Tohoku University

Research institute of electrical communication (RIEC)

Ishiguro - Kano laboratory

Physical realization of an earthworm robot with a force based feedback control algorithm

Rodrigo Senofieni
email: rodrigo.senofieni@mail.polimi.it

February 2020

Contents

1 Preface

Earthworm are very interesting animals but yet not a lot studied in robotics. In comparison with other animal-like robots, earthworm robot have seen a small development and not many applications in real life cases. Their small size and their adaptability could play a fundamental role in many real-life scenario where humans and traditional robots struggle to work. If we think for example about inaccessible place like pipes, small holes or dangerous areas (inaccessible for other robots), earthworm robot could play a fundamental role. Another interesting application could be for medical examination with non-invasive approach.
Several works have been carried out by other researcher. We have a great realization by [1] or a similar example by [2]. Other realization were carried on also by [3] and [4].
In our realization we are focusing more on different aspects: modularity, behaviour with respect to different situations (hill or slope conditions) and adaptability depending on the terrain. As we will explain later on, thanks to a force/friction based feedback control algorithm, the robot is capable of moving efficiently in different situations.

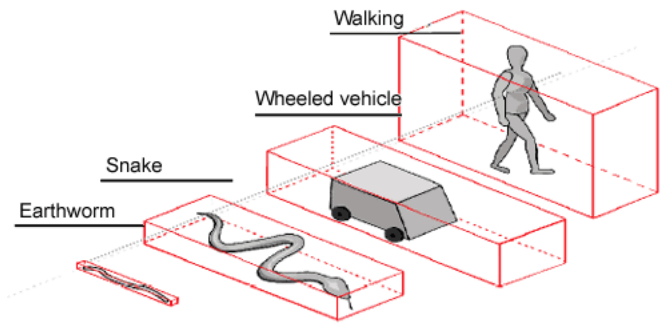


Figure 1: Difference in scale between different object. Earthworm can easily work and live in environments where others may struggle. For the picture credits to [3].

This project was carried out from October 2019 to February 2020, during the exchange period studying at Tohoku University. The laboratory that hosted me and provided the resources for the robot is the Ishiguro - Kano laboratory, see [5].

2 Introduction to the problem

The main work was subdivided and spread over all five months. The idea was to first develop some prototypes and test the various sensors and motors, and exploiting different mechanical structures. This gradual approach gave the possibility to understand the pros and cons of different and possible solutions. After having a clear idea of how the robot should be made, the construction and assembly phase were ready to go. The work was subdivided during the five months as follow:

- First month: understanding of the possible and different realization of an earthworm robot, the prof and cons between the different mechanical design.
- Second month: testing of the various force sensors and motors required for the earthworm to move.
- Third month: beginning of the construction and assembly of the first section. Implementation of a dummy control law.

- Fourth month: construction and assembly of other segments of the earthworm. Implementation of a force-based feedback control law
- Last month: testing in different operating conditions and small improvements/changes.

In this academic realization the idea was to keep things as simple as possible and to avoid non-ideal working conditions. As we are going to see in the next paragraph, the force-based control law is thought to be working only in one moving direction, thus the earthworm in our realization was modelled and built in order to move only straightforward. A more detailed description of the mechanical structure will be provided in chapter 4. The main idea of this project is to show that even the simplest force-based control law shows an adapting behaviour and a propagating wave, perfectly exploiting the movement of an earthworm.

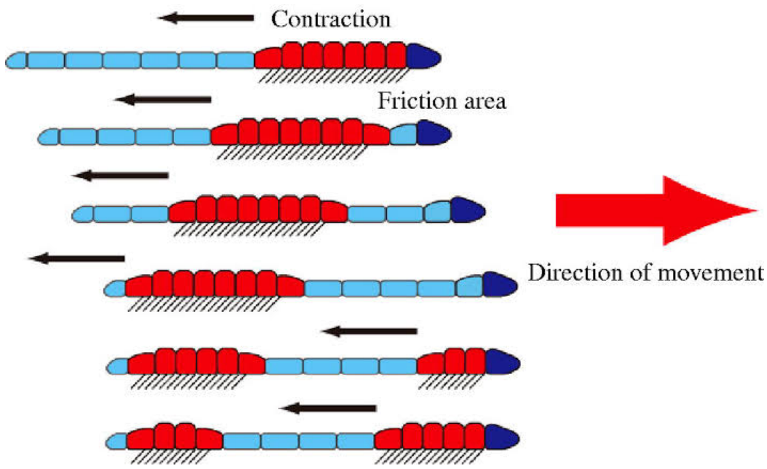


Figure 2: Simple representation of the propagating wave of contraction/expansion along one direction of an earthworm

It is clear that an earthworm achieve movement thanks to a series of contraction and propagating waves. However, it is still unknown which is the mechanism inside the "brain" of an earthworm that brings it to have such specific movement. The idea of this work is to show how a simple design based on informations coming from external world (such friction from the ground) can replicate efficiently the natural movement of an earthworm. As we are going to explain in the next section, the control algorithm is mainly based on informations coming from the outside world (i.e. the reaction force coming from the ground above) and generate a series of contraction and propagating waves such as a real earthworm.

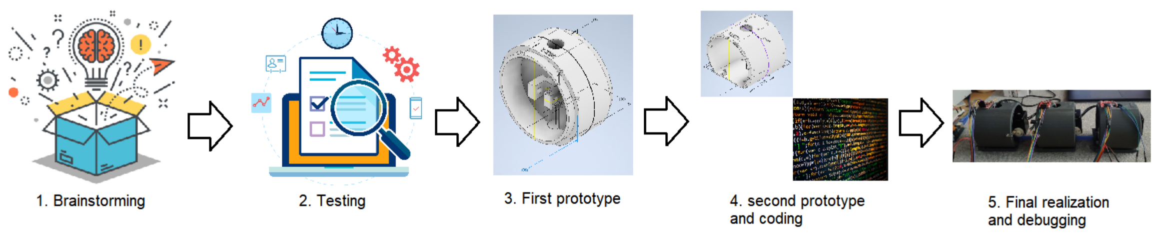


Figure 3: The five steps visualized

3 Mathematical model and control law

The main inspiration for the mathematical model comes from past works conducted by the laboratory professors in [6]. An earthworm moving in only one direction (i.e. only straightforward) can be modelled, from a theoretical point of view as follow:

