Poppy: open-source, 3D printed and fully-modular robotic platform for science, art and education

Matthieu Lapeyre

Poppy: open-source, 3D printed and fully-modular robotic platform for science, art and education.

Matthieu Lapeyre

September 26, 2014
Version: 0.13

Inria Bordeaux Sud-Ouest

Flower team

Submitted in fulfillment of requirements for the degree of Doctor of Philosophy Specialized in Computer Science

Poppy: open-source, 3D printed and fully-modular robotic platform for science, art and education.

Matthieu Lapeyre

1. Reviewer Fethi Ben Ouezdou
Vice-président à la Valorisation de la Recherche et de l'Innovation
Université de Versailles Saint-Quentin-en-Yvelines

2. Reviewer Jacques Droulez
Institut des Systèmes Intelligents et de Robotique
UPMC CNRS

Supervisors Pierre Yves Oudeyer

September 26, 2014

Matthieu Lapeyre
Poppy: open-source, 3D printed and fully-modular robotic platform for science, art and education.
Submitted in fulfillment of requirements for the degree of Doctor of Philosophy Specialized in Computer Science , September 26, 2014
Reviewers: Fethi Ben Ouezdou and Jacques Droulez
Supervisors: Pierre Yves Oudeyer

Inria Bordeaux Sud-Ouest
Flower team
200 Avenue de la vieille tour
33401 and Bordeaux

Abstract

This thesis **suggests novel approaches and design processes to create and produce robotic platforms, the control and morphology of which can be freely explored through experimentation in the real world, that are easy to diffuse and reproduce in the research community.** Especially, this alternative design methodology is driven by the desire to:

- freely explore morphological properties,
- reduce the amount of time required between an idea and its experimentation on an actual robotic platform in the real world,
- makes experiments that should be easy to do, actually easy to do,
- make the work easily reproducible in any other lab,
- keep the work modular and free to use in accordance with open source principles, so it can be reused and extended for other projects.

Our approach follows novel design methods for both design and production, for all technological aspects of the robot (i.e. mechanics, actuation, electronics, software, distribution). In particular these methods relies on 3D printing for all mechanical parts, the Arduino electronic architecture for the sensors acquisition, an easy to use Python API called pypot for the control and finally the disctribution of all our work under open source licenses.

Using this methodology, we create the Poppy Humanoid robot, a fully modular robot allowing exploring freely the role of morphology and adapting its body to specific experimental setup.

We experiment the use of this robot for several application. First, as a scientific tool and we show that Poppy can be easily and quickly modified to either explore the role of morphology or to be adapted to different experimental setups. Based on this work, but from another perspective we investigate the potential impact of such platform for educationnal and artistic applications.

Keywords: Morphology, Humanoid.

This work has been supported by INRIA and the ERC grant EXPLORERS 24007.

Résumé

Acknowledgement

First of all, I would like to thank Pierre-Yves Oudeyer for supervising me along this PhD thesis. Apart from his excellent advice and open minded supervision, he has been an example both as human-being and as a professional researcher. Moreover, he strongly supported the development of the Poppy project and is a main motor for the dissemination of the platform.

Numerous thanks for Pierre Rouanet, Jonathan Grizou, Didier Roy, Steve N'Guyen, Nicolas Rabault, Clement Moulin-Frier, Stephane Ribas, and Patrick Guillaud who actively participated to the technological development of the Poppy platform and its open source diffusion. I would like to thank all members of the Flowers laboratory for the amazing and unique atmosphere of the team, for sharing their knowledge and allowing me to learn every day. I also thank our team assistants Nicolas, Nathalie, Catherine, and the numerous internship students. In addition, I would like to thank Jasmine Garside for her amazing work and help as a first proofreader of this manuscript.

Many thanks to the members of my jury, Dr Fethi Ben Ouezdou, Dr Jacques Droulez, Dr Thierry Vieilleville and Mrs Anne-cecile Worms for accepting to review this work and taking part of the PhD defense [1].

I would like to thank the Inria staff for their support during the Poppy Project birth. IT people, Aurelien Dumez and Nicolas Sulek, for their work to set up all web platforms, needed to the creation of a successful open source project. General services people, J-P.Brumaud, P.Dieudoné, L.Dufayet for their daily support. The communication service, S.Valerius and L.Kovacic for their excellent advices and work to make the Poppy project known and understood outside the laboratory. The technology transfer service, P.Moussier and M.Cromer for their work on the trademark protection. Finally, I would like to thanks the Inria general direction for supporting and promoting the project, making this possible the continuation of this PhD thesis work as an applicative engineering project for research and education people.

[1] video of the PhD defense: `https://www.youtube.com/watch?v=6f0D-HHyqho`

During these three years, I had the chance to engage in different collaborative works. I would like to thank everyone who take the risk to start unique projects with a completely new and under development robotic platform. Thus, I thank the LPPA Laboratory for being the very first external entity to build a Poppy Humanoid at the early stage of the development process (only 5 months). I thank the Comacina Capsule creative Association, A.Braconnier and M-A.Villard, for their work as the first artistic project with the Poppy platform. I thank the Citée des Sciences museum for the autonomous organisation of Hackathon dedicated to our project. I thank the Saintonge Sainte-Famille high school and especially J.Claverie for being the first educational establishment to use Poppy as a tool for young students. I thank the ENSAM Paris and Bordeaux engineering school for including Poppy in their course. And finally, I would like to thank all members and actors of the growing Poppy community.

Unlimited thanks to my family for always have been supportive with all the projects I started since I was a young child, and especially to have always allowed me to do what I wanted. Among all, I thank my dear Rafiaa for being here along all these years, her open-minded ideas and the inexhaustible source of inspiration she is to me, also I guess the Poppy project creation would not have been possible without the unlimited support she gave to me.

Contents

1 Introduction

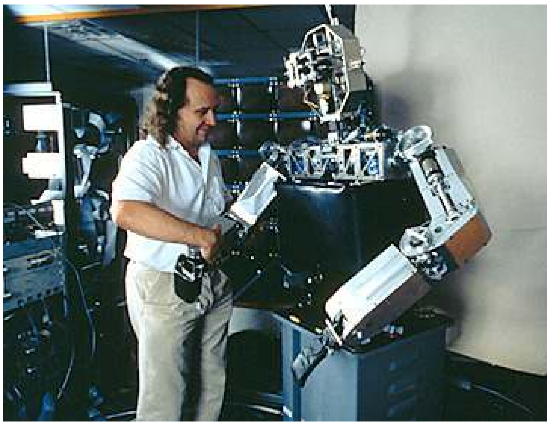
Research in humanoid robotics has been thriving in recent years (Hirai et al. (1998), Kaneko et al. (2008)), both due to their predicted relevance as personal and assistive robotics (Tapus et al. (2007), Oztop et al. (2005)), and because of the scientific challenges raised by robotics with regards to cognition (Asada et al., 2001), natural communication (Stiefelhagen et al. (2004), Breazeal and Scassellati (2002)), bipedal locomotion (Yamaguchi et al. (1999), Chestnutt et al. (2005), Collins and Ruina (2005)) and full-body physical interaction with the environment (Ude et al., 2004).

In the same way as the LHC is an experimental platform for exploring quantum mechanics and the origin of our universe, humanoids can act as simplified and controllable human simulators. Thus humanoid robots can be amazing tools for studying human beings and eventually contribute to a better understanding of human behaviour and abilities (Atkeson et al. (2000), Cheng et al. (2007), Brooks (1986), Oudeyer (2010)).

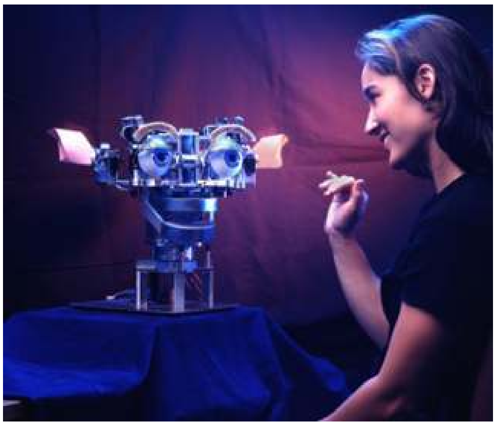
A famous example of such uses of humanoids was the Cog project (Brooks et al., 1999) at the Humanoid Robotics Group of the Massachusetts Institute of Technology. This research project had two goals: an engineering goal of building a prototype general-purpose flexible and dextrous autonomous robot and a scientific goal of understanding human cognition (Brooks and Stein, 1994). In particular, this project concentrated on embodiment and interaction intelligence with four aspects of a novel methodology: developmental structure, physical embodiment, integration of multiple sensory and motor systems, and social interaction. For this purpose they built several robotic platform such as a humanoid (Brooks et al., 1999) (see Fig. 1.1a), and a very expressive multi-articulated head named Kismet (Breazeal, 2003) (see Fig. 1.1b).

The context of this PhD thesis is grounded in the same scientific motivations as the work of R. Brooks, R. Pfeifer, T. McGeer and initiative such as the Cog project i.e. **exploring the role of morphology, cognition and embodiment intelligence in several ways using real experimental robotic platforms**.

The scientific approach of the Cog robots is oriented toward the exploration of embodiment in several ways, from the low level mechatronics to head design for social interactions, but robots were built 15 years ago and using classic manufacturing techniques (see Fig. 1.1) that made them expensive, complicated to modify and



(a)Rodney Brooks and the Cog humanoid



(b)Cynthia Breazeal with Kismet

Fig. 1.1.: The Cog project was about the use of computer and robotic technologies to better understand and emulate human intelligence.

especially difficult to diffuse in other laboratories. We are now in 2014, the makers revolution is in progress (Anderson, 2012) and novel technologies allow a rethink of the way we design robotic platforms, especially humanoid ones.

In the Inria Flowers team[1], we are interested in the study of mechanisms that can allow robots and humans to autonomously and cumulatively acquire repertoires of new skills over extended periods of time. This includes mechanisms for learning by self-exploration, as well as learning through interaction with peers, for the acquisition of both sensorimotor skills (e.g. locomotion, affordance learning and active manipulation) and social skills (e.g. grounded language use and understanding, adaptive interaction protocols, and human-robot collaboration).

An interesting evolution over the last decades has been the demonstration of the importance of robot morphology for sensorimotor control, cognition and development (Kaplan and Oudeyer (2008) Steels and Brooks (1995) Pfeifer and Bongard (2006)). Indeed, the actual behaviour of a robot is emerging from a complex interaction between the control algorithm, the robot's morphology and the environment (Steels, 1990). Moreover, it is clear that a well-adapted robot morphology using specific properties can greatly reduce the complexity of a given task by ensuring implicitly a part -or the entirety- of the control required (Pfeifer and Iida, 2005). Finally, as Rodney Brooks argued, *the world is its own best model* (Brooks, 1991) and simulators cannot realistically handle the complexity of real physics with multi-point contacts, soft material compliance, friction or unpredicted multi-modal interactions.

[1] `flowers.inria.fr`

Exploring mechanisms of sensorimotor tasks acquisition requires us to also focus on robot morphology. Therefore, we **should consider robot morphology[2] as an experimental variable (Kaplan and Oudeyer, 2008) that can be tuned, and conduct experiments in the real world**.

While it is straightforward to explore and experiment with the variation of certain software parameters (e.g. algorithms, simulator), experimenting with morphological variables on a real robot is much more challenging:

1. how can we have an experimental robotic platform that allows for morphology to be easily and quickly changed while acting robustly in the real world?

2. how can we make this project, mainly the hardware, diffusible and reusable in the research community?

Unfortunately current robotic platforms are not suitable for addressing such challenges. On one hand, commercial robots such as Nao (Gouaillier et al., 2008), Darwin Op (Ha et al., 2011), Nimbro Op (Schwarz et al., 2012) or iCub (Metta et al., 2008) are easily accessible and easy to use. Yet they provide a "traditional" morphology (e.g. limited compliance, rigid torso, large feet, over actuation) and modifying their morphology is impractical or impossible. On the other hand, lab prototypes are mainly handcrafted and specifically tuned which make them almost impossible to reproduce in another lab (Wisse et al. (2007), Nakanishi et al. (2013), Ly et al. (2011), Niiyama et al. (2010), Radkhah et al. (2011)). Moreover in most case, they are not open source and/or the hardware is to complicated/expensive to modify.

The main issue of these robots is the approaches and technologies chosen to design and produce them. Indeed, the classic way of designing and producing robots is a complicated, time-consuming and expensive process involving specific upfront tooling and complex manufacturing processes.

In this thesis, we **suggest novel approaches and design processes to create and produce robotic platforms, the control and morphology of which can be freely explored through experimentation in the real world, that are easy to diffuse and reproduce in the research community.** Especially, this alternative design methodology is driven by the desire to:

- freely explore morphological properties,

[2] robot morphology is defined as any characteristic which defines the physical structure of the robot such as link sizes, number of links, joint characteristics, mass distribution, actuator characteristics, material properties, sensor characteristics and sensor placements (Paul, 2006)

- reduce the amount of time required between an idea and its experimentation on an actual robotic platform in the real world,
- makes experiments that should be easy to do, actually easy to do,
- make the work easily reproducible in any other lab,
- keep the work modular and free to use in accordance with open source principles, so it can be reused and extended for other projects.

To reach these goals, we decided to follow novel design methods for both design and production, for all technological aspects of the robot (i.e. mechanics, actuation, electronics, software, distribution). In particular these methods relies on:

3D print mechanical parts: Since few years' novel techniques, especially 3D printing, are revolutionizing the way we can produce objects. 3D printers open new horizons as they are able to produce parts which were, until now, either not possible or extremely expensive to produce using classical techniques. Especially 3D printing techniques are fast, low-cost and accessible. It allows everyone to produce complex mechanical part in just a couple of hours without requiring any specific upfront tooling. These properties of the 3D printing process enable for the first time the exploration of morphological variant for mechanical parts. Indeed, it is now fast and low cost to create alternative designs. Associated with modular architecture, we can easily and quickly change robot parts and conduct experiments.

Electronic architecture based on Arduino Exploring the role of morphology does not only concern the mechanical properties but also the sensors apparatus i.e. which sensor is used and where it is placed on the body. While it is not yet possible to print complex electronic circuit, we preferred to rely on the Arduino hardware and software environment, which make electronic board easily reconfigurable and compatible with a wide range of sensors. Also, low-level embedded programing skills are not necessary because the board micro-controller can be programmed using Arduino programming language, which abstracts most of the complexity.

Easy to use python API: We designed a robust sensory-motor control API adapted to the hardware variability we have. We choose to use Python as it allows fast development, easy deployment on all operating system and quick scripting by non-necessary expert developers. It also offers a large variety of scientific and machine-learning libraries used in robotics (e.g. Numpy, Scipy, Scikit-learn).

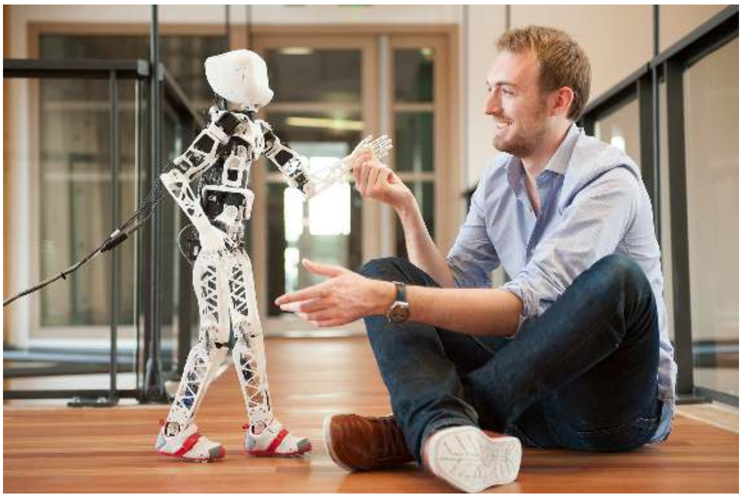


Fig. 1.2.: Poppy is a humanoid prototyping platform, which design has been made following the methodology presented in this thesis. It allows for a rich and easy exploration of the robot morphology and its impact on control and cognition. As open source and modular platform, it permits relevant applications in Science, Art and Education contexts. Also it strongly shares open collaboration and cumulative science philosophy

Open source diffusion: Finally, while the main aspect of such an approach is to allow variability, reuse and modification of the initial design, it is necessary to not only diffuse our work through scientific publications but also to distribute the material needed. This means anyone outside the Flowers lab should have access to the actual source files and be free to make any changes suitable to their own research. Therefore in addition to the technological choices previously presented, we decided to distribute all our work (both software and hardware) under open source licenses. This is an essential step toward building new research tools that facilitate both scientific results reproducibility and cumulative science in robotics.

We think this design methodology can contribute to 1) the construction of better experimental robots while making the modification of robot morphology both easy and low cost, and 2) the transfer and reuse of scientific work in other laboratories through the use of open source diffusion.

Within this context we have built a whole new humanoid robot called Poppy™ (see Fig. 1.2). This humanoid robot was designed to conduct scientific experiments easily and quickly on sensorimotor learning, exploring morphological properties, and human-robot interaction. As an experimental robotic platform, Poppy was designed to be **affordable**, **lightweight**, **robust and safe**, **modular**, **easy to use**, **highly-hackable** and **fast and easy to duplicate or modify** with the goal of being easily reproducible and used by other labs thanks to an open source distribution (hardware and software).

Poppy makes possible exploring new body shapes in just a few days. It enables and simplifies the experimentation, the reproduction and the modification of the morphology in research laboratories. It also allows collaborative working, sharing and replication of the results on these issues between laboratories. The ambition is to become a reference platform for benchmarking and dissemination of scientific results.

Thanks to the fact that it integrates advbanced and yet easily accessible techniques in an embodiment that motivates students and the wider public, this platform also meets a growing societal need: education and training in technologies combining computer science, electronics and mechanics, as well as a training tool to the emergent revolutionary 3D printing process. Poppy provides a unique context for experimentation and learning of these technologies in a Do-It-Yourself (DIY) approach. Finally, the possibility to easily modify both the hardware and the software also makes Poppy a useful tool for artistic projects working with interactive computerized installations.

Proceeding

The proceedings of this thesis will be structured along 4 main parts.

Firstly, the related work will present in chapter2 some inspirational scientific work made over the last 20 years showing the paramount importance of the robot morphology; in chapter 3 a quick overview of current robotic platform especially those exploring the role of morphology and humanoids and in chapter 4 the recent emergence of 3D printing techniques and open hardware projects.

The second part will describe the development of Poppy. In chapter 5, we will present the chosen approach to build robotic platform allowing both a free exploration of the morphology and the diffusion in the research community. Then in chapter 6 we will present how we actually used this approach for the design of a complete humanoid robot. Finally we will describe in the chapter 7 the development of an easy-to-use control library for modular robots called pypot.

The next part will deal with Poppy's applications. In chapter 8 we will discuss experiments we made to demonstrate how the Poppy's design can be used for exploring the role of morphology. In chapter 9 and chapter 10, we will present first experiments for educational and artistic purposes.

We will cloture this thesis with a discussion part. We will in particular discuss the challenge raised by the open source diffusion and the creation of community in chapter 11. We will then conclude the thesis in chapter 12.

Part I

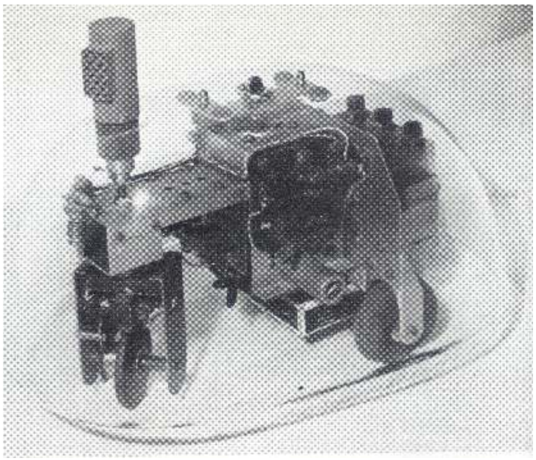
Related Work

2

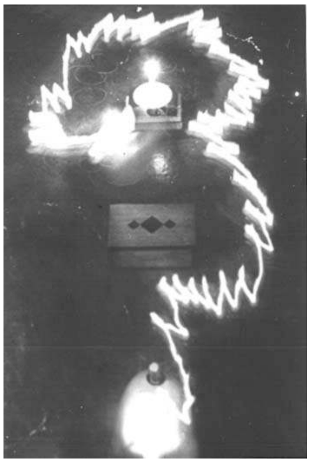
Exploring robot morphology: some fascinating work

2.1 Introduction

In 1949, Elmer and Elsie, also known as the turtle robots (see Fig. 2.1a), created by the cybernetic pioneer W. Grey Walter, could be considered as two of the earliest robots in the era of modern robotics history (1950-now). Back at this time, the transistor had just been invented (1948) (Brinkman et al., 1997) and calculus was done with mechanical machines. The turtle robot was entirely analog but was able to demonstrate complex behaviours (see Fig. 2.1b). Without any "reflection" or internal representation of itself and the world, this robot, thanks to its mechanical design and the direct analog interaction between sensors and actuators was able to avoid obstacles and reach its charging station (Walter, 1950). These complex behaviours, which can be compared to the ones found in nature, were in fact done without any kind of intelligence and were actually emerging from the interaction between the



(a)The turtle robot.



(b)Demonstration of obstacle avoidance behaviour.

Fig. 2.1.: The W. Grey Walter's turtle was a really simple robot using a direct analog connexion between light sensors and wheel actuators. The way sensors and actuators were plugged determined the robot's behaviour. It could demonstrate complex behaviour such as obstacle avoidance or returning to its recharging station.

robot's morphology (i.e. where sensors are placed and how they are connected with actuators) and the robot's environment (i.e. light sources).

2.1.1 The limits of the cognitivist approach

With the arrival of digital computers, researchers imagined the opening up of a field where it could be possible to replace pre-wired analog electronic behaviours by the use of computer-run programs. Not dependent on the hardware platform, robots would be therefore more versatile. Artificial intelligence (AI) term was introduced in a workshop organized in 1956 by a MIT professor John McCarthy (McCarthy, 1978). On the whole, participants were convinced, that by using the notion of computation or abstract symbol manipulation, it would be possible to reproduce interesting abilities similar to human ones (Kaufmann and McCorduck (1979), Haugeland (1989)). The symbol-processing paradigm or cognitivist paradigm sees cognition as pure computation. In other words, the abstract algorithm or the program doing calculus constitutes the actual process of intelligence. Eventually, researchers following this paradigm no longer saw physical incarnation as a relevant component. Cognitive and computationalist hypotheses state that thought is reducible to a set of symbolic calculations being established (Fodor, 1987). The body, on the other hand, is forgotten, irreparably separated from the mechanisms of intelligence (Kaplan and Oudeyer, 2008). Moreover, the robot body became a handicap, which often ruined the efficiency of algorithms and programs created by AI researchers. Indeed, the real world body is non-perfect, there is some noise in sensory acquisition, there is gravity, friction and inertia acting on actuators, and the environment is always changing and unpredictable.

To overcome these issues of real world applications, the other side of the robotics community, still interested in the hardware challenges strove to design more reliable and powerful robots which can react as fast and as closely as possible to the model used for its control. To do that far more precise sensors and actuators powerful enough to overcome inertia and mechanical friction are needed. Thanks to this work on hardware, industrial robots have become increasingly fast and precise, enough so to outclass any human on certain assembly tasks.

However, even with really efficient robots, artificial intelligence failed to show results comparable with the expectations researchers and society had. Robots are able to solve incredibly complex task such as chess games or able to achieve highly precise tasks in manufacturing but require a perfectly controlled and predictable environment. Going beyond this known environment seems impossible to program and none of them are able to act fluently in the real world. Unlike virtual worlds, the real world is challenging in various ways. It is not possible to enable omniscience: we do not access the knowledge of all world states and parameters, the measures a robot can take are limited, take time and are noisy while the action taken is always different. Finally real world states are never clearly defined as precise discrete states: "the weather is never simply good or bad" (Pfeifer and Bongard, 2006).

While the classical approach has known great successes in solving abstract problems such as chess games, search engines and text processing, it has failed in the understanding of natural forms of intelligence which requires a direct interaction with the real world. This is especially the case when we consider the current state of the art for interaction with humans (natural language) or objects (grasping) and locomotion in an open environment (walking, running, riding a bicycle).

2.1.2 Emergence of the embodiment paradigm

Stuck on these major issues that crop up when acting in the real world, a kind of crisis of artificial intelligence happened in the 1980s and the cognitivist paradigm was questioned. While some researchers in the field introduced new tools such as neuronal networks, others questioned the "cognition is computation" approach and the irrelevance of the body. Thanks to researchers such as Rodney Brooks (Brooks, 1986), Rolf Pfeifer (Pfeifer and Scheier, 2001) or Luc Steels (Steels and Brooks, 1995), a novel paradigm emerges: cognition needs a body to think. Embodied artificial intelligence rejects the symbolic approach and postulates that it is not possible to have intelligence without the body and the environment (Pfeifer and Scheier, 2001). Rather than postulating that there is a hierarchical structure in which the brain controls the body, the new theory focuses on the interaction between the two systems, even for mathematical thinking, which we could assume to be purely abstract (Lakoff and Núñez, 2000).

Following this paradigm, several researchers tried to tackle challenges in which the classical cognitivist approach failed i.e. the understanding of natural forms of intelligence, which requires a direct interaction with the real world. Locomotion is a great example of a task where classical robotic approaches did not yield expected results.

Animals are incredibly skilled. Even if we consider an insect with a brain a thousand times smaller than the human one, their abilities to move in an open world is simply incomparable to the most advanced current robots. One important reason for this is that in the classical view, the ability to figure out where you are is based on detailed inner models or representations having been either programmed into the robots or learned by interacting with the environment and continuously updated. The more complex these models are, the more effort is needed to acquire the relevant data to maintain them, leading to major problems when learning tasks in highly dimensional spaces. Brooks even argued that intelligence always requires a body and that we should forget about complex internal representations and models of the outside world; that we should not focus on sophisticated reasoning processes but rather capitalize on the system-environment interaction (Brooks, 1991) (Brooks, 1995). Then he started to work on insect locomotion because if we understand insect-level-intelligence it will be much easier and faster to understand and build human-level intelligence (Brooks, 1996).

Exploring the role of morphology and how it shapes the ways we think appears to be a fascinating open field. Indeed, exploring the interaction between body properties and cognition could lead to both a better understanding of animal behaviours (human being in particular) and to building robots more adapted and robust to an open environment with unpredictable interactions.

Thus, an interesting evolution over the last decades is the demonstration of the importance of morphology for sensorimotor control, cognition and development. The research community exploring the embodiment paradigm has grown, but surprisingly not as much as we could imagine given that the classical paradigm is failing. However, new work has appeared introducing new principles we will describe in this chapter such as morphological computation, compliance or ecological balance and emergence.

In the context of this thesis we will talk about intelligence as meaning the ability to move in a natural environment, and interact with people and objects.

2.2 Exploring the role of robot morphology

The achievement of robust locomotion in a natural environment is one of the major current challenges for robotics researchers. For decades and it is still mostly the case, the challenge of locomotion for robotic agent was only tackled through symbolic, abstract and complex computation of internal models and representations of the world. The body is reduced to a noisy interface between the abstract algorithm and the real world. However, if we look at nature, it appears obvious that an animal's morphology deeply changes the way it can act in its ecological system and so it has evolved in an attempt to optimize its body properties.

For some reason, in the fields of robotics and artificial intelligence the link between the body properties and the ability for a robot to move in an ecological environment does not seem as obvious. The fact that the ability to act and achieve complex tasks is due to brain computation is so deeply grounded that it even affects the general public.

However, while we may think there are indeed calculi necessary to achieve complex tasks, there is no reason they should be explicit, with precise internal models or representations of the physical world; they could be directly done through body properties.

Therefore since the 80s, considering robot morphology, defined as *any characteristic of the physical structure of the robot such as link sizes, number of links, joint characteristics, mass distribution, actuator characteristics, material properties, sensor characteristics and sensor placements* (Paul, 2006), novel research topics appeared exploring the role of robot body morphology in the achievement of complex tasks in the natural world, especially robot locomotion.

2.2.1 Morphological computation principle

Introduced by Rolf Pfeifer (Pfeifer and Iida, 2005), the morphological computation principle states that part of the computation needed in the achievement of a given task can be done implicitly through the interaction of a physical form with the ecological niche environment. Indeed, the morphology of a robot affects its control, because it not only determines the behaviours that can be actually performed, but also the amount of control required to achieve this behaviour correctly.

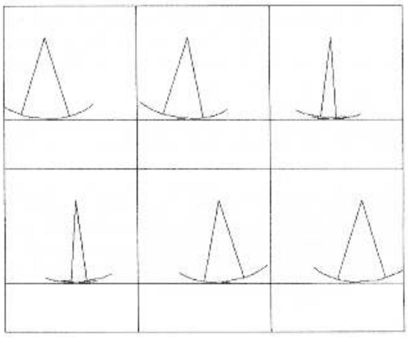
A great illustration is the achievement of flight by the airplane. Most of the control permitting the plane to fly is done by the interaction between the wings and the air. Indeed in a plane, the shape of its wings is critical. Their profile generates the lift while their shape and position determine the stability of the flight.

In this way, the interactive relationship between sensory-motor apparatus, morphological properties, environment and control is of prime importance. This relationship was first observed and characterized by Pfeifer as the morphology and control tradeoff (Pfeifer and Scheier, 2001), but the mechanisms underlying this relationship have been unclear. The fact that simple physical interactions give rise to computation indicates the theoretical possibility for the dynamics of morphology to play a computational role in the system, and thereby to subsume part of the role of control (Paul et al., 2004).

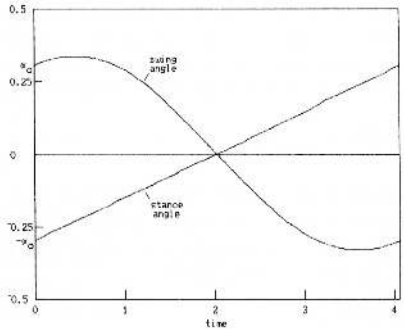
2.2.2 Passive and semi-passive walking

The role of morphology in robot biped locomotion has been particularly explored through the research on passive dynamic walkers (Wisse et al., 2007).

