

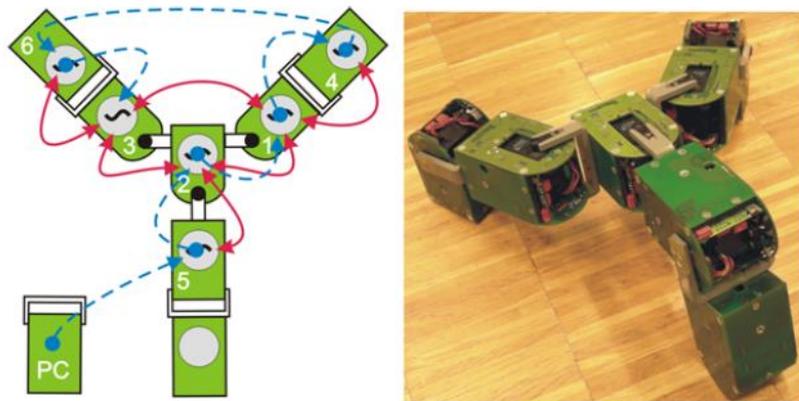


Figure 2-12: Example of tripod YaMoR robot, on the right and corresponding CPG configuration, on the left, in red lines [29]

2.2.2 State of art: control in tensegrity robotics

Tensegrity structures exhibit a lot of interesting properties: lightness, robustness to failures, impact tolerance and compliancy. However, as other soft robots are difficult to control with standard methods, due to their non-linear dynamics and complex oscillatory motion [6].

2.2.2.1 NASA ReCTeR

Reservoir Compliant Tensegrity Robot (ReCTeR) **Figure 2-13** is a highly compliant, lightweight (1.1 kg), underactuated tensegrity icosahedron robot. It is composed of 6 rods and 24 cables and only six of the latter are actuated.

The properties seen (compliance, multi-path load distribution, nonlinear dynamics, etc.) make tensegrities ideal for physical interaction with the environment, however on the oth-

er hand present a significant challenge to traditional control approaches. In fact, the first successful results in tensegrities motions are not based on analytical methods [2].

Starting from this, NASA researchers have approached the problem using CPGs instead of traditional strategies. CPGs present interesting features, for instance, distributed control, robustness to perturbations, inherent tolerance to redundancies, fast control loops and the ability to modulate locomotion using control signals [2].

The first and easiest implementation of the controller, both in software and in hardware, is the open loop, which has shown a basic rolling behavior, but it has failed in presence of external forces and unexpected terrain conditions. To cope with failures researchers have added to the robot ground contact sensors and these have improved the motion on various terrain [2].

The better performance obtained with sensors lead the researcher thinking to full state estimation of the robot using omnidirectional ranging sensors, that is something easy to implement in simulation, but it is quite hard to develop in real world.

The conclusion they have achieved is that the biologically-inspired control approaches explored appeared naturally suited for biologically-inspired tensegrity structures [2].

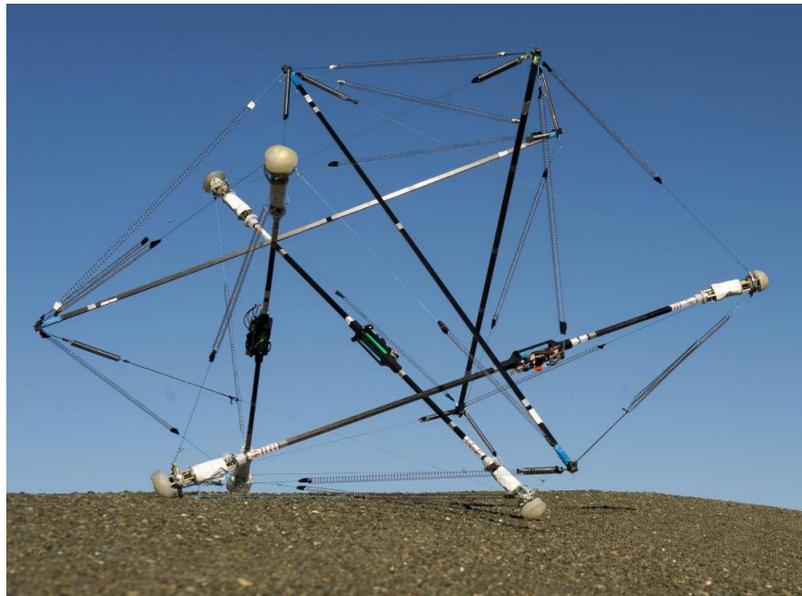


Figure 2-13: NASA ReCTeR [2].

2.2.2.2 Tensegrity quadruped

Another example of tensegrity robot is a quadruped developed by Hustig-Schultz et al. in collaboration with NASA Ames research center. It is a bio-inspired robot called MountainGoat **Figure 2-14**, which is based on locomotion via motion of the spine. In fact, in early steps of evolution, vertebrates locomoted using spine, while legs were developed later to enhance mobility. Due to counter intuitive nature of the structure's movements, instead of hand-designing the controller, a machine learning approach has been considered more effective [24]. The controller adopted is a CPG based on coupled oscillators equations with feedbacks. Researchers have introduced some feedback forces in the oscillator's equations and these have been computed using a neural network with two inputs (the length and the tension of the cables), one hidden layer made of four neurons and three outputs. The result is a robot able to explore, in simulation, spine-locomotion.

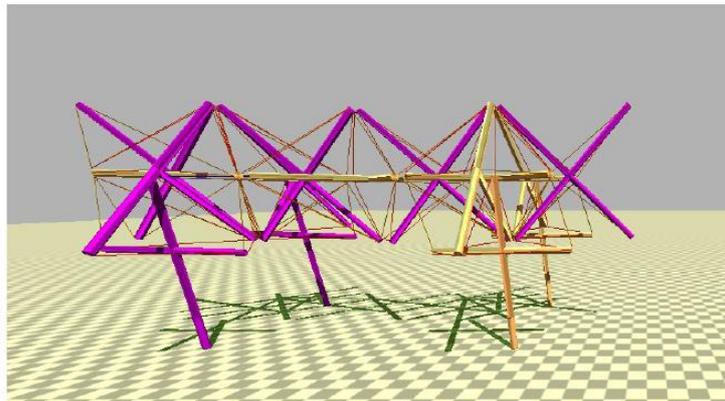


Figure 2-14: NASA ReCTeR [2].

2.2.2.3 Spine-like tensegrity robot

In this research [22] Mirletz et al. have started investigating whether CGP could have been used for whole robot control and they have addressed the central role of spine in allowing robots to adapt to rough terrain and unstructured environment. Spine compliant properties have been mimicked through a spine tensegrity structure **Figure 2-15**.

For what concerns control, the attention have been focused on CPG. In particular, a comparison between closed-loop CPG and open-loop CPG, in fact the latter is supposed to limit the type of terrain that robots can handle [22].

The CPG equations implemented are similar to the ones used for the tensegrity quadruped, in fact feedbacks forces are obtained using a neural network with two inputs (the length and the tension of the cables), one hidden layer made of four neurons and three outputs.

The performance test has been carried out only in simulation and on two different terrains, the first one is flat ground and the second one a ground with 2 [cm] hills.

The results are shown in the table below **Figure 2-16**. On flat ground the performance of the controller in open-loop is higher than the closed-loop, but when obstacles are added to the ground the closed-loop performed more than three times better than the open-loop controller. As conclusion the researchers stated that with feedback it is possible to improve the performance of CPG controller on rough terrain and, moreover, CPG is a strong candidate to control tensegrity structures.

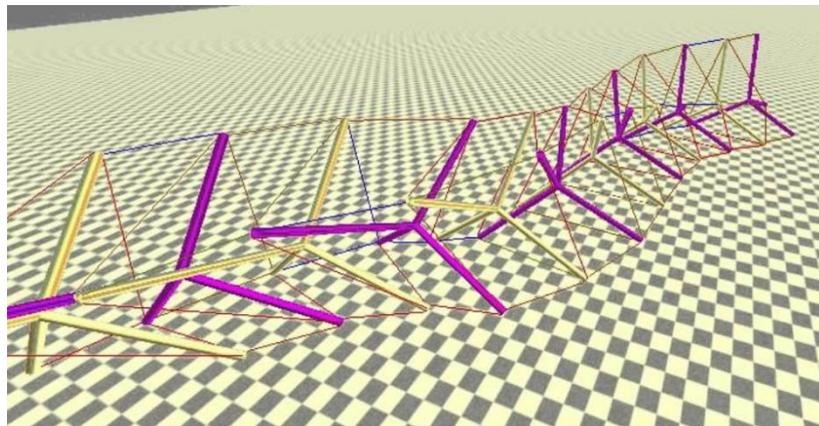


Figure 2-15: Spine-like tensegrity robot [22].

Terrain	Without Feedback	With Feedback
Flat Ground	492.8 [cm]	487.1 [cm]
2 cm Hills	30.4 [cm]	105.3 [cm]

Figure 2-16: CPG closed-loop and open-loop performances on different terrains [22].

2.2.2.4 Skelton approach to closed-loop control of tensegrity structures

Skelton and de Oliveira have proposed in 2009 an analytic approach to address the design of closed-loop control for tensegrity structures. In literature, as far as they knew, this topic has been addressed only for planar structures, due to simpler equations involved. For what concerns tridimensional tensegrity control, one has to deal with differential-algebraic equations, or singularities of the mass matrix and modelling the cables as elements that cannot take compression.

They have developed a Lyapunov-based control design that can drive the system from an initial configuration toward the desired target coordinates. But the control solution proposed is not admissible and furthermore, the controllability is lost in some configurations due to singularities [31].

2.2.2.5 ULTRA Spine robot

The controller implements a linearized model of the dynamics of the spine **Figure 2-17**, computed at each time-step, it is a model-predictive control (12 states per rigid body, 3 vertebrae, 36 states), in which the 24 cables are treated as inputs of the model. Even if the sum of errors trend toward zero, the length of time taken to solve the optimization problem for the controller (0:5-1 sec.) was longer than the timestep of the simulation (0:001 sec.). Thus, the optimization procedure need to be made more efficient before its application to hardware robots [32].

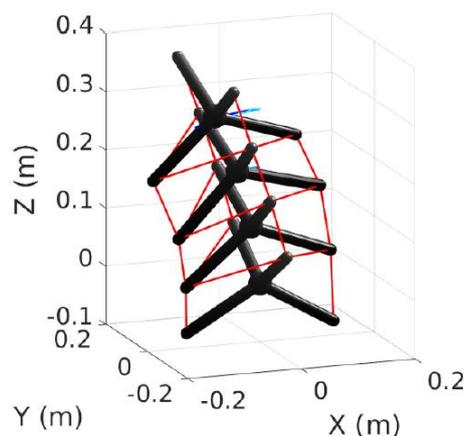


Figure 2-17: ULTRA Spine robot [32].

2.2.2.6 Neural Networks and tensegrities

Traditional engineering approaches strive to avoid non-linear dynamic couplings among components. In contrast, biological systems are characterized by non-linearities in their dynamics. An example of how non linearity has advantages are tensegrities structures and in the following research has been addressed the control of complex and highly dynamically 15-bar tensegrity structure **Figure 2-18** using a relatively simple controller: ANN with spiking neurons [13].

Each strut contains a single SNN with two inputs, corresponding to the tension sensed at the single actuated string on each end, two hidden nodes and two outputs. Using a simulator and genetic algorithms the SNN has been trained and after this step several different gaits, in term of speed, have been observed [13].

These results have demonstrated how the coupled dynamical properties of a complex mechanical system can be exploited for benefit rather than ‘engineered away’. Simultaneously, they have given an insight into why biological systems often contain the kind of complex coupled dynamics that are so often assiduously avoided in engineering.

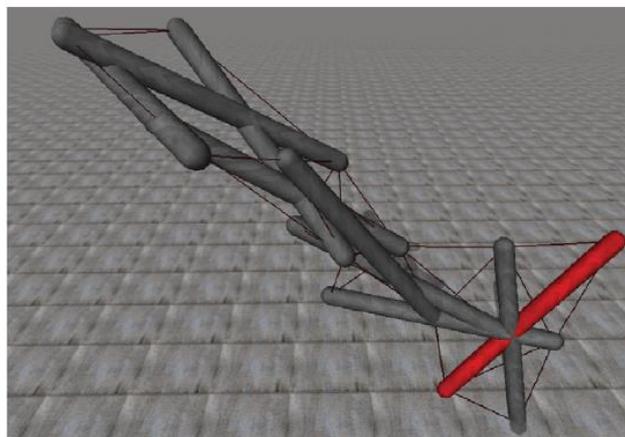


Figure 2-18: A complex and highly dynamically coupled 15-bar tensegrity structure [13].

3 Validation environment

A simulator for tensegrity robot was not available, hence the research has been carried out in hardware. Anyway, some interactions are very difficult to reproduce in simulation, so a hardware validation is suggested to prove the effectiveness of the results achieved [19].

4 Proposed sensor technologies

4.1 Single module approach

The first step has been understanding which quantities should have been tracked using sensors and in this direction the analysis has been focused on a single module.

4.1.1 Definition of the variable measured by the sensor

Considering a tensegrity module, one can notice that the main difference between a classic rigid modular robot and a tensegrity one is its ability to deform when subjected to external forces.

Compliance to the external environment is the reason why tensegrity has been chosen for this Bio-inspired robot.

4.1.2 Sensing deformation in literature

Reconfigurable robots have been traditionally designed to demonstrate basic capabilities of self-reconfiguration and locomotion without the context of an application. Hence, sensors are added only if they are useful in showing these capabilities [30]. This is the reason why the literature in modular robotics have not shown a lot of diversity related to them and in particular regarding sensors for deformations detection. While for what concern tensegrities, the need of measuring deformations has clearly emerged [2]. It is the case of SUPERball, an icosahedron tensegrity (6 rods and 24 cables) in which each end-cap of each rod is equipped with ranging sensors to estimate accurately the distance between them (proprioceptive state estimation) and between the structure and a fixed world frame [5] **Figure 4-1**.