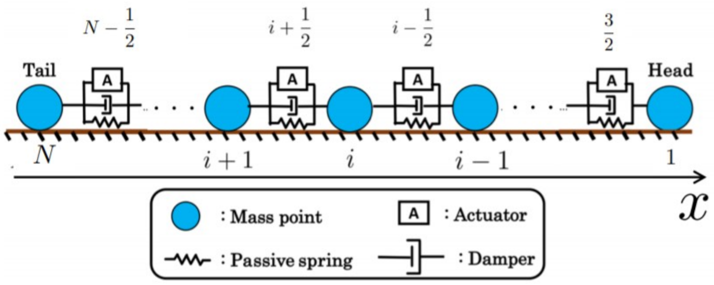


Figure 4: Theoretical model of the earthworm. Credits to [6]

As we can see each section is composed by a mass point i, which is connected to the other sections thanks to a spring/damper system and an actuator. The original idea proposed in the paper was to have this mechanism, however in a practical realization in order to avoid any kind of non linearity due to spring stiffness or dampening, only an actuator was fitted in each section, giving the capability of contracting and expanding when required.
the simple decentralize control rule is expressed as follow:

$$\dot{f}_{i+1/2} = h \tanh\left(\sigma_1 \sum_{j=i-n_f+1}^{i} S_j - \sigma_2 \sum_{j=i+1}^{i+n_b} S_j\right) - \sigma_3 f_{i+1/2} \tag{1}$$

The time evolution of the actuation force to be applied to each actuator in each section can be described as follow: whenever n_f mass point in front of the one I am considering receive a propulsive reaction force, the current actuator increase a contraction force such to pull the posterior part. On the other hand, when n_b mass points receive a propulsive force, the current actuator generates an expansion force such that the front sections are pushed forward. Further and more detailed explanation are contained in [6].
Since the decentralized control law is based on feedback informations coming from n_f and n_b section (respectively front sections and back sections), a series of sensors are required to sense the feedback signals.
The first obvious requirement is to measure the force S_j coming from the ground. S_j represents the reaction force coming from the ground and acting on the bottom part of the robot, positive in the direction of movement. A precise estimation of this force is necessary to measure at each time instant the amount of force which is being applied on each section, both from the ground and from other actuators connected to the section, contracting or expanding and thus, applying a force.
Now, since the dimension of the robot will be very small, the components of the force which have been taken into account are only the longitudinal ones. The motor and the screw mechanism definitely generates some yaw movement when contracting and expanding, but these effects have been neglected thanks to the tuning and calibration of the force sensors used in the robot.
Other important considerations have to be made for what is concerning the friction between the ground and the robot surface. Since the purpose of this work is not to built

a perfect crawling robot but instead show as, from a simple decentralized control rule some non-trivial behaviours emerges, the friction between the bottom of the robot and the ground has been assumed to be *anisotropic*. In section four, a detailed description of the approach used to achieve this goal will be discussed.
It is now clear that the main components required in each section are:

- A good way of measuring the horizontal reaction force coming from the ground, in order to have a good estimation of the time evolution of the force for each actuator in each section.
- A proper actuator, preferably with the possibility to be current/force or torque controlled.

Having these two fundamental aspects in mind, we can now proceed to the next section.

4 Modelling of the mechanical structure

4.1 Realization and materials

The mechanical structure of each section has gone through several design phases with improvements, changes or adding of new parts, in order to best fit all the desired components. The main aspects that were taken into account were: accessibility for assembly, dimensions (small as possible), enough space for anchoring the motor and the force sensors and a shape at least similar to an earthworm. As a first step, the idea was to keep the dimensions as small as possible, and thus the first prototype was the one below:

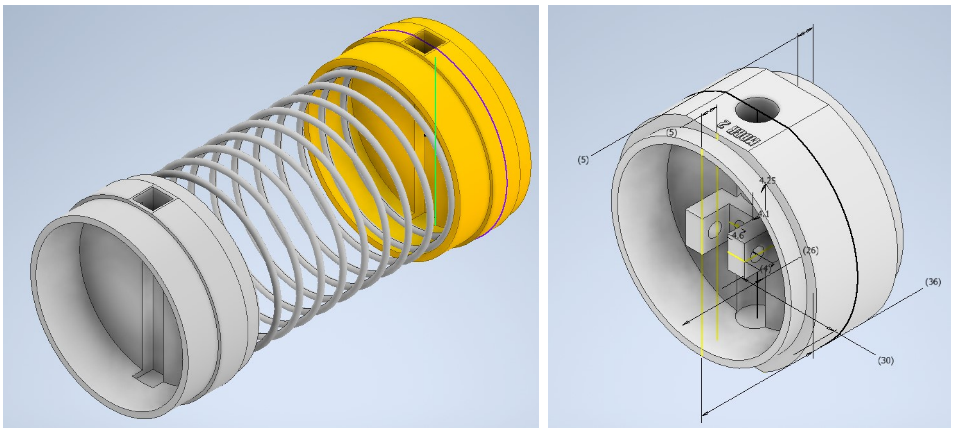


Figure 5: Mark 1 on the left, Mark 2 on the right. MK1 was the very first one and was meant to help exploring different design aspect of the possible final robot. In this case, a spring was inserted between two section. Even though, in the final realization no spring was necessary for the functioning of the robot. On the other side, the MK2 was though to be housing a linear actuator by *Actuonix*. However, this design was later abandoned due to hardware necessities.

The first practical realization of a section was the MK3. This helped a lot understanding the new feature to be added and the possible changes to be made. As it is visible, the key feature of this design where the dimensions (very small) and the flat bottom (in order to increase the friction area). The big contact area on the bottom was fundamental since the robot needs a lot of friction for moving forward. As already introduced in section 3, the usage of anisotropic material mounted on the bottom helped a lot the robot while moving:

when elongating (in the direction of movement) the anisotropic material gives very little friction, while when contracting (movement against the direction of motion), the material gives a big friction coefficient, making the movement possible.

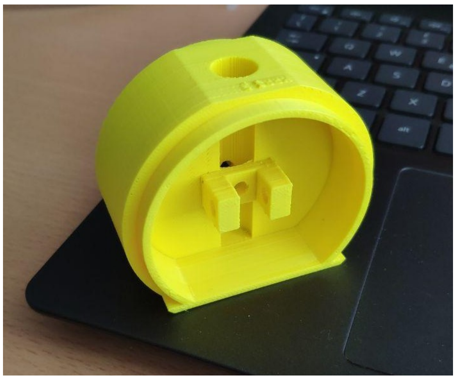


Figure 6: Physical model of the MK3. Material used: ABS plastic

In order to obtain such a behaviour, a material with a friction coefficient depending on the direction of movement was required. We can see an interesting study conducted by [7], in which a perfect anisotropic material is constructed. However, as a first step, a more simple approach was adopted. This kind of materials are widely used by professional skiers: by attaching an isotropic material under their ski they can efficiently walk with their skies on, without loosing the sliding capability when moving forward.
There are several companies providing this kind of materials, but the one used here is from Colltex [8].