Biped walking on slightly inclined planes appeared as toys in the early twentieth century. Their legs are straight and they rock from side to side to allow feet to lift off



(a) The simplest of walking models is the synthetic wheel, a biped with straight legs and semi-circular feet



(b) The stance leg rolls forward steadily like a spoke in a wheel, while the free leg swings ahead like a pendulum. Support is transferred between legs when their speeds and angles match.

Fig. 2.2.: The cycle of a passive walker is naturally stable and will repeat continuously, thus synthesizing the motion of an ordinary wheel (McGeer, 1992).

the ground. The analysis of the behaviour of such systems, purely passive, is much more recent. Indeed, an advantage of such an object is its low energy consumption. The energy supplied to the system comes from the variation of potential energy due to the slope. It compensates for the energy lost during impact.

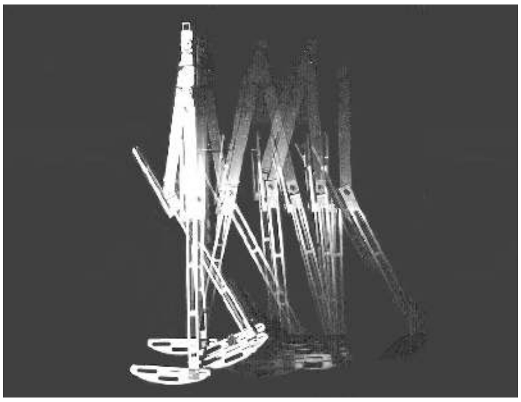
The unipodal transfer movement is similar to a passive pendulum and arises from the correct combination of an initial pulse and coupled gravity-inertial effects. The behaviour of the walker is therefore an inverted pendulum.

In the early 90s, Tad McGeer, coming from an aeronautic background, formalizes the idea of a compass biped with free articulation by the concept of the synthetic wheel (McGeer, 1990). The dynamics of the system are formalized by an equation of motion linearized close to an average vertical position of the legs, and an equation of shock for foot/ground contact modelling energy dissipation (McGeer, 1992).

The tuning of initial conditions conducive to passive movement is performed numerically, after one step, the robot should be back to its initial state. This model allows the robot to obtain a completely passive, cyclical walking gait (without motorization) , which is stable on a plane, slightly inclined by a few degrees. The potential energy gained during the descent exactly compensates for the energy dissipated during impact.

Passive walkers

Tad McGeer has also showed that passive walking can be obtained on a bi-articulated robot (McGeer, 1992). Appropriately shaped feet and a judicious mass distribution



(a)Tad McGeer with his prototypes

(b)Passive walker robot

Fig. 2.3.: Passive walker robots are just composed by mechanical elements, there is no controller nor motors, yet thanks to a clever mass repartition and feet shape, they demonstrate a very human-like and stable biped walking motion.

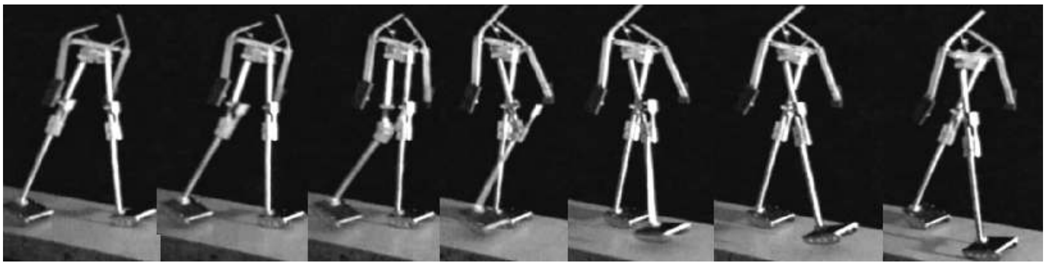


Fig. 2.4.: While fully passive walker are limited to 2D walking, the Cornell walker robot created by Steve Collins (Collins and Ruina, 2005) has demonstrated the 3D bipedal walking ability thanks to the addition of low-power actuation and arms.

generate a footstep combining a forward pendulum swinging movement on its stance leg and a swing with spontaneous flexion of the transferred leg. To make this motion possible, a device must prevent the leg from bending during the stance phase. The dynamic behaviour is mainly determined by three dimensionless parameters: the length ratio, the mass ratio and the slope of the planar support (Garcia et al., 1998).

Semi-passive walkers

Passive robots are limited to walking on inclined ground, they cannot have a passive torso and they are locally stable robots as the domain of attraction for limit cycles is small.

Thus the work on passive dynamic walkers has been pursued with the appearance of semi-passive walkers combining both specific passive properties and low power actuation to increase their robustness (Anderson et al., 2005). We can note the work of Collins (Collins and Ruina, 2005) which explored the case of a semi-passive 3D biped robot. Its morphology is based on a particular mass distribution, knee locking, round feet and springs on the legs to generate an efficient walking gait while keeping its lateral and frontal balance. The concept of a 3D semi-passive robot has been pushed even further with the realization of a complete humanoid robot with torso, arms and head: the robot Denise (Wisse, 2005) and Flame presented in (Hobbelen et al., 2008) created by the Delft university.

2.2.3 Emergence of complex behaviours

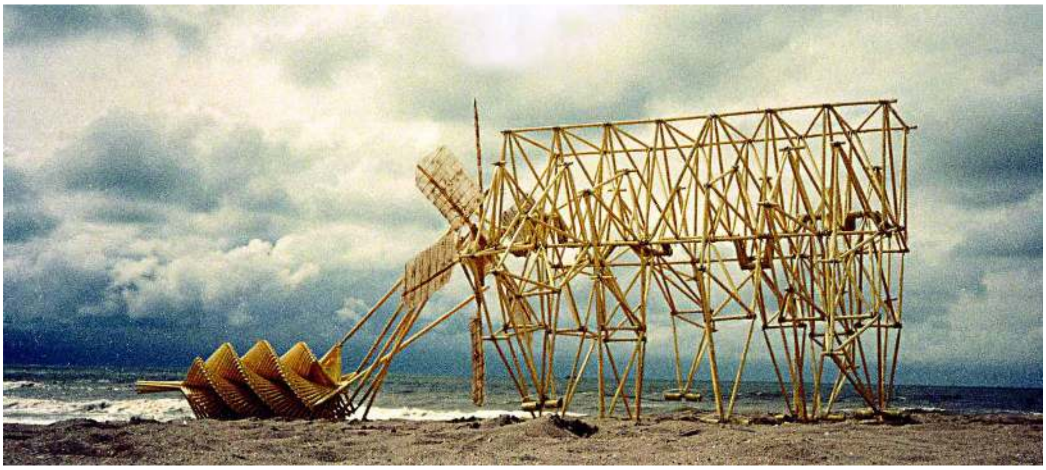
Finding the rules that can lead to a desired behaviour is more difficult than explaining a real complex behaviour we can observe when an agent is interacting with its environment. Since the behaviour itself cannot be preprogramed but is always the result of an agent-environment interaction, we must design for emergence rather than directly for a specific behaviour (Pfeifer and Bongard, 2006). It is called the design of emergence (Steels, 1990). It remains an open question how this can be done systematically. At the moment, design for emergence is rather an art than a real engineering discipline.

It is precisely in the artistic field that we find one of the most fascinating examples. Theo Jansen is a kinematic sculptor. This artist plays on the border of several fields, between engineering, research and art, is the designer of the sand beasts (see Fig. 2.5).

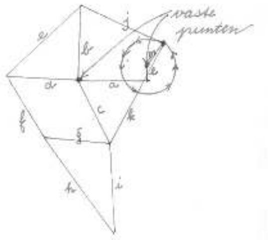
These giant structures move using a really clever mechanism composed of eleven rods, the lengths of which have been tuned by numerical optimization. This system produces a walking motion(see Fig. 2.5c) with a center of rotation always remaining at the same level. For this reason Theo Jansen likes to say he "reinvented the wheel" but adapted to the environmental niche of his creatures .i.e. the beach.

Since the beginning of this work, Theo Jansen has created dozens of creatures that are increasingly evolved. However, the basic mechanism remains the same, both simple because it is composed of only one degree of freedom and complex because the length ratio between the eleven rods is critical and must be equal to specific numbers. Thus, using only really basic material, electric plastic tubes, Theo Jansen created multi-legged creatures capable of moving in the sand, powered by the wind (Jansen, 2007).

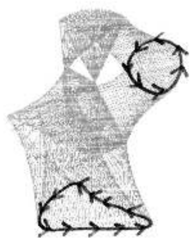
The evolution of his work, led to several improvements. In this video: `http://youtu.be/rWbU3eV4ZpQ`, 72 legs move at the same time using one crank. But he also extended the mechanism to add independence of sorts. For instance, he added lemonade bottles to store energy. These bottles are used as pressure tanks



(a) One of the Theo Jansen's creature "living" on the beach



(b) Leg mechanism



(c) Actual trajectory

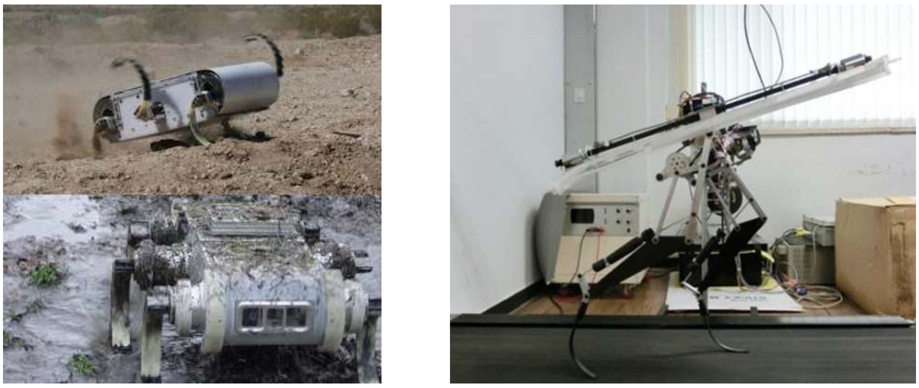
(d) Actual leg are based on basic plastic rods.

Fig. 2.5.: Based on a very clever mechanism, the Theo Jansen's creature can walk robustly on terrain as complex as the beach. Moreover, he managed to add sophisticated behaviours such as avoiding the water, changing direction and store energy. As in the passive walkers robot, there is no control or motor, these behavior are only emerging from the actual morphology and the interaction with the environment.

filled using pumps powered by the wind. Beasts can utilize this stored energy in case the wind fades away.

Also, a natural enemy of these beasts is the sea. Using the same basic material, Theo Jansen created sensors able to detect water and reverse the way beasts move. The same principle allows also these beasts to avoid obstacles.

Thus the work of Theo Jansen goes beyond kinematic art and is really instructive for robotic and IA research fields. Indeed, thanks to a specific morphology adapted to their environmental niche, his creatures are able to act autonomously and "survive" in the real world. No computation, no abstraction, the apparent intelligence of these creatures comes only from a direct interaction between their particular morphologies and the environment. Based on low cost materials, yet clever mechanisms, his work is a meaningful proof of concept showing how the morphology of an agent can lead to the creation of complex behaviour like those we could call "intelligent".



(a) The Rhex robot is a compliant hexapod able to move quickly and robustly over complex terrain.

(b) Raptor has both compliant actuators and feet. It is currently the fastest bipedal robot.

Fig. 2.6.: Example of robots those impressive behaviours were achieved thanks to a compliant morphology.

2.2.4 Compliant robotics

Compliance describes the stiffness of a system. It is mostly used in robotics to describe how an actuator or a mechanical part reacts to external forces when trying to reach a position. The real output of a compliant actuator will be modified by its interaction with the environment while a rigid actuator will force the output to be the desired one. Compliant actuators can be obtained with several kind of actuators (e.g. impedance control (Park, 2001), hydraulic (Alfayad et al., 2011), SEA (Pratt and Williamson, 1995)). Compliance can also be achieved by the use of soft materials for the mechanical structure (also called soft robotics), whereby the link or the shape of the robot can be deformed by its interaction with the environment.

Several projects have already shown the importance of an adequately compliant morphology in achieving complex sensory-motor behaviour such as legged locomotion in complex environments with relatively little control. This is illustrated by the quadruped Big Dog those compliance relies on hydraulic actuators (Raibert et al., 2008) as well as the RHex robot (Saranli et al., 2001) using six compliant legs. Both demonstrate impressive adaptability and crossing behaviour over rough-terrain.

Also certain humanoid robots have shown the importance of a compliant structure for human robot interaction. For example the compliant structure of the vertebral column and legs of Acroban (Ly et al. (2011), Oudeyer et al. (2011)) was shown to permit a self-organized physical human-robot interface allowing non-expert users to lead the robot by the hand.

Furthermore, it has been shown that the compliance of the body explains the dynamics of walking and running (Geyer et al. (2006), Iida et al. (2007),) and appears to be a key-feature toward the achievement of the fastest current robots both for quadrupedal (DARPA Cheetah robot[1]) and bipedal running (see Raptor robot[2]).

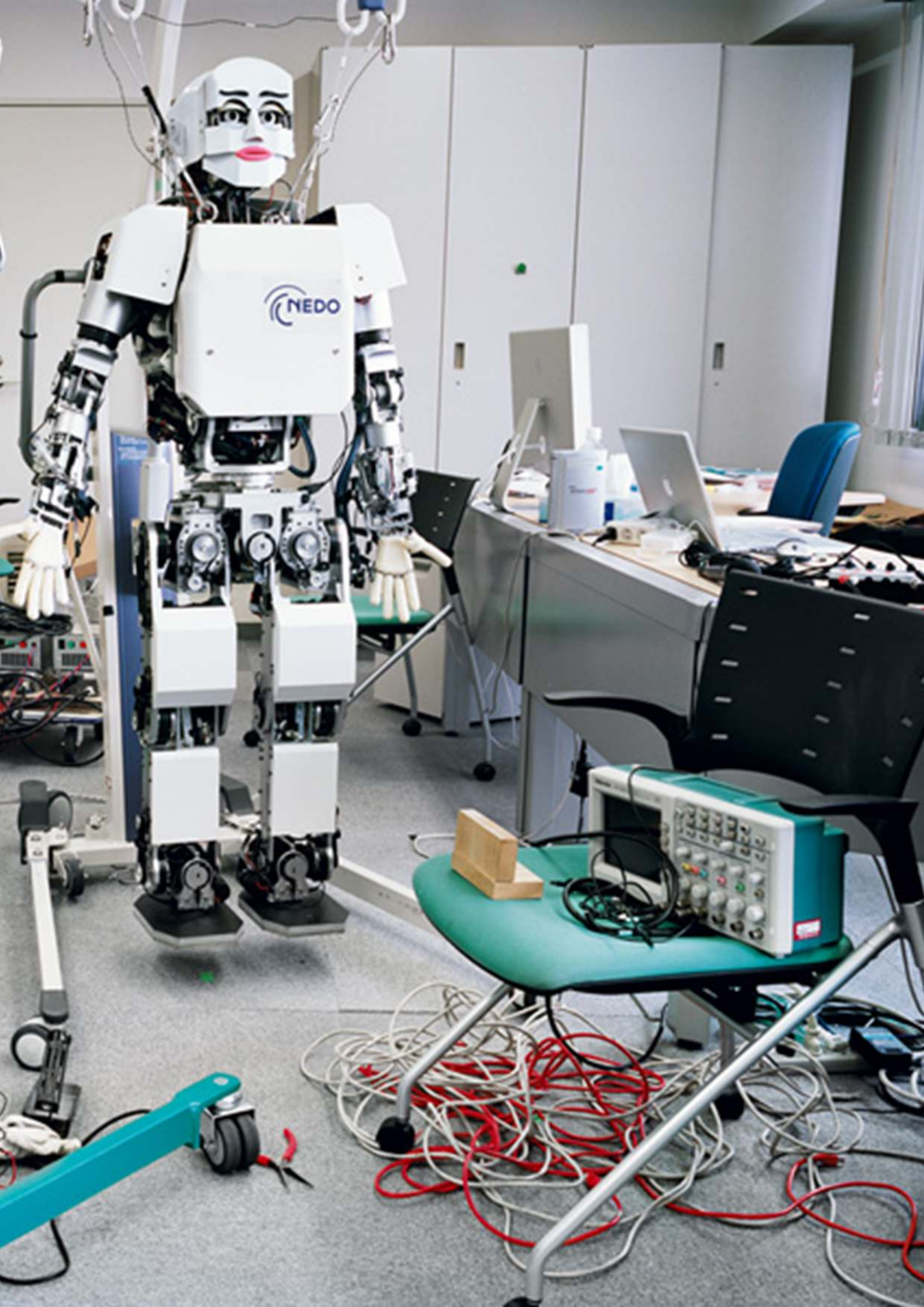
2.3 Conclusion

Researchers have shown the interest of having a well-adapted morphology, by exploring several aspects such as morphological computation, passive dynamics or compliance with this goal in mind. In addition, we could complete this review about the role of morphology with the ecological balance principle (Pfeifer et al., 2005), bio-inspiration (Scarfogliero et al. (2009), Pfeifer et al. (2007)) or the coupling of adequate morphologies with central-pattern generators that has been shown to generate robust locomotor behaviour (Ijspeert et al. (2007), Steingrube et al. (2010)).

The work presented in this chapter shows the recent awareness in the robotics field of the importance of morphology. It appears increasingly clear that the achievement of robust robotic requires an understanding of the interaction between robot morphology, control and environment.

However, as shown in the diversity of projects, the role of morphology is still a research field open to exploration. For this purpose, an abstract robot is not sufficient and it is necessary to have a real robotic platform to experiment on. This will be the subject of the next chapter, in which we will present an overview of the current robotics platforms.

[1] Darpa cheetah video: `http://youtu.be/d2D71CveQwo`
[2] Raptor robot video: `http://youtu.be/lPEg83vF_Tw`


NEDO

Review of experimental methods

3

> *The world is its own best model.*
>
> — **Rodney Brooks**

3.1 Introduction

As we showed in the previous chapter, an interesting evolution over the last decades in the robotics field has been the demonstration of the importance of robot morphology and its impact on cognition and control. This opens new horizons toward the achievement of more adapted and robust behaviour in open-ended environments with unpredictable interaction.

However as Rodney Brooks explained, exploring interaction between morphology, cognition and environment requires real world experimentation. Indeed embodied artificial intelligence (Pfeifer et al., 2007) needs to act in the real world in order for complex behaviour to emerge. The real world includes a large number of constraints such as inertia, multi-point physical contacts, friction, multi-body dynamic and impacts which are complicated to realistically model without involving considerable engineering resources. Moreover, if one is interested in exploring robotic behaviour in an open-ended environment with human interaction, managing the unpredictability will be limited to a small subset of cases.

From another angle, "The world is its own best model" and if we can use simulation for exploring well-defined concepts, the exploration of emergent complex behaviours based on interaction between robot self-dynamics and the environment appears easier and less expensive if conducted directly in the real world. Moreover, as Luc Steels explained (Steels, 1990), actual behaviour emerges from the interaction between the controller, morphology and the environment. Some could argue that adding actual morphology and an ecological niche add unnecessary complexities to a problem we already have difficulty solving, even just in terms of the control. Yet it appears some behaviour cannot be achieved without real physical complexity. Brilliant examples are the passive/semi-passive walking robots (presented in section 2.2.2). These robots are technically rather simple, just composed of a mechanical structure with the proper link size and foot shape. The establishment of a model should be rather easy but their real dynamic is very difficult to simulate on a classical physical simulator. Indeed, all complex physical effects (e.g. shocks, friction, inertia) contribute to the achievement of passive walking.

One of the best state of the art works in this domain was achieved by Delft University with the different passive and semi passive walkers they built (Wisse, 2005). Desiring to explore biped locomotion, we had a discussion with Martijn Wisse on simulation strategies for adding semi-passive abilities to Poppy. Here is his testimony about their attempt to make their robot walk in simulation after managing to create the real one:

> Even after obtaining a successful walking motion, we did not manage to create a simulation that walked successfully using the same controller parameters. We tried very hard with some of the best people, but we didn't succeed. The reason was, I think, that our type of control (using the emergent behaviour of a set of simple reflex-like controllers) was highly sensitive to hardware effects like friction. Normally, one uses a local joint controller to make the joint follow a desired trajectory independent of the exact amount of friction. The local controller "abstracts these hardware effects away", if you know what I mean. This makes the behaviour of the whole system quite predictable. However, in our robots, we did not have this kind of abstraction as we were not following trajectories, and thus a little bit of extra friction has an effect on the entire motion.
>
> We did spend a long time making a high-fidelity model in Adams, and also using other methods, but eventually we gave up without success.
>
> Martijn Wisse - Associate professor at Delft University of Technology

Thus results obtained in simulators are difficult to transpose on a real platform and vice-versa. One of the main reasons seems to be the complexity of realistically handling non-perfect components e.g. non-linear friction in joints, feet/ground reactions and so on. Yet the interesting contributions of robot morphology on its behaviour are precisely those we currently have difficulties modelling correctly.

This raises a major limitation for the reproducibility of results in the scientific community. While it is rather simple to transfer experiments conducted in a simulator by sharing the software material, sharing real world experiments is far more challenging.

In this chapter, we will review the current state of the art of robotic platforms, in particular how they are made and how the results can be demonstrated or transferred in the scientific community. Yet there are many robotics platforms, from robot arms (Jako, LWR, Kuka) to wheeled platforms (Pioneer 3-AT, P3-DX) or even submarines (AQUA2). In this thesis we are particularly interested in the exploration of morphology for locomotion and interaction so we will restrain the platform review to the ones actually exploring particular morphologies and humanoid platforms.

3.2 Platform exploring the role of morphology

As we discussed in the previous chapter, several robots have been made to explore the role of morphology, each exploring particular aspects of this challenge.

3.2.1 Bio-inspired robot

The ECCE ROBOT (Marques et al., 2010) investigates the role of morphology for cognition and human-robot interaction through a bio-inspired and compliant anthropometric design, which copies the inner structures and mechanisms (bones, joints, muscles, and tendons). Thanks to a design based on polymorph mechanical structures and wire-driven actuation, they managed to produce a really complex structure mimicking both the mobility of the human upper body and the intrinsic compliant properties of the human muscular system. While polymorph[1] is a convenient material to easily create custom shapes by hand, the diffusion of robots based on this technology is limited because this manual process cannot be reproduced outside the lab by someone else.

The Kojiro robot (Mizuuchi et al., 2007) also involves a bio-inspired morphology, but unlike the ECCE1, it is a complete humanoid robot with an advanced leg design. The project aims to show the musculoskeletal humanoid's advantages by involving many degrees of freedom and sensors, a multi-articulated spine and compliance. Moreover, it implements the concept of modular reinforceable muscles Mizuuchi et al. (2004), which means the actuation required can be explored by changing the muscle unit. Each muscle unit consists of a DC motor with a gearhead, a pulley, a tension sensor using strain gauges, a thermometer, a sensor-amplifier circuit board, and a rounded outer shell. However, while the robot seems promising for exploring both locomotion and human-robot interaction thanks to its advanced musculoskeletal system and compliant actuation, the data permitting the duplication of Kojiro has not been distributed and the structure of the robot appears really complex, with numerous components.

[1] Polymorph is a thermoplastic polymer which melts at 62°C and consists of small off-white plastic granules. By heating these granules in hot water the user can easily melt the pellets to form a transparent flexible material. Once melted the opaque white pellets fuse together, become transparent and soften, allowing the user to form the plastic by hand into unique shapes.

3.2.2 Passive dynamic walkers

There are numerous passive and semi-passive dynamic walkers. We can indeed cite the one from Tad McGeer (McGeer, 1990), the work of Steve Collins (Collins et al., 2001) and Russ Tedrake (Tedrake et al., 2005), the robots made in Delft University Denise (Wisse, 2005) and Flame (Hobbelen et al., 2008). Also more recently a passive walking robot designed and built by the Nagoya Institute of Technology (Japan) walked non-stop for 13 hours and 45 minutes on a treadmill, completing some 100,650 steps over a distance of around 15.2 km. All these robots demonstrate impressive results and show the interest of using clever morphologies for the achievement of tasks as complex as bipedal walking.

However these robots are really difficult to transfer and reproduce in the robotics community.

Firstly, their mechanical structure have mechanical parts that are either handcrafted or produced with classical machining techniques based on milling or casting metal alloys. These techniques require specific upfront tooling, which makes the production of a small batch really expensive.

Secondly, while the control of this robot is rather simple, the mechanical design is far more subtle and requires expertise few people in the robotics community have. Unfortunately, the descriptions we can find in the associated papers are limited to theoretical models. It is necessary but not sufficient. Again the talk we had with Matijn Wisse is really representative of the way passive and semi-passive robot are created:

> We never actually produced a high-fidelity simulation. We made very simple simulations only. From them, we learned how to tune parameters. Then, we designed the real robots, without running full-blown optimizations. Rather, we used our intuition for a large number of decisions on design trade-offs, using lessons from the simple simulations combined with other limitations such as available motors etcetera. Then, we (again) used our intuition and large amount of experience to tune the robot's controllers, and make design improvements, until it walked.
>
> Martijn Wisse - Associate professor at Delft University of Technology

Thus there is a big step between the model and the achievement of a functional semi-passive robot. Indeed the engineering design of such a platform has a strong impact on the achievement of passive dynamic walkers.

(a)ATRON

(b)Roombots

Fig. 3.1.: Modular robots

It is a problem for the diffusion of such an idea as the laboratory desiring to explore passive principles has to take the risk of spending time developing a passive robot without any guarantee it will succeed in finding the appropriate tuning.

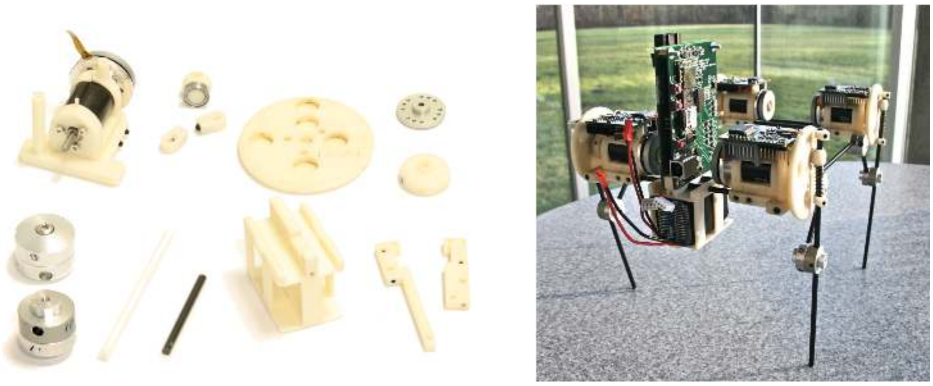
3.2.3 Modular robotics

Despite the numerous robotic platform developed, there are only a few whose hardware design can be completely reconfigured.

We can find some modular robots (Murata and Kurokawa, 2007) examples such as Molecubes (Zykov et al., 2007), M-Tran (Murata et al., 2002), Superbot (Salemi et al., 2006), ATRON (Jorgensen et al., 2004) or Roombots (Sproewitz et al., 2009). They are independent robot modules, which can be assembled together to create various robot form and applications (see Fig. 3.1). However, these kinds of robot are not suitable for exploring morphological properties or creating humanoid robots.

Actually, to our best knowledge there is only one example of a modular kit that makes real exploration of the role of morphology possible. The Locomorph project (Moeckel et al., 2013) offers a multi-purpose hardware kit called LocoKit (Larsen et al., 2012)). This kit uses carbon-fiber rods assembled with Locokit parts (see Fig. 3.2a). It allows to quickly create robots and study the impact of several morphological properties such as link length, joint stiffness or mass distribution (see Fig. 3.2). Also it permits to add spring over rods to create a linear damping system and therefore add compliance to robots. The kit is distributed for €2500 and includes the components needed to create a quadruped robot (see Fig. 3.2b).

It appears to be the only existing solution allowing both the exploration of the role of morphology and the transfer of results between laboratories. While being very interesting, it is also limited asthe robot created must be rod-based so multi-body



(a) (b)

Fig. 3.2.

articulations such as those we can find in human leg or torsos seem complicated to produce.

3.3 Commercial humanoid platforms

Robotic prototype platforms appear to share the same issue as regards the reproducibility of science. Most of them are constructed using classical manufacturing techniques and involve a complex morphology, which makes them expensive to reproduce both in term of human resource and material cost. Finally, they are in most cases not open source and no material is shared that allows others to reproduce them easily.

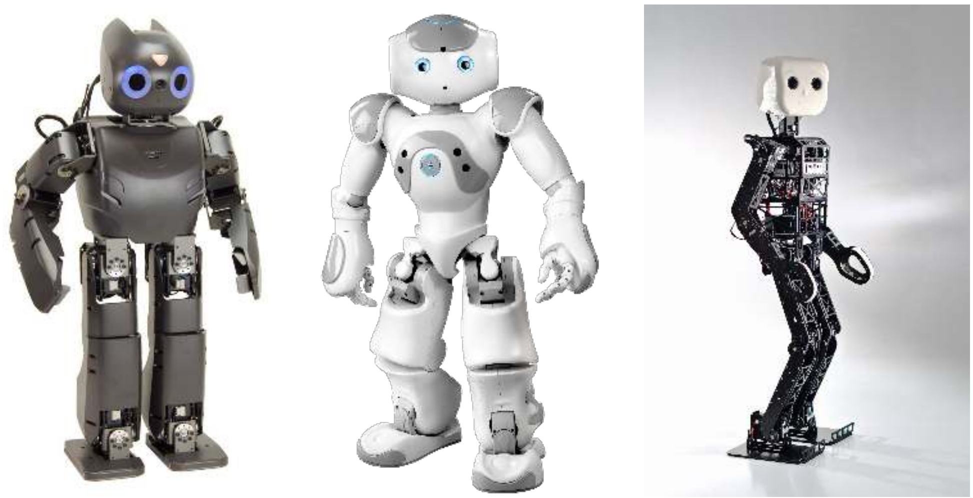
The use of commercial platforms could be an alternative as they are easily available, and have a constant and reproducible morphology.

3.3.1 Advanced research platforms

The two most famous humanoid robots are iCub (Metta et al., 2008) and HRP-2. These robots involve very advanced technologies:

ICub is an open source[2] robot measuring about 100 cm in height for an overall weight of 22 Kg. It has 53 degrees of freedom designed specifically for manipulation and locomotion, and powered by high-ended actuators based on a harmonic drive reduction system and brushless frameless motors (Natale et al., 2013). In addition, in the majority of cases torque is transmitted from the motors to the joints using steel tendons routed in complex ways via idle pulleys.