The working principle behind that is the time-of-flight of the packages exchange among sensors, which is used to compute the distance. Unfortunately, the large number of packages exchanged during the communication among sensors have made the system inefficient as soon as the number of them has grown. To solve this problem NASA researchers have introduced “time broadcast messages”: when a module starts a measurement sequence it sends out a poll message and when another module receives this message, after a fixed delay, it emits its own message, followed by response and final message after additional delays.

One downside of this approach is that takes longer (60 m) than the previous one (3 ms). Then the data coming from ranging sensors, IMUs and rest length information are processed together by an unscented Kalman filter and after this step the state is sent to the controller **Figure 4-2**.

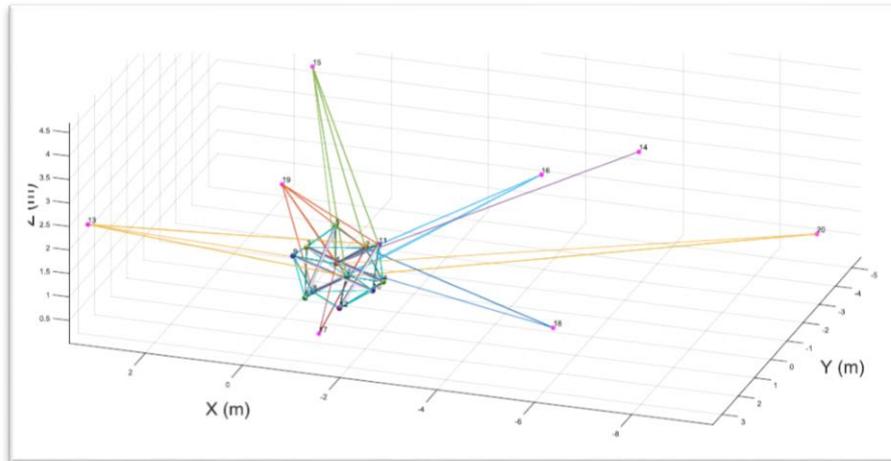


Figure 4-1: SUPERball sits in the middle of the plot surrounded by 8 ranging base stations [5].

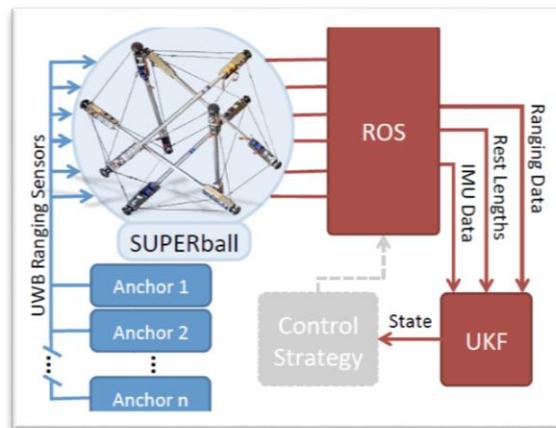


Figure 4-2: SUPERball data flow [5].

4.1.3 Conclusions

The approach described above has worked well for NASA SUPERball, but its application in our case study can be trickier.

In fact, applying ranging sensors to a theoretically infinite number of modules can be difficult due to interferences, it has to be considered that SUPERball has a diameter of more than 1[m], while the tensegrity built here has a diameter of 0.07[m] and this leads to a

very small time-of-flight. Moreover, for the sake of this project knowing the position of each vertex can be useless and this is the reason why a brand-new solution tailored for it has been investigated.

4.2 Multiple modules approach

4.2.1 Multiple modules approach definition

Instead of considering just one single module, till the beginning, multiple modules are taken into account.

4.2.2 Advantages and disadvantages wrt single module approach

Considering multiple modules, one can focus his attention on the *shape* reconstruction of the robot, taking into account the minimum data necessary to roughly understand where the end effector or in general the parts of interests are located. For example, the COG of each tensegrity could be enough to understand where the manipulator is **Figure 4-3**.

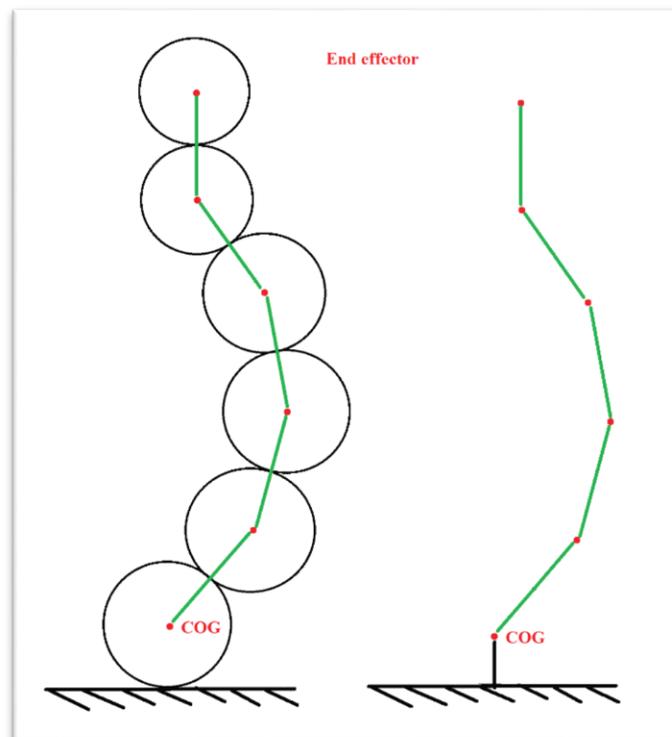


Figure 4-3: Multiple module shape reconstruction using COGs.

This is more convenient than investigating where each of the vertex of each of the modules are, because allows to place less sensors and consequently feed the controller with less data.

On the other hand, the information is less precise and depending on which is the intended task, it could be necessary to implement, locally, a different approach. For example, grasping task may require more sensorized modules to perceive more accurately objects.

4.3 Candidate sensors technology

The candidate technologies able to sense deformations are wireless sensor network and ultra-stretchable strain sensors.

Wireless sensors network has been successfully implemented by NASA to control SUPERball [5], the downside of this solution is that it requires beacons for triangulation, making the total architecture bulky. Then, the dimensions of tensegrity modules considered here are small (diameter $\cong 6,5\text{cm}$) and constrained by the 3D printing manufacturing process, hence the distances could be too small to be detected by this technology. Furthermore, the scalability, intended as the number of modules used, could be a problem, due to noise and interferences and it may lead to a more complicated signal processing.

Ultra-stretchable strain sensors **Figure 4-4** have been mechanically characterized in the LIS Lab. at EPFL with 10'000 cycles, the manufacturing process is well known, no beacon or other bulky equipment is needed, they can be embedded directly in the structure of the robot and they present a linear characteristic over the whole domain.



Figure 4-4: Capacitive stretchable sensor tested at EPFL [23].

4.4 Conclusions

The technology selected for the sensors are the ultra-stretchable strain sensors, because they can be directly embedded in the structure, in this way it has both mechanical and sensoristic functions. Furthermore, they are extremely lightweight, can detect external forces and the inner volume of the robot is left free.

Using ultra-stretchable strain sensors, the shape of the robot will not be directly reconstructed using the positions of COGs, however it will be deducted by the controller sensing the deformations.

Ultra-stretchable strain sensors can be divided into two classes: resistive and capacitive sensors. An entire research study about their technical features has been conducted by Shintake et al. [23] to compare these two technologies.

The capacitive sensors have displayed superior performance compared to the resistive ones except the gauge factors: high linearity **Figure 4-5** and low hysteresis under different cycles, strain amplitude ranging from 50% to 500%, high repeatability and durability up to 10100 cycles **Figure 4-6** and good temperature tolerance. On the other hand, the resistive strain sensors exhibited variable, but higher gauge factor [23].

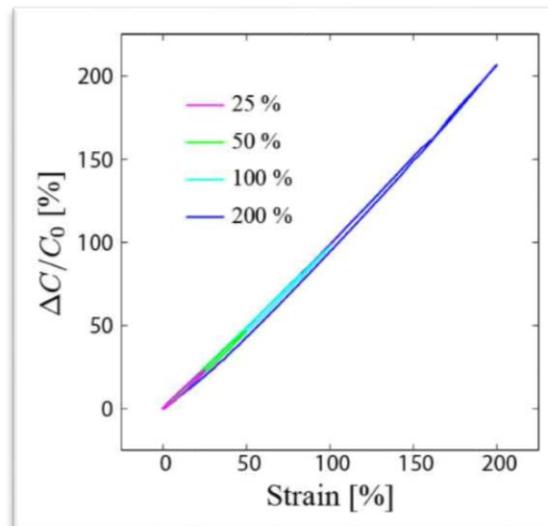


Figure 4-5: One cycle test with different elongations [23].

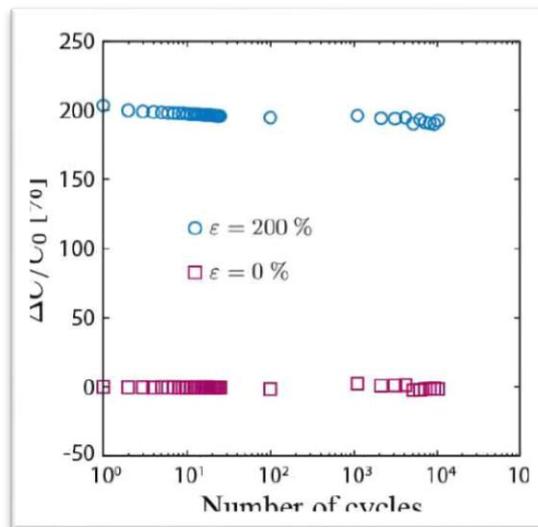


Figure 4-6: Ten thousand cycles extremum points [23].

4.5 Sensor positioning

After having defined the sensor technology that will be implemented, it is fundamental to choose the features of the robot that will be used for the validation.

Aiming to demonstrate that the sensor technology proposed can be used to detect the deformations of the robot, the easiest possible architecture has been considered.

The prototype on which sensors have been implemented and tested is a robot made of three modules **Figure 4-7** and able to move only in one direction, without the possibility of steering.

Furthermore, always for simplicity, the sensor has to detect only the forces imposed on the module along the direction of the actuation, external actions applied along other directions will not be detected.

The number of sensors and their placement is critical, especially with more complex forces. In the scenario described, one sensor per module is needed, because only one direction has been considered noteworthy, while in general, complex stresses may require more than one sensor per module to be detected.

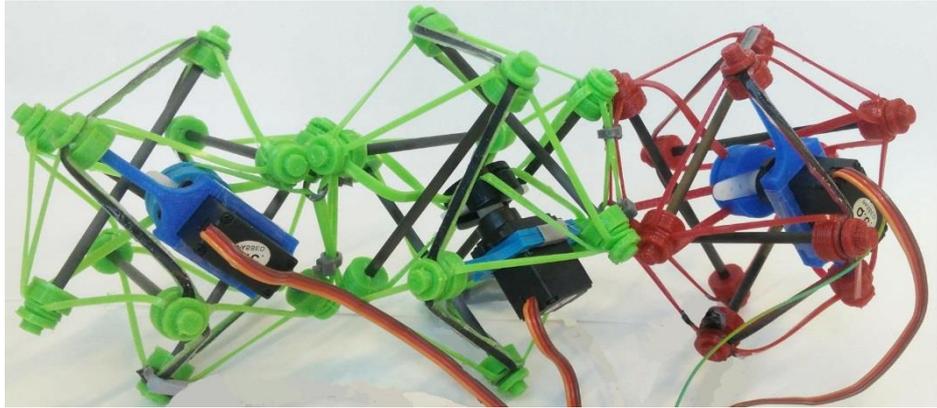


Figure 4-7: Tensegrity modular robot made of three modules and motion direction.

After having fixed the number of sensors needed, the next step is to set where they have to be positioned and in this sense an algorithm has been developed:

1. Identify which are the forces that affect the module (in this case one, along the direction of the actuation).
2. Apply each of these forces one by one to the module in simulation or on sensorized prototypes.
3. Then, identify which are the most stretched cables for all the forces applied.
4. Place the sensors on those cables.
5. Multiple sensors can be required to differentiate forces in more complex cases.

In the considered case the most stretched cables lie on the intersection between the module's surface and a plane perpendicular to the direction of the actuation **Figure 4-8, Figure 4-9**. At the intersection there are six cables and among these it would have been enough to select one and place there the sensor.

To identify them, a static simulator for tensegrity structures has been adopted, consequently the deformations on these six cables do not present any kind of difference among them due to real effects.

If a similar experiment would have been carried out on a real prototype, it would have been necessary to define a metric to select the most stretched cables, because it would have been very unlikely that all the cables would have had exactly the same length.

However, in capacitive strain sensors, variation of capacitance and deformation are linear dependent, hence increasing the deformation, the total variation of the capacitance is incremented and can be easily detected

Figure 4-10: $\frac{\Delta C}{C_0} = \frac{\Delta l}{l_0} \times 6$

Indeed, on all the six cables a sensor has been placed and in this way the total length is six times bigger.