Figure 7: General working principle of Colltex ski skins

The MK3 design was first though to be housing the linear actuator produced by Actuonix [9] and a simple FSR attached to the bottom of the section to sense horizontal reaction force. However, this design was later abandoned due to two main problem:

- First, the linear actuator did not provide any feedback information about the force, current, speed or position, but only a feedback signal when limit elongation was reached. Thus, for our project this was not enough.
- Second, the measurement of the horizontal reaction force coming from the ground was subjected to many uncertainties and thus was not giving correct measurements. As a consequence, a more complex and accurate design was required.

4.2 Final design and assembly

The turning point in the desired mechanical structure was obtained with the MK4. After some consultations and some research, we found that the best solution for our robot was to equip it with Dynamixel servomotor: they offer great peak power, several feedback informations and a built-in controller. Furthermore, the Dynamixel motor in question could have been also current-controller, which suited perfectly the aim of the project. Further informations about the hardware used will be provided in section 5.1. For what is concerning the measurement of the horizontal reaction force, the solution was to adopt a load cell mounted vertically, attached to the base in contact with the ground and with the main structure. Thus, whenever a movement or a force is applied to the base (i.e. the horizontal reaction force coming from the ground), the load cell would have give the correct measurements. The final design of the MK4 can be observed in the pictures below.

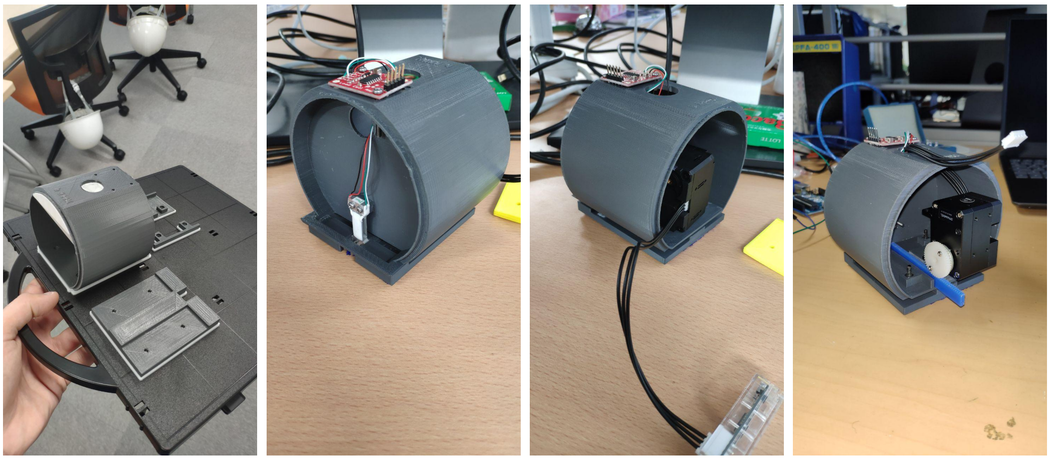


Figure 8: MK4 assembly. From left to right: parts printed, structure of load cell anchoring mechanism, mounting of the motor and final section

All the parts where printed in ABS plastic using the Stratasys Fortus 250mc 3D printer. An advantage of this design is its fully modularity. The section, once assembled, can be easily added and removed. One drawback is that, in order to house the hardware required (servomotor and screw/nut system), the overall section is about double in size with respect to the MK3 design. On positive aspect though is that, being bigger, the contact area in the bottom is larger, meaning that the robot can move better and more efficiently.
By printing another segment and putting the two together, I obtained the first section:

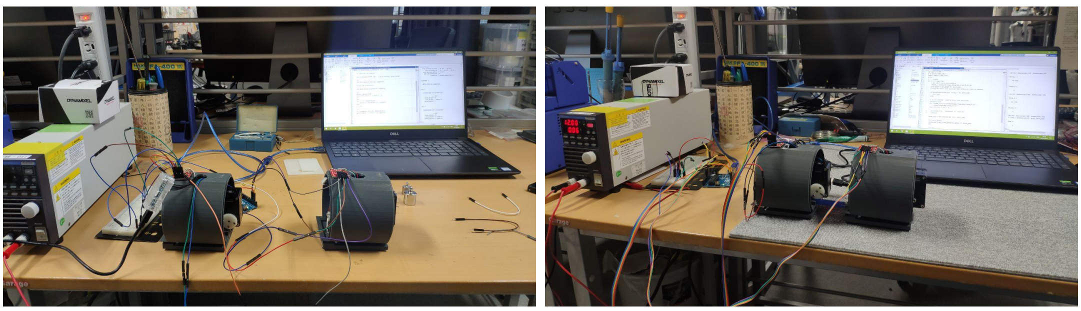


Figure 9: First section of the earthworm robot.

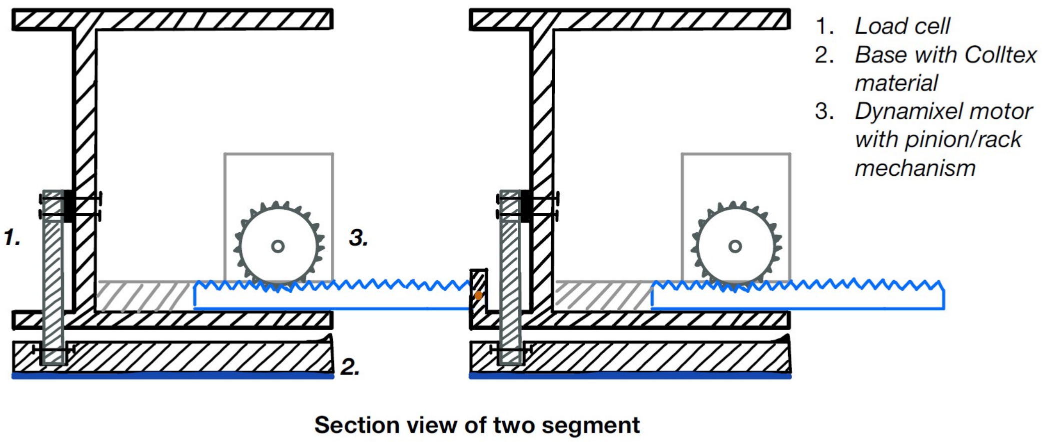


Figure 10: Lateral section view of 2 segments of the robot

The initial tests were performed only with this section. The aim was to test the prototype of the code and to solve any possible problem that would have come up. After we obtained satisfying results, we come up with further improvements to make to the mechanical structure, in order to improve performances. First of all, the overall robot was not enough rigid and sometimes it would have deviate the path, not going perfectly straightforward. This would have affected the force sensed from the load cells and as a consequence, more rigidity was required. To solve this problem, one linear guide for each section was added. As it can be seen from the pictures below, the linear guides not only provided the required rigidity, but also allowed the robot to have enough mobility and pitch movement in order to give it the flexibility of going on ground with changes of inclination.