[2]The hardware design, software and documentation are released under the GPL license.



(a)Darmin Op (b)Nao (c)Nimbro OP

Fig. 3.3.: Three humanoid robots made by KumoTek (USA).

HRP-2 is a complete humanoid 150 cm in height weighing 58 kg and comprising 30 DoFs. It also involves high-end actuators with harmonic drives and cooling systems installed in both the computer and actuator drive systems to improve temperature control, yet contrary to the ICub robot, they made the choice to have the robot as stiff as possible.

Both robots involve a very complex and advanced design, which has required the work of dozens of engineers, also their production techniques make them very expensive (i.e. more than €200,000). Therefore, these robots do not permit to freely change their morphologies. Moreover, their high cost and their fragility limit the risk researchers can take in exploring behaviour in the real world.

3.3.2 affordable platform

On the other hand, there are small and affordable commercial humanoids platforms, that are easily accessible and easy to use such as Nao Gouaillier et al. (2008), Darwin Op Ha et al. (2011), Nimbro Op Schwarz et al. (2012) or iCub Metta et al. (2008).

DARwIn Op is an open source humanoid research platform created by the Romela lab at Virginia tech (Ha et al., 2011) and distributed by Robotis for about $12,000 (see Fig. 3.3a). It is 45cm high and weighs 2.9kg, and has 20 Dynamixel MX-28 actuators (6 for each leg, 3 for each arm, and 2 for the neck). Its mechanical structure is composed of aluminium parts.

Nao is 55cm high, 5.2kg and 25DoFs humanoid robot with a plastic mechanical structure (ABS, PA, XCF) (see Fig. 3.3b). It is a very famous robot, a few thousands units of which have been sold to labs and universities. Its cost was around €12,000, but recently it was halved.

Nimbro-OP is a tall humanoid measuring 95cm in height and weighing 6.6 kg, it has 20 powerful MX-106 and MX-64 actuators (6 per leg, 3 per arm and 2 in the neck). It costs $20,000 and its structure is based on an aluminium and carbon composite (see Fig. 3.3c).

Yet, they provide a "traditional" morphology (e.g. limited compliance, rigid torso, large feet, over actuation) not really appropriate for exploring the interesting morphological properties we showed in the chapter 2. Also as they use classic manufacturing techniques such as metal milling and plastic casting, the modification of their morphologies is made difficult.

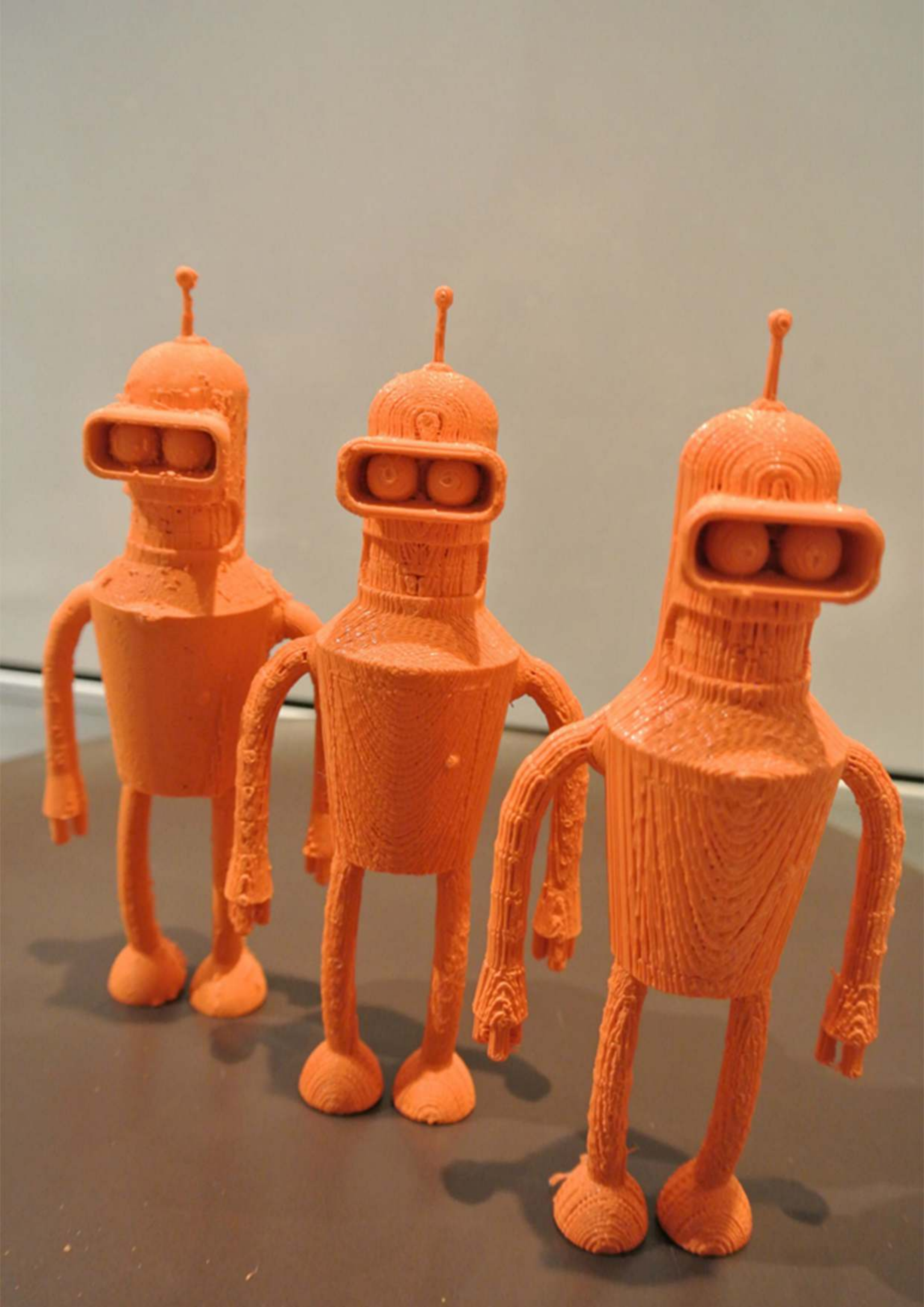
3.4 Conclusion

In the previous chapter, we presented several work showing the importance of robot morphology and the need to continue research in this domain. As we explained in section 3.1, this research area requires areal robotic platform to explore.

There are many robotic platforms, we could have a more exhaustive description, yet the main objective of this review was to show an overview of the landscape of possibility with the current robotic platforms. In particular that we have on one hand, some prototype robotic platforms designed for specifically exploring one aspect of robotic morphology but whose design methodology strongly limits their reproduction in the robotics community, mainly because they are not open source and produced with expensive techniques. And on the other hand, commercial robots that are easily accessible so the results should be reproducible from one lab to another. Yet the design method used also involves classical production techniques, therefore modifying their morphology would be too expensive and time consuming.

Finally, the current research practices in the Robotics field limit diffusion and the impact of contributions. Indeed, in most cases, there is no material associated with a published paper. This means only the theory is shared with the community but not the actual framework allowing for the results to be reproduced.

In the next chapter, we will present novel production techniques and modes of diffusion, which can solve both problems, exploring the role of morphology and reproducibility between research labs.



4 The open hardware and 3D printing revolution

4.1 Introduction

With the democratization of personal computers and the development of the internet, computer science and related applications have seen a great expansion. Open source software played a major role, indeed most of the web servers are running on the Linux operating system and Apache, while open source software like WordPress, Qt, Firefox, VLC and so on have permitted the realization of a wide variety of applications in daily life (Peeling and Satchell, 2001).

However, while copying and sharing bits of software is virtually free and can often be run on any computer, producing the atoms of a real object both has a potentially high cost and requires expert tooling. Thus the production of mechanic or electronic hardware components is limited to two options: either it is handcrafted or mass produced. Also the step between the handcrafted prototype and mass production is so large that only big companies can achieve it. Conventional manufacturing processes require the production of specific tools, the programming of complex machines, the human intervention to put the part along the different tool and so on, most of the costs are in the up-front tooling, and the more complicated a product is, the more it costs. Thus, most companies would not accept to run a whole production process just for a few units and if they accept the cost will be so high than most prototypes never find a way of reaching people outside the workshop where they had been created. So until now, production in small or medium series has been extremely difficult to achieve because it has not been profitable. Only big companies have been able to raise enough money to produce new hardware, meaning niche products and personalization are left aside.

Over the last few years, a novel evolution has begun that is going to completely change the current rules of production and distribution of goods. This evolution is acting on both technological and political/societal front, which tends to confirm the claims of those who argue it will be the next industrial revolution (Anderson, 2012). Indeed, new rapid prototyping technologies are emerging and make production cheap, fast and easy to anyone. At the same time the associated machines and tools are diffused under open source hardware licenses, acting as an unexpected

(a)Nylon (b)Steel (c)Odd Guitar

Fig. 4.1.: Some examples of 3Dprinted parts, which were until now impossible to produce. In addition, it can be done with a large range of material from wax to titanium.

lever arm toward the diffusion of these technologies to anyone. All these radical changes are contextualized by the phenomenon of *makers* and the exponential growth of associative production spaces (e.g Fab Lab, Makerspace, Tech Shop or Hackerspace).

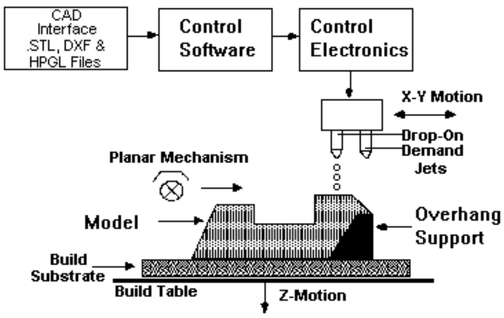
In this chapter, we will first present and discuss ground-breaking 3D printing technology. Then we will present the open source hardware movement and how its interaction with 3D printing is changing the current production paradigm.

4.2 The 3D printing revolution

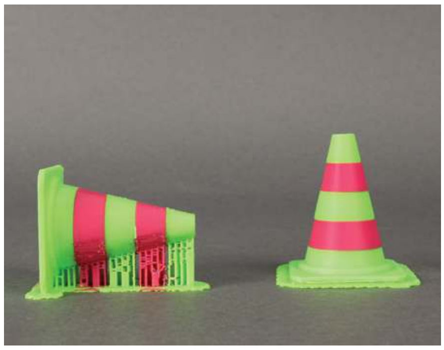
Prototyping is an essential step in a product development and manufacturing cycle. It allows for the form and the functionalities of a new product to be tested before large investment in tooling for mass production. Until the last decade, prototypes were largely handmade by skilled craftsmen, adding weeks or months to the product development time.

The term 3D printing encompasses various processes for making a three-dimensional object from a digital model, primarily through additive processes in which successive layers of material are laid down under computer control. Accurate parts can be produced right from a digital model in few hours and with minimum handling tasks. Consequently, errors are minimized and product development costs and lead times substantially reduced. It has been claimed that rapid prototyping can cut new product costs by up to 70% and the time to market them by 90% (Waterman and Dickens, 1994).

Recent progress in the 3D printing process mean 3D printing can be considered not only as a way to produce prototypes but also as an actual production technique. Indeed the layered method of assembly allows intricate designs - geometries, which



(a)FDM process



(b)FDM result example

Fig. 4.2.: The FDM technique relies on melting and selectively depositing a thin filament of thermoplastic polymer (ABS - PLA) in a cross-hatching fashion to form each layer of the part.

are either impossible or too expensive to achieve with conventional metal casting (see Fig. 4.1).

In this section we detail the different 3D printing techniques available with pros and cons, then we will discuss the expected changes in the industrial sector.

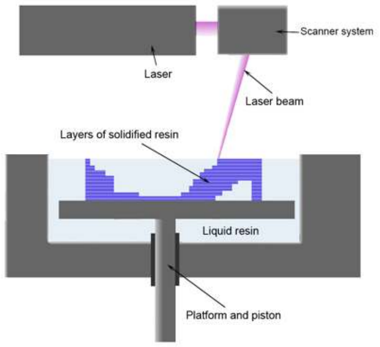
4.2.1 Multiple techniques

The term 3D printing encompasses several different additive processes.

Fused Deposition Modeling (FDM)

The FDM technique relies on melting and selectively depositing a thin filament of thermoplastic polymer (ABS - PLA) in a cross-hatching fashion to form each layer of the part. The material is in the form of a wire supplied in sealed spools, which is mounted on the machine, and the wire is threaded through the FDM head. The head moves in the horizontal X and Y directions to produce each layer through zigzag movements. The supporting table moves in the vertical direction and is lowered after the completion of each layer (see Fig. 4.2a).

It is a technique that is really low-cost and friendly to the office environment but is limited by its slowness on dense parts, the need for supports and its lack of precision for detail, thin walls and surface finish (see Fig. 4.2b).



(a)SLA process

(b)Some parts made with a SLA printer

Fig. 4.3.: The stereo-lithography technique: a UV laser beam polymerizes the top layer of a photosensitive resin in a horizontal direction (X/Y) while the platform moves in a vertical direction.

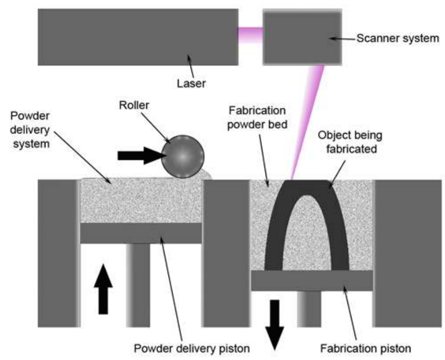
Stereo-Lithography (SLA)

This technique relies on a photosensitive monomer resin, which forms a polymer and solidifies when exposed to ultraviolet (UV) light. The UV laser beam moves in a horizontal direction and is focused on the top layer to polymerize the photosensitive resin (see Fig. 4.3a). Due to the absorption and scattering of the beam this reaction only takes place near the surface. Then the cured layer of polymer is lowered by the platform (Z axis) so that a fresh layer of liquid resin covers the part.

This technique has several advantages, such as a good surface finish (see Fig. 4.3b), and is capable of producing highly detailed parts with thin walls, but is limited by material (only photo polymers) and support structures are always needed which can be difficult to remove.

Selective Laser Sintering (SLS)

The Selective Laser Sintering process uses a high-power (25-50W) CO2 laser beam that melts and fuses finely powdered material spread on a layer. Before the powder is sintered, the entire bed is heated to just below the melting point of the material in order to minimize thermal distortion and facilitate fusion with the previous layer. While the laser moves on the horizontal plane, the platforms move along the vertical axis -through a distance corresponding to the layer thickness (usually 0.01 mm) - and a counter-rotating roller spreads a precise amount of fresh powder above the sintered layer. The unsintered powder serves as the support for overhanging portions, if there are any in the subsequent layers.



(a) SLS process

(b)

Fig. 4.4.: The Selective Laser Sintering process uses a high-power laser beam that melts and fuses finely powdered material spread on a layer.

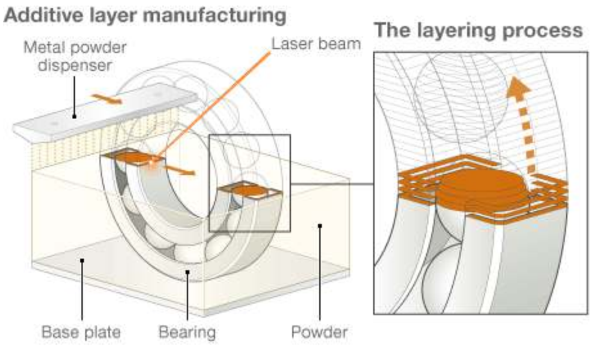
This technique has a really interesting advantage: it is compatible with different materials (Polyamide, Alumide), morevore it does not require support and mechanical properties of parts are homogeneous (the same in any direction). However, this technique requires some handling to extract extra powder from the part and SLS 3D printers are very expensive (+$50K).

Direct Metal Laser Sintering (DMLS)

Direct Metal Laser Sintering is very similar to the SLS process but uses a high-powered 200 watt Yb-fiber optic laser in order to fuse metal powder into a solid part by melting it locally (see Fig. 4.5a). Parts are built up additively layer-by-layer, typically using layers 20 micrometres thick. This process allows for highly complex geometries to be created directly from the 3D CAD data, fully automatically, in hours and without any tooling. DMLS is a net-shape process, producing parts with high accuracy and detailed resolution, good surface quality and excellent mechanical properties (see Fig. 4.5b). However, this is obviously the most expensive technique.

4.2.2 Major impact expected in days to come

The 3D printing is revolutionizing the way people can produce parts. On one hand, it brings very affordable numeric production tools to a wide-audience. Using these tools, communities have emerged with Fablab, makerspace or hackerspace, where people can create together new project involving specific and home-made hardware components. As computers have allowed people to produce in their garage new software or websites and eventually develop a novel e-economy, the 3D printing open the landscape of possibilities for hobbyists to innovate by developing novel hardware



(a)DMLS part


(b)DMLS part

Fig. 4.5.: Direct Metal Laser Sintering is very similar to the SLS process but uses a high-powered 200 watt Yb-fiber optic laser in order to fuse metal powder into a solid part by melting it locally.

prototypes and niche products at a fraction of the cost of classical manufacturing techniques.

On the other hand, these techniques also open new horizons for the industry. At The University of Southern California, Professor Behrokh Khoshnevis has built a robot equipped with a nozzle that spews out concrete. This colossal 3D printer can build a house in 24 hours and therefore dramatically reduces its cost.

Another example is the SuperDraco engine (see Fig. 4.6a), this rocket engine has its combustion chamber 3D-printed by direct metal laser sintering (DMLS). The regeneratively-cooled combustion chamber is made of inconel; a family of nickel-chromium alloy that is noteworthy for its high strength and toughness.

> Through 3D printing, robust and high-performing engine parts can be created at a fraction of the cost and time of traditional manufacturing methods. SpaceX is pushing the boundaries of what additive manufacturing can do in the 21st century, ultimately making our vehicles more efficient, reliable and robust than ever before.
>
> Elon Musk, SpaceX CEO/CTO and Tesla Motors CEO.

Thus parts for vehicles can be optimized to be lighter and - simultaneously - incredibly robust (see Fig. 4.6b). The AMAZE project has been able to print airplane wing sections as well as jet engine parts, and the ESA hopes to one day print a satellite as one piece:

(a)The SuperDraco rocket engine has a combustion chamber: 3D-printed - SpaceX.

(b)A conventional hinge is seen in the background and a 3D-printed metal hinge is seen in the foreground - EADS.

Fig. 4.6.: Thanks to its special properties, 3D printing is hitting the industry

> This novel technology offers many advantages. 3D printing, formally known as additive manufacturing, can create complex shapes that are impossible to manufacture with traditional casting and machining techniques. Little to no material is wasted and cutting the number of steps in a manufacturing chain offers enormous cost benefits.
>
> European Space Agency (ESA)

4.2.3 Conclusion

3D printers open new horizons as they are able to produce parts which were, until now, either not possible or extremely expensive to produce using classical techniques while adding several key abilities:

- **Accessible:** 3D printed parts can be obtained everywhere, either by personal printing or by using an online service[1].
- **Low cost:** from tens of cents if produced on personal printer to tens of euros if outsourced through web services. Also the cost is not proportional to the part's complexity, meaning designers are free to explore the shape they want with almost no constraints.
- **Fast:** Production takes only a few hours from scratch and does not require any specific upfront tooling.

[1]examples: i.materialise, shapeways or sclupteo

- **Skill-free:** while the production process is fully digital, few or no specialist skills are required.

- **Multi-material, precise and robust:** the current 3D printers can create precise (up to 0.1mm) parts from different materials such as Polyamide, PLA, ABS and even titanium or flexible material. The obtained parts are robust and can often be used as final parts for several years.

- **Reduces the number of parts:** 3D printing permits the printing of complex parts and even assembled parts as complex as bearings or gearboxes. This means we can replace multiple parts that have to be assembled by a single ready-to-use part right after production.

4.3 The open hardware movement

The concept of "open-source hardware" or "open hardware" is not as well known or widespread as the free software or open-source software concepts yet. However, it shares the same principles: anyone should be able to see the source (the design documentation in case of hardware), study it, modify it and share it.

4.3.1 Open hardware in the industrial history.

In the 18th century London and Lyon (France) were two majors silk manufacturing towns. Because London was on the way to taking the lead, Lyon tried a new and original policy for innovation. They decided to freely diffuse new techniques. Inventors were invited to the city hall to present their innovations publicly. They were then rewarded a first time for the presentation, and a second time when the innovation was actually implemented on Lyon machines. This policy was followed by decades of cumulative inventions such as: perforated paper tape (1725 B. Bouchon), punched card programming (1728, J-B. Falcon), the Jacquard loom (1801, J-M. Jacquard), and the sewing machine (1829, B.Thimonnier). Meanwhile, silk production in London was governed by patents, techniques were kept secret and monopolized by theirs inventors (Alain, 1997).

The impact of these two opposed political choices turned out largely in Lyon's favour. In 1815, Lyon had 14,500 looms and London 12,000. But in 1853, Lyon had 60,000 looms while London fell to only 5,000 and became a silk importer.

Lyon stimulated inventions and disseminated innovation: looms became programmable, order processing and production agile, parts were standardized, counter-tops parts appeared, services grew, the Lyon loom park was up to date and operational. In

London, the industry was controlled by investors, customers had to order large series, the choice available decreased, waiting periods were longer, the artisans became employees, wages fell and the state of the London looms park deteriorated.

The Lyon policy created a win-win ecosystem creating both job opportunities and advanced technology. Thanks to this choice, the city took the lead over London.

4.3.2 Definition of Open Hardware

The Open Source Hardware Association (OSHWA) aims to be the voice of the open hardware community. It promotes the use and development of open source hardware for education and economic development, to collect, compile and publish data on the open source movement and organize the movement around shared principles.

Also the Open Source Hardware Association defines[2] open source hardware as:

> *Hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.*

Fig. 4.7.: The open source hardware logo

4.3.3 Open source hardware licenses

The Open Source Hardware Association definition is not enough, a legal framework is needed to both protect and promote open hardware projects. This is the role of open source licenses which will be discussed in this section.

In general, there are two broad classes of open-source licenses: copyleft and permissive. Copyleft licenses (sometimes referred to as "viral") are those that require derivative works to be released under the same license as the original; common copyleft licenses include the GNU General Public License (GPL) and the Creative Commons Attribution-ShareAlike license. Other copyleft licenses have been specifi-

[2] Complete definition available on `http://www.oshwa.org/definition/`.

cally designed for hardware; they include the CERN Open Hardware License (OHL) and the TAPR Open Hardware License (OHL). Permissive licenses are those that allow for proprietary (closed) derivatives; they include the FreeBSD license, the MIT license, and the Creative Commons Attribution license

Creative Commons licenses

Founded in 2001, Creative Commons is a non-profit organization that enables the sharing and use of creativity and knowledge through free legal tools. They provide free and understandable licenses standardizing the way to share and use creative work. Thanks to the use of several modules, which can be combined, the Creative Commons licenses permit the creator to modify his copyright terms to best suit his needs. First intended for artistic and cultural content such as music and writing, the Creative Commons are now used also to share open source hardware files.

Fig. 4.8.: Creative Commons logo

The Creative Commons licenses are based on four major condition modules:

Attribution (BY) : requiring attribution to the original author.

Non Commercial (NC) : requiring the work to not be used for commercial purposes.

No Derivative works (ND) : allowing only the original work, without derivatives

Share Alike (SA) : allowing derivative works under the same or a similar license (later or jurisdiction version).

The combination of these modules leads to six licenses (see Fig. 4.9) but related to open hardware, only two of them are considered as open source following the OSHW definition:

Attribution CC BY People can distribute, remix, tweak and build upon the licensed work, even commercially, as long as they credit the authors of the original creation.

Attribution-ShareAlike CC BY-SA People can distribute, remix, tweak and build upon the licensed work, even commercially, as long as they credit the authors and license their new creations under identical terms. **This license is often compared to the "copyleft" free and open source software licenses.**

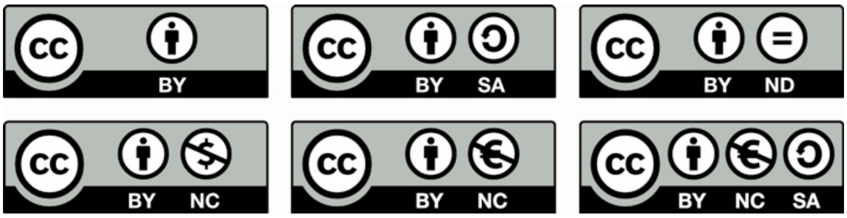


Fig. 4.9.: The combination of the 4 Creative Commons modules give 6 licenses allowing creators to choose how they want to share their work and how "open" they are.

CERN OHL

Inspired by the open source software movement, the Open Hardware Repository[3] was created to enable hardware developers to share the results of their R&D activities. The recently published (March 2013) CERN Open Hardware License offers the legal framework to support this knowledge and technology exchange.

The CERN–OHL is to hardware what the General Public Licence (GPL) is to software. It defines the conditions under which a licensee will be able to use or modify the licensed material and is compliant with the OSHWA definition criteria. In the spirit of knowledge sharing and dissemination, the CERN Open Hardware Licence (CERN OHL) governs the use, copying, modification and distribution of hardware design documentation, and the manufacture and distribution of products[4].

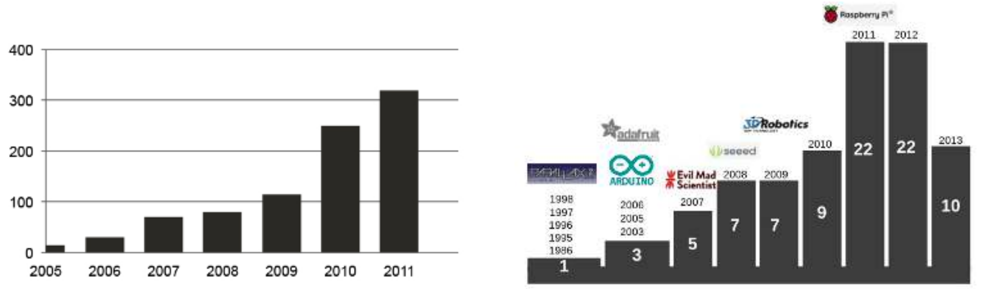
TAPR Open Hardware License (OHL)

Specifically designed for open hardware, and avoids the issues other licenses have with focusing on copyright protection of documentation instead of the right to make, distribute, or use a product based on that documentation[5]. It requires that all derived works use the same license and include before and after documentation if any changes were made.

4.3.4 Some famous open hardware projects

Based on the open source hardware philosophy, several companies and projects have been created over the past ten years (see Fig. 4.10).

[3] `http://www.ohwr.org/`
[4] License details available on `http://www.ohwr.org/projects/cernohl/wiki`
[5] The full text of the TAPR Open Hardware License (OHL) is available here: `http://www.tapr.org/ohl.html`



(a) Creation of new open hardware projects per year between 2005 and 2011.

(b) Start-up creation based on open hardware distribution

Fig. 4.10.: Evolution of the open source hardware movement in the past decades. Graph extracted from *HOPE 2010 - How to run an open source hardware company*

Several kinds of object have began to have an open source version, even the most advanced ones such as laptops (Novena project[6]), reflex camera (OpenReflex[7]) or even cars (LocalMotors[8], OSVehicle[9]).

Arduino

One of the most meaningful open hardware project is Arduino. Massimo Banzi was a teacher from the Interaction Design Institute Ivrea in Italy. His students were using *BASIC Stamp*[10] for a cost of $100 and often complained they couldn't find an inexpensive, powerful microcontroller to drive their arty robotic projects.

In 2005 Banzi and David Cuartielles, a Spanish microchip engineer, decided to design their own board. The Arduino project aimed to offer an affordable and easy to use electronics board for a student-friendly price: $30. The first wiring design was done during the PhD thesis of Hernando Barragan (Barragán, 2004) and the software by another student: David Cuartielles. After the wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open source community.

Fig. 4.11.: The Arduino logo

The Arduino story was one of the first hardware projects with a real desire to promote innovation through open source, so to make it work they had to find an appropriate licensing solution that could apply to their board. After some investigation, they

[6] https://www.crowdsupply.com/kosagi/novena-open-laptop
[7] http://www.instructables.com/id/3D-Printed-Camera-OpenReflex/
[8] https://localmotors.com/vehicles/
[9] http://www.osvehicle.com/
[10] A BASIC Stamp module is a single-board computer that runs the Parallax PBASIC language interpreter in its microcontroller.

realized that if they simply looked at their project differently, (i.e. considering the source files as documentation[11]), they could use a license from Creative Commons normally used for cultural works such as music and writing.

They define their work as:

> Arduino is a platform for prototyping interactive objects using electronics. It consists of both hardware and software: a circuit board that can be purchased at low cost or assembled from freely-available plans; and an open-source development environment and library for writing code to control the board. Arduino comes from a philosophy of learning by doing and strives to make it easy to work directly with the medium of interactivity. It extends the principles of open source to the realm of hardware, supporting a community of people working with and extending the platform. It has been used in universities around the world and in numerous works of interactive art.
>
> Mellis (Mellis et al., 2007)

By 2006, Arduino has sold 5,000 boards, the next year 30,000. Following the idea of open source collaboration the community eventually grows until 100,000 people and thousand of side projects and derivatives emerged. In 2013 Arduino has registered over 700,000 official boards, but has estimated that there is at least one derivative or clone board per every official one.

Today, Arduino is a very successful project with more than 1,000,000 boards sold and a wide range of low cost electronics boards[12]. Moreover, there are now dozens of open-hardware orientied companies building new products on top of Arduino environment, the most famous ones being Sparkfun[13] and Adafruit[14].

Shapeoko

Designed by Edward Ford, Shapeoko (see Fig. 4.12) is a simple, low cost ($685) and open source (CC BY-SA) CNC milling machine. It is based on two other open hardware projects: MakerSlide[15] for linear motion and an Arduino board for the control.

[11] "*You could think of hardware as piece of culture you want to share with other people*", Banzi.
[12] http://arduino.cc/en/Main/Products
[13] `http://www.sparkfun.com/`
[14] `http://www.adafruit.com/`
[15] Open a source linear bearing system under Creative Common BY-SA licenses: `http://makerslide.com/`

Fig. 4.12.: The shapeoko v2 is a low cost and open source CNC 2.5 axes.