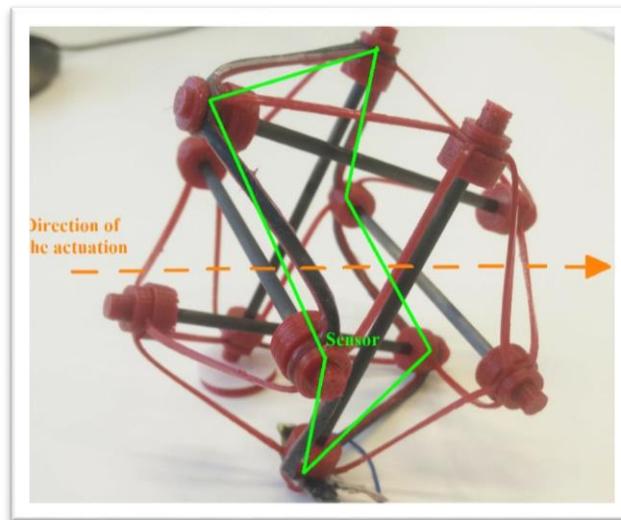


Figure 4-8: Single module picture and sensor positioning

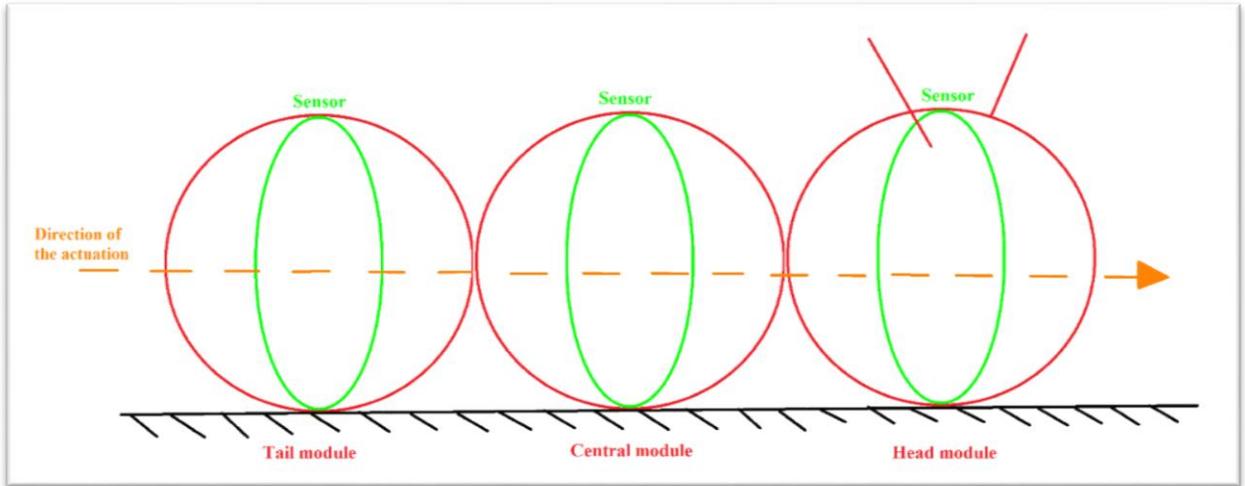


Figure 4-9: Sensors positioning in a stylized three modules robot.

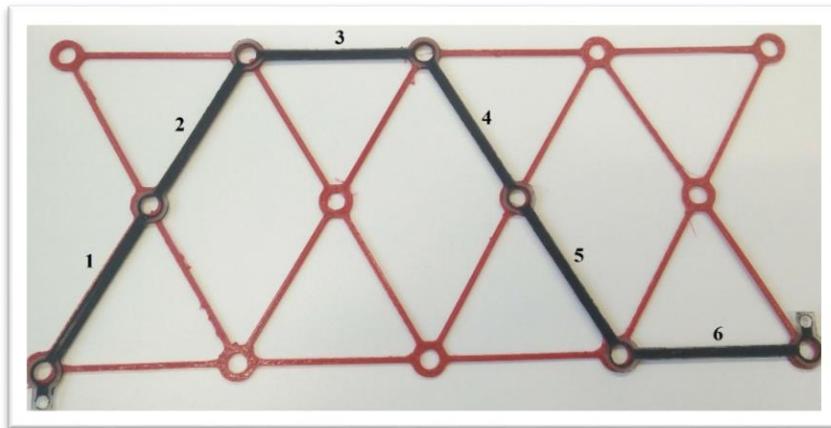


Figure 4-10: Tensegrity net and sensor in black and the six cables that compose it.

5 Hardware components realization

The next chapter is focused on the hardware needed for the validation of the concepts, the problem encountered during the manufacturing process and their solution.

5.1 Capacitive strain sensors

The diffusion of soft robotics has led to the development of new technologies to detect large deformations and able to follow the kinematics and the dynamics of the system on which they are mounted.

Researchers have developed highly stretchable sensors made of compliant elastomers and conductive materials, however those technologies cannot be manufactured in large scale and low cost, while carbon black filled elastomer sensors meet these criteria [23].

The sensors consist of a carbon black filled elastomer and are manufactured using film casting techniques and CO₂ laser ablation, this process speed up the multi-layered manufacturing of both, capacitive and resistive elements.

Capacitive sensors working principle is based on the change of the capacitance of a flat plates capacitor, due to the reduction of the thickness of the dielectric caused by mechanical strain. While, resistive sensor exploits the piezo resistive effect and geometrical changes of electrodes, where mechanical strain causes a change in the electric resistivity [23].

The capacitive sensor has showed superior performances compared to the resistive sensor, except the gauge factors. In fact, it has demonstrated high linearity and low hysteresis under different cycle strain amplitudes (50 % to 500 %), high repeatability and durability up to 10100 cycles and good temperature tolerance from 25 to 80 °C. On the other hand, the resistive type sensor exhibited a higher, but variable gauge factor [23].

Due to its higher performances, the capacitive sensor has been chosen for the implementation.

5.1.1 Manufacturing process

The manufacturing process consist of several steps **Figure 5-1** and it involves two machines: one the Zehnter ZAA2300, used to cast silicone layers and the other one is the Laser cutter, used to engrave and cut the sensors.

1. **First silicone layer casting:** the Zehnter machine as well as the applicator coater is cleaned using iso-propanol to remove dust and silicone residue, then a PET film is positioned on it.

The liquid silicone mixture of Sylgard 184 is prepared by mixing the two components at 2000 [rpm] for 1 minute.

The silicone mixture is casted on the PET film with the target thickness at 5 [mm/s] and then it is cured at 80 [°C] for 30 minutes in the oven.

2. **First electrode layer casting:** carbon KET-300 (1 g), trimethylpentane (32 g), silicone Sylgard 184 (10 g) and 3 metal balls are mixed together for 10 minutes at 2000 [rpm].

The liquid mixture is casted with the target thickness directly on the silicone layer produced in the previous step and then it is cured for 60 minutes at 80 [°C].

3. **Laser engraving 1th electrode:** after having designed, using a CAD, the first armature of the capacitor it is engraved with the Laser cutter.

The process consists in the removal by laser ablation of the conductive silicone from all the surface, but the one of the electrode.

4. **Second silicon layer casting, dielectric:** conductive residue from laser ablation are carefully removed from the upper layer using iso-propanol. Then silicon casting and curing, as in step 1, is repeated.

5. **Second electrode layer casting:** a new electrode is casted on the previous step silicone layer as in step 2.

6. **Laser engraving 2nd electrode:** an electrode, different from the previous one, is laser ablated as in step 3.

7. **Last silicone layer casting:** as in step 1 and 4, the last silicone layer is casted and cured.

8. **Sensor realization:** using laser cutter the sensors are separated from the silicon sheet and the connection with the armature are exposed **Figure 5-2**.

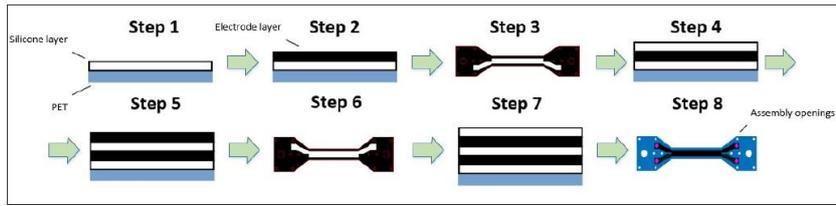


Figure 5-1: Sensor's fabrication steps [23].

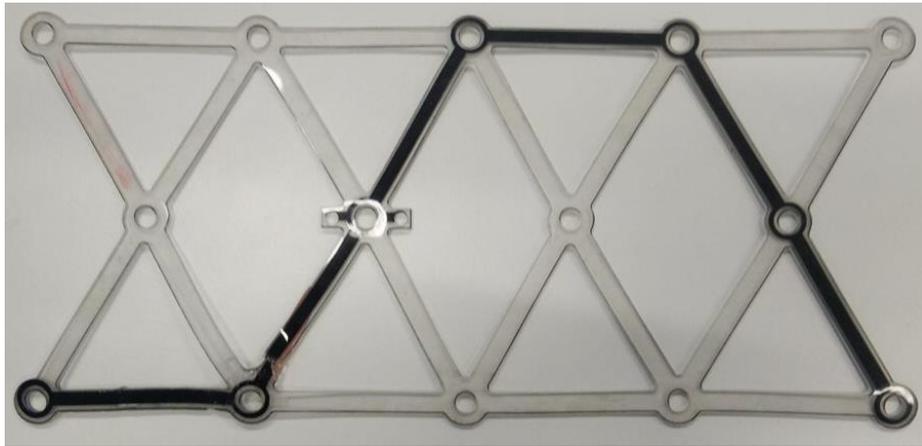


Figure 5-2: Net with embedded sensor.

5.1.2 Electrodes connection

Wires are added to the electrodes in order to connect them with other electronics.

After several trials, the most effective way to do it is: wrapping the electric wires to the connection holes, closing these ones only from one side with sellotape, without removing PET film and then, pour the conductive silicone used for the electrodes in the holes **Figure 5-3**.

After having waited for 60 minutes that the silicon is cured into the oven, it is possible to apply silicone glue on the silicone and on the wires to strengthen the connections.

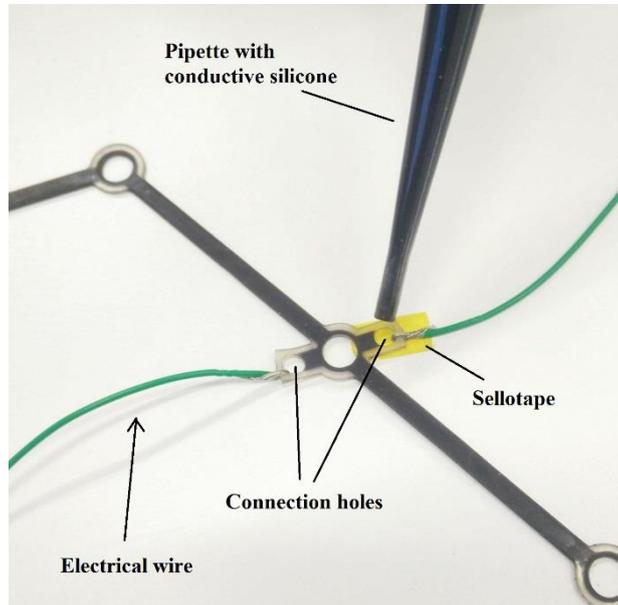


Figure 5-3: Electrodes connection

5.2 Design of supports for tensegrity's struts

The new sensor technology has required a new design to be connected to the struts. Sensor net can be considered flat hence, sockets to connect the struts and the sensor net cannot be embedded in latter **Figure 5-4**.

Indeed, two new components, cap and collar, **Figure 5-5**. have been designed and printed in NinjaFlex using Lulzbot.

Furthermore, this new design has improved the resistance of the net, compared to the previous NinjaFlex one [10], solving a systematic weakness **Figure 5-6**. in correspondence of the strut-net connections.

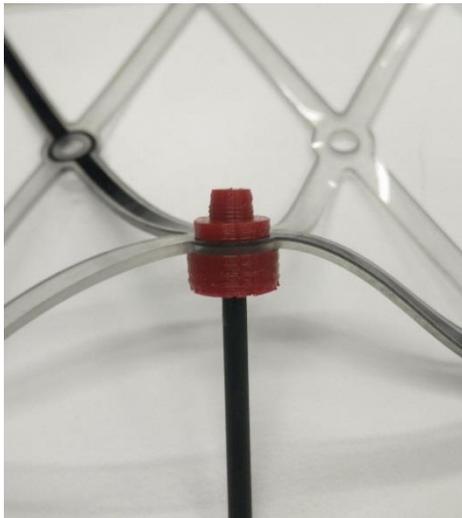


Figure 5-4: Flat net and new support system.

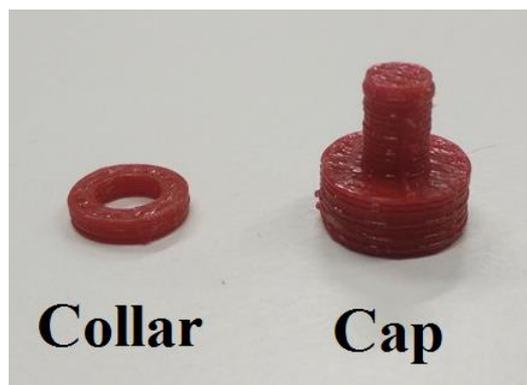


Figure 5-5: Cap and collar printed in NinjaFlex.

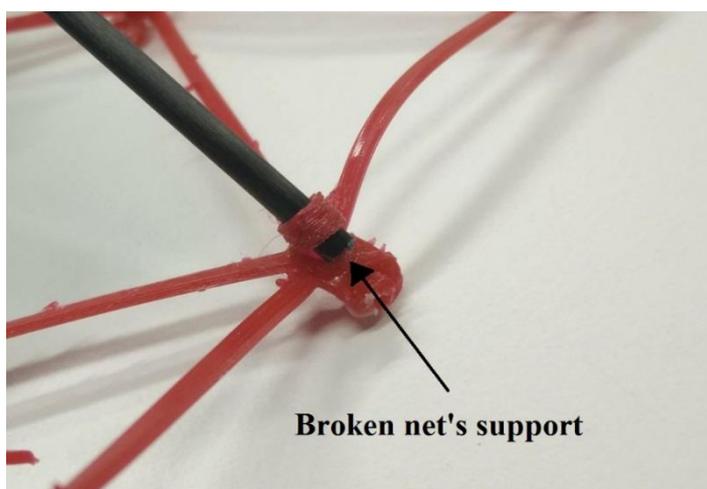


Figure 5-6: Old net and broken support system

5.3 Sensor integration

5.3.1 Sylgard silicone

The silicon initially used to fabricate the ultra-stretchable strain sensors is the EcoFlex 00-30. It has a low Young's modulus (125 [kPa]) while Sylgard 184 has a Young's modulus $\cong 2$ [MPa], to increase stiffness the latter has been used to cast the tensegrity net.