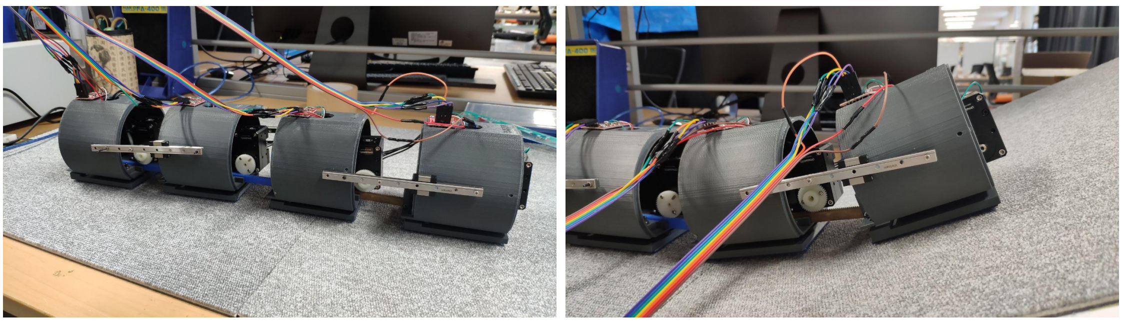


Figure 11: Structure of the MK4 in two situations: on the left we have flat ground, on the right a change of inclination.

5 Hardware used

In order to easily control the robot while moving, a set of different hardware was fitted inside. Even thought, the overall hardware complexity is very low.

5.1 Dynamixel XM430-W350

Each section is fitted with one of this motors, mounted upside down and current controlled. More Dynamixel motors can be connected in series and can be easily addressed thanks to their unique ID, which will identify them.

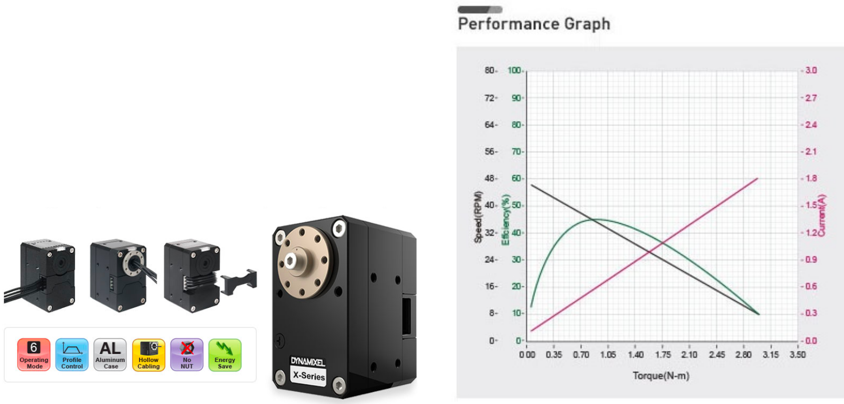


Figure 12: On the left, Dynamixel XM430-W350R. On the right, performance graph of the motor. More infos and data sheet are available at [10]

Each motor is mounted inside the section thanks to apposite fixture designed in inventor and 3D printed. Since the overall robot is fully modular and each section can be added or removed without any limit, to add a new section it is just necessary to connect the new motor to the previous one. Only the first motor will be connected via USB to the computer thanks to the provided tool from Dynamixel U2D2. To power up everything, an external power supply was used (12V, 1.8A).

Figure 13: Connection schema of more Dynamixel motors to the pc thanks to the U2D2 usb adaptor.

In section six, a more detailed description of the communication protocol, libraries and software environment will be given.

5.2 Load cell with Sparkfun amplifier and Arduino Uno

The other important hardware component is the load cell. This element is fundamental to have an estimation of the horizontal reaction force coming from the ground. In this way, the robot can have an idea of what is going on in the environment surrounding him and

efficiently evaluate the actuation force to be applied in each section.
However, in order to have the load cell properly working, it requires an amplifier since the signal is extremely weak and non measurable. A quick and efficient solution was to use Sparkfun load cell amplifier ([11]).

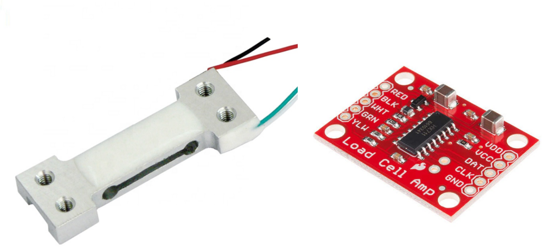


Figure 14: Load cell and amplifier respectively. Each section is equipped with one of this element.

In order to control and read data coming from the amplifier, an Arduino UNO was used. Considering that each loadcell requires two digital pin, a total of six section can be connected to a single Arduino UNO. After that, another arduino should be used. However, in our case only one was enough since the number of total section is five.
In the next section we will give a detailed explanation of libraries, tools and software used, with a detailed description of the code implemented.