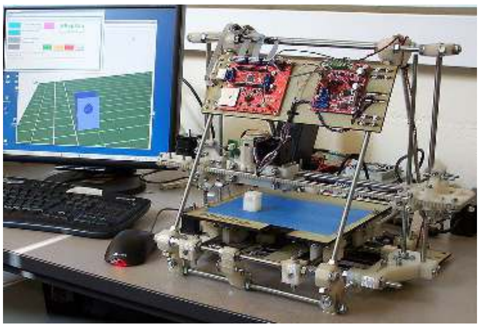
RepRap

The RepRap project started in 2005 and based on the Fab@Home[16] principles, developed a multi-purpose open source 3D printer using Fused Deposition Modeling, a technique with the particularity of being largely self-replicating:

> RepRap is an open-source, self-replicating, rapid prototyping machine. It is a robot that uses fused-filament fabrication1 to make engineering components and other products from a variety of thermoplastic polymers. RepRap has been designed to be able automatically to print out a significant fraction of its own parts. All its remaining parts have been selected to be standard engineering materials and components available cheaply worldwide. As the machine is free and open-source anyone may – without royalty payments – make any number of copies of it ether for themselves or for others, using RepRap machines themselves to reproduce those copies.
>
> Jones et al. (2011)

Distributed under GNU General Public License, the RepRap (see Fig. 4.13a) was one of the first low-cost 3D printers. Thanks to the the fact that it is genuinely collaborative, this project generated such a large number of variations and interpretations that is difficult to even count. One of them is the famous Makerbot Replicator (see Fig. 4.13b). Now Makerbot is one of the major worldwide general public 3D printer distributors.

[16] http://www.fabathome.org/



(a)RepRap v2

(b)MakerBot Replicator v1

Fig. 4.13.

4.4 Conclusion

The emergence of new and accessible rapid prototyping techniques changes ways of producing things. Because it is now quick, simple and cheap to make things, it opens the realm of possibility of sharing hardware because anyone can produce it.

The 3D printing and open source hardware are complementary. Indeed, open source projects are more relevant when, as the software, people contributing can build the project at home. In this way 3D printing can act as a lever arms for open hardware projects while open hardware projects are meaningful to widespread the use of 3D printers.

Also, Arduino has shown how useful can be open hardware projects and opened new perspectives allowing non-expert people to easily create interactive objects. While ones could think open sourcing is dangerous for economic viability, Arduino demonstrated that it actually generates new kind of economy, derivative contributing to the development of the community rather than stealing commercial parts.

Following this example, Tesla Motor have recently released open source all their patents about electric cars. Elon Musk[17] is genuine inventor, yet also a very good business man. This act is not a philanthropic idea, it will actually permit Tesla to develop its economy. Indeed, they are the only active actor on the electric car market. To increase the market, more actor are needed. By releasing their patents, Tesla hope to create dynamic favorable to the expansion of their market. By this way, they target to have a good part of a big market rather than having a big part of a small market.

Therefore, using 3D printing technique and open hardware distribution model appear to be promising for the future of technology and economic development.

[17] SpaceX CEO/CTO and Tesla Motors CEO

Part II

The Poppy project


R. STAHL
STUTTGART
WANDERER

5 Motivations and Methodology

5.1 Introduction

In chapter 2, we discussed the emergence of a novel paradigm in the field of robotics that appeared in the late eighties. Embodied artificial intelligence rejects the symbolic approach and postulates that it is not possible to have intelligence without an actual robot body associated with its ecological niche (Pfeifer and Scheier, 2001). Following this paradigm, several researchers have tried to tackle challenges in which the classical cognitivist approach failed (see (Brooks, 1986)) e.g. the understanding of natural forms of intelligence that require direct interaction with the real world.

Thus, an interesting evolution over recent decades is the demonstration of the importance of the morphology for sensorimotor control, cognition and development (Kaplan and Oudeyer (2008), Steels and Brooks (1995), Pfeifer and Bongard (2006)), which can be defined as follows:

> The morphology of a robot thus refers to the physical structure and form of a robot. Specifically, the focus is on characteristics such as link sizes, number of links, joint characteristics, mass distribution, actuator characteristics, material properties, sensor characteristics and sensor placements. In short, any characteristic that defines the physical structure of the robot is included in the term morphology. Chandana Paul (Paul, 2006)

Exploring the interaction between body properties and cognition could lead both to a better understanding of animals' behaviour (human beings in particular) and to build robots that are more adapted and robust to an open environment with unpredictable interactions. In particular, we can highlight the acquisition of sensorimotor tasks and the exploration of adapted bodies for natural, physical and social interactions with humans.

In this context, we should not only pay attention to the robot body design but **introduce morphology as an experimental variable and conduct experiments in the real world.** As Rodney Brooks said *the world is its own best model* (Brooks, 1991) and simulators cannot handle the complexity of real physics with multi-point contacts, soft materials and frictions. This is especially true for complex dynamic tasks such as physical interaction or legged locomotion.

Following the definition of robotic morphology given by C. Paul, we need to find a framework allowing easy and quick tuning of morphological parameters on an actual robot in order to explore and hopefully find new ways of improving robot behaviour in the real world. However considering morphology as an experimental variable raised two major problems:

- **how can we obtain an experimental robotic platform with both a morphology that can be changed easily and quickly and the capacity to act robustly in the real world?**

- **how can we make sure this platform, particularly the hardware, can be diffused and reused in the research community?**

In the next sections of this chapter, we will **suggest novel approaches and design processes to create and produce robotic platforms, the control and morphology of which can be freely explored through experimentation in the real world, that are easy to diffuse and reproduce in the research community.** We will detail the methodological and design challenges involved in creating robots with variable and modular hardware. Then we will present the design methods we chose to address these challenges and those we have used to create Poppy (see chapter 6). And finally, we will discuss the importance of open source distribution for creating open and cumulative science.

5.2 Challenges

The role of morphology appears to be a fascinating open field of research but until now it has been under-explored. We presented in chapter 3, a review of robotic platforms, both commercial and lab prototypes. It appears the current platforms are not suitable to tackle these challenges.

Firstly, for most the electronic and mechanical structures are produced using classic manufacturing techniques, which makes them too complicated and expensive to modify. Indeed, the classical way of designing and producing robots is a complicated, time-consuming and expensive process. The development of current robotic platforms requires dozens of engineers working for many years and significant fundraising for production. Such techniques make creating variant parts impossible, mainly because of the approach and technologies chosen to design and produce them.

Secondly, beyond the restriction on exploring morphological variants, one of the fundamental aspects of the scientific research is to demonstrate facts, which should therefore be reproducible. Unfortunately in the robotics field, the amount of material resources and the techniques involved makes it difficult to transfer robot platforms from one lab to another. While commercial robots can be easily accessible (subject to appropriate funding) because they are relatively mass-produced, lab prototypes are mainly handcrafted and specifically tuned, which makes their reproduction in another lab impossible. Therefore, scientific validation is limited and researchers cannot build novel work upon the one of their peers.

Finally, robot hardware of both types is very rarely open source, which simply prohibits any modification and reuse of the work (we will discuss more in detail the importance of this point in section 5.4).

Therefore, allowing experimental platforms to be transferred and shared is a way of ensuring the scientific validity of experiments, and also of promoting and accelerating scientific research by reducing time lost in development, instead concentrating research resources on the exploration of novel ideas.

In this context, creating a platform reproducible everywhere without special tooling or skills, the morphology of which can be freely explored, raises methodological and design process challenges that we will describe in the following points.

5.2.1 Make the morphology variable

Current robotic platforms, in particular humanoid ones, have mechanical parts either handcrafted or produced with classic machining techniques based on milling or casting metal alloys or plastic. These techniques require specific upfront tooling which make the production of a small batch really expensive. Also, to keep the cost of the robot rather low, mass production is needed to achieve economy of scale. In this context, the morphology of current robotic platforms cannot be modified because it would require redoing most of the production process. In addition, the design of such mechanical parts is limited because the manufacturing process implies constraints and the complexity of a part greatly increases its cost. The same issues appear with electronics and the robot sensor space which is, in most cases, frozen. Thus the classic way to design and produce robot is not adapted to the free exploration of the robot morphology, novel design and production paradigm have to be used.

5.2.2 Create reproducible robot prototypes

Most labs must reinvent the wheel by developing whole new robotic platforms even though functional setups have already been developeded by other laboratories and should/could be reused.

For example, several interesting robotic platforms explore key aspects of robot morphology. We can cite Kenshiro (Nakanishi et al., 2013) which has a complex and bio-inspired artificial muscles actuator network, or semi-passive walkers such as Denise (Wisse, 2005), demonstrating impressive walking ability with little control and power actuation. Unfortunately, none of these robots can be and have ever been transferred to another lab. Indeed even if they were open source theirs productions require specific tooling and hand tuning that only few skilled people have.

Therefore some constraints have to be applied on hardware platforms to make them reproducible:

Precision, stationary Experiments should be repeatable, implying that the robot's morphological properties should be stationary. This means that robot performance should not be dependent on where it was built or the users' skills.

Easy and fast to duplicate: In order for the platform to be reusable, it needs to be easy and fast to duplicate and not rely on specific tooling or exotic components.

Affordable: To ensure widespread use, a key aspect is to keep the cost of the platform relatively low. The more labs involved, the better the scientific impact.

5.2.3 Keep robotic platform simple and easy-to-use

The field of robotics is intrinsically multidisciplinary. A robot itself requires technology from mechanics, electronics and computer science, but the scientific impact of robotics can be far larger and reach non-engineering fields such as humanities, social or biological sciences. Thus, robotics is a specialised field in which nobody can be an expert in all required skills. We have to take into account the fact that the end user can certainly be expert in one specific subfield but a beginner in others. This mean that for each subfield, the designed robot has to be simple enough to be understood and used by beginners while having, at the same time, enough potential to not constrain users in their domain of expertise.

5.3 The chosen design methods

To address these challenges, we suggest exploring an alternative design methodology that is driven by the desire to:

- freely explore morphological properties,
- reduce the amount of time required between an idea and its experimentation on an actual robotic platform in the real world,
- make our work easily reproducible in any other lab,
- keep our work modular and free to use in accordance with open source principles, so it can be reused and extended for other projects.

To reach these goals, we decided to follow some design methods for both design and production, for all technological aspects of the robot (i.e. mechanics, actuation, electronics, software, distribution).

5.3.1 3D-printed mechanical parts

We could envisage simply having classical mechanical parts that are reconfigurable and adjustable, allowing for example, the exploration of different lengths of a link or different centre of mass positions. However, this limits the morphological exploration to only a few dimensions with a limited range.

As we discussed in chapter 4, over the last few years, novel techniques, especially 3D printing have been revolutionizing the way we can produce objects. 3D printers not only open new horizons for the production of mechanical part, they are able to produce parts that were either not possible or extremely expensive with classical techniques (see Fig. 5.1).It completely changes the paradigm associated with production. Indeed the cost does not change with the quantity or the complexity, meaning designers are free to explore the shape they want with almost no constraints.

Also, these novel techniques come with the open hardware and makers revolution, which has brought low cost 3D printers into the home. The production of mechanical parts can be now done in few hours directly on site with limited human handling (see Fig. 5.2).

3D printers have several key abilities:

(a) Example of object those shape would be impossible to produce without additive manufacturing

(b) 3D printed metal heat exchanger

Fig. 5.1.: Example of complex parts those production has been made possible by 3D printing techniques.

Worldwide: 3D printed parts can be obtained everywhere, either by personal printing or by ordering parts on other web services, such as i.materialise, shapeways or sclupteo.

Low cost: The cost of producing 3D parts is rather low, ranging from tens of cents if produced on a personal printer to tens of euros if ordered though a web service.

Fast: In a couple of hours a whole part can be created from scratch. When using web services, queuing and shipping delays have to be added, increasing the production time to several days.

Skill-free: Since the production process is fully digital, few or no specialist skills are required.

Multi-material, precise and robust: current 3D printers can create precise (up to 0.1mm) parts in different material such as nylon, PLA, ABS or even titanium and flexible material. The parts obtained are robust and can be used as final parts for several years.

Reduce the number of part: 3D printing can be used to print complex parts and even assembled parts as complex as bearings or gearboxes. This means we can replace multiple parts that have to be assembled into ready-to-use ones.

These properties of the 3D printing process allow for the first time to really explore morphological variants of mechanical parts. Indeed, it is now fast and low-cost to create alternative designs. Associated with modular architecture, we can easily and

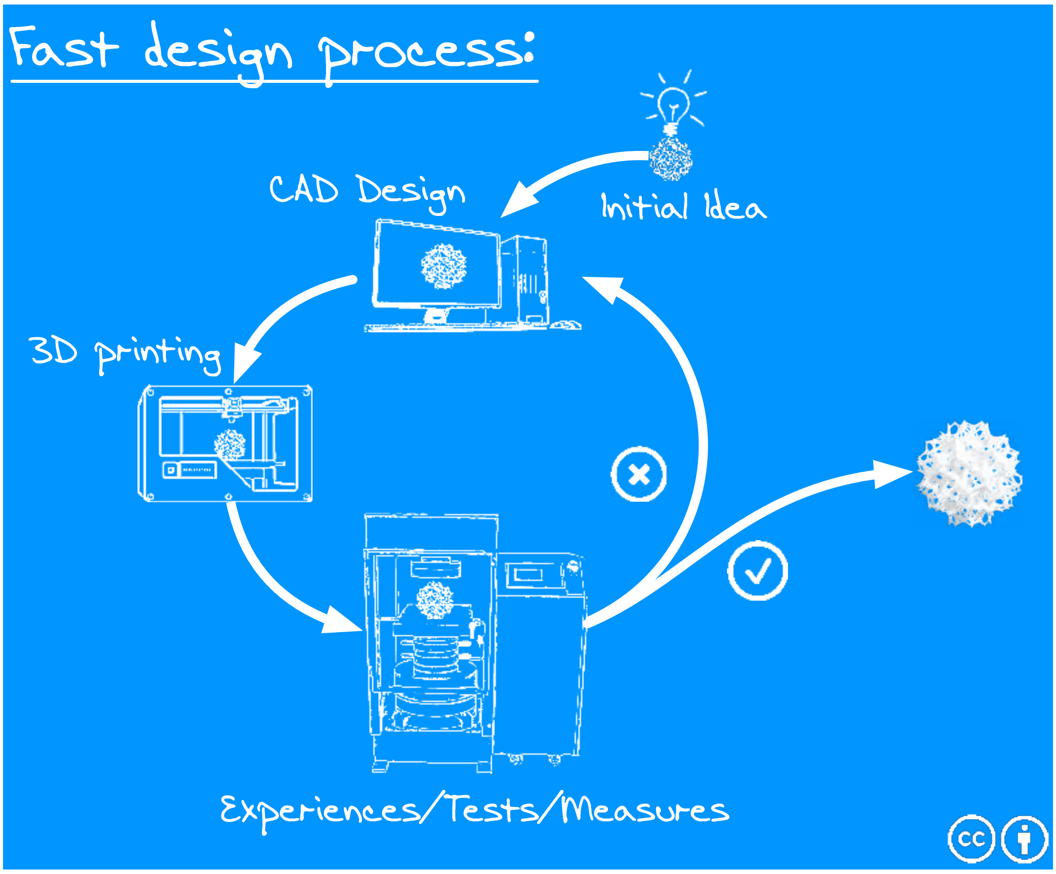


Fig. 5.2.: The 3D printing technique allows for fast iterative conception loop while it permits to directly and automaticcaly produce a prototype on site for a fraction of the cost of classical manufacturing techniques.

quickly change robot parts and conduct experiments. Also this process is compatible with our diffusion goals since it is simple, and accessible anywhere with an internet connection and a mailing address. Also parts can be produced directly in the lab if it is equiped with a 3D printer.

5.3.2 Electronic architecture based on Arduino

Thanks to 3D printing, exploring morphological variants of mechanical parts is now easier than ever before but unfortunately, the printing of electronic components and boards is not yet available. However, exploring the role of morphology does not only concern the mechanical properties but also the sensors apparatus i.e. **which sensor is used and where is it placed on the body**. The Swiss Robots (Maris and Boeckhorst, 1996) is a great example of the impact of the sensors' positioning on robot behaviour[1].

[1]Swiss Robots are wheeled robots composed by two motors and two distance sensors. If we set 3 or 4 of them in an area with randomly distributed cubes, they will eventually shuffle most of the cubes into two or three clusters with few pushed against walls. This behavior depends only on the

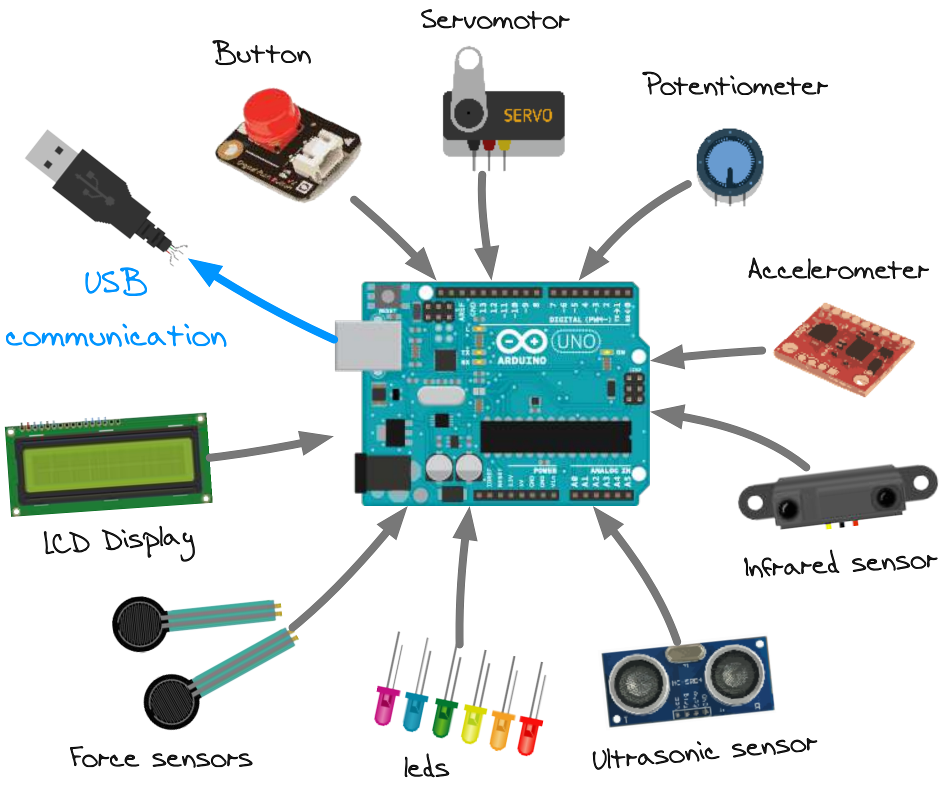


Fig. 5.3.: The use of Arduino as electronic architecture allows for sensors to be easily added and/or changed, while keeping the same electronic board. In addition, it permits to add expressive components such as LEDs, LCD or sound systems, allowing users to easily explore human-robot interaction.

To permit the exploration of sensor-system variants, we suggest basing the electronic architecture on Arduino. As presented in section 4.3.4, Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is intended for anyone doing interactive projects. The Arduino board can sense the environment by receiving inputs from a wide variety of sensors, and affects its surroundings by controlling lights, motors, and other actuators. Low-level embedded programing skills are not required, since Arduino boards can be programmed using the Arduino programming language[2] which abstracts all the complexity.

The Arduino community is very active and expanding, more and more sensors are designed to be directly plugged onto Arduino boards. Thus, using Arduino adds modularity to robot electronic architecture, allowing the reconfiguration of the sensors space by easily adding new ones (see Fig. 5.3).

position of the distance sensors. If they are placed on the front, robots will avoid cubes, if placed on the side, robots will clean the room and create cluster of cubes.

[2] http://arduino.cc/en/Main/Software

5.3.3 All-in-one actuators

Robots are actuated using various techniques from classic and cheap servomotors to the highly powerful and dynamic hydraulic actuators powering the Atlas humanoid robot. While some actuator technologies such as Series Elastic Actuator (SEA), cable-driven or artificial muscles are really promising to create more robust and efficient robots, they are still work-in-progress solutions and require advanced skills both to assemble and use. These technologies are not yet compatible with the creation of diffusible and reproducible robotic platforms in a multidisciplinary research community.

To permit the diffusion, we need off-the-shelf and stationary solutions, easy to assemble, easy-to-use and available anywhere. Also, to allow the exploration of morphology, actuators have to be modular and allow the tuning of several parameters.

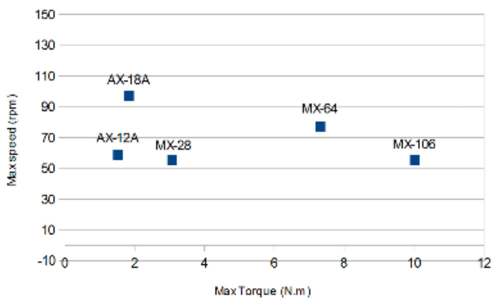
We therefore chose to use Robotis Dynamixel servo-motors[3] for robot actuation (see Fig. 5.4a). Dynamixel motors are easily accessible, as they are mass produced and shipped worldwide. Also they are commonly used actuators in the robotic field and many robots are powered by them, including Darwin-OP (Ha et al., 2011), Myon (Hild et al., 2012) , Acroban (Ly et al., 2011) or Nimbro (Schwarz et al., 2012).

The Dynamixel motors are not simple servomotors, they are all-in-one-modules that contain drivers, encoders and communication buses. They are also quite powerful, robust and rather precise. This is achieved by the combination of Maxon motors, metal gearbox and precise magnetic rotation sensor (resolution: 0.1^o). They embed a 32bits micro-controller dedicated to communication (serial port TTL or RS232), the control of the joint (position, speed or torque) and the measurement of internal data such as the real position, speed, load or temperature. They also allow tuning the internal PID or limitation of the maximal torque. This enables rich behaviour, useful both for physical interaction and locomotion.

Different models are available and permit the adjustment of the actuation to the power required by the joint (see Fig. 5.4b). They are different in size and power but their API remains the same and we can easily switch from one to another without changing the code or the electronic integration. Yet, even if the size changes, the footprint keeps the same pattern (see Fig. 5.5) which make easy-to-configure parametric mechanical parts, it just takes a couple of minute to transform a part designed to be compatible with Dynamixel MX-28 to one compatible with Dynamixel MX-64.

[3] http://www.robotis.com/xe/dynamixel_en

(a)Robotis Dynamixel AX and MX series



(b)Power of each Dynamixel model

Fig. 5.4.: The Robotis Dynamixel come with different models from low cost ones such as AX-12/18 to the most powerful MX-series with maxon motor and magnetic encoder.

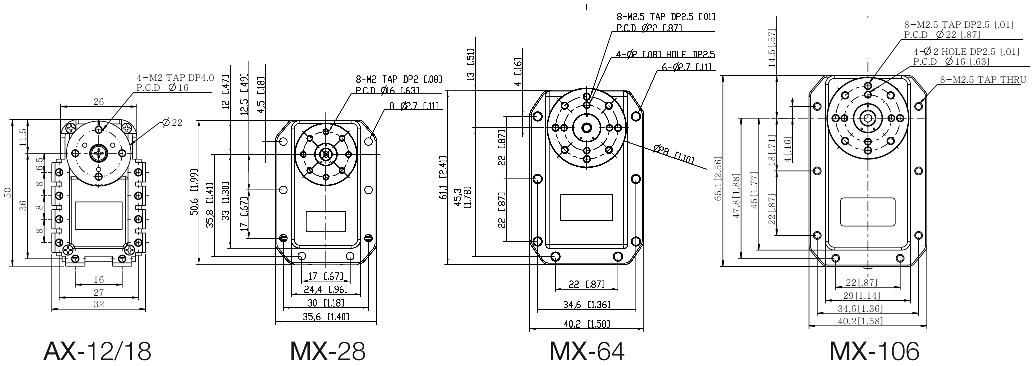


Fig. 5.5.: The footprint of Dynamixel motors keeps the same pattern, only the dimensions are increased following the power of the motor. Thus, switching from one model to another only requires changing the dimension and not the design of a part. With parametric CAD software such as as Solidworks, it takes a couple of minutes to modify a mechanical part to be compatible with another Dynamixel motor.

5.3.4 Accessible and extensible software

Having variables in software is more classical, yet using a non-stationary robot hardware (i.e. those link length or sensorimotor space can change) requires to have control library adapted to this low-level modularity. Here the choice has been made in favour of ease-of-use and modularity. We designed sensory-motor control API adapted to the hardware variability we have. We choose to use Python as the main programming language as it allows fast development, easy deployment on all operating systems and quick scripting by non-necessary expert developers. It also offers a large variety of scientific and machine-learning libraries used in robotics (e.g. Numpy, Scipy, Scikit-learn). This language is rather slow compared to C or Java, but sensorimotor control is done using serial bus communication and as the serial communication is handled through the standard library we can still achieve rather high performance.

5.3.5 Open source distribution

Finally, while the main aspect of such an approach is to allow variability, reuse and modification of the initial design, it is necessary to not only diffuse our work through scientific publications but also to distribute the material needed. This means anyone outside the Flowers lab should have access to the actual source files and be free to make any changes suitable to their own research. Therefore in addition to the technological choices previously presented, we decided to distribute all our work (both software and hardware) under open source licenses. This is an essential step toward building new research tools that facilitate both scientific validation and cumulative science in robotics. We will discuss this in detail in the next section.

5.4 Allowing cumulative and Open science

As we explained previously, new design approaches and methods should be used to create robots with morphology that can be explored by the user. In addition, by choosing the relevant technologies, we can permit both the easy exploration of morphological variants, and the transfer and exchange between research laboratories.

To head in this direction, an unfettered access to knowledge and the components associated (articles, data, software, materials, methods) is needed. Also, it is preferable that work can be built upon without asking permission and where the methodology is increasingly based on open collaboration.

A very well adapted tool is the open-source license, which allows the source code, blueprint or design to be used, modified and/or shared under defined terms and conditions. The terms and conditions are defined by several different licenses and the author can choose among them the one that best suits the level of freedom with which he wants to distribute his work. These licenses are famous and widespread in software development and have started to be used for hardware over the last few years (see chapter REF).

Nevertheless, in Science the preferred distribution channel is still primarily based on paper publishing and only a few researchers distribute their work under open source license. It is surprising as the use of open source collaboration seems very desirable for scientific research, especially in the robotic field:

Scientific validation : Similarly to publishing detailed mathematical proofs, sharing materials associated with a robotics experiment permits serious peer-reviewing, fundamental for the scientific validation of our field. Indeed, robotics experimentation involves a large amount of material (both software and hardware), reviewers should be able to evaluate if the material and experimental setup are coherent with the results submitted.

Open Science: We often use only one part of the data collected in an experiment. The open distribution of all material allows the reuse of experiments by other researchers, who can use the same data to extract alternatives or extend the initial results. It also permits access to all details and especially to the constant parameter tuning, very sensible for a number of algorithms.

Cumulative Science Most of the time, only a scientific paper is published. If this paper presents an algorithm or a mechanism, interested researchers have to reverse-engineer the entire development process. Either the researcher will have to waste time on doing this or he will not use this work at all. Finally, it permits mutual aid between researchers, which helps to debug or improve performance.

Yet placing all material we have on the web with an open source license is not enough to achieve the goals previously mentioned. As a paper has to be well written in a clear, precise and concise manner, associated material has to be understood and directly usable. Therefore, there is a considerable amount of extra work required to permit fluent and effective open collaboration:

- Since the work is intended to be reused by external and hopefully numerous of people, the sources have to be clean, robust and well-documented. In addition, some how-to tutorials are very welcome.

- A versioning tool should be set up to track changes and efficiently manage a collaborative workflow.

- Online community tools should be set up to host discussions between researchers.

This work can increase the overall development time by a factor of 3 but participates both in building cumulative science in the research community and increasing the actual impact of our work.

In the Poppy project we decided to distribute all the hardware under "copyleft" licenses, which let users freely use the sources as they want on the condition that they share the derivative work with the same license. The open source distribution and community management will be discussed with more details in the chapter 11.

5.5 Conclusion

In this thesis, we aim to enable both the free exploration of morphological variants on real robotic platforms and their diffusion in the research community. To do so, we suggest exploring an alternative design based on 3D printing for mechanical parts, Arduino electronic architecture for sensors, Robotis Dynamixel motor for actuation and Python API for control.

This design process permits the creation of low-cost and highly hackable experimental robotic platforms thanks to a fully modular and open source approach.

The tools used form part of the makers revolution and the emergence of the new rapid prototyping tools, sometimes called the novel industrial revolution (Anderson, 2012). Therefore we can rely on the hundreds of Fablabs around the world as a lever arm to increase the dissemination and reproducibility of robotic platforms designed with such methods.

Yet the chosen approach raised some limitations. Indeed, since we want to keep our work reproducible, we have to reduce the complexity of the assembly as well as the use of our robotic platforms. This means we need to spend more time developing and testing our design to make it as easy to use as possible. Also we are limited in the components we can use, they have to be easily accessible i.e. easily available and in large quantities in online stores.

Also, for the open source distribution, essential in creating a research community platform, a lot of effort is required to create an efficient and pleasant workflow.

In the next chapter, we will explain how we applied the methodology presented to the design of a whole new humanoid robot called Poppy. Then, the design of an easy-to-use modular Python library will be describe in chapter 7.


Papaver Rhœas

6 The development of Poppy

6.1 Introduction

In 2012, when we started this project, none of the existing humanoid platforms were suitable for exploring the role of morphology. There were two kinds of platform. On one hand commercial robots that are rather easy to use and accessible but with a static and frozen morphology. On the other hand, prototype robots produced in labs to address specific experimentation needs, studying interesting morphologies but complicated to use and impossible to reproduce outside the lab. In both cases, only a few are open source, limiting hacks, extensions or modifications of their morphologies even further.

In the Flowers Lab, we had both kinds of robot. We used Nao (see Fig. 6.1a) to study human robot interaction (Rouanet et al. (2009) Rouanet (2012)). It was really convenient for use by researchers who are not interested in hardware issue since they are addressing more high-level research challenges. Yet such a platform is limited as it is not possible to modify the robot if it is not strictly adapted to our experiment. For example, back at this time the Nao camera was not efficient, with a closed field of view and a slow framerate. We have difficultly achieved 5 frames/seconds. Although we had the necessary skills to hack Nao and change the camera to fit our needs, its hardware was not designed to be changed. Improving the vision performance would only be possible with the addition of an external camera on the Nao head which could ruin the user experience. In addition, it would have been interesting to explore how the camera parameters (FOV, framerate, resolution...) can change the user experience but again, it is not possible with this robot.