The characterization of the stretchable sensors has been performed only with EcoFlex 00-30 and the cyclic stress test has been performed only with it [23]. Unfortunately, after exactly two cycles the tensegrity net made in Sylgard 184 broke **Figure 5-7**.

The broken point was far enough from the corners to exclude strain intensification, anyway corners have been properly fitted to avoid it till the beginning.



Figure 5-7: Sylgard-184 broken net.

5.3.2 Solution 1: sensor made in Ecoflex 00-30

The first solution tested is the realization of the tensegrity net in EcoFlex 00-30, the original silicon with which characterization has taken place.

The tensegrity structure without the motor weights 19 [g], while the motor itself weights 15 [g]. When EcoFlex 00-30 has been used for the net of the tensegrity, the structure has collapsed under its weight without the motor on board **Figure 5-8**.

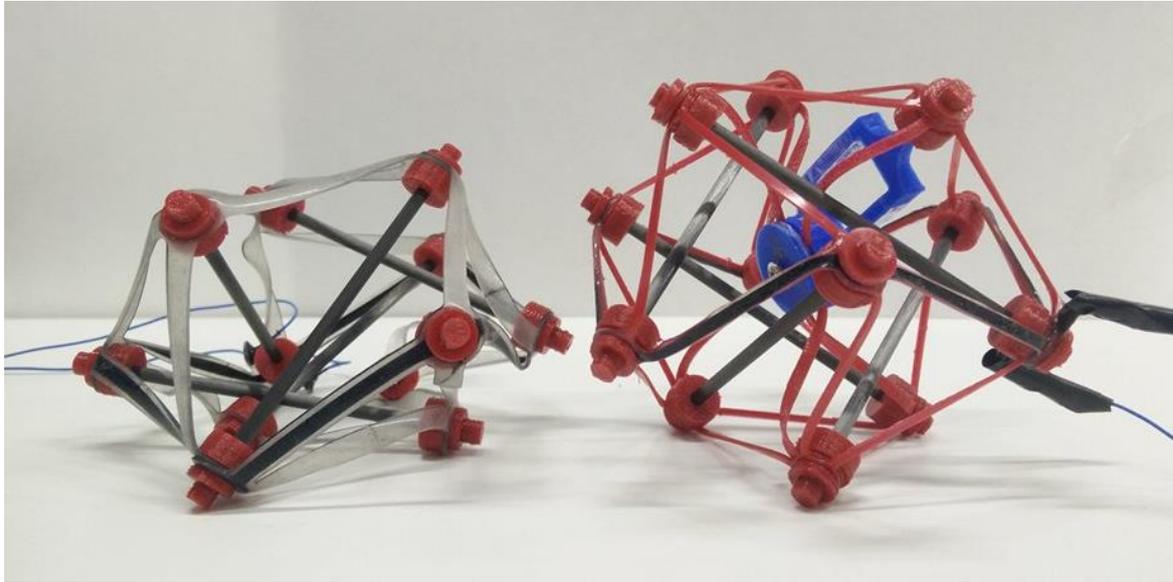


Figure 5-8: Comparison between a tensegrity made in EcoFlex 00-30, on the left and one made in NinjaFlex, on the right.

5.3.3 Solution 2: NinjaFlex net + sensor made in EcoFlex 00-30

Using SolidWorks, a net 1 [mm] thick has been designed and printed in NinjaFlex using LolzBot, then an ultra-stretchable capacitive sensor made in EcoFlex 00-30 was manufactured with a new design and it was glued on top of the NinjaFlex net using silicone glue **Figure 5-9**.

The new design consisted in having only the sensor, without the rest of the net and in this way the alignment of the printed net and the sensor, when gluing them together, has been simplified.

Combining the mechanical properties of NinjaFlex and EcoFlex 00-30, a much stiffer module with embedded sensor is available. On the other hand, NinjaFlex has hysteresis, hence the sensor itself is affected by it too, hence a characterization is needed to check whether it is still a viable solution.

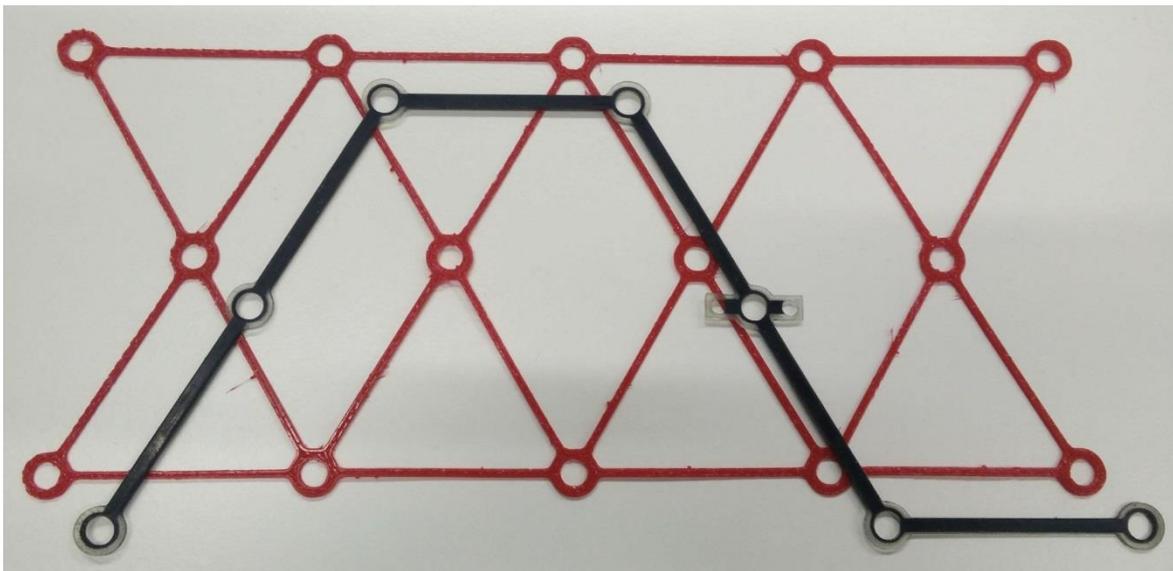


Figure 5-9: Tensegrity net in NinjaFlex, in red and EcoFlex 00-30 sensor, in black, before being glued together.

5.4 Capacitive strain sensor test

5.4.1 LRC meter results

To check whether the capacitive sensor was working or not the LRC meter (HIOKI IM3523) has been used. Five consecutive measurements of the capacitance have been taken actuating the sensorized tensegrity using a servo motor. Differences in the five measurements of the capacitance have been observed only in decimals figures, the results are provided in the table below.

Servo motor angle [°]	Sensor's Capacitance [pF]
-----------------------	---------------------------

0	149
10	149
20	150
30	151
40	152
50	153
60	154
70	155
80	156
90	157
100	158
110	159

Even if the servo motor can reach 180°, after 110° no change in capacitance has been detected.

To produce a characterization to hysteresis of the capacitive sensor is necessary to synchronize the motor movements and the log of capacitance values, in order to perform it a portable component able to measure capacitance, connectable to the microcontroller is needed.

The Freescale MPR121, which is a proximity capacitive touch sensor controller, can be used for general purpose capacitive detections, it has 12 inputs, an I2C interface, a resolution of 0.1[pF] and an operative range between 0.45-2880 [pF].

Considering the measurements of the sensor taken with LRC meter and the nominal capacitance around 150[pF] the FreeScale component seems to fit the needs.

A semiconductor company, Adafruit, embeds this component in a ready to use board, that can be easily ordered online. When tested the board has not worked as expected. In fact,

the variations in the capacitance, across the whole servo motor interval 0-110°, have not been detected.

According to the technical support of the company, the component purchased was not the proper one to detect capacitance variations in stretchable strain sensors.

The **Figure 5-10** shows how capacitance, measured by LRC meter, varies when the module is deformed from 0 to 110°. Using the AdaFruit board a similar plot was expected, while as one can observe in **Figure 5-11** the behavior of the variation of the capacitance is completely different. In fact, the values measured by the board, which are proportional to the real capacitance, oscillate around the value 218 without any correspondence to the contraction and the relaxation of the module.

The fastest solution to cope with this hardware issue is the implementation of resistive sensors. They do not require any other external equipment; hence the variation of the resistance can be directly measured using a microcontroller.

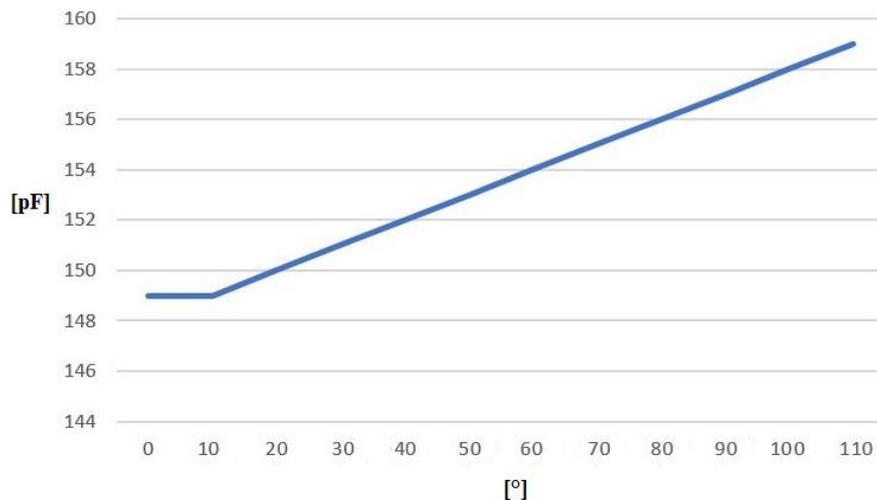


Figure 5-10: Capacitance measurements taken with LRC meter.

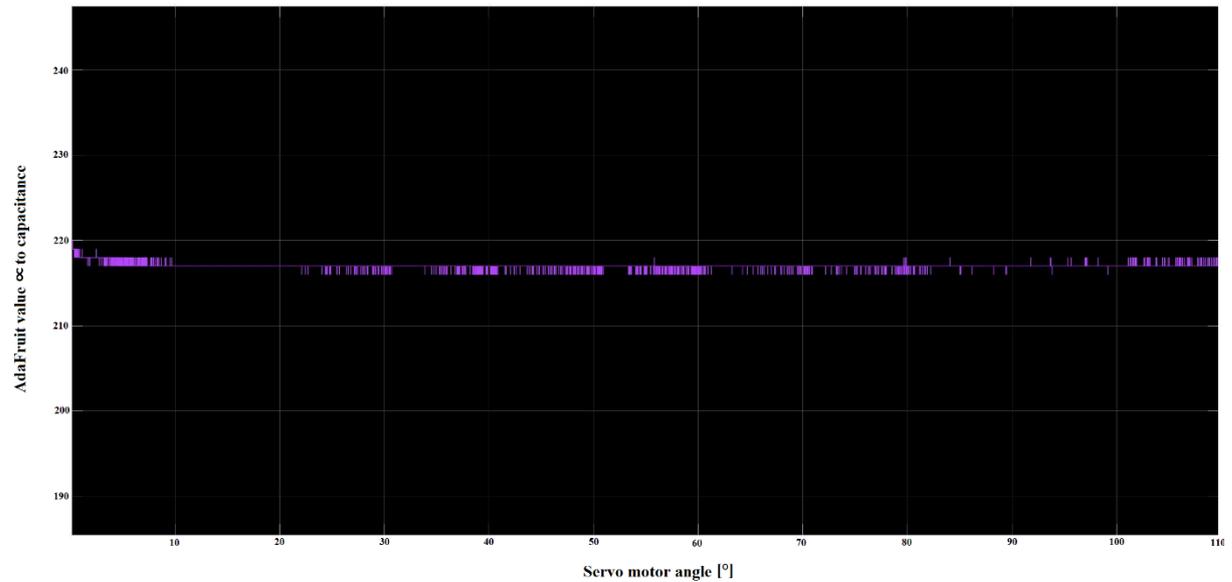


Figure 5-11: Capacitance recorded using AdaFruit board. Between two vertical lines there are 10[°], starting from 0 till 110.

5.4.2 Resistive sensor characterization: hardware and results

Ninjaflex is affected by hysteresis, below it is described the hardware used to perform the characterization.

5.4.2.1 Voltage divider

The easiest way to measure a resistive-stretchable strain sensor is through Arduino and a voltage divider **Figure 5-12**. The nominal resistance, when no stress is applied, has been measured with a LRC meter (HIOKI IM3523) and it is around 2 [MΩ]. In order to obtain a finite gain, a known resistance of 2,2 [MΩ] has been selected.

Using MatLab and a servo motor to deform the structure (HK939MG), a script was able to log and plot over time the value of the resistance.

In the voltage divider represented, V1 is the analog input to Arduino, R1 is the sensor and its value is computed as:

$$R1 = \left(\frac{5}{V1} - 1 \right) \cdot 2.2 [M\Omega]$$

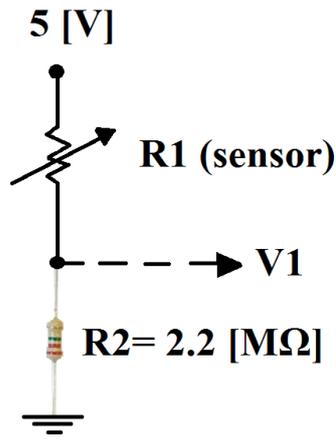


Figure 5-12: Voltage divider.

5.4.2.2 First order filter design

To improve the quality of the signal a simple first order passive filter **Figure 5-13** has been implemented, using as f_c the maximum frequency declared on the servo motor's data sheet:

$$f_c = \frac{1}{2RC\pi} = 1 \text{ [Hz]}$$

$$RC = \frac{1}{2\pi \cdot 1 \text{ [Hz]}} = 0.159$$

Hence: $C = 1 \text{ [}\mu\text{F]}$, $R = 150 \text{ [k}\Omega\text{]}$ and $\tau = 0.159 \text{ [s]}$.