6 Software implementation

The whole software for implementing the control algorithm has been developed in Matlab. Matlab provides a friendly environment and intuitive user interface. Also, thanks to the large amount of tools available, it offers great opportunities for testing and experimenting new features. In order to have the the motor working and properly communicating with the computer, some libraries were required. Thankfully, Dynamixel provides on their GitHub page plenty of libraries for several environments [12].
The key feature was compiling and addressing the c++ libraries in the Matlab directory. After that, all the built-in functions for communicating with the motor were available. Dynamixel motors uses TTL Half Duplex Asynchronous Serial Communication with 8bit. Furthermore, thanks to the built-in PID controller, the motor can achieve excellent performances in tracking references signals (in our case, the goal was to achieve a desired current).
The second most important thing was interfacing the load cells with Matlab. Again, thanks to the provided toolbox, communication with Arduino was very simple. The challenging steps was to interface and read correctly the values from the load cells. There are no official libraries for this hardware and furthermore, no support for Matlab. However, thanks to the work of [13], which provided on the Matlab file exchange website a full library with written instructions and an interactive toolbox, communicating and getting data from the load cells was very easy.
The main structure of the algorithm works as follow:

- Initialize/set the hyperparameters. The key feature of the implemented software is its easiness of adding new section to the robot. Thus, if a new section is added to the robot, there is no need of adding or modifying the code in any way. By changing the

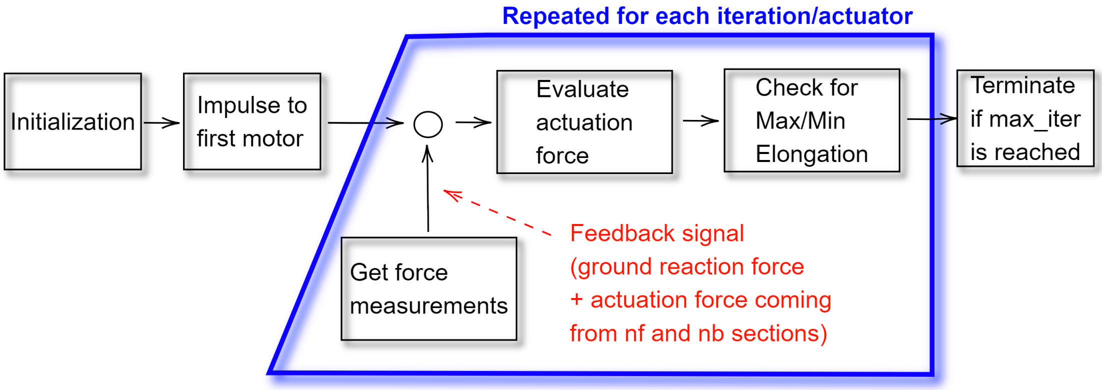


Figure 15: Main structure

2 hyperparameters $numAct$ and $numSec$ (i.e. number of actuators and number of sections, $numAct = numSec - 1$), the whole software is referred to this 2 parameters, thus, the robot will work perfectly.

- Self calibration of the load cells. The parameters for the calibration have been predetermined by using known weights. After that, for the calibrations phase the robot need to be steady and not touched in any way, in order to not interfere with the calibration.

- Initialize the serial communication between Matlab and the Dynamixel motors, get a success packet if everything is ok.

- Verify that all the motors and the racks are situated half way (this operation needs to be performed by the user manually).

- Set up number of maximum iterations (i.e. how much time the robot has to go straight forward).

- Main section of the force based feedback control algorithm. Matlab first read the values of the forces coming from the load cells, stores them in an bi-dimensional array (time and load cell in considerations, mainly used for plotting results at the end). After that, the force to be applied to each section is evaluated thanks to the solution of the ODE exploiting the movement of each section. Since we are dealing with discrete-time data, Forward euler integration was used to achieve the results. After that, Matlab sends the current command (proportional to the torque and thus, to the force) to the motor (figure 10, right plot).

- When achieved maximum number of iterations, stop.

- The overall algorithm is nested inside a "protection" control function that measure the positions of each encoder of the motors. The idea is that the movements of the earthworm are limited (i.e. cannot expand or contract as much as it wants, but there are some boundaries to be respected). Thus, this function prevents any damage to the robot (when contracting to much), or avoids that the rack gets out of the housing (when elongating to much).

- Close the communication with the serial port and disconnect Dynamixel.

In the next section results of the tests conducted will be discussed.

7 Tests and results

In order to test the performance of the robot, three main test were conducted: on flat ground, on a slope (18 and 25 degree of inclination respectively) and going down a slope (to test the energy efficiency feature).

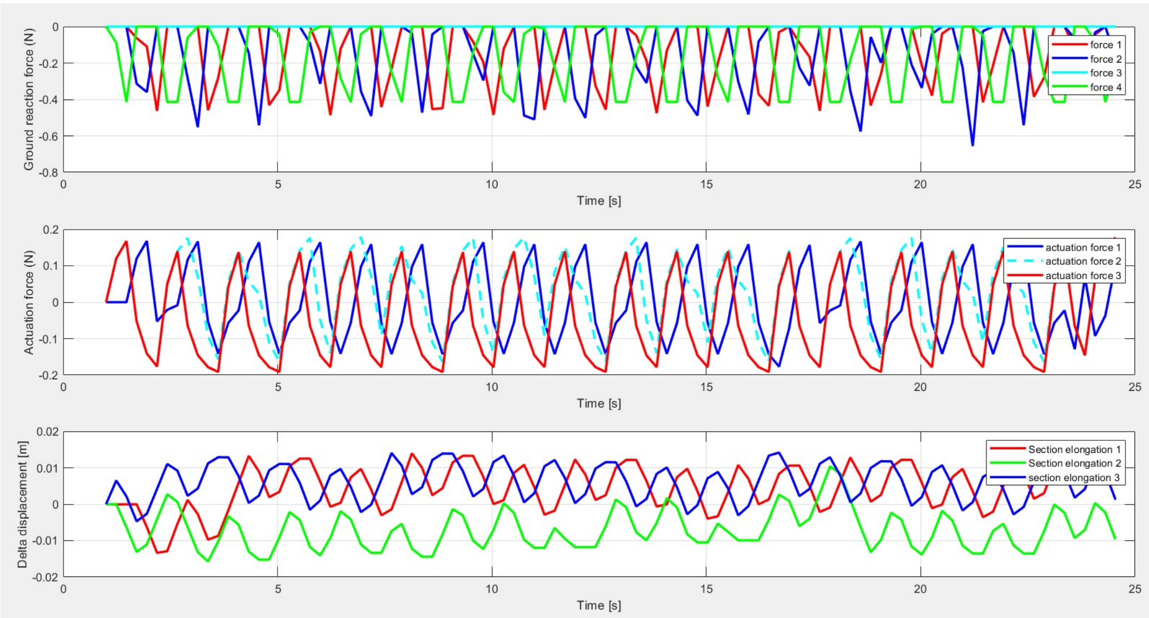
7.1 Flat ground

As expected, the robot can achieve the movement easily. Depending on the maximum number of iterations, the robot will go on straightforward. In the test, a total of 130 iterations were used (35 seconds simulation). The parameters used can be observed on the table below:

	Value
σ_1	10000 $[s^2/(Kg\ m)]$
σ_2	10000 $[s^2/(Kg\ m)]$
σ_3	6000 $[1/s]$
h	2000 $[Kg\ m/s^3]$
n_f	2
n_b	2
sim_time	25 $[s]$
k_p	600

Table 1: Set of parameters used

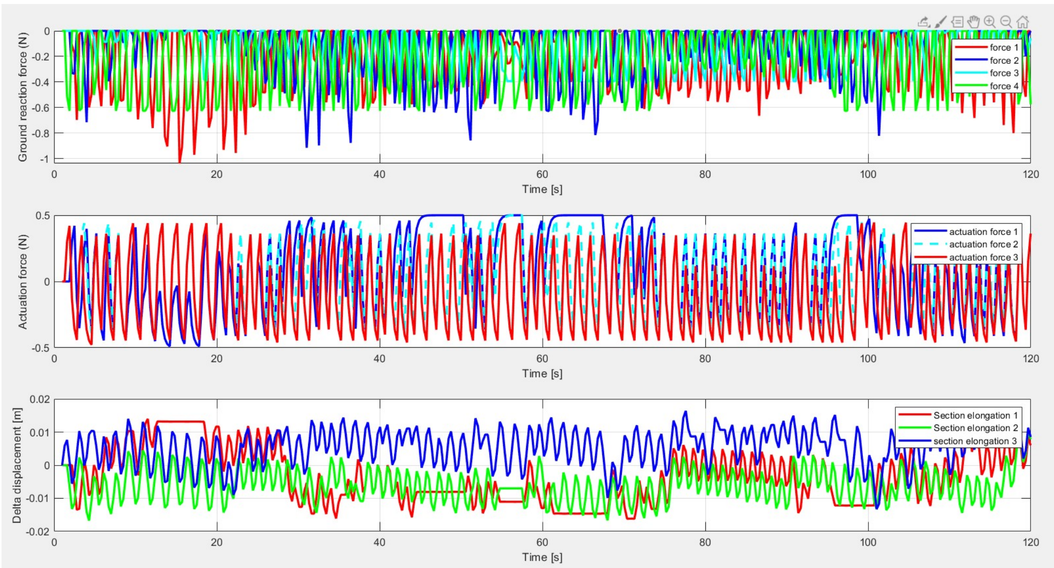
It is important to remember that the parameters are much bigger with respect to the ones used during simulations in [6]. This is mainly due to the fact that the overall robot is heavier (about ten times more, about 150 g/section) than the ones used for the simulations (10 g/section). By looking at the plots below, we can see that the robot can exploit movements efficiently.



The first plot shows the ground reaction forces coming from the ground. Since in our model the propulsive force is in the one that helps the robot moving, in our reference system is negative. Thus, since the other reaction force (the ones against the movement) are positive, they have been setted to zero because they do not contribute in any way to the movement of the robot. We can see that the load cell number 3 is not reading any force. This was founded to be normal, since n_f and n_b are equal to 2, the section number 3 is like "floating", always moving in the direction of movement. Changing the two feedback parameters n_f and n_b will affect the results and the plots.
The second plot is showing the evolution of the force of the three actuators. There is an evident non-trivial oscillating behaviour emerging from the decentralized control rule expressed in section 3. Notice that again, actuator number two and three are moving in the "same" way, due to the choice of the feedback parameters n_f and n_b.
The third plot shows the relative displacement of each section. Notice as the control law is making each section contracting and expanding, making the movement possible.
Although the simplifications and the assumptions taken into account with respect to [6], the overall results on flat ground are very satisfactory and in line with what was expected from the simulations.

7.2 Uphill (15 and 25 degree inclination respectively)

The goal is to test the adaptability of the robot with respect to external changes and relatively hard climbing conditions. Thanks to its flexible design, the robot can manage to face both scenario of 15 and 25 degree hill. However, there is a design choice that affects inevitably the results. The load cells used have a maximum reading capability of 100 grams. This value is definitely enough when going downhill or on flat ground, however when going uphill there might be some problems. As it can be seen from simulations results, the back section (on which the whole weight of the robot rests), at some point stops contracting and after some random time it starts working back again. This is due to the fact that, if the strain gauge built-in inside the load cell saturate (more than 100 g of force applied), the load cell will stop detecting weight. This situation is more visible in the case of the 25 degree inclination, while in the 15 degree case is less visible. Overall the robot can still manage to climb efficiently uphill. Below there is the resulting data from the performed experiments (only 25° inclination showed, parameters were not modified).



7.3 Downhill: energy saving feature

Earthworm and other animals that exploit movements with crawling locomotion show an interesting behaviour when going downhill. It can be observed a non non trivial behaviour emerging from earthworm or other crawling animals in particular situations: their adaptability to the surrounding world is amazing and yet not expected from such "simple" animals.

Figure 16: Snapshot of an earthworm going downhill.

The behaviour is the one that will minimize the force used going down, by simply using gravity force. The main idea of the control law proposed in [6] is also to show the energy saving of which the robot is capable. From the simulations it emerges that the robot, when going downhill, stops moving and simply "slides down", like a real earthworm does. In the simulations the assumption of ideal friction conditions when going downhill helps a lot the control algorithm exploiting the energy efficiency feature. However, in a real world experiment, these conditions are far from being replicable. Thus, the robot in the transition phase (from flat ground to downhill) has some struggle due to the fact that the frictions between the ground and the bottom of the robot are far from being ideal.
Nevertheless, in the experiments which were performed, the robot once it starts sliding it stops moving - which is what it was expected -. Once it arrives at the bottom of the slope, it starts moving again. There are several improvements which will be taken into considerations in section 8, and some of them are already being developed. The "ideal" situations would be to have a material on the bottom of the robot capable of giving maximum friction when giving propulsive reaction force and as low friction as possible in the direction of movement. Further considerations will be taken into account later on.
In order to try overcoming the transition problem between flat ground and going downhill, we take into consideration the idea of making the control law "less ideal", by adding a white Gaussian noise (WGN) to the evaluated actuation force. The idea is that, even if the robot detect a slope and the actuation force decreases, there is still some contribution coming from the WGN, trying to replicate possible disturbances coming from the surrounding environment. However, adding a WGN makes the algorithm less robust and the robot moves in a quite "irregular" way. Thus, this idea was later abandoned.