We also used Acroban (see Fig. 6.1c) designed by Olivier Ly (Ly et al., 2011). It is a handcrafted humanoid platform created to explore certain morphological properties, especially compliance, with the aim of achieving dynamic locomotion and playful physical human robot interaction. While it actually allows modification of its is morphology, it is manufactured from aluminium mechanical parts, Robotis Dynamixel motors, scotch, and rubber bands cobbled together, and changing it requires lot of effort . The manufacture of aluminium parts required is especially complicated and requires either very good handiwork or a 3-axis CNC. In addition, its use was quite complicated and while several researchers could have been interested by Acroban to study human robot interaction and social acceptance, It was not

(a)Nao (b)Darwin-Op (c)Acroban

Fig. 6.1.: None of the existing platforms in 2012 were suitable for exploring the role of morphology. Nao was impossible to modify. Darwin Op and Acroban used aluminium parts that are really difficult and expensive to produce.

possible to use it without significant mechanical work. Finally, the material and manufacturing process make the platform non-stationary. Even if a lab manages to reproduce it, there is a high probability that the physical properties will not be the same. Therefore, the diffusion and the reproducibility of results are limited.

A last alternative would be the use of Darwin Op robot (see Fig. 6.1b) which is both open source and easily accessible (Robotis sells it already assembled for $10K), yet as Acroban its hardware consists of manufactured metal parts making its morphology very difficult and expensive to modify. Moreover, even if Darwin is open source and very popular, to our knowledge its morphology has never be modified by the research community[1].

Thus one of the main goals was to successfully design a humanoid robot which can merge the advantages of both kind of robot, i.e. simple, accessible, reproducible and allowing to easily change and hack its morphology for scientific experiments that can be both customized and shared.

6.1.1 An experiment-proof robot

Most researchers can attest to the difficulty and frustration faced while conducting robotic experimentation in the real world. We are challenged daily by bugs, technical issues, unpredicted events and side effects. While a bug in software can be fixed, an error with a hardware platform can cause damage to the robot and postpone the results of an experiment by several weeks.

[1] We can also notice the lack of community management tools such as wiki, forum and correct versionning system. If someone create a variation of Darwin, there is no place where he can share it.

Therefore many researchers in robotics avoid technical issues associated with the real world experimentation by using simple models and physical simulation. But the real world is extremely more complex and richer than the virtual one. Some high-level behaviour experiments are conducted in simulators based on the hypothesis that real-world constraints are not relevant, yet it is really certain? Indeed, while the real world constitutes a lot of constraints, it is also rich in complex physical effects (gravity, friction, inertia), which should be taken into consideration and could be very useful if interacting with the agent.

As we saw in the related work (chapter 2), the emergence of complex behaviours appears thanks to the interaction between the real world and simple robotic systems. We cannot program behaviour because behaviour is the result of interaction between the program and the real world. Thus we cannot design behaviour without the ecological niche of the robot Steels (1990).

While using simulators can be helpful as a first step to design robots, it appears incomplete when showing results on the role of morphology without real world experimentation. Therefore, when one wants to study the role of morphology on robot behavior, being able to explore it in the real world is of paramount importance. Unfortunately, current tools make the experimental step really hard to achieve for researchers.

Throughout our work on building cognitive and developmental learning algorithms (Oudeyer et al. (2007),Moulin-Frier and Oudeyer (2013)), we have experienced these issues, especially while building and using Acroban Ly et al. (2011) and during the Ergo robot experience (see chapter 10). Much time has been spent debugging non-robust technologies but it has been very instructive for understanding those that are efficient and those that should be avoided. Therefore Poppy has been designed based on the background experience we have acquired building using robots acting in the real world.

Robustness and Safety: Demanding and lengthy real-world experimentation necessitates that the robot be robust and safe. It should be able to sustain experiments and fall down without easily breaking. At the same time, one should ensure that physical interaction with the robot is safe for humans.

Precision, stationary: Experiments should be repeatable, implying that the robot properties should be stationary.

Breakable, repairable: Breaking should not be costly and the robot should be easily repairable.

Transportable: To allow for experiments in natural environments, possibly involving interaction with non-technical humans, the robot should be transportable outside the laboratory.

Easy and fast to duplicate: If the robotic platform is to be reused in this way, it must be easy and fast to duplicate.

Affordable: To ensure widespread use, a key factor is to keep the cost of the platform relatively low. The more labs can be involved, the greater the scientific impact.

To respond to these needs we created Poppy (see Fig. 6.2).

6.1.2 Overview

Poppy is a small and lightweight 25-DoFs humanoid robot (see section 6.2), whose morphological design allows for quick and simple modification by non-expert people. This is achieved thanks to the use of the 3D printing technique for the mechanical structure and Arduino based electronics architecture (see section 6.7). Its motors are common and widely used off-the-shelf Robotis actuators, and allow for compliant control and soft physical human-robot interaction. The pypot library (see chapter 7) enables programming beginners as well as advanced roboticists to control the robot and is adapted to the modularity of Poppy's hardware.

Its current morphology takes inspiration from the human functional morphology: a large number of joints (25 motors), the limbs respect human proportions, it has five joints in the torso and its thighs are bended by a $6\,\mathrm{deg}$ angle, similar to human ones (see section 6.4.3).

All the work and material involved in building, creating and using Poppy is distributed under open source and open hardware licenses. The 3D-printed skeleton and the electronics boards are under Creative Commons licenses while the pypot library is distributed under GPLv3 licenses.

Poppy is the first open source and 3D-printed complete humanoid robot, the design of such a novel platform will be discussed in the following sections.

6.2 Exploring morphological variants

The whole structure must be easily reconfigurable both for the purposes of repairing and hacking. This means the process of replacing Poppy's parts must be simple,

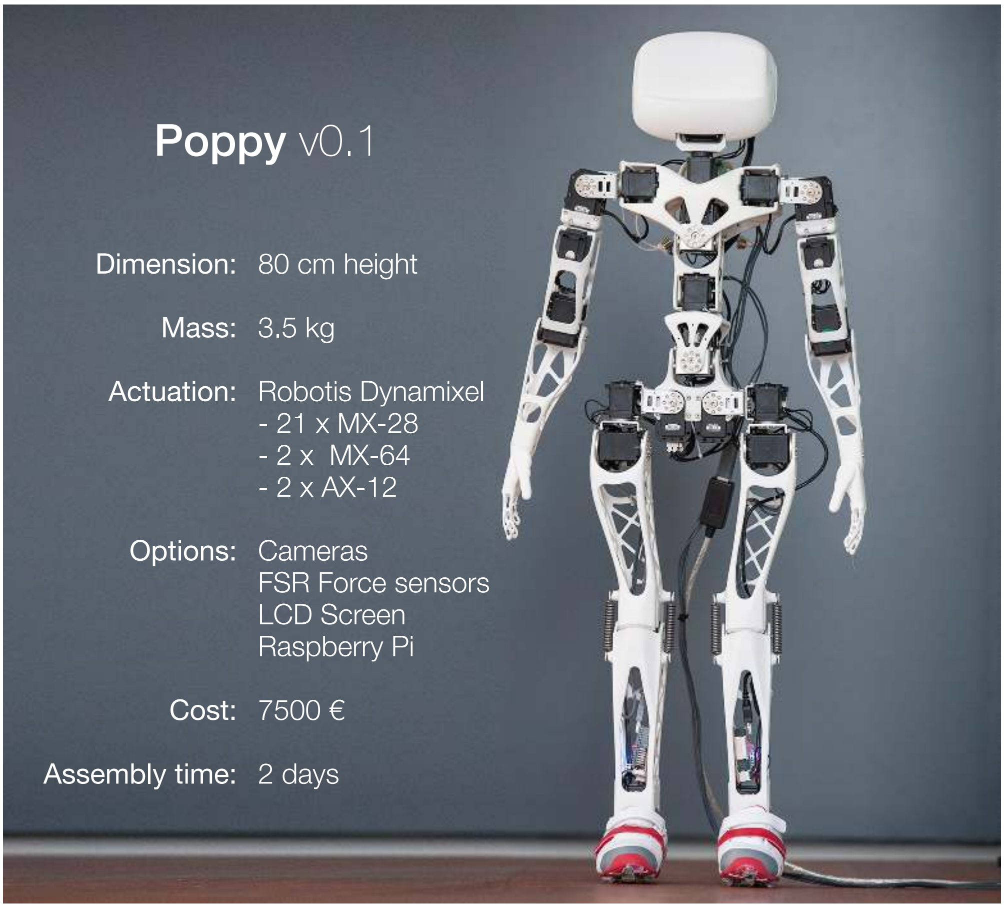


Fig. 6.2.: Overview of Poppy beta with the main specs and features. This figure will be updated when the final release of Poppy is ready.

low-cost and not require time or special tooling. Also, in order to have a real impact in the open hardware community, special attention is given to the modularity and the reusability of our technological bricks. Poppy is fully modular (mechanic, electronic, software) allowing freely exploration and modification of Poppy's body. Its modularity and the use of 3D printing make Poppy highly hackable. Therefore it can be easily adapted to particular experimental setups and allows diverse exploration into its morphology.

6.2.1 3D printed parts

We introduced for the first time the use of 3D-printed mechanical parts in our work when we built the ergo-robot installation (August 2011 - see chapter 10). The result was impressive as the parts were robust, precise, low-cost and fast to produce. Very convinced by this technology and with a keen desire to be free to explore robot morphology, we decided to build the whole mechanical structure of Poppy based on 3D printing techniques.

Material	Mass Density ρ (kg/m^3)	Yield strength σ (MPa)	Young Modulus $E(GPa)$
Polyamide	930	49	1.65
Aluminum	2700	200	70
Steel	$7500 - 8000$	350	200
Titanium	4500	1200	114

Tab. 6.1.: Comparison of material properties. The Young modulus represents the stiffness of the material while the yield strength corresponds to the maximal stress tolerable before plastic deformation.

Technique used

Several 3D techniques exist and were presented in the related work (see chapter 4). The Stereo-Lithography[2] (SLA) is very precise yet the material is not well adapted to support mechanical stresses.

Alternatively, we can use Fused Deposition Modelling[3] (FDM), which has the great advantage of being very low cost (2000$ for the printer + 40$/kg of material) and therefore, accessible directly in the lab. The parts produced are good yet the finish is not perfect and often needs to be reworked by hand. Also the process creates non-uniform parts, less resistant on one axis. Above all the FDM printers have low reliability, leading to a large number of printing failures. Nevertheless low-cost FDM printers are really useful when we just want to produce initial or single-use parts.

We prefered the use of Selective Laser Sintering (SLS)[4]. This 3D printing process allows the production of almost any shape without constraint. In addition, the price of the part depends on the total size and not on the complexity of the shape. This permits the production of very optimized shapes without increasing the total price of the robot. Moreover the use of polyamide material produces high quality parts with very good mechanical properties: uniform, lightweight, flexible and robust.

The table 6.1 compares mechanical properties of polyamide with classic metallic materials. We can see the relatively good properties of the polyamide material. The young modulus represents the stiffness of the part. The polyamide one has a very low young modulus meaning it is very flexible while keeping correct yield strength and very low density.

[2] This technique relies on a photosensitive monomer resin, which forms a polymer and solidifies when exposed to ultraviolet (UV) light.

[3] The FDM technique relies on melting and selectively depositing a thin filament of thermoplastic polymer (ABS - PLA) in a cross-hatching fashion to form each layer of the part.

[4] The process uses a high power (25-50W) CO2 laser beam, which melts and fuses fine powdered material spread on a layer.

A SLS printer is much too costly for a lab, but outsourcing the production to an external company[5] is really easy[6] and relatively low cost[7]

6.2.2 Exploring morphological variants

3D printing is a key feature of Poppy that permits the exploration of morphological variants. Indeed it is now cheap and easy to produce custom parts, and because Poppy is open source, anyone has access to the source files and can freely change the parameters he or she wants. Indeed Poppy is designed using Solidworks, a parametric CAD Software very widespread in small-size engineering companies. The parametric modeling allows to create mechanical parts those design are based on a set of parametric sketches and functions. Parameters can be modified and the final part will be changed accordingly[8]. Therefore it is possible to easily change the mechanical structure and properties just by tuning associated parameters in the source file and re-printing the part (see Fig. 6.3).

Moreover, 3D printing does not only permit the shape of a part to be changed, it can actually produce it with different materials. In particular, Direct Metal Laser Sintering (DMLS) - very similar to the SLS process - permits to produce the same parts using materials such as steel[9] or titanium[10]. It is therefore possible to extend the exploration to material mechanical properties (e.g. flexibility, density).

6.2.3 Scalable actuation

As explained in chapter 5, the chosen methodology relies on the all-in-one Robotis actuator. They are really convenient to use as they directly embed drivers, encoders and communication buses. They are also quite powerful, robust and rather precise. This is done by the combination of Maxxon motors, metal gearbox and precise magnetic rotation sensor (resolution: 0.1°).

Also Robotis offers a range of motors with different actuation power (see Fig. 5.4b). They are different in size and power but their API remains the same and we can easily switch from one to another without either changing the code or the electronic integration. Yet, even if the size changes, the footprint keeps the same pattern (see Fig. 5.5).

[5] `http://i.materialise.com/`
[6] In most cases, the company offers automatic scalable orders through an on-line platform
[7] Printing all the parts to build a Poppy costs about 1200€HT.
[8] If designed correctly
[9] http://i.materialise.com/materials/stainless-steel
[10] `http://i.materialise.com/materials/titanium`

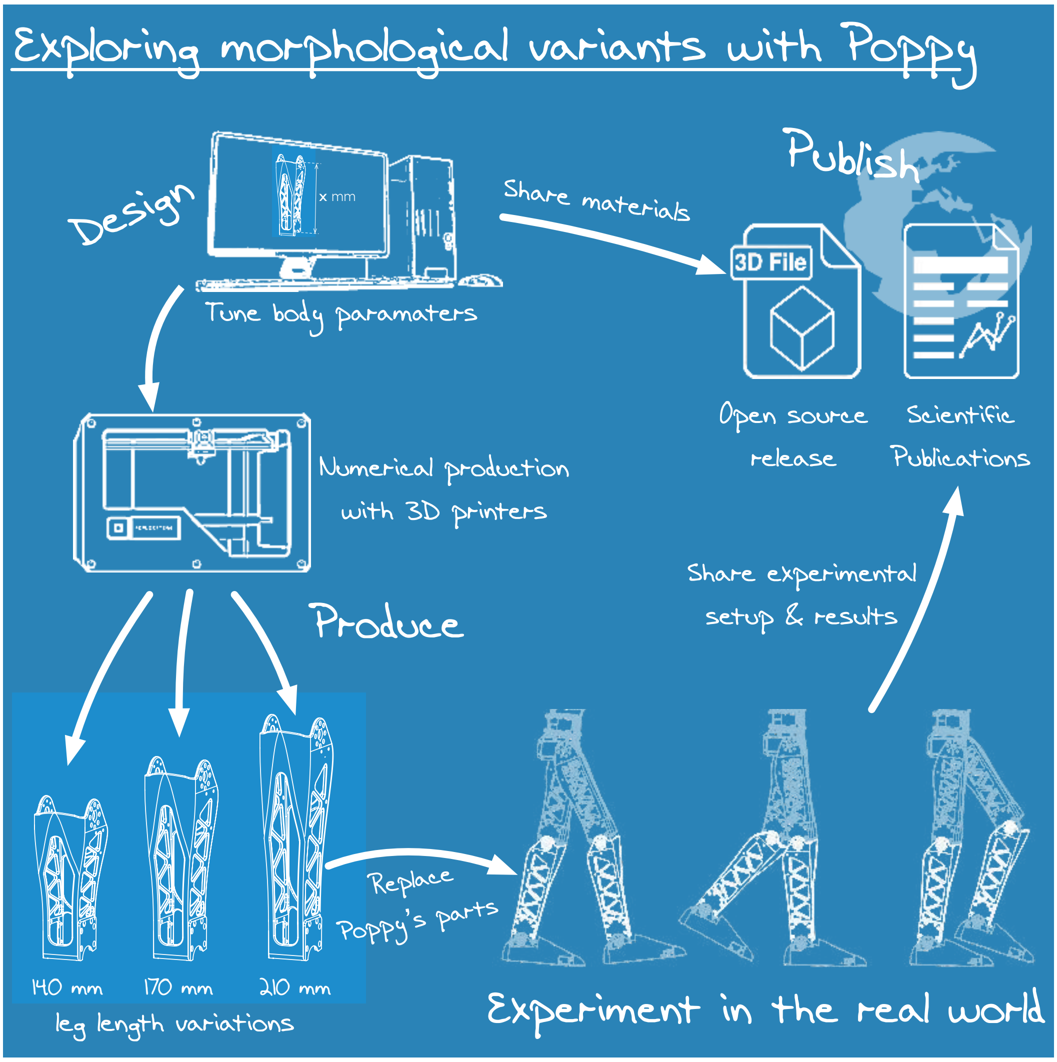


Fig. 6.3.: The use of 3D printing open new perspectives to explore the role of morphology in the robotics field. Indeed we can now easily produce on site new mechanical parts, therefore it is possible to create numerous variations of body properties. Poppy's modularity makes the part substitution easy and fast so we can directly iterate with real world experiments. Then we can share online a reproducible experimental setup.

Poppy is designed with Solidworks, a parametric modeler that offers features like configurations which define sets of parameters. Configurations allow to create multiple variations of a part or assembly model within a single document. Configurations provide a convenient way to develop and manage families of models with different dimensions, components, or other parameters. It is therefore possible to create, for each part, a configuration compatible with each motor just by setting the suitable parameters. The Fig. 6.4 shows an example with Poppy's leg.

It just takes a couple of minute to transform a part designed for Dynamixel MX-28 to one compatible with Dynamixel MX-64 as we only have to change the parametric dimensions accordingly to the Robotis motor shape. Then it is possible to switch from one to another with one click. On Poppy, most of the parts are already distributed with multiple configurations suitable for different motors power. It allows the actuation power to be scaled and introduces it also as an experimental (discrete) variable.

6.2.4 Electronic

Unlike the mechanical parts, there is no quick and low-cost solution for producing custom electronics yet. However, exploring morphology may also require the sensors space (e.g. number, type, properties, positions) to be varied. As we explained in the chapter 2, we address this challenge through the use of the Arduino environment.

Arduino has developed both hardware and software, so creating and programming electronics systems becomes very easy. Their boards have plenty of I/O pins (digital and analog) suitable to power and control almost any electronics components. Also these pins can be used to handle low-level communication such as UART, SPI and I2C, useful to plug sub-module (e.g. IMU, LCD Matrix, tactile interface and so on). The software they developed abstracts the complexity of low level control[11] and communication[12] very well. Therefore, it allows wide variety and flexibility in the extension of the electronic system, while keeping an ease of use adapted to a non-expert audience. In addition, Arduino has a growing community - already relatively large - which creates, shares and produces low-cost, various and multipurpose electronic components. Actually almost all kinds of sensors have an Arduino version with ready-to-use hardware and software.

Being able to change the morphology easily is of paramount importance in the Poppy project. Using Arduino-compatible architecture permits electronic modularity, which means the sensory-motor space can be considered as an experimental variable.

[11] we can turn a led on/off with just one line of code.
[12] Using just print-like functions we can communicate on serial bus.

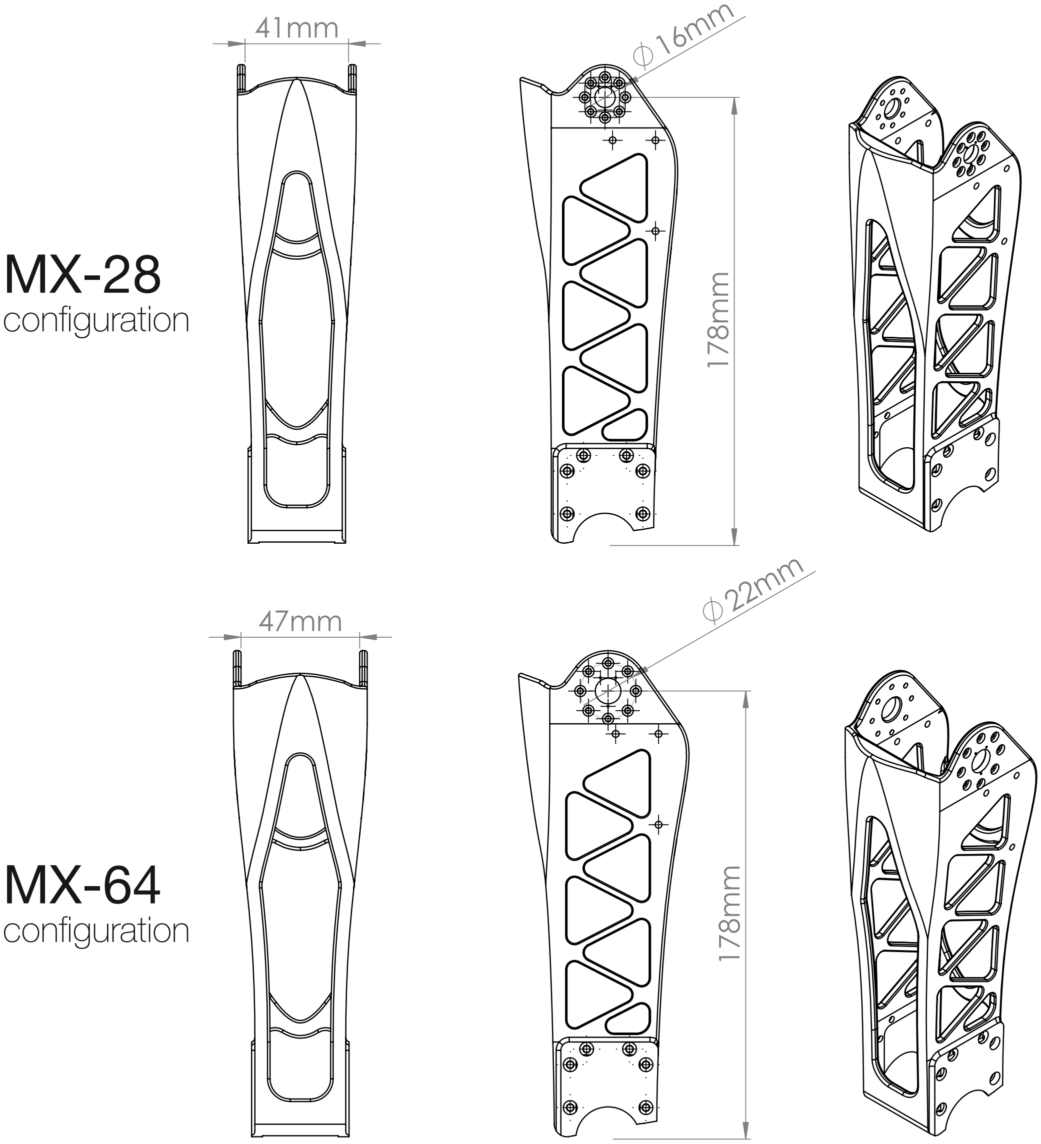


Fig. 6.4.: Example of the use of mechanical configuration. Here the CAD source of the leg involves 2 configurations so it can be compatible with Dynamixel MX-28 (up) or Dynamixel MX-64 (down). Configuration define the set of parameter suitable for both and can be changed just in one mousse click.

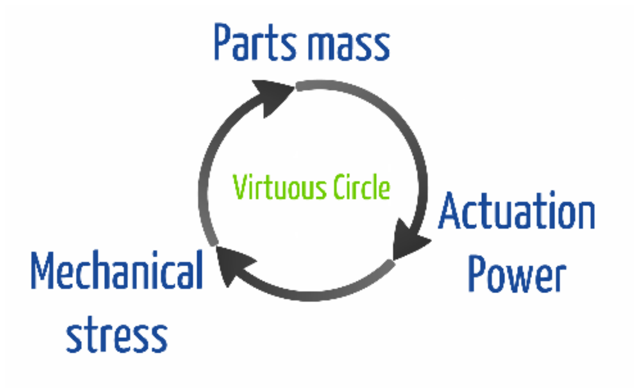


Fig. 6.5.: Using small actuation power reduces the amount of stress applied on the mechanical structure. Thus it is possible to reduce the robustness of mechanical parts by removing matter. Because the global weight is reduced, we can use smaller actuation power, and so on.

6.3 Lightweight

Many humanoid robots use powerful motors often associated with highly accurate sensors. This has a cost, both in terms of weight and computation resources. Moreover, to ensure the accuracy of the sensory-motor space it is necessary to design very rigid mechanical parts. The whole structure obtained is powerful but very heavy and not very agile due to inertia. In the Poppy platform, lightness is very important both for dynamic properties and safety:

- for a given actuation power, reducing the link mass reduces its inertia and permits the agility and the responsiveness to be increased,
- makes Poppy a platform that is easier to manipulate and transport outside the lab,
- makes the robot safer for people as well as for itself when it falls (and it will definitely fall a number of times).

The lightness was achieved by combining low-power actuation and optimized structure. Indeed when combined, it creates a kind of virtuous circle where the reduction of the maximum torque reduces the strength on mechanical parts. Because less force intensity is applied, we can remove material from parts. Because we have a lighter mechanical structure we can reduce the actuation power required and so on.

Also using low-power actuation has several interesting advantages:

- The actuation being the main cost of the robot (>60%), using the least powerful motors significantly reduces the total cost of the robot.

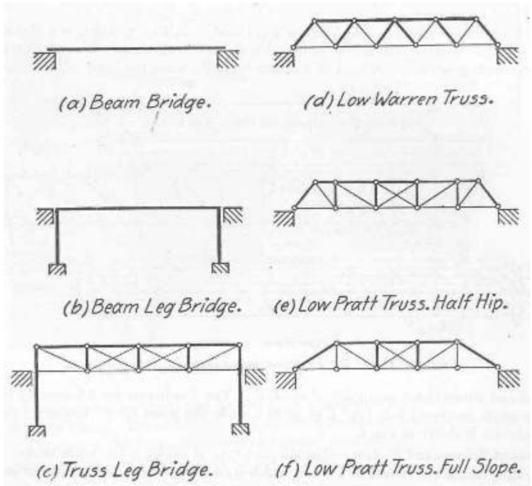


Fig. 6.6.: The truss structure has been massively used in civil engineering, especially to construct bridges. Here is shown several truss structure configurations.

- Low-power actuators mean a safer robot. Indeed, in the case of a programming error, the robot is not powerful enough to hurt someone or itself.
- on a research challenge level, it constrains the possible movements to the ones requiring little strength, the result being certainly more human-like.

Therefore Poppy was designed using the weakest and lightest motors i.e. MX-28[13], except for a few particular joints (such as the hips) while the mass reduction of the mechanical parts was achieved by using truss design.

Truss is a well-known design technique from structural mechanics to create lightweight yet very robust structures. It is mainly used in civil engineering (see Fig. 6.6) but can also be used in planes, which require lightness and strength resistance.

The principle is based on beam theory and permits to increase the second moment of area of a beam cross-section (see Fig. 6.7), which is an important property in the calculation of deflection, the main weakness of a long beam.

The second moment of area is computed as follow:

$$I_x = \iint_s y^2 dxdy$$

$$I_y = \iint_s x^2 dxdy$$

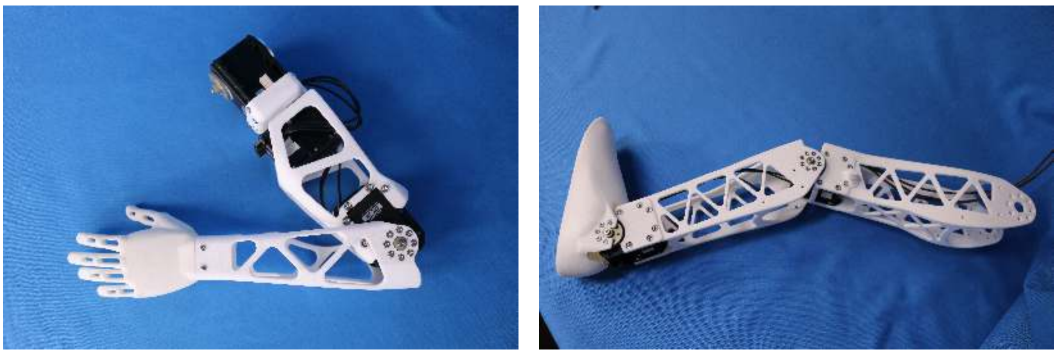
[13] Robotis motors are quite heavy (72, 126 and 153g respectively for MX-28, MX-64 and MX-106) in comparison of the Futaba servo-motors, 20-50g for a comparable output torque see http://www.futaba-rc.com/servos/brushless.html

where $s = dxdy$ is the surface integrated along the two axes of the cross-sectioned surface. We can see the on each dimension varies with a quadratic factor meaning the variation is not linear. Therefore matter placed far away from the origin is much more effective in increasing the second moment of area. Thus the main idea is to remove -the no effective- matter at the center and place it on the rim. In truss structure, matter is assembled by linkage avoiding local deformation.

Fig. 6.7.: Comparison of poppy's leg section and a rectangular beam having the same second moment of area.

The Fig. 6.7 shows the comparative cross section of two beams with the same second moment of area value. More precisely, the Fig. 6.7a is a cross section of Poppy's leg while the figure Fig. 6.7b is a basic beam with a rectangular profile. It would require a section such as $b = 27.72mm$ and $h = 27.59mm$ for the rectangular to have the same quadratic momentum as the truss design (i.e. $I_x = 54.862mm^4$ and $I_y = 53.260mm^4$ measured with Solidworks). Considering the length of the leg part (i.e. $190mm$), the total mass would be equal to $142g$ instead of $47g$ for the actual leg. This corresponds to a reduction of 70% of the mass.