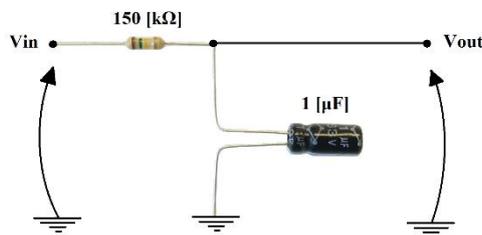


Figure 5-13: Voltage divider.

The result is shown below, **Figure 5-14**.

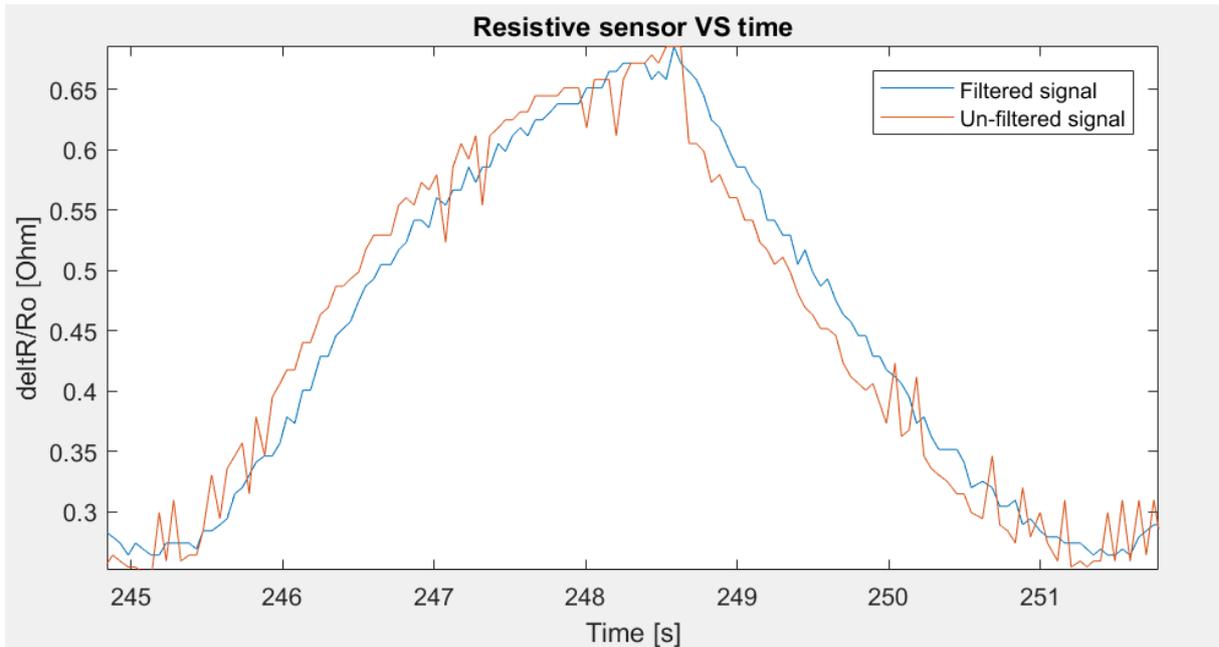


Figure 5-14: Filtered signal, in blue, compared with the unfiltered one, in red.

5.4.2.3 Modified servo motor for hysteresis analysis

In order to link resistance values with the position of the servo motor, the latter has been modified to extract the potentiometer's analog signal. Which is used in the servo motor's controller as angular position feedback.

Hence, the servo motor has been dismantled, an electric wire has been soldered to the output pin of the potentiometer **Figure 5-15a** and then, thanks to a hole accurately drilled in the servo cover, the wire has been pulled out and connected to Arduino board for position log **Figure 5-15b**.

Using a MatLab script both, sensor's resistance and servo's position, have been logged and plotted to characterize the sensor to hysteresis.

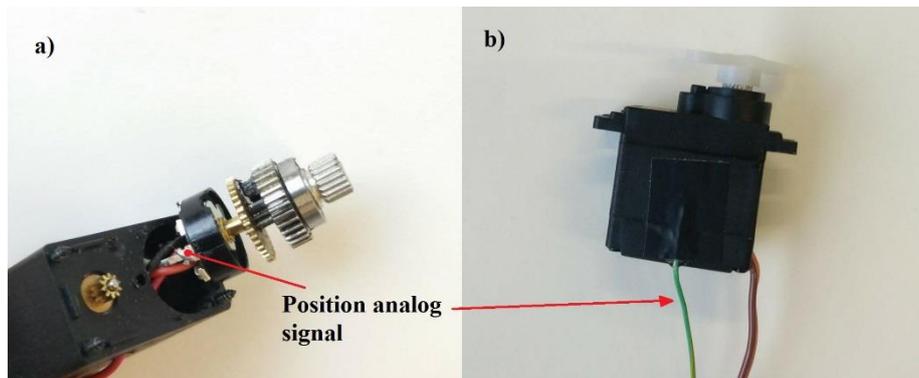


Figure 5-1: a) Potentiometer analog signal b) Modified servo ready to use.

5.4.2.4 Resistive sensor characterization: results

Data have been collected for more than 1000 cycles using a servo motor to deform the module. Every 100 cycles the module has been left to recover for 10 minutes. The characterization is shown through four different graphs: R vs Time, $\Delta R/R_0$ vs time, $\Delta R/R_0$ vs ServoAngle and $\Delta R/R_0$ vs compression percentage.

The compression percentage has been computed considering a linear relationship between the servo angle and the compression % of the module, it spans from 0%, when the servo motor is at 0° , to 23%, when the servo motor is a 130° .

The resistance is stable **Figure 5-16**, the range of variation of $\Delta R/R_0$ is included between 0.7 and 0.3. Considering the two pictures **Figure 5-16**, **Figure 5-17**, a slightly decreasing trend can be observed. To cope with this problem an automatic calibration procedure repeated for few cycles can be designed.

Moreover, sudden variations are evident in the resistance value and this is due to the vibrations produced by the motor **Figure 5-17**. In fact, phenomenon similar to slipping between motor's shaft and gears has been observed. Anyway, this proves that the sensor worked correctly and that it was able to detect fast variations in the deformation.

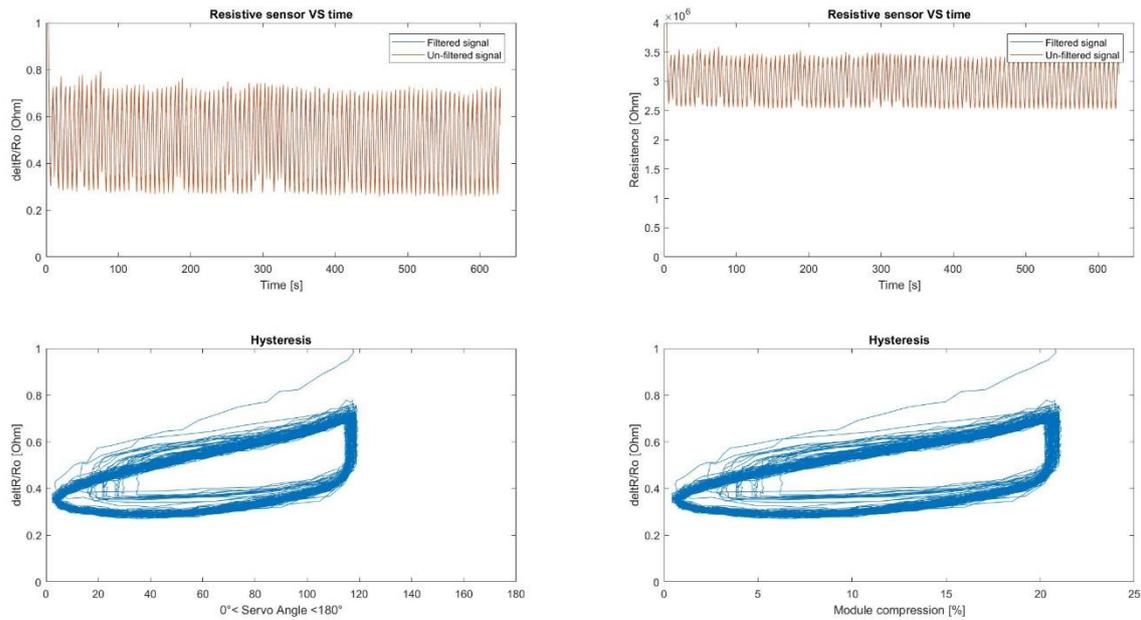


Figure 5-2: In the picture are shown the resistance's variation and the hysteresis under 100 cycles, in particular from cycle 200 to cycle 300. Despite some vibrations induced by the motor, the signal is quite stable.

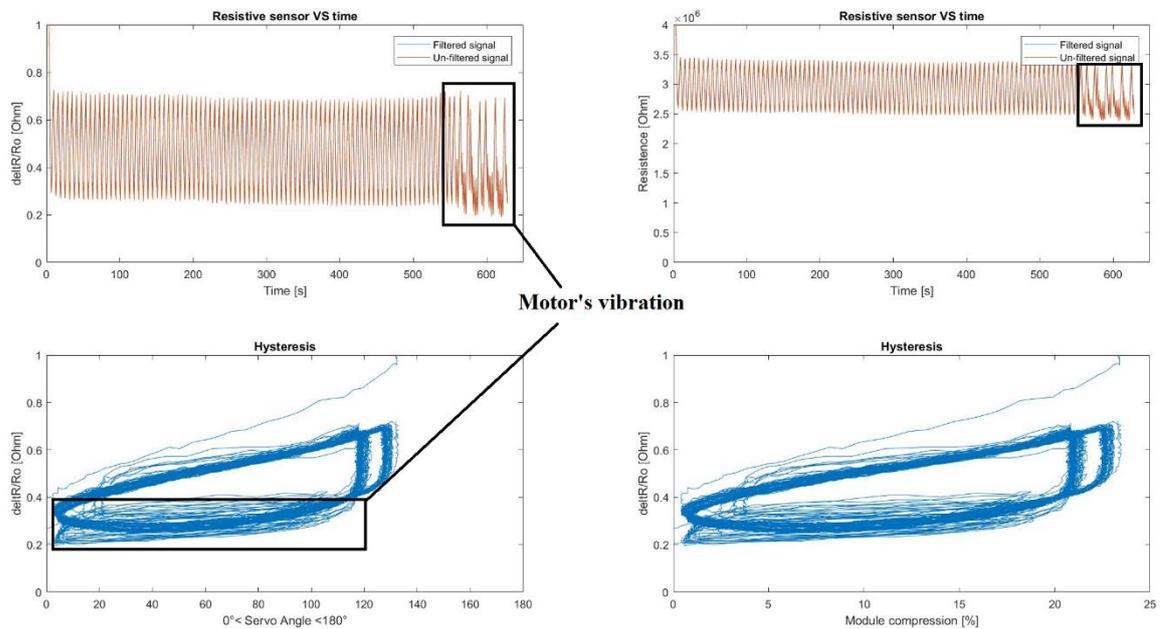


Figure 5-3: In the picture are shown the resistance's variation and the hysteresis under 100 cycles, in particular from cycle 800 to cycle 900. In the final part of the plot the vibrations of the motor are tracked by the sensor.

6 Proposed controller

The controllers reviewed for modular robots are very simple architectures, namely gait table, hormone controller, sinusoidal controller and their common issue is synchronization among modules. To solve it closed-loop structure have been implemented, as in the case of the hormone.

For what concern tensegrities, their properties have non-negligible drawbacks in terms of modelling non-linear dynamics and control. Non-traditional controllers have achieved successful results in controlling them. Moreover, it has been shown, in simulation and hardware validation, that the introduction of sensory feedback has improved the performance of the robot in case of external forces and unexpected terrains.

Among the controllers shown, CPG has been implemented successfully in both, tensegrity and modular robotics, but the first that has been explored is a simple neural network.

6.1 Artificial Neural Network

Artificial Neural Networks can be considered a class of computational models able to represent non-linear input/output mappings. A characteristic feature of artificial neural networks is that learning occurs by training with examples rather than explicit programming, hence they result very attractive for controlling robots. For instance, some parameters of a robot cannot be determined precisely, as inertia matrix and friction, resorting on analytical methods, therefore obtaining these relations measuring input/output data result very appealing [41].

Neural networks are parallel computational structures inspired by neuroscience. They consist of very large number of simple processing unit called neurons, connected to a multitude of other neurons. Neurons themselves are very simple unit, but the power of neural networks relies on the interconnections among them. The learning procedure modify the strength of these interconnections during the training phase. The result is that the desired mapping is encoded in these interconnections and give them immense processing capabilities [41].

ANN has three different kind of layers, the input layer, the output layer and between these two there are the hidden layers. The input neurons receive signals from the environment,

these are processed and propagated through the network and then the output is produced by the output neurons.

The artificial neuron is characterized by synapses, that connect it to other neurons, by an activation level and by an activation function [42]. The neurons receive inputs, either from external environment or from other neurons, the inputs are multiplied by the weights of the synapses, a bias weight is added and finally an activation function is applied to obtain the output of the neuron [41].

When the controller has to deal with nonlinear robot kinematics, dynamics, complex interactions with the environment and uncertainty in available information, the application of conventional controlling methods is difficult, if not impossible. Hence, the learning capabilities of neural networks seem ideal in such situations, since they can learn with a minimum amount of a priori knowledge [41].

One of the most used ANN architecture in robotics is the feedforward, which is characterized by unidirectional connection between neurons [41].

Learning means the minimization of the error function obtained between the desired output and the real one. The most popular algorithm to update weights is the backpropagation based on the gradient descent methods with chain rules.

Backpropagation needs the knowledge of the desired output; indeed, it cannot be applied to our case. The learning phase in this research is based on the definition of a fitness function, which make possible to score the performance of the controllers.