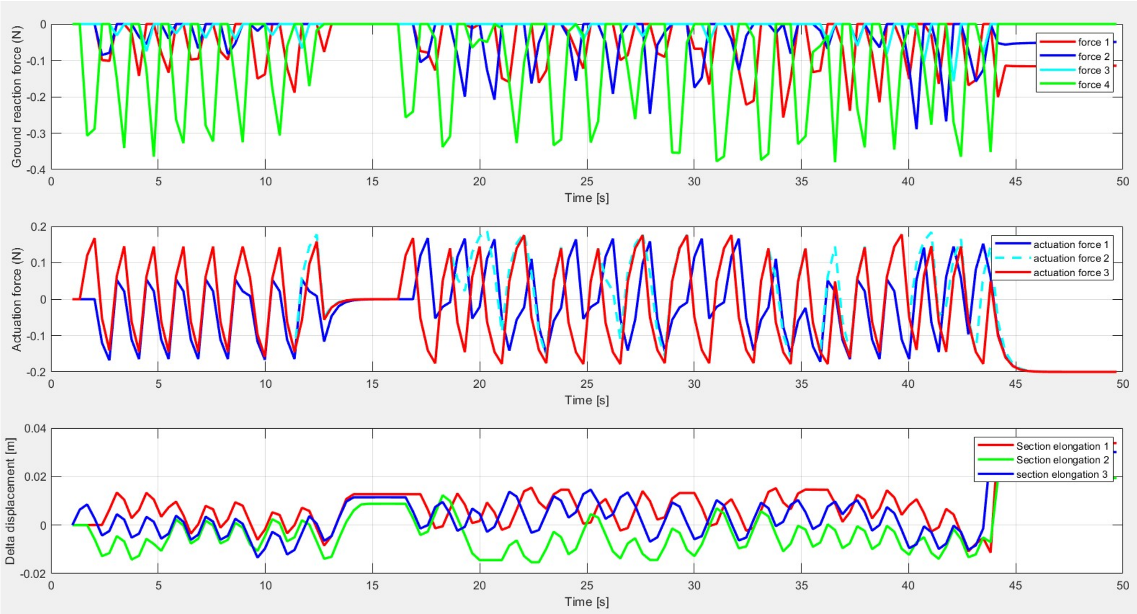
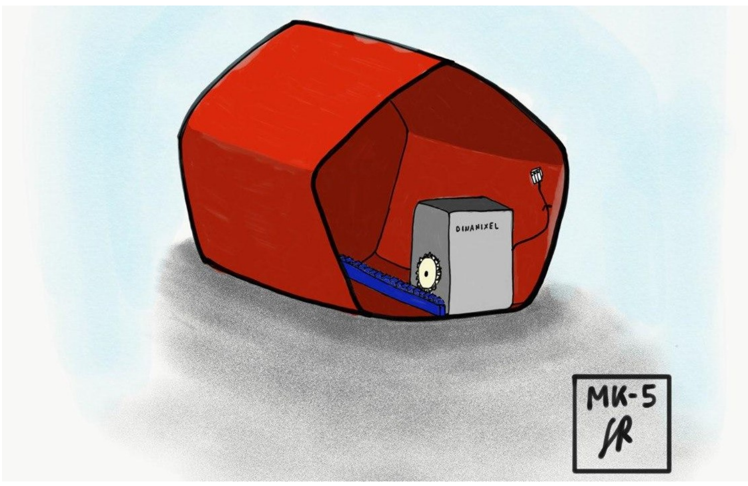


Figure 17: Downhill experiment. It can be observed that, around 13 seconds of simulation time, the feedback algorithm correctly stops the actuation force, making the robot sliding down the hill without wasting any energy.

8 Possible future improvements

In this section new ideas and further possible improvements will be taken in consideration. The current design can be improved still by a lot and these are some ideas.

- Different shape of each section. The current design was focused on making the assembly phase as easy as possible and to grant, in case of necessity, full repairability of any part. However a possible improvement is to make the overall robot "all-around", meaning that it is no more constrained of standing on only one side, but can easily be rolled over and still continue to work. A possible design that was theorized is the one below:



As it can be seen the hexagonal shape grants full versatility of the robot in any condition. However this design requires a new way of sensing the horizontal ground force.

- One weak point of the current design that can be definitely improved is the force measuring system. The usage of load cells makes the measurement too much dependant on the working conditions.

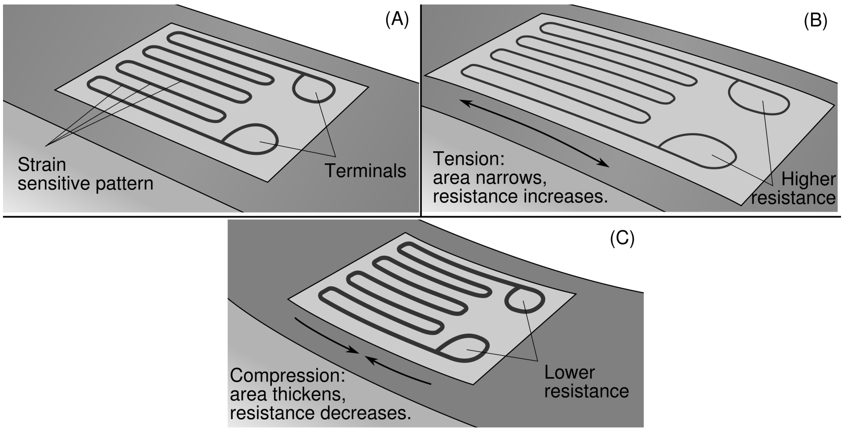


Figure 18: Working principle of a strain gauge. Notice that inside the load cell there is a similar hardware built in the mechanical structure.

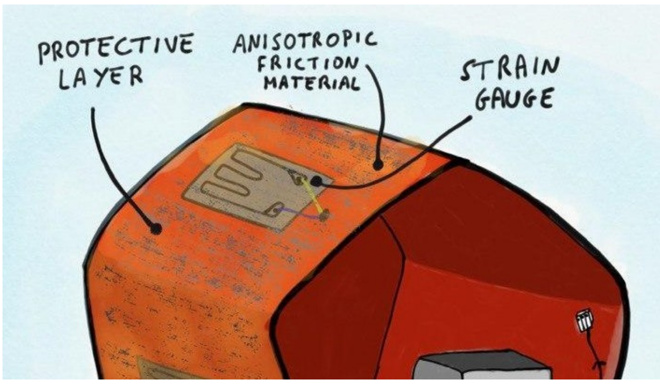


Figure 19: Possible implementation of the strain gauges on each side of the hexagon section. This guarantees fully adaptability in any working scenario

In order to have a robot capable of exploiting the movement in different situations and still measuring the correct horizontal reaction force is to implement on all the lateral surfaces of the section (in this case we are considering a possible hexagonal design) some strain gauges built in the mechanical structure or directly attached on the surface (ad of course adequately covered by some protective layer).