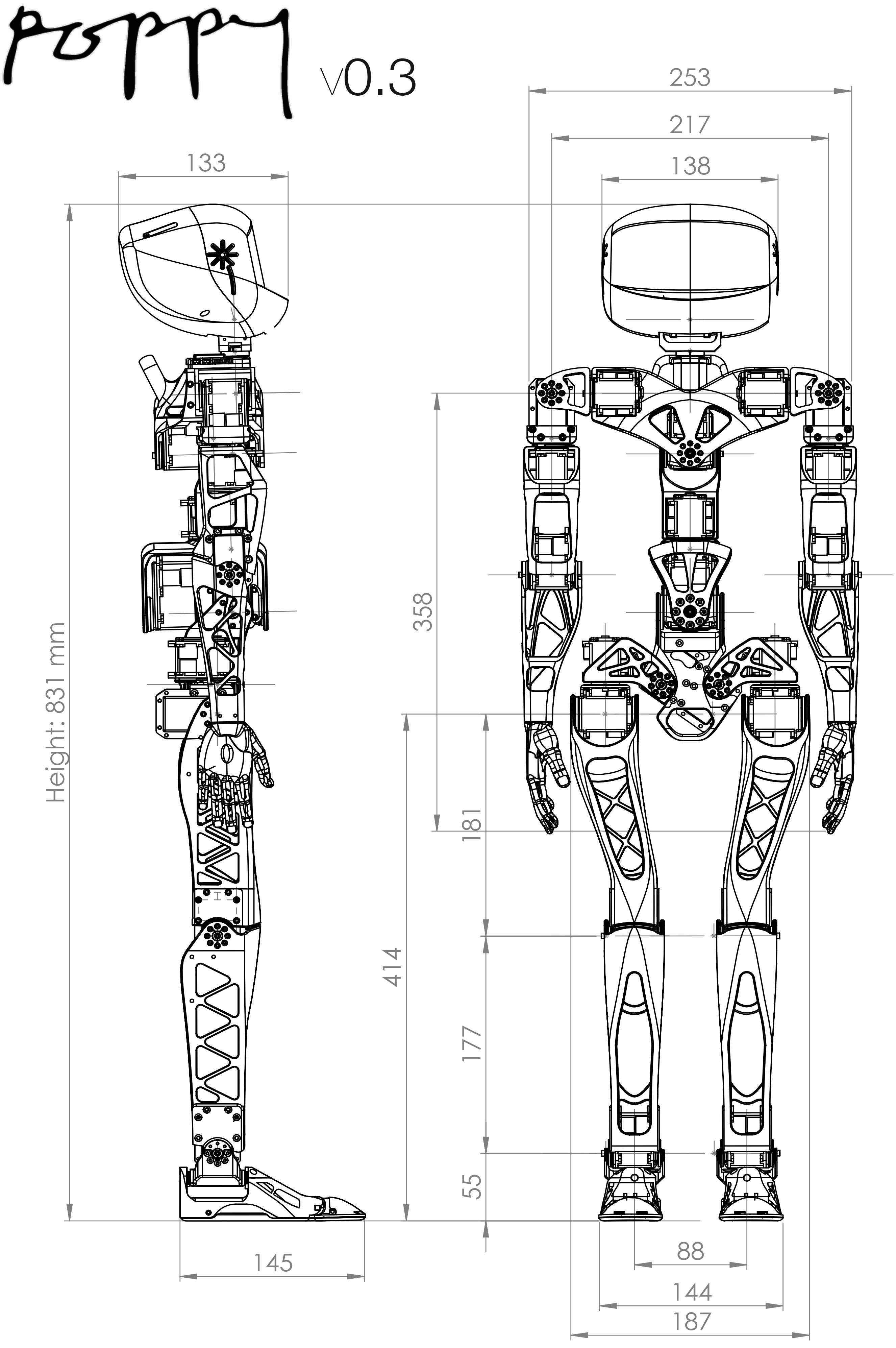
All of Poppy's limbs are based on this structure and have been optimized using finite element analysis (FEA) to perform structural simulation and validate the performance and safety factors of parts. Thanks to the use of this design on all of Poppy's limbs (see Fig. 6.8), we managed to have -certainly- the most lightweight humanoid robot with 3.5kg relative to its 83 cm height.



(a)Poppy's arm mechanical structure

(b)Poppy's leg mechanicalt structure

Fig. 6.8.: The whole design of Poppy is based on the use of truss structure to reduce its weight while keeping sufficient robustness.


Poppy
v0.3
253
217
138
133
Height: 831 mm
358
181
414
177
55
145
88
144
187

6.4 A little robot

To reach the goals presented in the introduction, the size of the robot is a really important aspect.

On one hand, small size makes the integration of complex, powerful and accurate mechatronics very difficult. Therefore it reduces the scope of technology we could use for the robot. In particular, the integration of hydraulic and pneumatic actuators, as well as advanced mechanisms involving several moving parts is really challenging.

On the other hand, having a small robot is really convenient for exploring morphological properties in the real world.

Firstly, it changes significantly the experimental process: the ratio between weight, and thus energy and torques enforced by movements, and the mechanical robustness of the structure and of the actuators, is such that the robot can fall without breaking itself. Moreover, it is lightweight, which allows people to handle it directly without additional infrastructure and in a safe way. On the one hand, all of this speeds up the experimental process. On the other hand, it deeply changes the methodology of movement and motor skill design by allowing the creation of movements directly on the robot by real-world experiments without a simulation process. This includes for instance adjusting motor primitives in real time, even critical ones.

Secondly, this brings advantages regarding human interaction, which is an important focus in this work: on one hand, from the above reasons, this rules out the problem of physical security in Human/Robot interaction. On the other hand, the size of the robot plays an important role in the psychological representation that people have of it.

However exploring morphological properties requires having a robot those morphology has an actual impact on its dynamic. Being too small reduces this impact because it reduces the inertia and the role of intrinsic structural frequency.

Thus Poppy's size is a compromise to facilitate at the same time easy testing in the real world, the integration of a large number of degrees of freedom (25 DoFs) and a structure those dynamic properties cannot be neglected.

6.4.1 Morphological proportions

From an anatomical point of view, Poppy reproduces the human proportions as described in the literature (Dufour and Pillu, 2005) (see Fig. 6.9) and the sensorimotor space organization: i.e. the main degrees of freedom (actuated and passive).

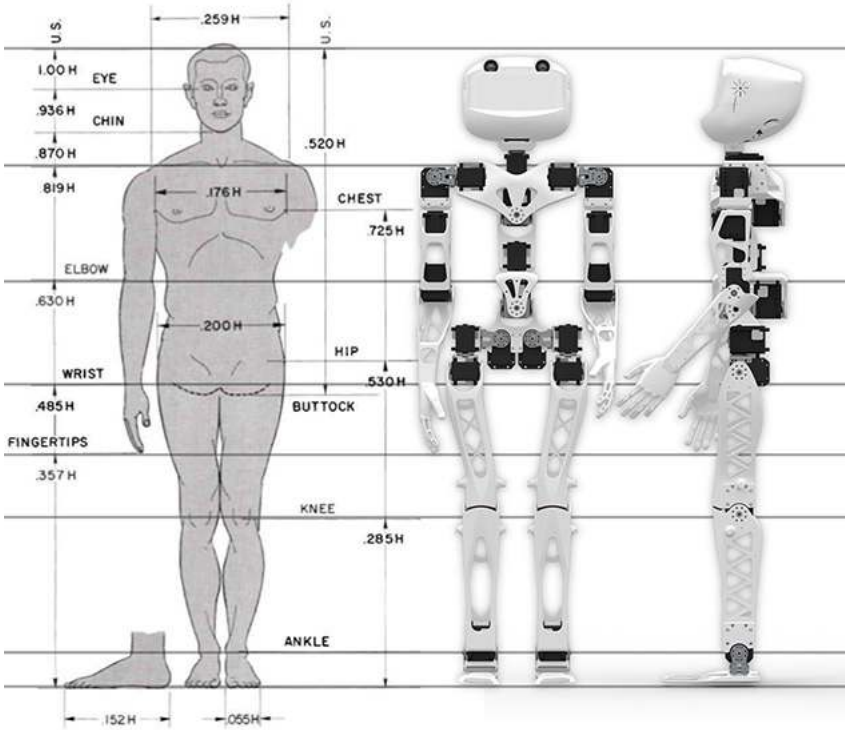


Fig. 6.9.: Human proportion used for the design of Poppy (Dufour and Pillu, 2005)

As the human grows from child to adult, his body proportions change (Bogin and Varela-Silva, 2010). One hypothesis we made is that human proportions converge toward an optimal link ratio. Based on this hypothesis, we decided to respect the human body proportion of an adult, thus the size of the robot described previously (see section 6.4) has defined all[14] dimensions of Poppy's links with the proportions presented on Fig. 6.9.

Of course, this hypothesis is contestable, and for example, mimicking the proportions of a child the same size as Poppy could be as relevant. Yet this choice has been taken as a starting point. Thanks to the design methodology we have and the open source diffusion, anyone can easily explore other choices and compare them.

6.4.2 Small and lightweight feet

With the goal of achieving biped locomotion, most humanoid robots have big, flat and rectangular feet. This design is indeed really convenient to simplify balance problems while easily increasing the sustentation polygon, but this design choice carries some constraints:

[14]An exception was made for the head, which is bigger to make it cuter and will be discussed in section 6.8.

Firstly, increasing the foot size increases the lever arm applied to the ankle. It can be useful as it extends the impact of the ankle control over the whole body. Yet given the potential high-torque applied to the ankle, achieving such control requires very powerful actuators. Powerful actuators are heavier and therefore the whole actuation design of the robot needs to be powerful in order to be able to lift and move the feet.

Secondly, some interesting dynamic controllers for biped locomotion seem to require mass-less legs (Hyon and Mita, 2002). Indeed, in this case there is no moment of inertia due to the motion of the legs, therefore controlling the whole body is simplified. While this case is a theoretical trick, it can still be transferred to the real world if there is a strong trunk/legs mass ratio. Therefore, either the overall mass of the robot should increase in order to make the mass of the legs negligible or we should design more lightweight legs. Because they are the furthest element from the torso, the mass of the feet has a strong impact on the inertia of the legs.

Finally, while current state of the art robots show that it is still simpler to achieve biped locomotion with big and heavy feet, some projects show impressive results thanks to the use of small or flexible feet (Bruneau et al., 2001). We can cite in particular Petman who demonstrates cutting-edge skills in biped locomotion over very complex terrain[15]. Moreover this aspect seems to be coherent with multi-legged animal species, indeed most animals have really thin legs and very small feet.

So even if common humanoid robots still use big and powerful feet, for the above mentioned reasons, we decided to explore bio-inspired, small, and lightweight feet (see Fig. 6.10).

While the human foot is very complex system involving dozens of bones, muscles and tendons, we simplified the design by extracting a few relevant functional human foot properties such as toes, which are key features concerning in both human walking Hughes et al. (1990) and biped robots with a human-like gait Sellaouti et al. (2006). On Poppy's feet, to reduce the weight and complexity, we designed a passive articulation with torsion spring (see Fig. 6.10). Also Poppy has very small feet compared to common humanoids, the ratio of height/length of feet is about 17%, close to the human one (see Fig. 6.9) while robots such as Nao have feet that represent 27% of the height.

Finally, unlike most humanoids, Poppy has only one active DoF. Indeed, while Poppy has small feet, it appears the actual moment it can transmit to the ground is really little for the lateral motion.

[15] Link youtube

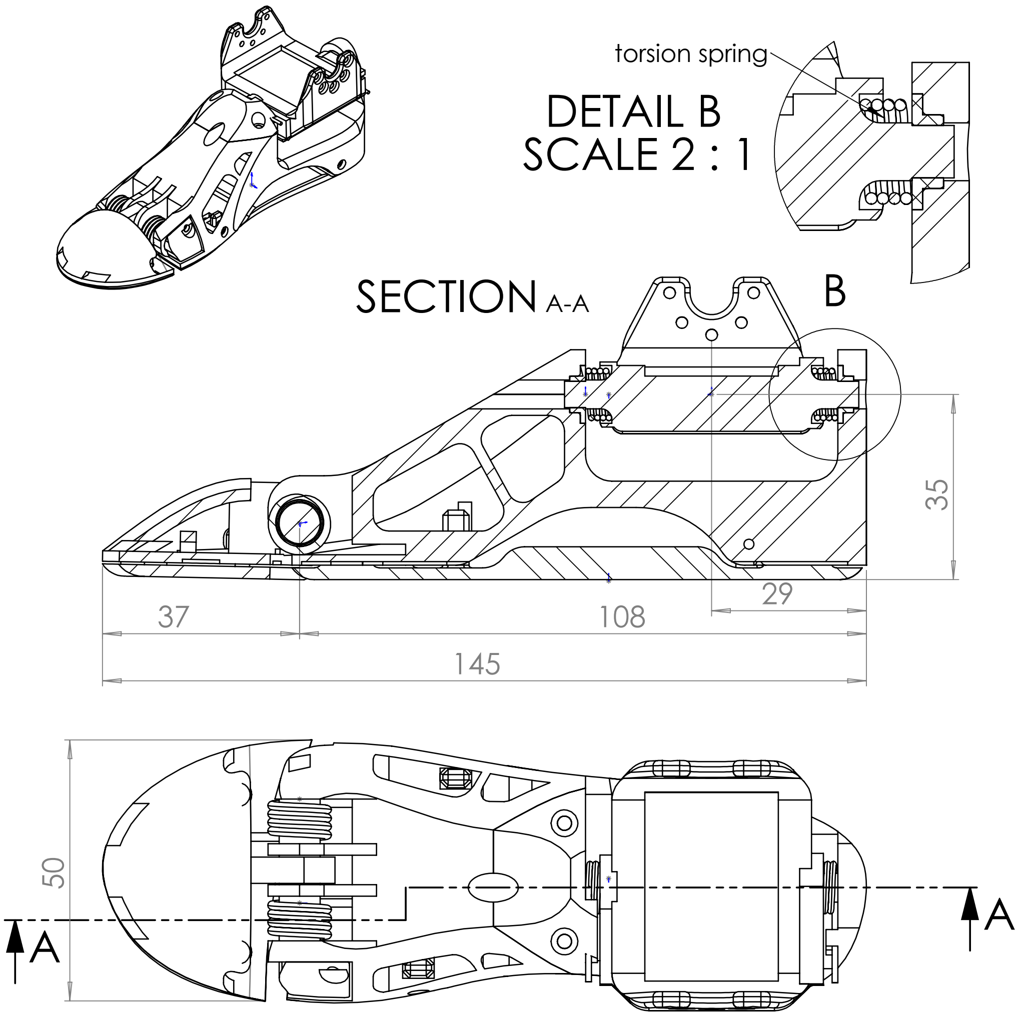


Fig. 6.10.: Blueprint of Poppy's foot. This foot has two passive joints, one for the toes and one for the ankle roll. As we can see on the section A-A and detail B, both use torsion springs.

Also creating an active articulation would add a lot of mass (72 gr) for little actual effect. Yet this DoF is still useful for ground adaptation. We therefore preferred to design an articulation based on a passive joint with torsion springs (see Fig. 6.10).

This design choice has a strong impact on the overall design, because we have lightweight feet, the power required to make the legs move is reduced, we can therefore use smaller motors which are also lighter.

6.4.3 Legs

Poppy's legs have 6 DoF each, three on the sagittal plane (ankle, knee, hip), one on the horizontal plane (hip) and one on the frontal plane (hip). These joints allow reproduce the main DoF of the human legs. In addition, if we look closely at the morphology of the human femur, it appears that it is inclined by 6 degrees (see Fig. A.1a). This particularity is reproduced on Poppy's thigh (see Fig. 6.11).

This slight bending makes the feet closer to the projection of the center of gravity and therefore changes the dynamic behavior. In this thesis we describe both a theoretical model (see appendix A) and real experiments (see section 8.1) showing that this bio-inspired thigh allows the reduction of falling speed by almost 60% (during single support phase) and the decrease of the lateral motion needed for the mass transfer from one foot to the other by 30% (double support phase).

Also, following the principles previously mentioned, the leg design is made to be as light as possible. This was achieved by using truss structure (see section 6.3) and the minimum required amount of power, involving two Dynamixel MX-28 for the ankle and knee joint, and a Dynamixel MX-64 for the hip joint (see Fig. 6.11). Yet as explained in section REF, most of our parts have several configurations allowing the actuator to be changed. Thus it is possible to easily replace the knee actuator by a Dynamixel MX-64 (see Fig. 6.2.3).

6.5 Pelvis

Poppy's small feet increase the challenge of the balance of the robot. Also, to keep the projection of the center of gravity (CoG) inside the support polygon, defined by the feet geometry, it is necessary to control the weight distribution of the robotic structure. In particular, we wanted that in its initial upright posture, Poppy stays balanced without any control.

Robotis actuators are among the densest elements in the Poppy platform ($1700kg.m^3$) and are the main source of weight ($1.8kg$). Their spatial distribution represents

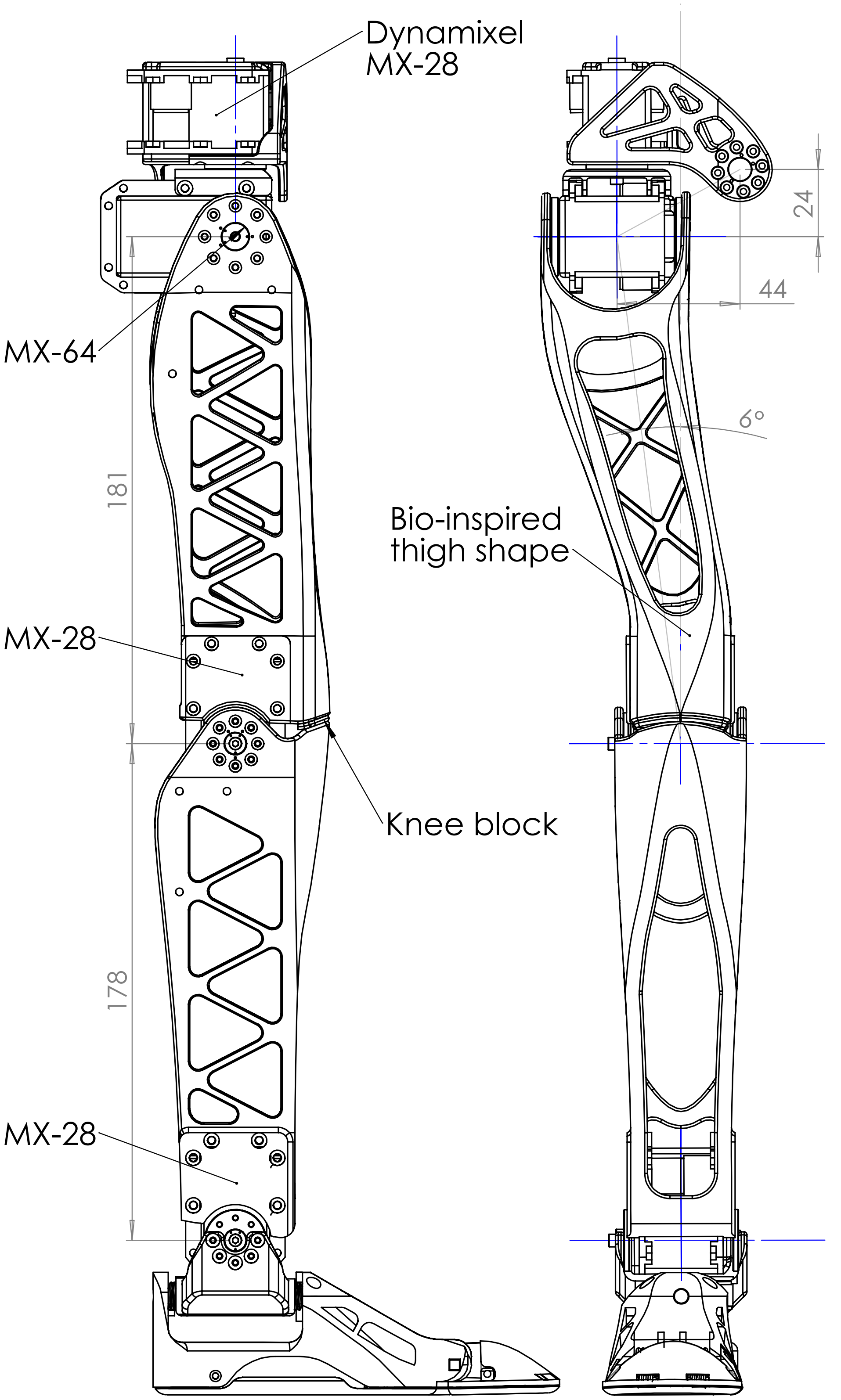


Fig. 6.11.: Blueprint of Poppy's right leg. Parts have a truss structure to reduce the mass, and each leg has 6 DoF. Only the one on the hip use a powerful motors (MX-64), others are powered by MX-28. This choice makes the robot could not support its weight if knees keep bended during walking. Therefore it constraints the research of more human-like gait with straight support leg.

therefore the major part of the distribution of mass in Poppy. In order to limit the displacement of the mass at the back of the robot, we decided to avoid a conventional ball joint assembly for the hip joint common on most robots based on Robotis motors such as DarwinOP or Acroban (i.e. distributed on a plane parallel to the sagittal plane). Instead, we placed them on the frontal plane as the left to right stability is greater than the rear to front stability. By doing so, the hip joint is not a real ball joint anymore. Yet to keep a wide range of motion, we used an original unsymmetrical motor configuration (see Fig. 6.12).

This configuration allows us:

- to create a compact multi-articulated pelvis,
- have hip rotations (frontal plane) leading to slight vertical motions of the leg, which act as a damper during walking. This damping can be tuned by adjusting the stiffness of the actuator.
- to reduce the hip joint lever arm and thus reduce the torque required to maintain position in single support phase,
- to reduce the distance between rotation axes to stay close to a ball joint,
- the resulting V shape frees up room to increase the range of motion of the legs on both the frontal and horizontal plane.

In addition, to reduce the shifting of the center of gravity to the back of the robot, the connection with the abs motors is slightly shifted toward the front. By doing so, we . This enables us to keep the CoG within the support polygon and to increase the range of motion of the abs motor when the robot leans forward.

6.6 Multi-articulated torso

Humanoid robots mostly have a rigid torso without any joints (e.g DarwinOp, Nao, NimbroOP) or few DoFs (e.g. two for Icub, one for HRP-2). However if we look at the human trunk and in particular at the spine, it has a complex mechanical structure and a large network of dense muscles controlling a very large number of DoFs. It allows for complex motions in several directions while keeping balance. Its movements are regulated by a complex combination of anticipatory and reactive muscle actions.

Before 1982 and the work of Thorstensson (Thorstensson et al., 1982), few scientists had really approached the subject. Since then, several studies have investigated

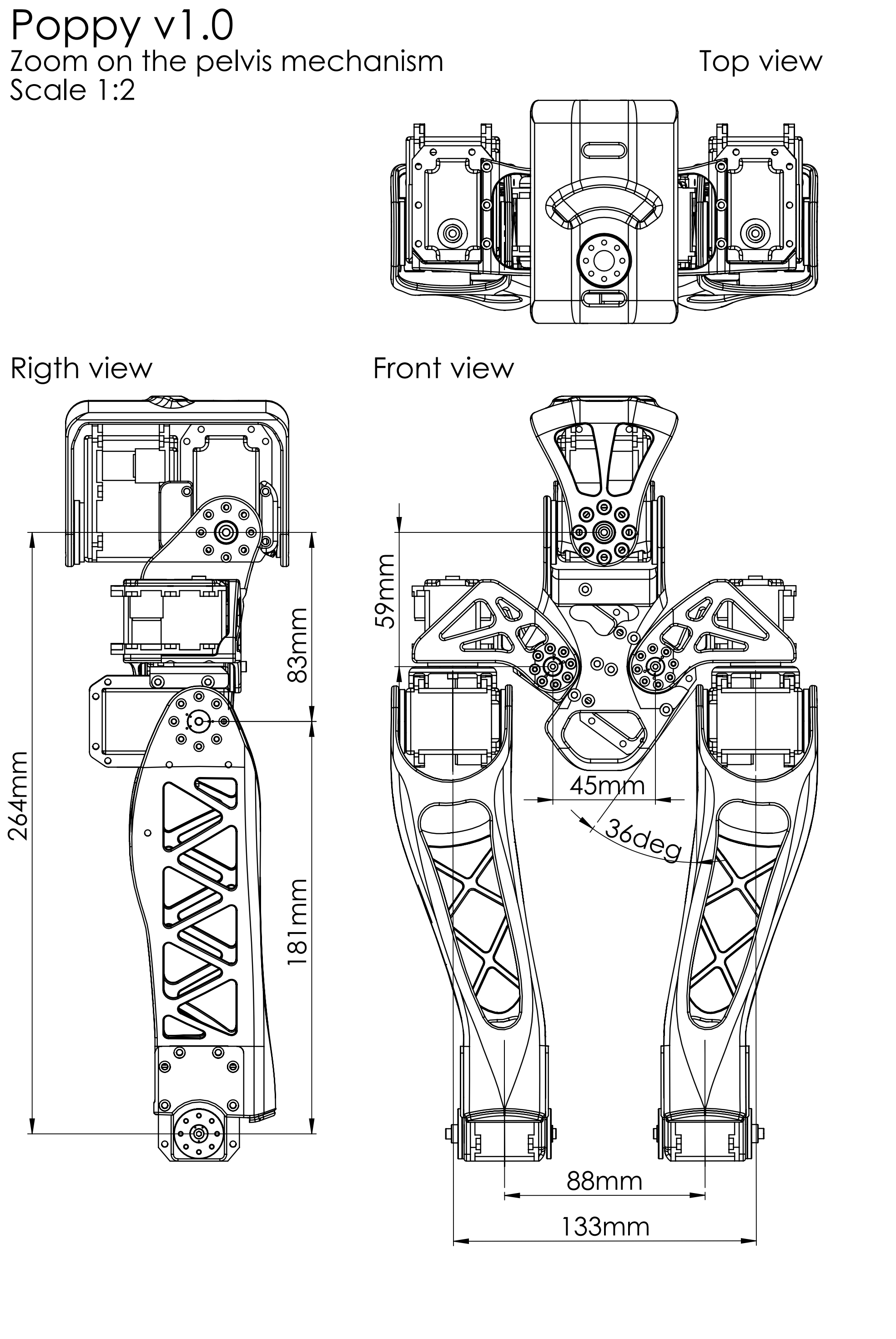


Fig. 6.12.: This blueprint shows the design of Poppy's pelvis. The use of a non-symmetrical configuration of motors permits to embed 6 DoFs in a compact space while keeping a wide range of motion in all direction.

the activity of the trunk during walking, and showed that the trunk is not only an additional mass but for example, participates actively in the human walk. Electromyographic studies have shown the importance of the erector spinae muscles in the organization of motor patterns during walking (Anders et al., 2007) but also of other rhythmic tasks (Sèze et al., 2008). Like the salamander (Ijspeert et al., 2007), a sequential activation of erector spinae muscles was found (Prince et al., 1994). They also show that the trunk leans forward and oscillates from 1.5 to 6 degrees during walking. In addition, lateral flexion during a gait cycle on the frontal plane promotes the weight shift and opposite rotations of the lumbar and thoracic belts on the horizontal plane allow for the extension of the footstep (Feipel et al. (2001), Lamoth et al. (2006)).

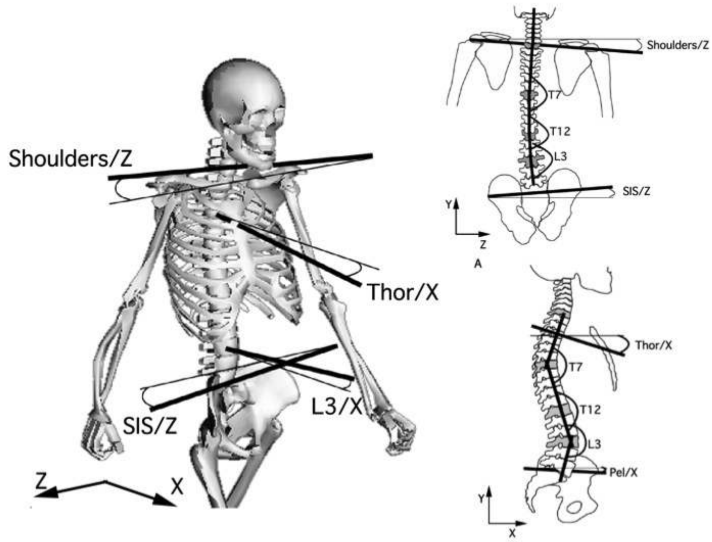


Fig. 6.13.: This figure shows the main human spine mobility in horizontal, frontal and sagittal plane. Figure extracted from (Ceccato, 2009)

Thus the human torso is complex and seems to play an important role in walking; it is essential for all human movements but especially for walking. The movements of the spine can facilitate the transfer of weight from one leg to the other, improve the balance but also participate in the dynamics of walking.

It seems therefore interesting to equip a humanoid robot that is an attempt to explore the role of morphology, with an articulated trunk in order to evaluate its impact on several tasks, from dynamic walking to physical human-robot interaction. Yet the human torso is difficult to replicate on a small robot using servo motors and therefore simplification is needed.

Interestingly, Ceccato (Ceccato, 2009) studied the role of the trunk during walking and highlighted that there are some places in the spine where the displacements are the most important, i.e. that the apparent high dimensionality of the trunk could be factorized down to a few essential components/dimensions.

Accordingly, it appears we can replicate the essential degrees of freedom of human torso with two DoFs on the sagittal plane, two on the coronal plane, placed in the pelvis and shoulder/thoracic and one on the horizontal plane placed in the middle of his torso.

These main degrees of freedom have been first introduced on Acroban (Ly et al., 2011) and continued on Poppy (see Fig. 6.14). Thus Poppy's trunk involves five degree of freedom, we use two Dynamixel MX-64 for the abs as they have to support and move the whole upper body mass, the 3 others joints being less subject to high constraints are powered by MX-28 (see Fig. 6.14a). This multi-articulated trunk allows a wide range of motion that can be useful to explore the role of the torso's motion on complex dynamics behavior (balance, walking), grasping and reaching task, or for human-robot interaction, extending to the expressive and emotional abilities.

6.6.1 Upper limbs

Poppy's arms were not designed for exploring grasping but rather for balancing, expressive and interactive purposes. Thus they only involve the minimum articulations required to produce a wide range of movements and they do not involve articulated hands.

A first version of the robot arms involved low-cost AX-12 motors($50), but these motors do not allow the same degree of compliance as MX-28 so the interaction was not smooth enough. We quickly replaced them with MX-28 motors, even if they are more expensive ($250), powerful and heavy (72gr rather than 50gr), the compliance ability is needed for playful physical interaction and demonstration.

This ability was especially useful for the experiments we made with Poppy walking while being socially guided by its hands[16] and experiments on chapter 8.