The fitness function considered is the maximum distance covered by the robot in a given time. The controller that will be actually implemented is the one that have travelled for the longest distance in the given time.

The definition of the network and its weights is a crucial problem and it is not known a priori. The network that has been considered at the beginning is very simple, it is made of three layers: an input layer, a hidden layer and an output layer **Figure 6-1**.

The number of neurons in the input layer is constrained by the number of inputs in the system, in this case one sensor per module, considering a three-modules robot means three neurons in the input layer. Similarly, the number of neurons in the output layer is con-

strained by the number of actuators that have to be controlled, in this case one per module, hence three neurons. While, the neurons in the hidden layer have been arbitrary set to three, as the number of inputs.

To find the controller with the best performances it is necessary to have a population among which the best one can be selected and to solve this optimization problem a genetic algorithm based on elitism can be used.

Before the implementation of the GA and the NN the process has been aborted due to the lack of a simulator, in fact the GA requires a lot of iterations to find the best controller and performing them in real world without simulators is very time consuming.

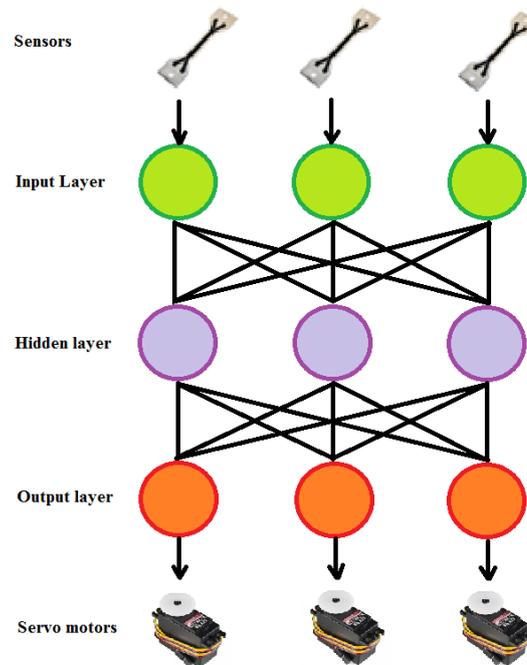


Figure 6-1: Neural Network.

6.2 Central Pattern Generator

The non-availability of a simulator for tensegrity structures and the successful results obtained with CPG in controlling modular and tensegrity robots have made them very appealing.

The CPG is a particular case of neural controller able to establish a rhythmic output pattern, reflecting on the simple NN described before it would have been difficult to induce a rhythmic behavior with one hidden layer made of three neurons.

The CPG that have been used for tensegrity robotics [23] or to mimic animals' gaits [43] are based on oscillators obtained using coupled differential equations, in this research it has been looked for the easiest oscillator available in literature.

In particular, the attention has been focused on the CPG proposed by Owaki et al. [44] for quadruped locomotion. It is a very simple CPG based on one equation that exploit sinusoidal function instead of coupled differential equations **Figure 6-2**:

$$\dot{\phi}_i = \omega - \sigma N_i \cos \phi_i,$$

Figure 6-2: CPG Owaki et al.

Where omega is the natural frequency of the system, N_i is the feedback coming from the pressure sensor on board of the i -th leg, sigma is a weight used to reshape the magnitude of the second term and F_i is the phase of the i -th leg.

The motors are controlled along x and y direction with the two equations below **Figure 6-3**:

$$\begin{aligned} X_i &= B \cos \phi_i \\ Y_i &= A \sin \phi_i \end{aligned}$$

Figure 6-3: Leg control.

Where A and B are positive constants used to tune the magnitude of the movement the i -th leg.

As one can notice this CPG presents a local feedback from a pressure sensor on the i -th leg and this means that sensors on other legs cannot influence the i -th one. In fact, this re-

search is based on demonstrating how coordination should rely on physical interactions through body dynamics rather than explicit interlimb neural connections.

In the case of the tensegrity modular robot it is difficult to forecast an interaction between modules so intense to not require any explicit intermodular neural connections and this is the reason why the neural connection has been explicitly considered.

The neural coupling considered comes from bio-robotics and it has been used for a CPG that reproduces the salamander's neural system [43], it consists of a constant c_{ij} that weights the sensory feedback s_j from the j -th module.

Finally, the CPG designed is the following:

$$\begin{cases} \dot{\varphi}_i = \omega - \sigma N_i \cos(\varphi_i) - \sum_j c_{ij} s_j & 1) \\ X_i = A \cos(\varphi_i) & 2) \end{cases}$$

$\dot{\varphi}_i$ Equation of the i th oscillator

ω Natural frequency.

σ Weight term for the sensor on the i th module.

N_i Sensor signal on the i th module

c_{ij} Weight term for the sensor on the j th module

s_j Sensor signal on the j th module

X_i Control signal for the i th motor

A Weight term to modulate the amplitude of X_i

After having defined the coupling in the formula, it is necessary to define also the neighborhood rules on which it is applied. In case of a modular robot it is very important to define a neighborhood that can be easily applied when the number of modules scales up or down. The first considered is very simple and related to how actually the robot is built. In fact, taking into account three modules: head, central module and tail, only adjacent module can influence each other, as represented below **Figure 6-4**.

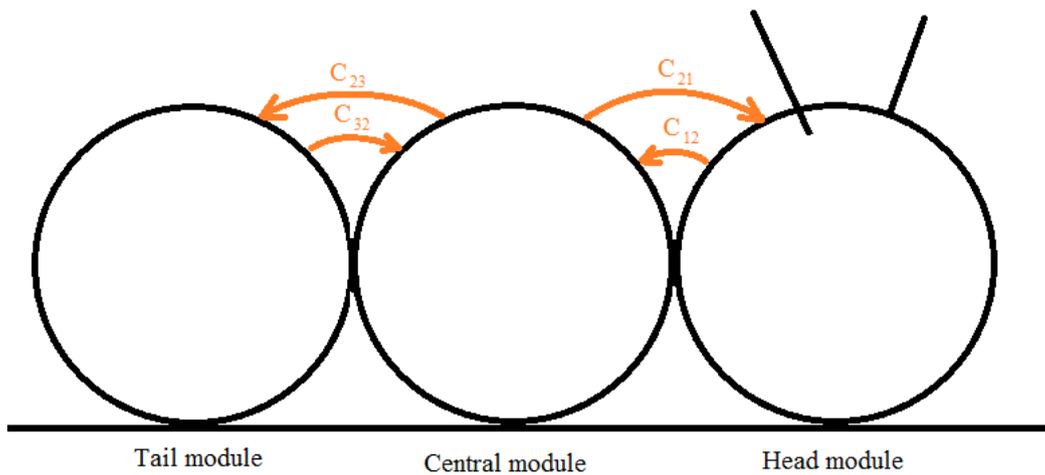


Figure 6-4: Interconnections scheme 1.

Using the connections scheme above the parameters of the controller that have to be tuned are seven: ω , σ , A , C_{12} , C_{21} , C_{23} and C_{32} .

To speed up the training phase, it is possible to reduce the parameters that depend on the configuration changing it. The example proposed below **Figure 6-5** is a configuration that is no more linked with the physical positions of the modules in the robot.

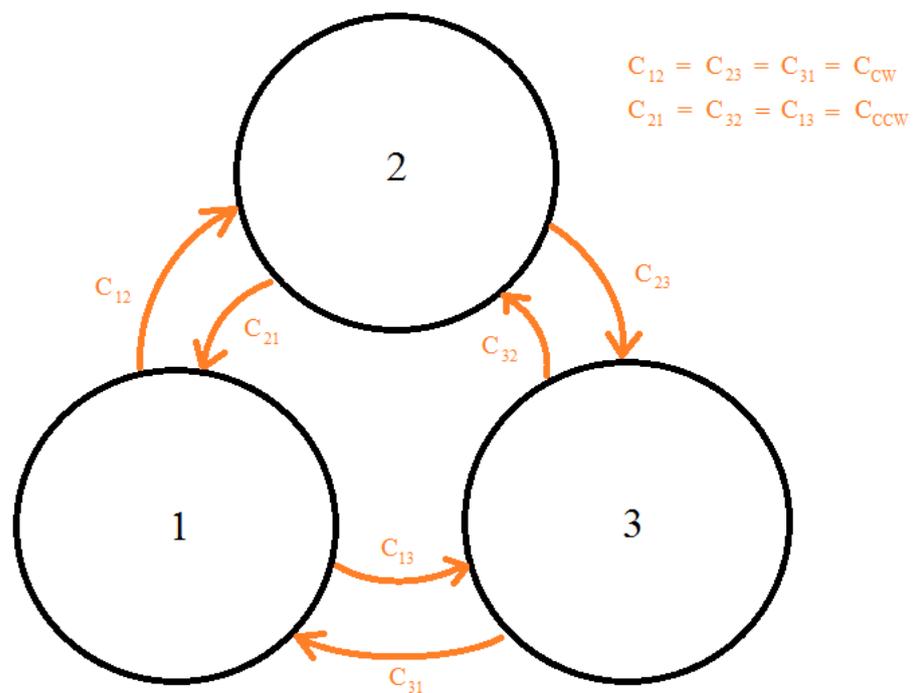


Figure 6-5: Interconnections scheme 2.

In the scheme above all the clockwise connections have the same weight and all the counterclockwise connections have the same weight, hence the number of parameters to tune is reduced to five: ω , σ , A , C_{cw} and C_{ccw} .

7 Controller implementation and tuning

The controller has been implemented using Simulink. The visualization of code elements in case of a modular robot, with a theoretically infinite number of parts, can be exploited to avoid errors and to debug it easily.

Equations 1) and 2) have been replicated in a Simulink's subsystem, respectively in the red and in the blue rectangles. The subsystem represents one module **Figure 7-1** and it communicates with the environment sharing with other modules input and output signals, as the natural frequency, sigma and the sensors info. Both architectures, the worm-like **Figure 7-2** and the circular-like **Figure 7-3** has been realized.

For what concerns equation 2), the cosine is used to limit in the interval $-1/1$ the value the angle φ . Instead of using exactly the same equation, in the Simulink model acos function has been applied to $\cos(\varphi)$ to obtain a value between 0 and 180, which is exactly the interval required by the servo block in the model to control the motor.

The *convert* block has been used to convert to the correct data type the signal entering and exiting the trigonometric functions, while the gains before *cos* and after *acos* are used to convert from radians to degrees and vice versa.

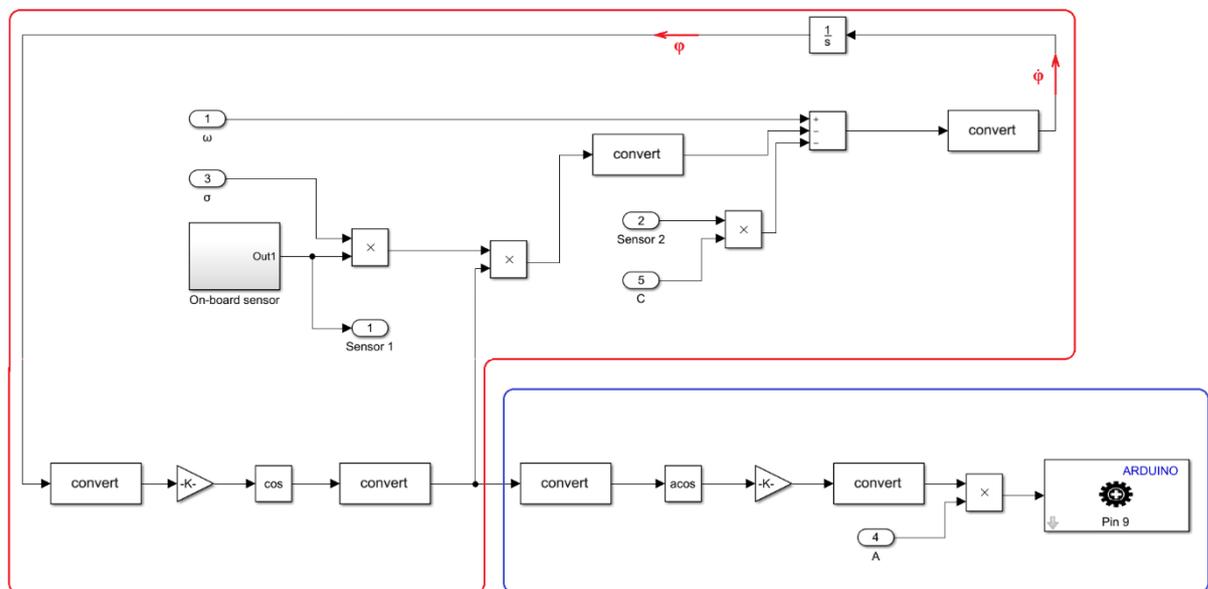


Figure 7-1: Module representation on Simulink.