- Different anisotropic friction material. The current usage of Colltex [8] ski skin is offering a great friction in the direction of movement (giving a good anchoring capability). However, first the material produces too much friction when moving and second, the material is strongly dependant on the working surface. The ski skins

works very well on rough surfaces (like carpet, terrain, snow), but struggles a lot on extremely smooth surfaces (like for example on a simple floor of the laboratory). The main idea is to design a similar surface with the same properties of [8], but with different structure and shape such that the overall working conditions are more ideal and less dependant on the ground roughness. The development has been already started by taking inspirations by other studies and trying to come up with the best possible product that perfectly fit our robot.
The inspiration was taken from the work of [7], and by adapting it to our needs, we obtained different prototypes.
The printer used is the Objet260 Connex 2, with VeroWhite material for the low friction part and TangoBlack for the rubber-like, high friction part. The first two printed parts were conceptually good, but not very well performing in real case scenarios. The pins had been underestimated in length and thus, there was not enough bending guaranteeing the anisotropic feature of the two materials. Furthermore, the first one had its base structured printed in VeroWhite, making it definitely too stiff. The second and third prototypes have the base structure made of TangoBlack, guaranteeing good bending and friction properties. However, since we were not fully satisfied by the MK2 design, we decided to undergo with a third try, with thinner and longer pins. The figure below shows the evolution of the printed materials.

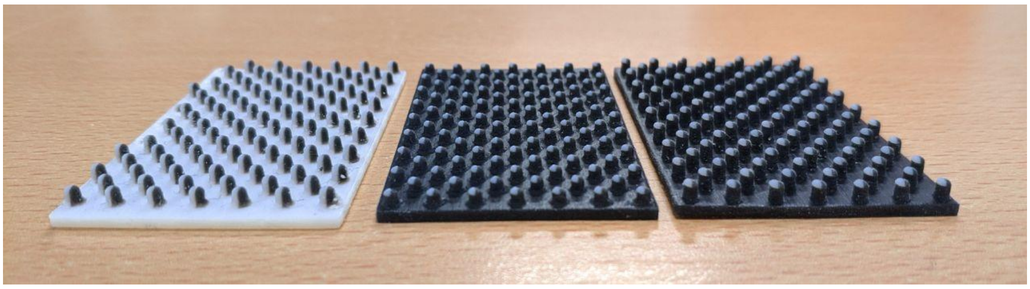


Figure 20: From left to right: MK1, MK2 and the MK3. The biggest difference between the MK2 and MK3 are thinner and longer pins, guaranteeing more bending.

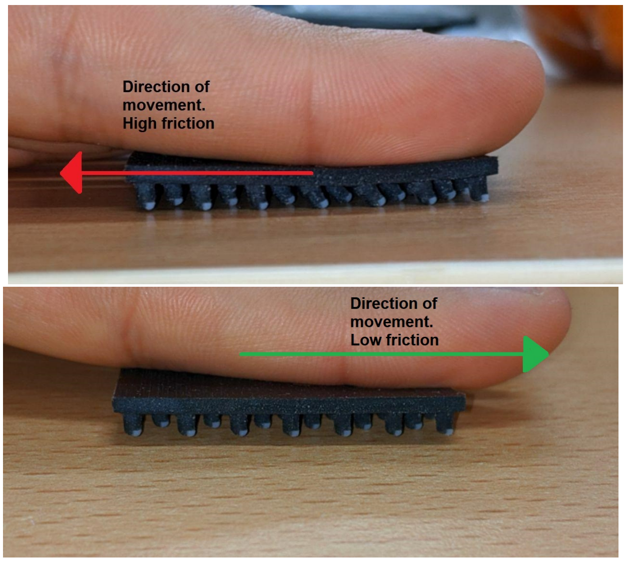


Figure 21: Working principle of the anisotropic material MK3. Credits to [7].

Even though the MK3 design and structure is the exact replica of the one proposed in [7], we did not obtain the behaviour we were hoping for. The material actually behaves well and gives two different horizontal reaction force depending on the direction of the movement, however it strongly depends from the vertical load applied on the material: the bigger is the load, the more the pins bend and as a consequence, the bigger is the difference in terms of anisotropic friction. However, since our robot is not very heavy, some other changes and improvements in the design are required. Overall, since the benefits were not considerable with respect to the usage of Colltex material, this material was not fitted to the robot.

9 Special thanks

The whole experience here at Ishiguro-Kano laboratory was supported by my home university Politecnico di Milano. Special thanks also to Tohoku university, which made all of this possible, my friends and my parents for always supporting me.
Thanks to all the laboratory members for helping me in this five months, for all the resources provided and for hosting me during this amazing experience.

References

[1] Joey Z. Ge, Ariel A. Calderon, and Nestor O. Perez-Arancibia, *An Earthworm-Inspired Soft Crawling Robot Controlled by Friction*

[2] MIT site: `http://news.mit.edu/2012/autonomous-earthworm-robot-0810`

[3] Source: `https://yab.yomiuri.co.jp/adv/chuo/dy/opinion/20140310.html`

[4] Sitangshu Chatterjee, Ryuma Niiyama, Yoshihiro Kawahara, *Design and Development of a Soft Robotic Earthworm with Hydrostatic Skeleton*

[5] Website: `http://www.cmplx.riec.tohoku.ac.jp/`

[6] Takeshi Kano, Daiki Kanto, Akio Ishiguro, *Non-trivial behaviours emerging from a simple decentralized rules (Part 1): A case study with one dimensional crawling locomotion*

[7] Tung D. Ta, Takuya Umedachi and Yoshihiro Kawahara, *Design of Frictional 2D-Anisotropy Surface for Wriggle Locomotion of Printable Soft-bodied Robots*

[8] Website: `https://www.colltex.ch/en/ski-skins`

[9] Website: `https://www.actuonix.com/`

[10] Dynamixel XM430-W350R reference: `http://emanual.robotis.com/docs/en/dxl/x/xm430-w350/`

[11] Sparkfun loadcell amplifier: `https://www.sparkfun.com/products/13879`

[12] GitHub page of Dynamixel: `https://github.com/ROBOTIS-GIT/DynamixelSDK`

[13] GitHub link forMatlab library: `https://github.com/GiacoboniNicholas/Adv_UNO_HX711_Matlab`