Also these arms combined with the complex spine make Poppy a particularly adapted tool for creating and studying emotions and gestural social communication (see Fig. 6.15). An example of such use has been demonstrated by two Cognitics students. Their goal was to study the transfer of emotion between robots and humans. Using Poppy's upper body and the pypot recording feature (see section 7.2.3), they were able to program a wide range of emotions (see Fig. 6.15).

[16] see video `https://vimeo.com/74557773`

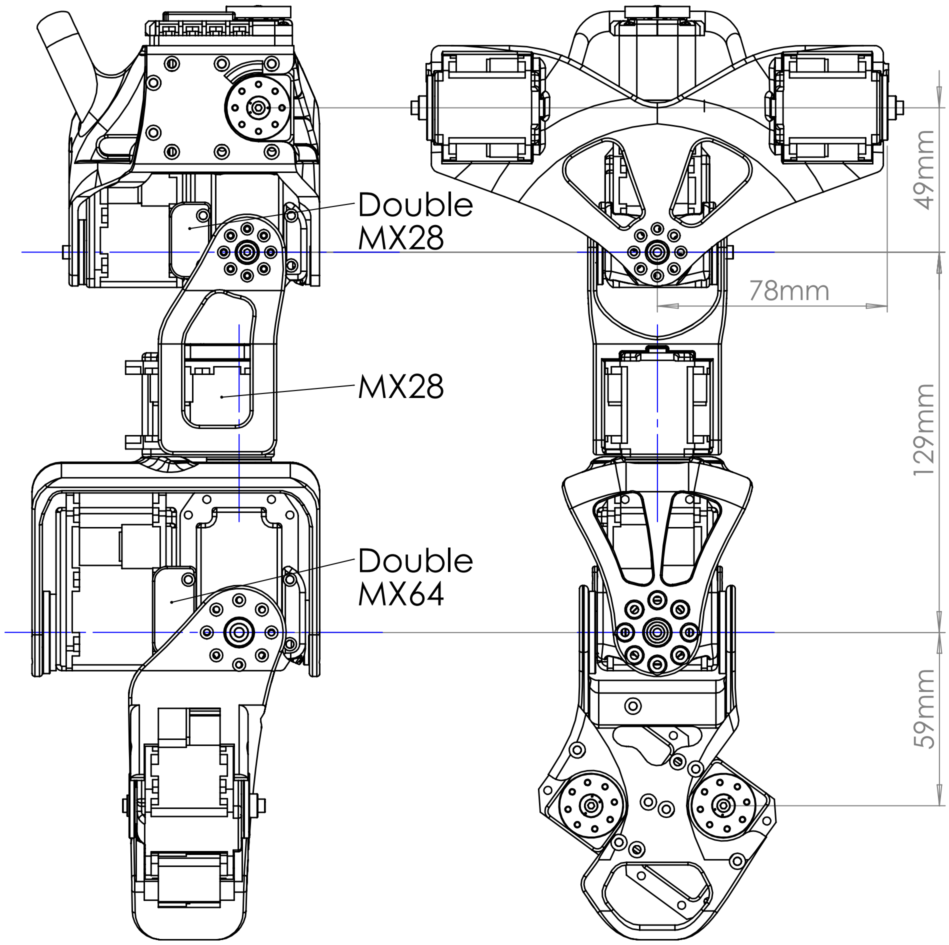
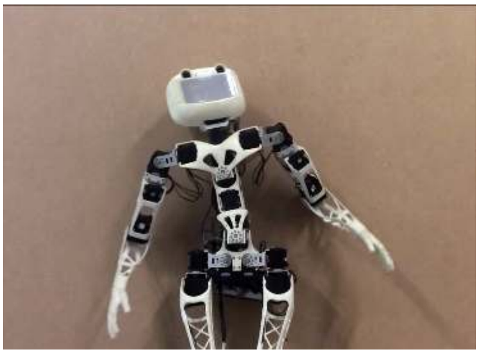
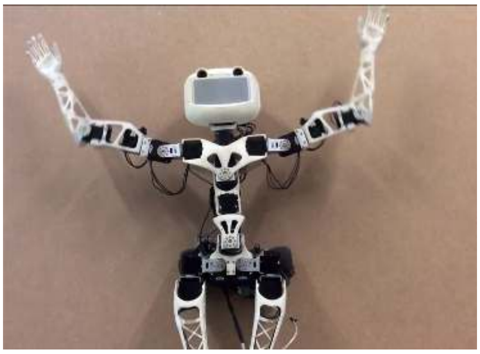


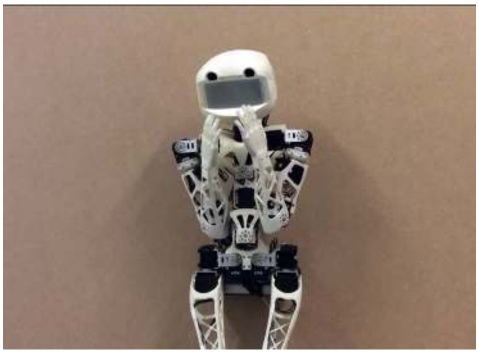
Fig. 6.14.: This figure shows the blueprint of Poppy's 5 DoF torso as well as illustration of the resulting mobility.



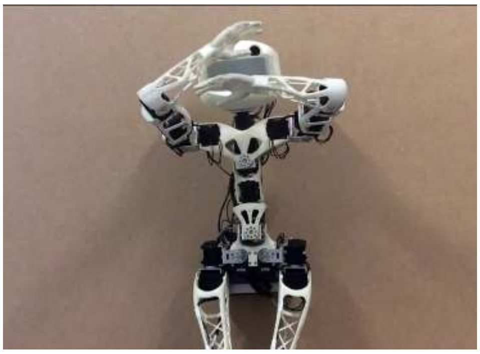
(a)http://youtu.be/StFIMuyz11M



(b)http://youtu.be/RwCtNwLk10E



(c)http://youtu.be/qrcmLXbpUVo



(d)http://youtu.be/ms2niFLevv8

Fig. 6.15.: The upper body mobility of Poppy allows for rich exploration of humanoid robot behaviour. Cognitics students have used Poppy to create and explore robotic emotion and study how people reacts. Using the multi-articulated torso and arms, they created a wide range of emotion those video links is displayed under each illustration.

6.7 Electronic architecture

To keep in the spirit of the project as described by its guidelines (see chapter 5), the electronic architecture has to be simple, easily reproducible, relatively low-cost and modular. Of course the performance of each component is very important and should be correctly dimensioned.

The first version of Poppy (beta) had a handmade electronics architecture, which required hacking several components before soldering them together. This design was not compliant with the design guidelines of Poppy (i.e. easy to use and to reproduce) and was actually the main reason Poppy was considered as a beta version. Recent work has been done toward the simplification and the reproducibility of the electronics part.

Yet the electronic integration is challenging. Indeed, because Poppy has 5 degrees of freedom in the torso, there is not enough room for all electronic components

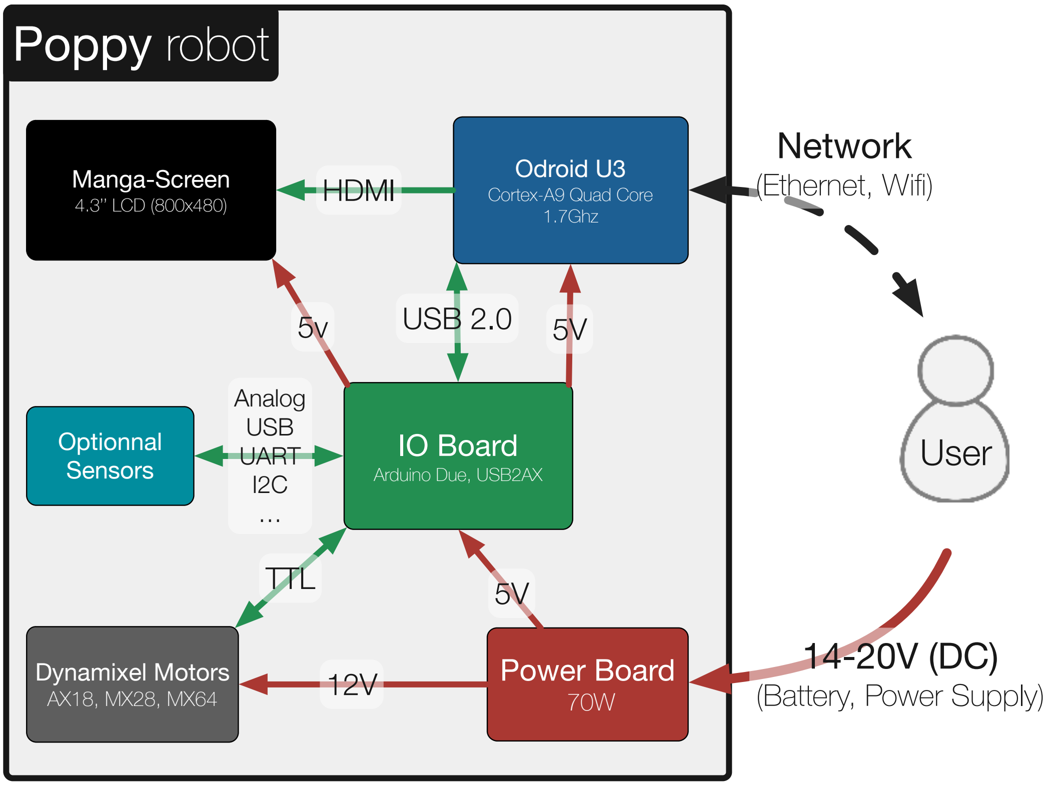


Fig. 6.16.: Overview of the Poppy electronics architecture.

needed. Therefore we had to embed most of them in the head which raises not only a problem of space but also of mass.

Poppy's electronic architecture is based on several key-components communicating with each other (see Figure Ref):

- an IO board controlling all the sensorimotor space,
- an embedded computing module to permit wireless communication,
- a head screen to display emotions or information,
- an alimentation board to provide the 12V needed for the motors and 5V needed for electronic systems,

We will describe in this section, the design choices we made for these various components.

6.7.1 Poppy IO Board

With the aim of offering easy-to-use modular electronics architecture and to make it fit in the head of Poppy, we decided to create a custom board[17]. We could argue it makes the diffusion of the platform more complex, since a custom board is too complex to make by hand and therefore may necessitate the kind of industrialization process we have been trying to avoid since the early days of the project. Nevertheless the maker's revolution brings novel solutions for producing electronics. Indeed, there are now companies (such as CircuitHub[18]), which offer scalable solution tools from 1 sample to a 10,000 batch. It is possible to upload our design and anyone can ask to have it produced. Of course ordering one part is more expansive but remains relatively low compared to the cost of the robot.

The board we designed included the basic elements needed both for the control of the robot and for its extensibility.

Motor control

Robotis Dynamixel are normally controlled by the *USB2Dynamixel* but we decided to replace it with USB2AX devices (see Fig. 6.17). The USB2AX is a small interface to control Dynamixel servomotors from a computer and was designed by Nicolas Saugnier. It plugs into a USB port and has a 3-pin molex connector compatible with the robotis ones.

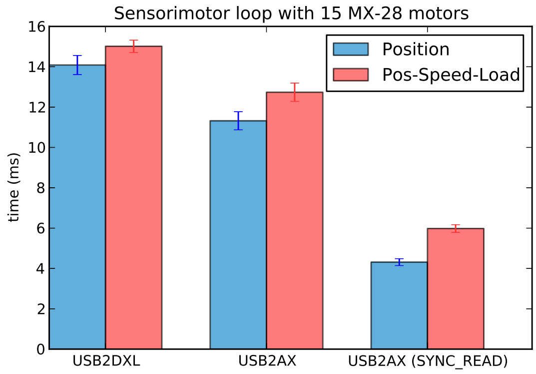
For use on Poppy, these devices have several main advantages:

1. They are a lot smaller than the standard USB2Dynamixel module (16x36mm instead of 35x90mm) (see Fig. 6.17b).
2. They can endure a short-circuit between the DATA and power-supply wire.
3. They have the sync_read instruction to read a lot of information very fast, which is not a standard Dynamixel instruction. The USB2AX converts SYNC_READ into multiple separate READ commands to get data from each servo, then sends back to the computer a single big packet containing all the data. This significantly decreases the effect of USB latency. A SYNC_READ command reads the same registers in each servo (see Fig. 6.17c).
4. It is open source so we can extend or adapt this solution to our needs.

[17] This board has been developped by Fabien Depraetre during his mster internship
[18] https://circuithub.com

(a)USB2AX dongle

(b)Comparison of size with the Robotis controller.



(c)Comparison of communication speed. We can notice the strong speed improvement the SYNC_READ instruction brings to the USB2AXS

Fig. 6.17.: These figures show comparison between standard Robotis controller and USB2AX dongle developed by Nicolas Saugnier.

This project has always been used on Poppy and greatly helped us to have an effective robot while keeping the space for electronics low. Because the project is open hardware, we have reused it and embedded it directly on a custom board so that we can make it even smaller by avoiding USB connectors.

Arduino integration

As we explain in the section REF, the modularity of Poppy's electronics is achieved thanks to the use of the Arduino environment. For the Arduino integration we decided to use the new Arduino Due based on the Atmel SAM3X8E ARM Cortex-M3 CPU. These board embeds both a powerful microcontroller (84 MHz 32-bit ARM core) and a large number of inputs/outputs: 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports).

Actual design of the IO Board

The IO board is an open hardware project aiming to simplify the use of Poppy for non-electronic-expert users. Therefore it integrates into one board all components it would be necessary to plug or solder.

This board is based on other open hardware projects previously presented and contains two usb2ax for communication with Robotis actuators, an Arduino Due to permit the extension of the sensors space plus convenient ports to easily plug external devices in. See Fig. 6.18b for complete details on all the available IO ports.

In addition, it integrates two sensors: an ADXL345 accelerometer, which has four measurement ranges (2g/4g/8g/16g) with up to 13-bit resolution and a data rate of up to 3200Hz; and a ITG-3200 gyroscope, which has a full scale range of +/- 2000°/s with 16-bit resolution and data rate of up to 36kHz.

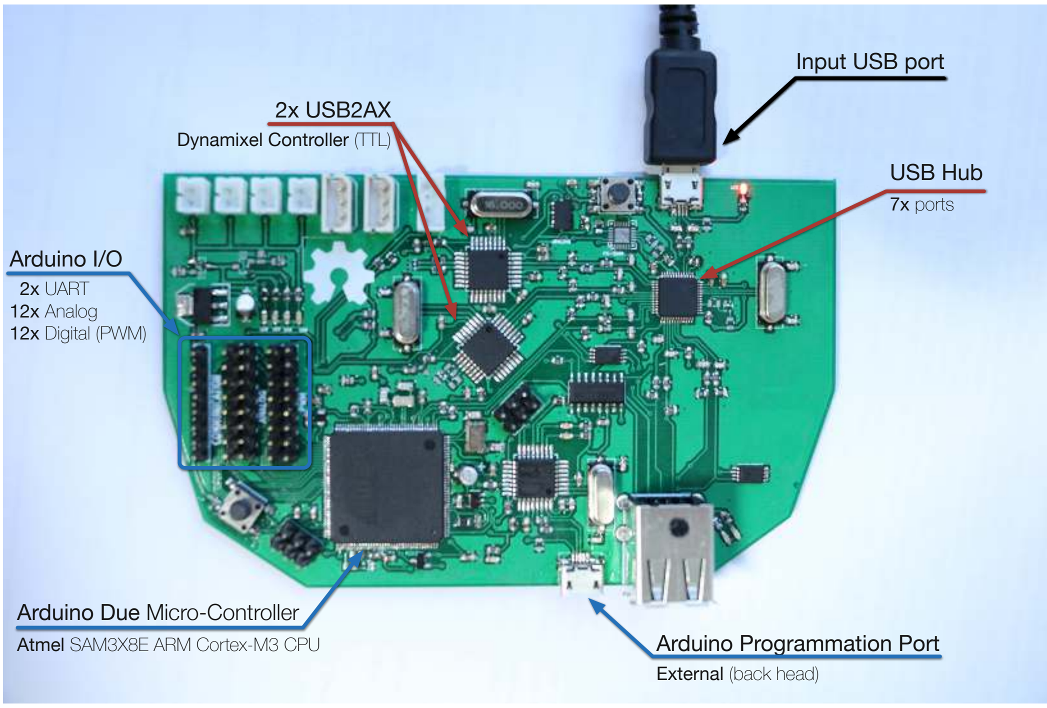
The board has been designed using KiCad[19], the source files are distributed under open source license[20] and available on our GitHub project[21]. The production of the board can be done using CircuitHub[22] and costs $250 for one board or $90 for ten[23].

6.7.2 Embedded computing module

The integration of a computing module is rather complex and not fundamental if the robot cannot walk for more than 5 m, therefore the Poppy beta version was controlled using an external computer connected by USB. However, Poppy is aimed at becoming a shared research platform with people addressing challenges in which embedded control could be necessary. Also as all users have different computer configurations (Windows, MacOS and wide Linux distribution), it is easier to maintain the control software if only one OS is used. Embedding linux allows us to have the control and ensure same performance for every Poppy.

Yet as I said previously, embedding control is complex. Indeed, the computer has to be small enough to fit inside the robot. With such a size, where are mostly ARM based computer. Most works are developed on x86 or 64 architecture and the switch to ARM architecture is not direct. Some software modules used do not exist or are not optimized, leading to major performance problems.

[19] Open source EDA software
[20] Creative Commons CC-BY-SA
[21] `https://github.com/poppy-project/poppy-electronics/tree/master/CarteIo`
[22] `https://circuithub.com/projects/Poppy_project/CarteIo`
[23] The cost decrease with quantity is up to $50.

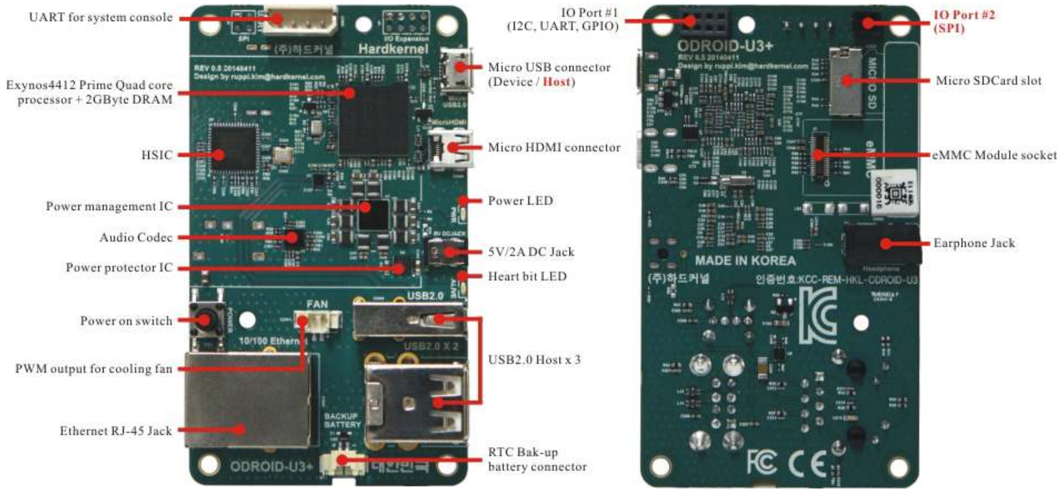


(a) Current design of the IO Board.

2x	high-speed motors buses
2x	UART ports
1x	I2C bus
2x	external USB Ports
1x	accelerometer
1x	gyroscope
12x	anlog pins
12x	digital pins allowing PWM control
4x	5v ports to supply power to external devices

(b) Main IOs available

Fig. 6.18.: The IO Board is a custom and open source electronic board encapsulating two USB2AX for motor control and one Arduino to easily extend the sensorymotor space if needed.



Processor	Samsung Exynos4412 Prime Cortex-A9 Quad Core 1.7Ghz with 1MB L2 cache
Memory	2048MB(2GB) LP-DDR2 880Mega data rate
3D Accelerator	Mali-400 Quad Core 440MHz
Video	supports 1080p via HDMI cable(H.264+AAC based MP4 container format)
Video Out	micro HDMI connector
Audio	Standard 3.5mm headphone jack HDMI Digital
LAN	10/100Mbps Ethernet with RJ-45 Jack (Auto-MDIX support)
USB2.0 Host	High speed standard A type connector x 3 ports
USB2.0 Device	ADB/Mass storage(Micro USB), Host mode is possible if the PCB Rev is 0.5 or higher.
Display	HDMI monitor
IO Port	GPIO, UART, I2C, SPI(Board Revision 0.5 or higher)
Storage (Option)	MicroSD Card Slot eMMC module socket
Power (Option)	5V 2A Power
System Software	Linux : Xubuntu 13.10 or latest version Android : u-boot 2010.12, Kernel 3.0.x, Android 4.x Full source code is available now.
PCB Size	83 x 48 mm
Weight	48g including the heat sink

Fig. 6.19.: Hardkernel Odroid U3 computer board.

It is the case with one of the most famous micro-computer, the raspberry pi. The first trials we did with Pypot were really disappointing on the performance level. As we can see in section 7.4.2, it takes about 6-10 ms just to read and write a motor position (mostly computing) while it is only 2 ms on a normal computer (mostly serial communication). Therefore we oriented our choice toward the Hardkernel Odroid U3 board those performance with pypot are similar to a classic computer (see Fig. 7.5).

The Hardkernel Odroid U3 (see Fig. 6.19) is a low-cost ($65) and powerful Linux computer embedding a 1.7GHz Quad-Core processor and 2GByte RAM while being very small (83 x 48 mm) and lightweight (48 grams) (see Tab. 8.2 for the detail of all specifications).

Among the plug-n-play small computers, the Odroid U3 is currently the most suitable board for our application with regards the size, the computing power and the I/O positions. Yet as we will explain in the limitations part (see section 6.9.4), this solution is still not perfectly satisfactory and the use of plug-n-play computers raises a lot of integration problems.

6.7.3 Display

The video out port on all new mini computer boards such as Raspberry Pi, Beagle board or Odroid boards is an HDMI port. Finding a small screen (< 7inch) compatible with an HDMI input is really hard and currently only one project exists. The manga-screen (see Fig. 6.20) is an open source (CC-BY-SA[24]) multi-purpose, HDMI-compatible LCD screen. This board is developed by Elias Bakken and works with a 4.3 inch screen (480x800px) made by Sharp[25].

The integration of a LED matrix panel would be easier but it would require creating drivers for the display. Using a HDMI display connected to a Linux computer allows users to easily display information or animation on the screen as if it were on their monitor. Therefore users are free to use any tools they like such as Processing, OpenGL, VLC or whatever. This flexibility would not be possible with a matrix LED panel which would have required controlling the information displayed with Arduino programming.

[24] see REF
[25] Sharp LQ043Y1DX07

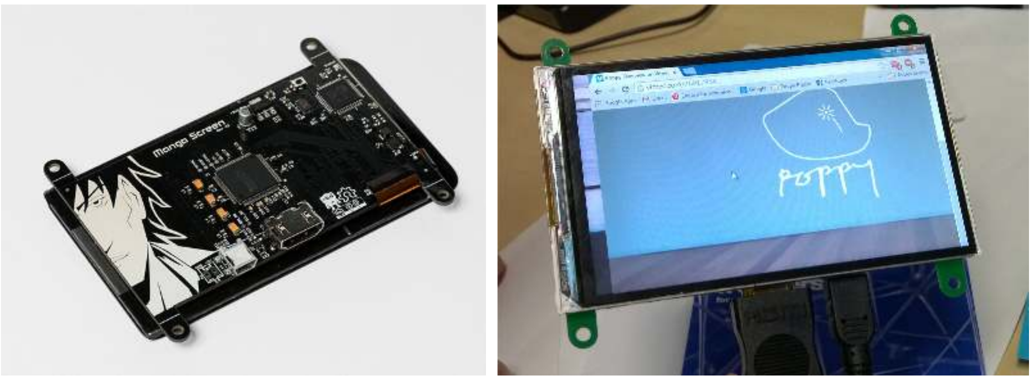


Fig. 6.20.: The future face display embedded in Poppy is an open source project called manga-screen (see `https://bitbucket.org/intelligentagent/manga-screen/`). This project is the only one we have found allowing to have both a 4.3inch display and HDMI connexion.

6.7.4 Power Supply

Power board

Power for the current Poppy is supplied by Robotis. This solution is really low-cost in the tens of euros range but it is limited. Indeed, the Robotis power supply provides directly 12V@5A dans is plugged directly into the robot. Yet the wire is short and not really convenient. It would be a better solution to have on board active components allowing the conversion of a wide range of power supplies to the one needed for Poppy. In particular, we could be compatible with a standard laptop power supply (18-22V).

This work is still in progress and the chosen solution will be presented in the final version of this thesis.

The battery issue

One issue associated with the batteries is the mass. Indeed 3.6V battery cell weights around 45 grams and we need at least 4 cells to supply the 12V needed for Poppy, thus a 14,4V pack weights almost 200 grams (see Fig. 6.21b). For comparison a MX-28 Dynamixels weights 72grams so a battery pack suitable for Poppy weighs approximately the same mass as 3 motors. In addition, the overall size is quite big with 18mm x 65mm x 18mm (see Fig. 6.21a) and makes the integration complicated in a multi-articulated and small robot like Poppy.

TYVA Part Number	Nominal Voltage	Nominal Capacity	Weight	Dimensions (L x l x h) mm	Discharge current max(23°C)	Charging current max(23°C)	AC Impedance (max)(1kHz mΩ)	Chemistry	Cycles Nbr	Energy Density (Wh/kg)	Volumetric Density (Wh/l)
ENERGIE											
[illegible]	3.2 V	1.7 Ah	40 g	18*65	1.7 A	3.4 A	<50	LFP	1000	138	331
VF 1865-SC1	3.6 V	2.2 Ah	45 g	18*65	10 A	20 A	<35	NCA	300	188	453
ME 1865-SC1	3.7 V	2.25 Ah	44 g	18*65	10 A	2.15 A	<35	NMC	700	185	468
VE 1865-LV1	3.6 V	2.8 Ah	47 g	18*65	5.2 A	2.6 A	<100	NMC	300	200	524
VE 1865 SS3	3.7 V	2.8 Ah	47 g	18*65	5.6 A	2.6 A	<65	LCO	500	220	580
ME 1865-PC1	3.6 V	2.9 Ah	46 g	18*65	6 A	2 A	<30	NCA	700	214	600
ML 1865-PC2	3.6 V	2.9 Ah	46 g	18*65	10 A	2 A	<25	NCA	700	214	600
[illegible]	3.6 V	3.35 Ah	47 g	18*65	6 A	[illegible]	<100	NMC	500	240	678
[illegible]											
FP1865-LN1	3.3 V	1.1 Ah	39 g	18*65	30 A	5 A	9	LFP	300	93	203
VP 1865 SS1	3.7 V	2 Ah	46 g	18*65	20 A	4 A	<18	NMC	500	-	-
VP 1865 SS2	3.6 V	2.2 Ah	46 g	18*65	20 A	4 A	20	NMC	700	-	-
VP1865-LV1	3.6 V	2.5 Ah	46 g	18*65	20 A	4 A	13	N-Hich	-	-	-

Fig. 6.21.: Example of batteries we can find on the market.

Poppy is not yet able to walk by itself, being energetically autonomous does not seem a high priority. Thus we chose to not include batteries in Poppy's current electronic architecture. Yet, we hope the community will try to address this challenge and maybe find original and suitable solutions.

However, using Poppy away from any power source is still possible. Is it indeed rather simple to connect external 12V batteries. These batteries can be easily found on the internet and have already been tested with Poppy for an artistic project where Poppy had to be surrounded by nature[26].

6.8 Aesthetic design of the head

A lot of effort has been put into the design and aesthetic of Poppy's head (see Fig. 6.23) because it is both its identity and main communication apparatus. From an aesthetic point of view, its design was inspired of course by existing robots, but also by animals, objects and art. Insights into our main inspirations are displayed on the board in Fig. 6.22. We tried to achieve a design that is cute, expressive and above all, simple.

Yet because of the multi-articulated vertebral column, there is only free room in Poppy's head to embed all electronics components needed. Therefore strong technical constraints were imposed because all electronic architecture plus the communication sensorimotor apparatus composed by a wide 4.3" screen, cameras, and audio devices all have to be embedded in the head. These components strongly constrained the design of the robot. Especially the screen, which necessitated a large flat part on the face. Obtaining a nice, rounded head shape with such constraints was rather difficult and require several iterations before obtaining the first correct finished version (see Fig. 6.23).