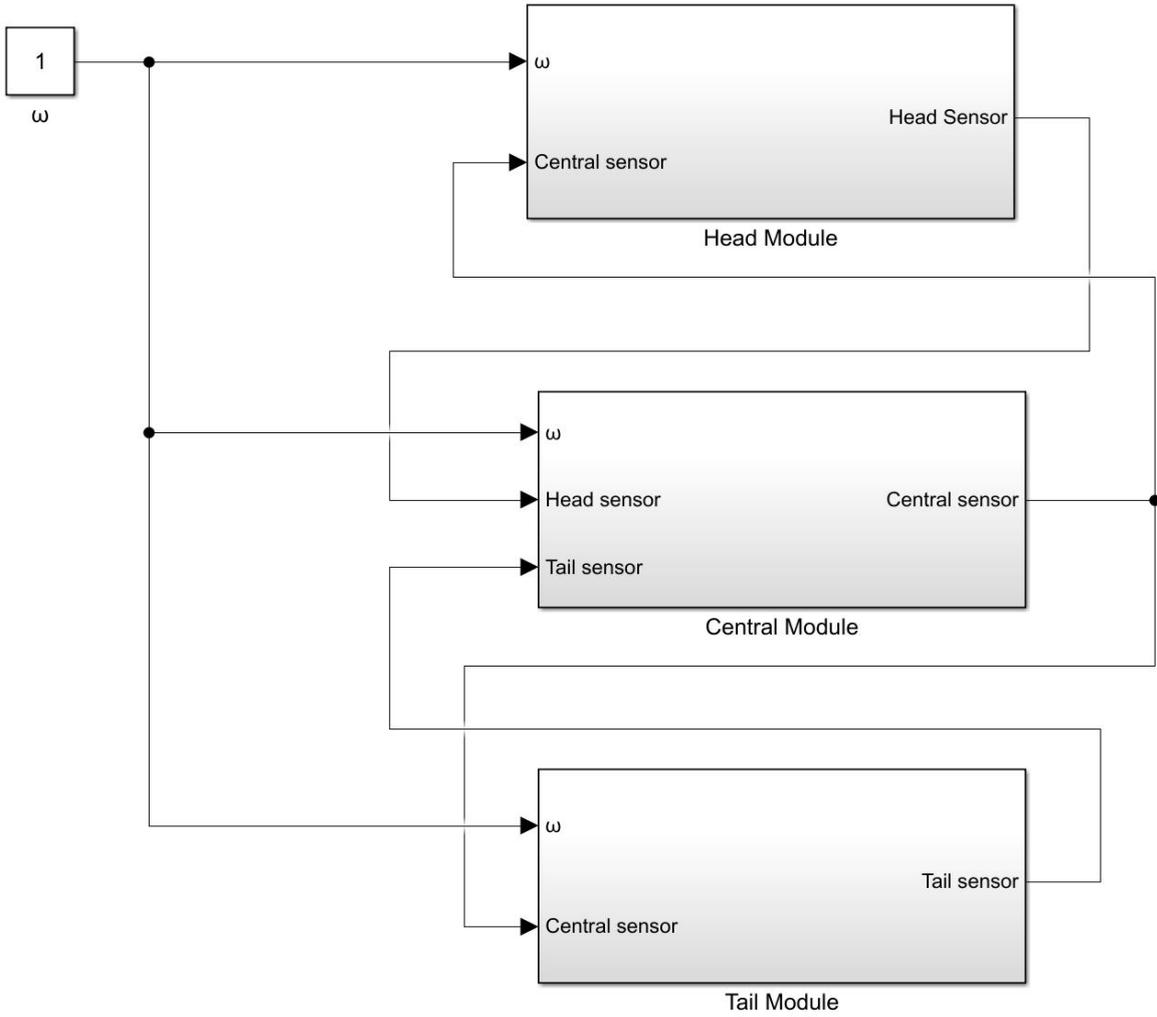
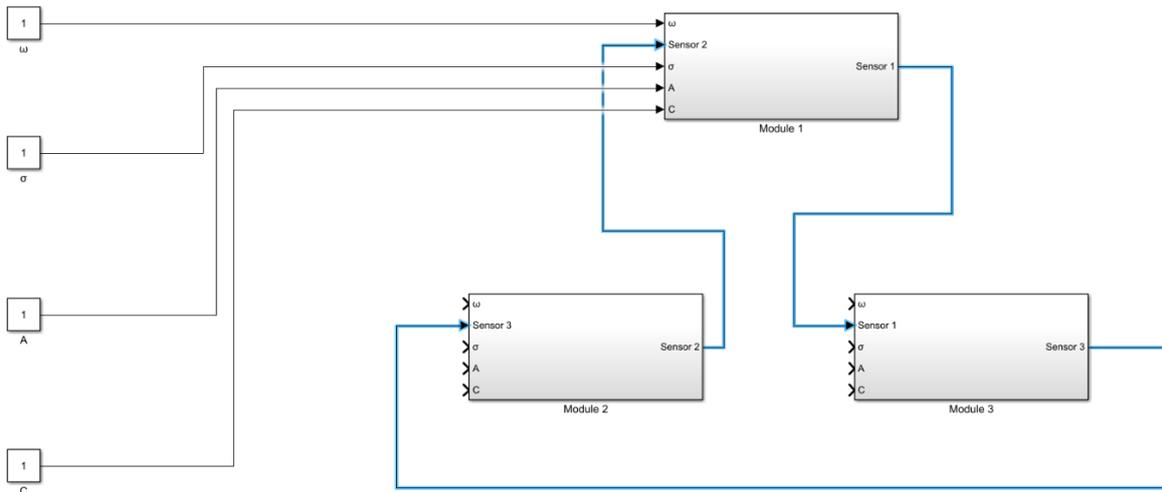


Figure 7-2: Interconnections scheme 1 Figure.



7-3: Interconnections scheme 2 implemented in Simulink.

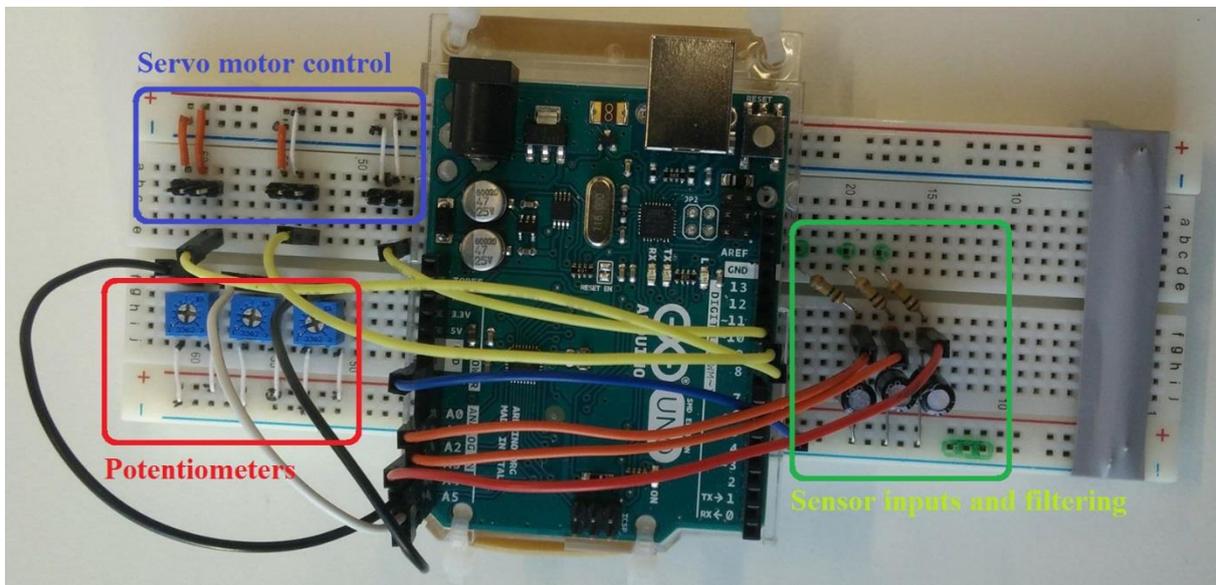
The parameters ω , σ , A and C_{ij} have to be tuned to produce a rhythmic movement and it is not something trivial to do in hardware.

Hence, not resorting on a proper system to evaluate in hardware the fitness function, but with the willing to test the controller in hardware, a very simple circuit has been built.

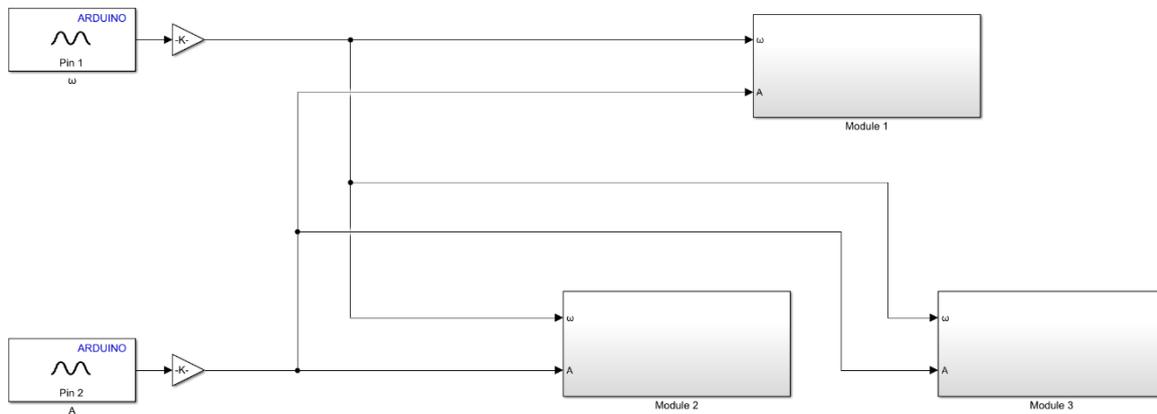
It consists of three potentiometers able to span 1024 values **Figure 7-4**, with a resolution decided by the designer. The potentiometers have been used to tune the parameters mentioned above, in the hope that the correct value would have been in the range $0 \div 5$, a priori fixed by the designer. The fitness function would have been evaluated just watching the different behaviors of the robot in correspondence of different potentiometers' positions.

The natural frequency is the first parameters that has been investigated. The system in open loop has been considered, as represented in **Figure 7-5** and then, the natural frequency found would have been used into the closed loop controller. Unfortunately, spanning all the available interval no changes has been noticed in the locomotion of the robot and detecting the natural frequency has been impossible.

Anyway, an arbitrary natural frequency of $1[\text{rad/s}]$ has been defined to proceed with the tuning of the closed-loop controller.



7-4: Arduino and circuitry for training and control the robot.



7-5: Finding omega using two potentiometers, one to change omega and the other one to change the gain of the servo.

However, also in this case no evident different behaviors have been noticed.

Resorting to a systematic approach in the future can prove the effectiveness of both controllers.

8 Conclusion

This research project has addressed the design of a controller for a new bio-inspired modular robot made using tensegrity structures.

Tensegrity structures have a lot of interesting properties for robotics. Compliancy and foldability are just a couple of them, but on the other hand they are difficult to control, due to their non-linear dynamics.

The analysis of the state of art related to controllers for tensegrity and modular robots has led to the necessity of a non-linear controller and also to the integration of sensory feedback to improve the performance of the robot.

The controller chosen is a central pattern generator, it is a bio-inspired neural controller that particularly fits to the problem. In fact, even if it can produce a rhythmic output without necessarily receiving rhythmic inputs, it can be designed to receive data from sensors, which are in charge of shaping the output and improving coordination in movements. Furthermore, it is a decentralized controller, a very important feature for a modular robot because it ensures that the growth in scale of the robot is not proportional to the growth of the controller's complexity.

For what concerns sensors, a theoretical study has been carried out to define, first of all, the variable of interest that sensors have to track and then where they have to be placed. But before this, another theoretical step has been accomplished: the definition of the robot on which this concept should have been validated.

The consequence of this reasoning is a robot made of three modules, that can move in one direction, no steering admitted and the sensors placed on it have to detect the contraction of each module during motion. These decisions have been taken with the goal of validating the effectiveness of the sensors in measuring the shrinkage of the modules in the simplest way possible.

The conceptual part has been followed by a practical one due to unavailability of a simulator. The state of art of the sensor in modular and tensegrity robotics has been investigated and in the end a new technology has been chosen to be actually mounted on board of the module.

Ultra-stretchable strain sensor is a soft technology that have been embedded on the modules and have been tested for several cycles.

To conclude, the signal produced by the sensors has resulted being stable after more than 1000 cycles, being easily read by a commercial microcontroller as Arduino, moreover the signal processing chain is very simple and consists of a first order filter and it can be directly used as input to the controller described above.

Furthermore, a fast tuning procedure for the parameters of the controller has been designed. It is based on the evaluation of the fitness function watching the behavior of the robot in correspondence of different values of the parameters.

Unfortunately, the variations have not been detectable just watching the robot, hence a more structured tuning procedure is needed to validate the controller.

9 Future work

Resistive sensors performed well after thousands of cycles, even if technically speaking capacitive sensors have better specifications and a comparison between the two on board of the robot can lead to a final decision about which shall be integrated. Obviously, a portable component able to measure capacitance is necessary, otherwise capacitive sensors cannot be implemented on prototypes.

The hardware validation in closed-loop of the robot is an important step and a structured training phase for the controller is needed.

For further development of larger assemblies of modules, it will be difficult to train the system in hardware, hence an accurate simulator to perform the learning phase is required or an automatic evaluator for the fitness function has to be designed.

Hardware validation is always suggested in case of highly non-linearity in the dynamics of the robot. Therefore, proving the effectiveness of the controller in hardware for bigger assemblies will be crucial.

10 References

- [1] Fuller, R. B. (1983). *Synergetics: Explorations in the geometry of thinking*. New York u.a: Macmillan u.a.
- [2] Caluwaerts, K., Despraz, J., Iscen, A., Sabelhaus, A. P., Bruce, J., Schrauwen, B., & SunSpiral, V. (2014). Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface*, 11(98), 20140520-20140520. doi:10.1098/rsif.2014.0520
- [3] Myers, T. W. (2009). *Anatomy trains*. Edinburgh: Churchill Livingstone/Elsevier.
- [4] Sabelhaus, A. P., Bruce, J., Caluwaerts, K., Manovi, P., Firoozi, R. F., Dobi, S., ... SunSpiral, V. (2015). System design and locomotion of SUPERball, an untethered tensegrity robot. *2015 IEEE International Conference on Robotics and Automation (ICRA)*. doi:10.1109/icra.2015.7139590
- [5] Caluwaerts, K., Bruce, J., Friesen, J. M., & SunSpiral, V. (2016). State estimation for tensegrity robots. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. doi:10.1109/icra.2016.7487331
- [6] Iscen, A., Caluwaerts, K., Bruce, J., Agogino, A., SunSpiral, V., & Tumer, K. (2015). Learning Tensegrity Locomotion Using Open-Loop Control Signals and Coevolutionary Algorithms. *Artificial Life*, 21(2), 119-140. doi:10.1162/artl_a_00163
- [7] Chen, L., Kim, K., Tang, E., Li, K., House, R., Agogino, A. M., ... Jung, E. (2016). Soft Spherical Tensegrity Robot Design Using Rod-Centered Actuation and Control. *Volume 5A: 40th Mechanisms and Robotics Conference*. doi:10.1115/detc2016-60550
- [8] Kim, K., Agogino, A. K., Toghyan, A., Moon, D., Taneja, L., & Agogino, A. M. (2015). Robust learning of tensegrity robot control for locomotion through form-finding. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. doi:10.1109/iros.2015.7354204
- [9] Ingber, D. E. (1998). The Architecture of Life. *Scientific American*, 278(1), 48-57. doi:10.1038/scientificamerican0198-48

- [10] Zappetti, D., Mintchev, S., Shintake, J., & Floreano, D. (2017). Bio-inspired Tensegrity Soft Modular Robots. *Biomimetic and Biohybrid Systems*, 497-508. doi:10.1007/978-3-319-63537-8_42
- [11] Chennareddy, S. S., Agrawal, A., & Karuppiah, A. (2017). Modular Self-Reconfigurable Robotic Systems: A Survey on Hardware Architectures. *Journal of Robotics*, 2017, 1-19. doi:10.1155/2017/5013532
- [12] Brunete, A., Ranganath, A., Segovia, S., De Frutos, J. P., Hernando, M., & Gambao, E. (2017). Current trends in reconfigurable modular robots design. *International Journal of Advanced Robotic Systems*, 14(3), 172988141771045. doi:10.1177/1729881417710457
- [13] Rieffel, J. A., Valero-Cuevas, F. J., & Lipson, H. (2009). Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of The Royal Society Interface*, 7(45), 613-621. doi:10.1098/rsif.2009.0240
- [14] Mirats Tur, J. M., & Juan, S. H. (2009). Tensegrity frameworks: Dynamic analysis review and open problems. *Mechanism and Machine Theory*, 44(1), 1-18. doi:10.1016/j.mechmachtheory.2008.06.008
- [15] Paul, C., Valero-Cuevas, F., & Lipson, H. (2006). Design and control of tensegrity robots for locomotion. *IEEE Transactions on Robotics*, 22(5), 944-957. doi:10.1109/tro.2006.878980
- [16] Paul, C., Roberts, J., Lipson, H., & Valero Cuevas, F. (n.d.). Gait production in a tensegrity based robot. *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005*. doi:10.1109/.2005.1507415
- [17] Shibata, M., Saijyo, F., & Hirai, S. (2009). Crawling by body deformation of tensegrity structure robots. *2009 IEEE International Conference on Robotics and Automation*. doi:10.1109/robot.2009.5152752
- [18] Koizumi, Y., Shibata, M., & Hirai, S. (2012). Rolling tensegrity driven by pneumatic soft actuators. *2012 IEEE International Conference on Robotics and Automation*. doi:10.1109/icra.2012.6224834

- [19] Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4), 642-653. doi:10.1016/j.neunet.2008.03.014
- [20] Conradt, J., & Varshavskaya, P. (2003). Distributed central pattern generator control for a serpentine robot. In International conference on artificial neural networks.
- [21] Kamimura, A., Kurokawa, H., Toshida, E., Tomita, K., Murata, S., & Kokaji, S. (n.d.). Automatic locomotion pattern generation for modular robots. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. doi:10.1109/robot.2003.1241678
- [22] Brian T. Mirletz, Roger D. Quinn and Vytas SunSpiral, CPGs for Adaptive Control of Spine-like Tensegrity Structures
- [23] Shintake, J., Piskarev, E., Jeong, S. H., & Floreano, D. (2017). Ultrastretchable Strain Sensors Using Carbon Black-Filled Elastomer Composites and Comparison of Capacitive Versus Resistive Sensors. *Advanced Materials Technologies*, 3(3), 1700284. doi:10.1002/admt.201700284
- [24] Hustig-Schultz, D., SunSpiral, V., & Teodorescu, M. (2016). Morphological design for controlled tensegrity quadruped locomotion. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. doi:10.1109/iros.2016.7759693
- [25] Rhode-Barbarigos, L., Schulin, C., Ali, N. B., Motro, R., & Smith, I. F. (2012). Mechanism-Based Approach for the Deployment of a Tensegrity-Ring Module. *Journal of Structural Engineering*, 138(4), 539-548. doi:10.1061/(asce)st.1943-541x.0000491
- [26] Brunete, A., Ranganath, A., Segovia, S., De Frutos, J. P., Hernando, M., & Gambao, E. (2017). Current trends in reconfigurable modular robots design. *International Journal of Advanced Robotic Systems*, 14(3), 172988141771045. doi:10.1177/1729881417710457
- [27] Shen, W., Lu, Y., & Will, P. (2000). Hormone-based control for self-reconfigurable robots. *Proceedings of the fourth international conference on Autonomous agents - AGENTS '00*. doi:10.1145/336595.336602

- [28] Sproewitz, A., Billard, A., Dillenbourg, P., & Ijspeert, A. J. (2009). Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture. *2009 IEEE International Conference on Robotics and Automation*. doi:10.1109/robot.2009.5152613
- [29] A. Spröowitz, R. Möckel, J. Maye, M. Asadpour, A. J. Ijspeert (2007). Adaptive Locomotion Control in Modular Robotics. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*
- [30] Christensen, A. L. (2012). Self-Reconfigurable Robots—An Introduction. Kasper Stoy, David Brandt, and David J. Christensen. (2010, MIT Press.), 224 pages. *Artificial Life*, 18(2), 237-240. doi:10.1162/artl_r_00061
- [31] R.E. Skelton, M.C. de Oliveira, *Tensegrity Systems*, 179 DOI 10.1007/978-0-387-74242-7_6, c_ Springer Science+Business Media, LLC 2009
- [31] R.E. Skelton, M.C. de Oliveira, *Tensegrity Systems*, 179 DOI 10.1007/978-0-387-74242-7_6, c_ Springer Science+Business Media, LLC 2009
- [32] Sabelhaus, A. P., Akella, A. K., Ahmad, Z. A., & SunSpiral, V. (2017). Model-Predictive Control of a flexible spine robot. *2017 American Control Conference (ACC)*. doi:10.23919/acc.2017.7963738
- [33] Jorgensen, M., Ostergaard, E., & Lund, H. (n.d.). Modular ATRON: modules for a self-reconfigurable robot. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. doi:10.1109/iros.2004.1389702
- [34] Chennareddy, S. S., Agrawal, A., & Karuppiah, A. (2017). Corrigendum to “Modular Self-Reconfigurable Robotic Systems: A Survey on Hardware Architectures”. *Journal of Robotics*, 2017, 1-4. doi:10.1155/2017/7139382
- [35] Salem, B. (2014). PetRo: Development of a modular pet robot. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. doi:10.1109/roman.2014.6926299
- [36] Fukuda, T., & Kawauchi, Y. (n.d.). Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator. *Proceedings., IEEE International Conference on Robotics and Automation*. doi:10.1109/robot.1990.126059

- [37] *Scanned Photographs of Polypod*. (1996). Retrieved from <http://robotics.stanford.edu/users/mark/photos.html>
- [38] Andres Castano, Wei-Min Shen, Peter Will. CONRO: Towards Deployable Robots with Inter-Robots Metamorphic Capabilities. *Autonomous Robots*, Springer Nature 2010.
- [39] Moeckel, R., Jaquier, C., Drapel, K., Dittrich, E., Upegui, A., & Ijspeert, A. (2006). YaMoR and Bluemove — An Autonomous Modular Robot with Bluetooth Interface for Exploring Adaptive Locomotion. *Climbing and Walking Robots*, 685-692. doi:10.1007/3-540-26415-9_82
- [40] Lyder, A., Garcia, R., & Stoy, K. (2008). Mechanical design of odin, an extendable heterogeneous deformable modular robot. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. doi:10.1109/iros.2008.4650888
- [41] Prabhu, S. M., & Garg, D. P. (1996). Artificial neural network based robot control: An overview. *Journal of Intelligent and Robotic Systems*, 15(4), 333-365. doi:10.1007/bf00437601
- [42] Floreano, D., & Mattiussi, C. (2002). *Manuale sulle reti neurali*. Bologna: Il mulino.
- [43] Ijspeert, A. J., Crespi, A., & Cabelguen, J. (2005). Simulation and Robotics Studies of Salamander Locomotion: Applying Neurobiological Principles to the Control of Locomotion in Robots. *Neuroinformatics*, 3(3), 171-196. doi:10.1385/ni:3:3:171
- [44] Owaki, D., Kano, T., Nagasawa, K., Tero, A., & Ishiguro, A. (2012). Simple robot suggests physical interlimb communication is essential for quadruped walking. *Journal of The Royal Society Interface*, 10(78), 20120669-20120669. doi:10.1098/rsif.2012.0669

11 List of figures

Figure 1-1: Tensegrity structure, cables in red and struts in black.	6
Figure 1-2: Cell picture obtained with fluorescence microscopy. In blue is the nucleus of the cell and at the bottom a sketch of the two main constituents of the cytoskeleton: microtubules and actin filaments [10].	7
Figure 1-3: Tensegrity models of the spine show how vertebrae float without touching [2].	7
Figure 1-4: ATRON robot.	8
Figure 1-5: A couple of modules belonging to a tensegrity modular robot [10].	9
Figure 1-6: Icosahedron tensegrity structure [10].	9
Figure 1-7: A complex and highly dynamically coupled 15-bar tensegrity structure [13].	9
Figure 1-8: Spine-like tensegrity robot [22].	10
Figure 1-9: Super Ball Bot all-in-one landing and mobility platform based on tensegrity structures. All credits to NASA research center.	11
Figure 1-10: Tensegrity deployability example, from the compact configuration to the expanded configuration [25].	11
Figure 2-1: ATRON System [34].	12
Figure 2-2: PetRo prototype on the left, caterpillar configuration on the right [35].	13
Figure 2-3: CEBOT robot [36].	13
Figure 2-4: Two polypod's segments, one expanded on the left and one angled on the right [37].	14
Figure 2-5: CONRO robot snake configuration [38].	15
Figure 2-6: Different configurations of YaMoR modules [39].	15
Figure 2-7: Odin robot with 21 modules: 8 telescoping link modules (black links), 6 rigid passive link modules (white links), and 7 joint modules.	16
Figure 2-8: SUPERball and the 8 beacons installed to locate it wrt a fixed reference frame [5].	17
Figure 2-9: SUPERball and the 8 beacons installed to locate it wrt a fixed reference frame [5].	18
Figure 2-10: Tensegrity modular robot [10].	19
Figure 2-11: Roombots modules.	20
Figure 2-12: Example of tripod YaMoR robot, on the right and corresponding CPG configuration, on the left, in red lines [29].	21
Figure 2-13: NASA ReCTeR [2].	22
Figure 2-14: NASA ReCTeR [2].	23
Figure 2-15: Spine-like tensegrity robot [22].	24
Figure 2-16: CPG closed-loop and open-loop performances on different terrains [22].	24

Figure 2-17: ULTRA Spine robot [32].	25
Figure 2-18: A complex and highly dynamically coupled 15-bar tensegrity structure [13].	26
Figure 4-1: SUPERball sits in the middle of the plot surrounded by 8 ranging base stations [5].	29
Figure 4-2: SUPERball data flow [5].	29
Figure 4-3: Multiple module shape reconstruction using COGs.	30
Figure 4-4: Capacitive stretchable sensor tested at EPFL [23].	31
Figure 4-5: One cycle test with different elongations [23].	32
Figure 4-6: Ten thousand cycles extremum points [23].	33
Figure 4-7: Tensegrity modular robot made of three modules and motion direction.	34
Figure 4-8: Single module picture and sensor positioning.	35
Figure 4-9: Sensors positioning in a stylized three modules robot.	36
Figure 4-10: Tensegrity net and sensor in black and the six cables that compose it.	36
Figure 5-1: Sensor's fabrication steps [23].	39
Figure 5-2: Net with embedded sensor.	39
Figure 5-3: Electrodes connection.	40
Figure 5-4: Flat net and new support system.	41
Figure 5-5: Cap and collar printed in NinjaFlex.	41
Figure 5-6: Old net and broken support system	41
Figure 5-7: Sylgard-184 broken net.	42
Figure 5-8: Comparison between a tensegrity made in EcoFlex 00-30, on the left and one made in NinjaFlex, on the right.	43
Figure 5-9: Tensegrity net in NinjaFlex, in red and EcoFlex 00-30 sensor, in black, before being glued together.	44
Figure 5-10: Capacitance measurements taken with LRC meter.	46
Figure 5-11: Capacitance recorded using AdaFuit board. Between two vertical lines there are 10[°], starting from 0 till 110.	47
Figure 5-12: Voltage divider.	48
Figure 5-13: Voltage divider.	48
Figure 5-14: Filtered signal, in blue, compared with the unfiltered one, in red.	49
Figure 5-15: a) Potentiometer analog signal b) Modified servo ready to use.	50
Figure 5-2: In the picture are shown the resistance's variation and the hysteresis under 100 cycles, in particular from cycle 200 to cycle 300. Despite some vibrations induced by the motor, the signal is quite stable.	51

Figure 5-3: In the picture are shown the resistance's variation and the hysteresis under 100 cycles, in particular from cycle 800 to cycle 900. In the final part of the plot the vibrations of the motor are tracked by the sensor.....	51
Figure 6-1: Neural Network.	54
Figure 6-2: CPG Owaki et al.....	55
Figure 6-3: Leg control.	55
Figure 6-4: Interconnections scheme 1.	57
Figure 6-5: Interconnections scheme 2.	57
Figure 7-1: Module representation on Simulink.	59
Figure 7-2: Interconnections scheme 1 Figure.....	60
7-3: Interconnections scheme 2 implemented in Simulink.	60
7-4: Arduino and circuitry for training and control the robot.	61
7-5: Finding omega using two potentiometers, one to change omega and the other one to change the gain of the servo.	62