[26] see associated topic on our forum: `https://forum.poppy-project.org/t/projet-artistique-en-pleine-nature-avec-poppy/298`

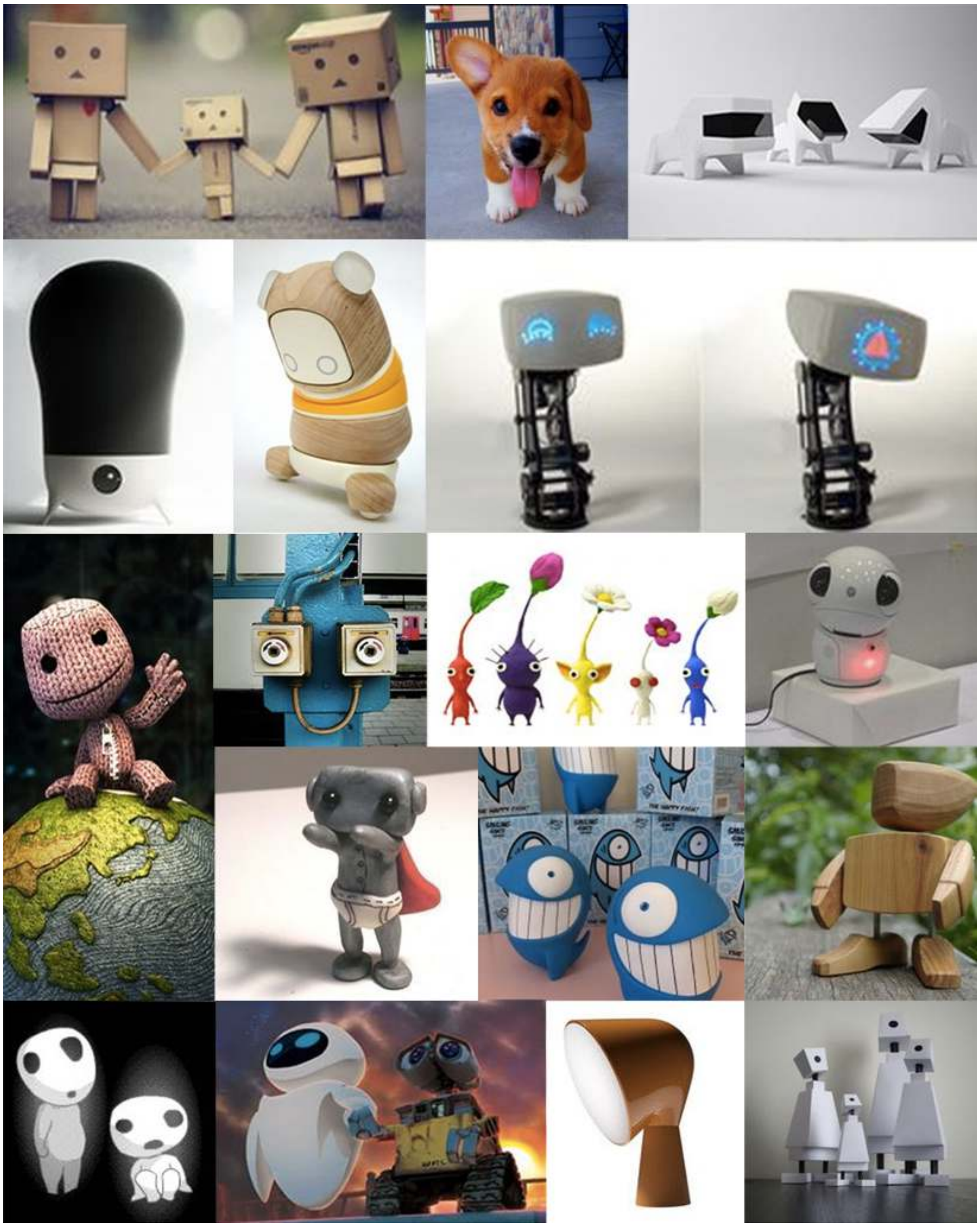
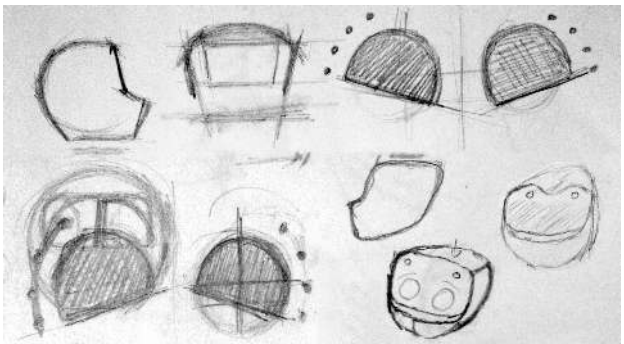


Fig. 6.22.: Complete board available on pinterest http://www.pinterest.com/matthieulapeyre/robot/



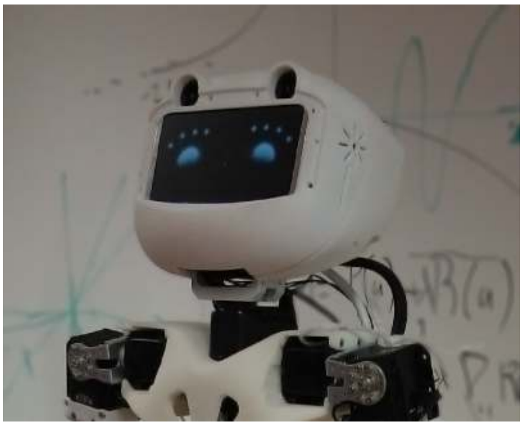
(a)

(b)

(c) First clay sculpture

(d) First CAO model

(e) Poppy beta



(f) Screen with basic eyes display

Fig. 6.23.: Evolution of Poppy's head from the first sketches to the Poppy beta version.

Fig. 6.24.: A preview of the new design for Poppy's head. As we want to show expressiv eyes on the screen, we have replaced the two cameras with big lenses of Poppy beta by a small one, thus cameras are not misunderstood as Poppy's "eyes".

This process firstly involved several sketches showing the main ideas of the desired design. But the transfer to CAD modelling was quite complex; these kinds of shape are rather difficult to design using parametric tools. The use of clay sculpting was very helpful in the transition from the 2D drawing to the 3D shape.

However, in the first beta version showed in Fig. 6.23, there is a major design error. Indeed our desire was to have a screen to create and explore freely expressive eyes but the use of two visible cameras changed the way people saw Poppy's head. Of course, when people see two cameras they consider them to be the eyes of the robot and therefore extrapolate that the screen may be the mouth or another face part.

We are currently working on the new design of Poppy's head and we simply addressed this issue by replacing the two big camera by a small one with a pinhole lens, which can be hidden on Poppy's face see Fig. 6.24.

While this work it is still in progress, we will update this section afterwards with a blueprint of the electronic integration and final design explanation.

6.9 Limitations

The design and the choices we made have raised several limitations:

6.9.1 Simplicity

Creating a robot for which one of the main objectives is to be easy to duplicate imposes strong constraints on the design. As we saw in chapter 3, humanoid robots often have a complex design involving many components. Achieving such complexity is not possible if we want to keep the robot easy and quick to assemble by non-expert users. Therefore, Poppy's mechanics are simple, with only one part per motor. However there are several joints whose performance could be improved by changing the design. For example, just by adding a complementary mechanism such as a reduction gear, we could increase the applied torque on critical joint (e.g. knee, ankle, hip).

6.9.2 No hands

The current version of Poppy does not have articulated hands. The grasping ability was not a top priority and is challenging, so we preferred to only design fixed hands. There are several laboratories working on this topic so we hope one of them will be interested in contributing to the Poppy project and design articulated hands. We have already been in contact with the Bristol Robotics Lab and the Open hand project for this purpose.

6.9.3 Motor modularity

Parametric modelling is great if we want to create variation of the same pattern with different dimensions. Thus, we can easily create configurations of all Robotis Dynamixel motors. Yet, if we are interested in using another motor brand, the configuration will not be direct and we will have to redo a part of the design process to ensure the compatibility.

6.9.4 Electronic

The electronics we developed are compatible with the exploration of morphology for sensory space because they have a lot of I/O ports to extend Poppy's sensorimotor space. Yet there are several limitations that make the final solution unsatisfactory.

Commonly used electronic components are not designed for robotic integration but rather for building small personal computers. Thus even if the electronic boards are often quite small, they have big common connectors such as USB, HDMI and Ethernet, which are of course never placed exactly where they should be for integration on the robot. Above all, cables are really annoying; they take up a surprising amount of room (connector size, the wire length and they are heavy) while being totally useless for our applications.

Also the size of the IO Board is finally fairly big, more so than expected and while it is still compatible with Poppy's design, it is too large, with a shape that is too strange to be appropriate for other open source projects.

Great open source projects keep their work modular so other projects can use one or several modules directly. Therefore the IO Board is not compatible with such principles and we are currently moving toward building modular Poppy electronics (we will discuss this new version in the discussion see section 11.4). Yet this IO Board was the first electronic board developed in the Poppy project and through the experience we acquired a better understanding of electronic integration.

6.10 Conclusion

Thanks to the methodology presented in the chapter 5, the design and production of a completely new humanoid robot has been very fast and low cost. Indeed, the project began at the end of May 2012 and the first fully-functional version of Poppy (the one we can see in Fig. 6.15) was presented at the end of September 2012 at Collège de France. The cumulative work of all the different people involved was equivalent to about 8 months and cost less than €10,000 in material.

Because of the lack of easy and cheap tooling to produce a custom electronics board, the new design of the electronic architecture took more time and several elements are still in progress for.

Yet Poppy cost about €8,000 and it only takes one or two days to assemble (see Fig. 6.25). Also its morphological design allows for modularity both for the mechanics, as the parametric parts can be easily customized and reprinted, and for the electronics thanks to the compatibility with the Arduino environment. This modularity is completed by the control library which will be presented in the next chapter.

Examples of variations in Poppy's morphology will be presented in chapter 8. Also thanks to its simple design, Poppy can be relevant for educational and artistic purposes. Applications will be presented in chapter 9 and chapter 10.

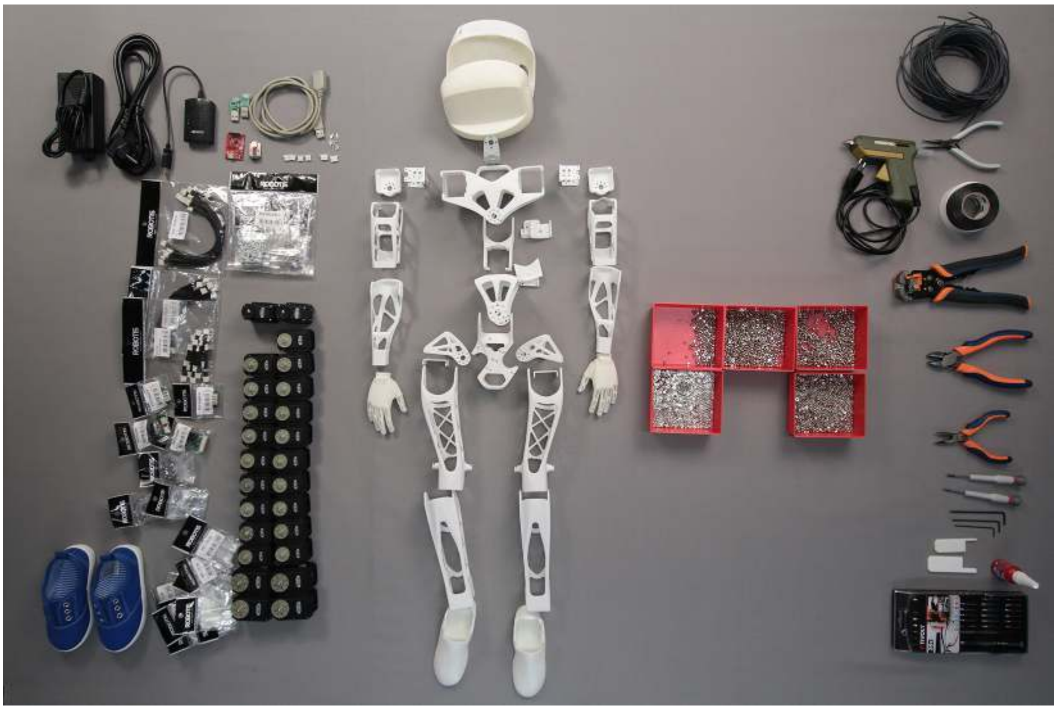


Fig. 6.25.: Time lapse of the assembly is available here: `https://vimeo.com/96262428`

```python
#import time
import pypot.utils.pypot_time as time
import logging

from ..primitive.manager import PrimitiveManager

logger = logging.getLogger(__name__)

class Robot(object):
    def __init__(self, motor_controllers=[], sensor_controllers=[]):
        """
        :param list motor_controllers: motors controllers to attach to the robot
        :param list sensor_controllers: sensors controllers to attach to the robot
        """
        self._motors = []
        self.alias = []

        self._controllers = motor_controllers + sensor_controllers

        for controller in motor_controllers:
            for m in controller.motors:
                setattr(self, m.name, m)

            self._motors.extend(controller.motors)

        for controller in sensor_controllers:
            for s in controller.sensors:
                setattr(self, s.name, s)

        self._attached_primitives = {}
        self._primitive_manager = PrimitiveManager(self.motors)

    def close(self):
        """ Cleans the robot by stopping synchronization and all controllers."""
        self.stop_sync()

    def __repr__(self):
        return '<Robot motors={}>'.format(self.motors)

    def start_sync(self):
        """ Starts all the synchonization loop (sensor/effector controllers). """
        [c.start() for c in self._controllers]
        [c.wait_to_start() for c in self._controllers]
        self._primitive_manager.start()

        logger.info('Starting robot synchronization.')

    def stop_sync(self):
        """ Stops all the synchonization loop (sensor/effector controllers). """
        if self._primitive_manager.running:
            self._primitive_manager.stop()
        [c.stop() for c in self._controllers]

        logger.info('Stopping robot synchronization.')

    def attach_primitive(self, primitive, name):
        setattr(self, name, primitive)
```

7 Pypot: An open source modular python library for robot control

7.1 Introduction

Poppy was designed to be a research platform for freely exploring morphological variations. Although Poppy meets "hardware" needs, control of the platform is also crucial. Whereas more conventional robots have a fixed mechanical and sensorimotor morphology, having a platform that can be fully modified changes the low-level control architecture issues. We decided to develop a new robotic control library. Called pypot (see Fig. 7.1), this library mostly developed by Pierre Rouanet, is adapted to the challenges of morphological exploration and experimentation. Moreover, development begun early in the design of Poppy, and shares the same guidelines and objectives i.e. being robust, modular, versatile and easy to use.

Fig. 7.1.: The logo of the pypot library.

To reach these goals, pypot is a library written in Python and developed to make it easy and fast to control custom and modular robots. In particular, pypot has been designed with a simple and easy to use API, a fully modular and customizable architecture, and key-features adapted to robot experimentation issues.

It also shares the open collaboration philosophy and is therefore distributed under an open source GPLv3 license. All sources are available on the associated GitHub repository: `https://github.com/poppy-project/pypot`.

Even if pypot has been developed within the context of the Poppy project, in the following sections, most of the code examples will involve an ergo-robot (see Fig. 7.2)

Fig. 7.2.: Ergo-robot are a 6 DoF serial robots with a stem shape those were developed in the Flowers Lab. In particular, they were involved for the Fondation Cartier exhibition (see chapter 10)

rather than Poppy to emphasize that pypot is a control library for any modular robot.

7.2 Control of custom robots made simple with pypot

One main preoccupation during the development of pypot was the achievement of a very easy-to-use library for the end user. For this purpose, pypot has been entirely written in Python to allow for fast development, easy deployment (cross-platform) and quick scripting by expert developers that are not necessarily expert developers. In particular, API is simple and permits to write complex behaviours with just a few lines of code.

This is made possible with a layered architecture based on:

- Fast and robust low-level API that directly encapsulates the communication protocol for setting and accessing hardware data.
- A controller that automatically ensures the update of sensorimotor data (get/set) at a predefined frequency. This method encapsulates the low-level API to prevent the writing of repetitive requests and optimize latency.
- Finally, a robot layer can generate automatically a whole robot API giving access to the whole sensorimotor space with just a few lines of code.

While low-level layers allow for modularity and customization, the high-level abstracts all the complexity into simple end user API. We will describe more in detail how this architecture works in the next sections.

7.2.1 The Dynamixel controller

Pypot handles the low-level communication with Dynamixel motors from Robotis. Using a USB communication device such as USB2Dynamixel or USB2AX, it opens serial communication with Robotis motors (MX, RX, AX) using communication protocols TTL or RS485. More specifically, it allows easy access (both reading and writing) to the different registers of any Dynamixel motors[1]. Those registers include values such as position, speed or torque.

While the Dynamixel Low-level IO provides access to all functionalities of the Dynamixel motors, it forces us to have synchronous calls, which can take a non-negligible amount of time. In particular, most programs will need to have a really fast read/write synchronization loop, where we typically read all motor positions, speeds, loads and set new values, while we would like to have higher level code that also computes those new values.

On top of the low-level, a Dynamixel controller can be added which defines synchronization loops that will read/write[2] the registers of Dynamixel motors at a predefined frequency automatically run in the background. Thus there is no need to wait for the answer of a read command to access data (this can take some time) so that algorithms with heavy computation do not encounter a bottleneck when values from motors must be known. The attributes of those "software" motors are automatically synchronized with the real "hardware" motors.

The controller is actually a module (see section 7.3.1) and can be changed according to the user's needs. Yet by default pypot has a base controller, which already defines synchronization loops, more exactly it:

- reads the present position, speed, load at 50Hz,
- writes the goal position, moving speed and torque limit at 50Hz,
- writes the PID or compliance margin/slope (depending on the type of motor) at 10Hz,
- reads the present temperature and voltage at 1Hz.

This controller embeds very useful synchronization loops, which should be enough for most users.

[1]The whole list of registers can directly be found on the Robotis website: `http://support.robotis.com/en/product/dynamixel/mx_series/mx-28.htm#Control_Table`

[2]Whenever one of the values is accessed, it is actually the most recent versions that have been read at the frequency of the loop.

7.2.2 Robot abstraction

The robot abstraction allows, from a configuration, to automatically generate both all low-level controllers and high-level accessors needed to control a whole robot. More precisely, through the use of the class Robot it is possible to:

- automatically initialize all connections (making the use of multiple USB2 serial connections transparent),
- define offset and direct attributes for motors,
- automatically define accessors for motors and their most frequently used registers (such as *goal_position, present_speed, present_load, PID, compliant*),
- define a read/write synchronization loop that will run in background.

The configuration, described as a Python dictionary[3], contains several important features that help build both the robot and the software to manage the robot. The important fields are listed below:

- controllers - This key holds the information pertaining to a controller and all the items connected to its bus.
- motors - This is a description of all the custom setup values for each motor. Meta information, such as the motor access name or orientation, is also included here. It is also there that we can set the angle limits of the motor.
- motorgroups - This is used to define the alias of a group of motors (e.g. *left_leg*).

To give a complete overview of what a config can look like, the Code 7.1 is an example of the config dictionary of a simple 6-DoF ergo-robot[4].

The robot abstraction encapsulates multiple Dynamixel controllers to read/write all the registers of a robot at a predefined frequency. The user only has to specify the configuration dictionary of his robot using the *from_config()* function. The robot configuration can also be loaded/saved as a JSON format.

Here is an example of how to create a robot:

[3]The configuration can be written in the Python script or can be loaded from any file that can be loaded as a dictionary (e.g. a JSON file).

[4]Since pypot 1.7, it is possible set the port to 'auto' in the dictionary. When loading the configuration, Pypot will automatically try to find the port with the corresponding attached motor ids.

```
ergo_robot_config = {
    'controllers': {
        'my_dxl_controller': {
            'port': '/dev/ttyUSB0',
            'sync_read': False,
            'attached_motors': ['base', 'head'], # You can mix motorgroups or individual motors
        },
    },

    'motorgroups': {
        'base': ['base_pan', 'base_tilt_lower', 'base_tilt_upper'],
        'head': ['head_pan', 'head_tilt_lower', 'head_tilt_upper'],
    },

    'motors': {
        'base_pan': {
            'id': 11,
            'type': 'RX-64',
            'orientation': 'direct',
            'offset': 22.5,
            'angle_limit': (-67.5, 112.5),
        },
        'base_tilt_lower': {
            'id': 12,
            'type': 'RX-64',
            'orientation': 'direct',
            'offset': 0.0,
            'angle_limit': (-90.0, 90.0),
        },
        'base_tilt_upper': {
            'id': 13,
            'type': 'RX-64',
            'orientation': 'direct',
            'offset': 0.0,
            'angle_limit': (-90.0, 90.0),
        },
        'head_pan': {
            'id': 14,
            'type': 'RX-28',
            'orientation': 'direct',
            'offset': 22.5,
            'angle_limit': (-67.5, 112.5),
        },
        'head_tilt_lower': {
            'id': 15,
            'type': 'RX-28',
            'orientation': 'indirect',
            'offset': 0.0,
            'angle_limit': (-90.0, 90.0),
        },
        'head_tilt_upper': {
            'id': 16,
            'type': 'RX-28',
            'orientation': 'indirect',
            'offset': 0.0,
            'angle_limit': (-90.0, 90.0),
        },
    },
}
```

Code 7.1: Example of a pypot configuration file. Here is the one of an Ergorobot.

```
import pypot.robot

# Load the configuration file
my_robot = pypot.robot.from_config('ergo_robot.json')

# Launch robot sensorimotor synchronization
my_robot = start_sync()
```

Then making the robot move is only one line of code:

```
my_robot.base_tilt_lower.goal_position = 120
```

In this example, the motor *base_tilt_lower* will not reach the 120 degree position, but actually 90 degree because its config file (Code 7.1) set the angle limit to *[-90, 90]*.

Therefore the user can set up their robot with just few lines of code and then use it safely just scripting the behaviour they want to achieve.

7.2.3 Move recording

Pypot involves a really convenient, yet simple, feature for recording movements. Indeed, when a robot's motor is compliant (Dynamixel property), a user can demonstrate a desired gesture by physically moving the motor position. Those Moves are simply defined as a sequence of positions.

The move module can be used to:

- record moves,
- play moves,
- save/load them on the disk.

Given the same ergo-robot configuration, the recording of a gesture at a 50Hz framerate moving on all motors for 5 seconds can be simply done using the following code:

```
import time
import pypot.robot

from pypot.primitive.move import MoveRecorder, Move, MovePlayer

ergo = pypot.robot.from_config(...)
ergo.start_sync()

```

```
move_recorder = MoveRecorder(ergo, 50, ergo.motors)

ergo.compliant = True

move_recorder.start()
time.sleep(5)
move_recorder.stop()
```

This move can then be saved on disk:

```
with open('my_nice_move.move', 'w') as f:
    move_recorder.move.save(f)
```

And loaded and replayed:

```
with open('my_nice_move.move') as f:
    m = Move.load(f)

ergo.compliant = False

move_player = MovePlayer(ergo, m)
move_player.start()
```

This feature seems very simple and anecdotal but is actually one of the most useful pypot features for non-expert users. Indeed it allows to physically "program" the robot and is very intuitive for artists. We will show a demonstration of such a use with Poppy in the chapter 10.

7.3 Modular environment

Figure Fig. 7.3 shows the pypot 2.x architecture and especially its modularity. Indeed pypot has a modular architecture both for the low-level communication with the robot and for the high-level behaviour control. This modularity allows pypot to be adapted to:

- morphological exploration because it permits to switch between several low-level controllers with respect to the hardware properties of the robot (i.e. sensors, motors),
- scientific experimentation because its high-level modularity allows to easily run behaviour and extend the control to other libraries.

This modularity is expressed through the I/O controllers and the primitive paradigms, which will be, discussed in detail in the next sections:

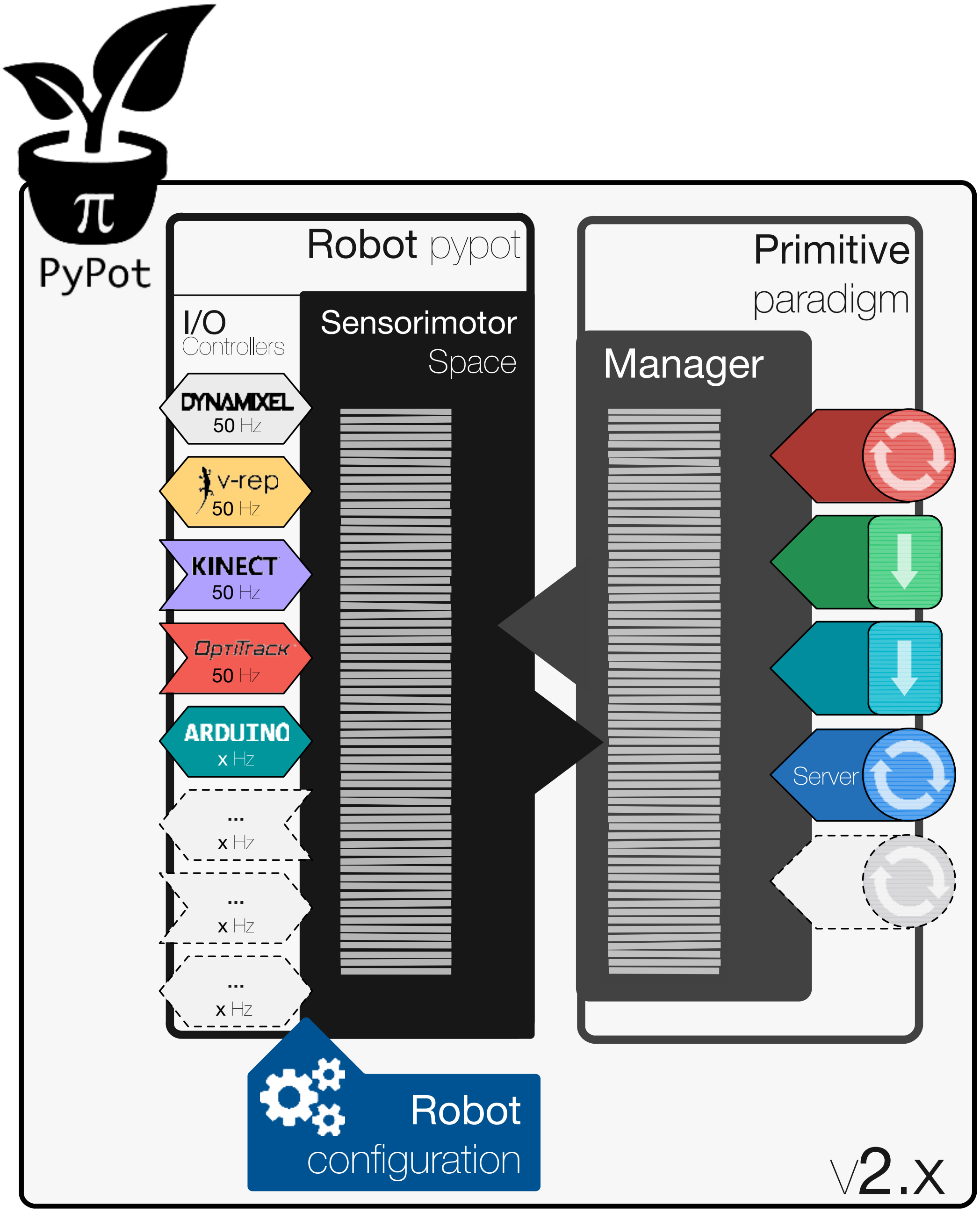


Fig. 7.3.: Diagram showing the modular structure of the pypot library. I/O controllers handle communication with devices while behaviour can be scripted thanks to the primitive paradigms. Between, pypot synchronizes all data and generates easy-to-use accessors.

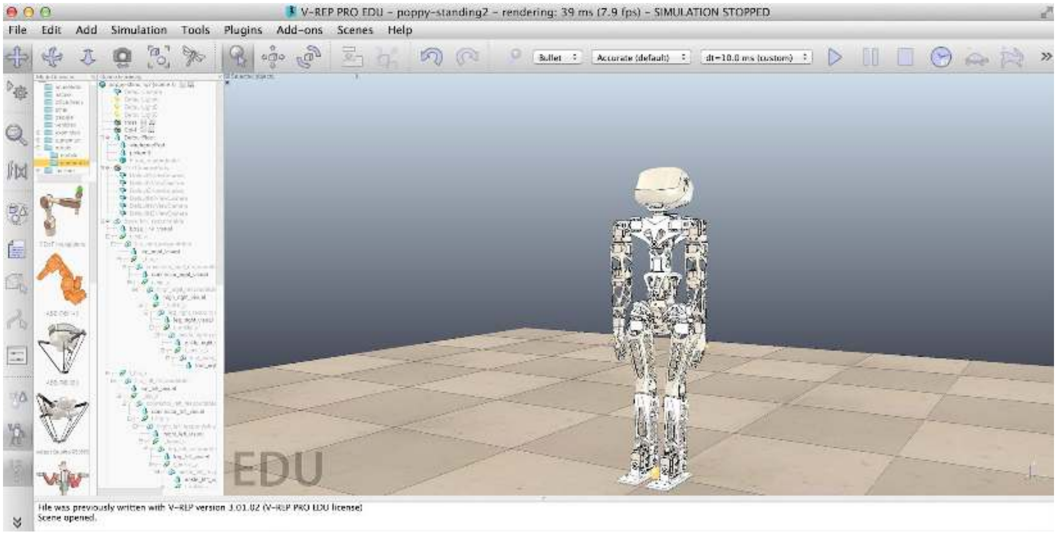


Fig. 7.4.: Pypot can be used on real robot as well as simulated ones. Its current version is compatible with the V-REP simulator and it is possible to switch from real world to simulation in just one line of code.

7.3.1 I/O controllers

The low-level Dynamixel controller presented previously (see section 7.2.1) is actually an instance of an I/O controller. The I/O controllers are the interface between real world data acquisition and the pypot core. They constitute the low-level modular part of pypot and allow for anyone to create a custom controller adapted to particular robot properties.

A brilliant example of this modular I/O controller architecture is the control of the robot either in the simulator or the real world just by switching between I/O controllers:

Switching between the simulator and the real world

As it is often easier to work in simulation rather than with the real robot, Pypot has been linked with the V-REP simulator[5]. It is described as the “Swiss army knife among robot simulators” and is indeed a very powerful tool to quickly (re)create robotics setup. Moreover, we chose to integrate this particular simulator first because it shares distinctive features with the Poppy project, i.e. being cross-platform, easy-to-use, extensible and open source[6]

[5] http://www.coppeliarobotics.com/features.html

[6] Actually V-REP has a double license Commercial/GPL so either one pays and can keep his customizations or has to release all modification under GPL.

The connection with V-REP was created using the I/O controllers presented previously and the V-REP's remote API. Thanks to the low-level modularity of pypot, it permits to seamlessly switch from your real robot to the simulated one because it only requires switching the low-level I/O controller from the Dynamixel (*DxlController* class) one to the V-REP one (*VrepController* class).

The switch between the simulation and the real robot is possible with a single line of code, and in most case, only requires changing the way the robot is instantiated:

```
# Working with the real robot
import pypot.robot

poppy = pypot.robot.from_config(config)
poppy.start_sync()

poppy.walk.start()

# Working with the simulated version
import pypot.vrep

poppy = pypot.vrep.from_vrep(config, vrep_host, vrep_port, vrep_scene)
poppy.start_sync()

poppy.walk.start()
```

In addition, it provides some extended features relative to the needs we have in a robot simulator, among them: load a scene, start/stop/restart a simulation, pause/resume the simulation, get an object position/orientation. Yet not all Dynamixel registers have their V-REP equivalent. For the moment, only the control of the position is used but it will be extended in the future. Also more advanced features can be easily added thanks to the controller abstraction.

Custom I/O controllers

The I/O controller is actually defined by two classes:

- the IO class defining how to communicate with an object (i.e. motor or sensor) and obtain its data,
- the controller class defining all object properties the robot can access and control.

It is therefore possible to extend the number of controllers to any connected object. For example, it could be used to change the type of motors used and replace the

Robotis motors with more low-cost ones, or to control a robot with a simulator other than V-REP, and the switch should be as straightforward.

Automatic generation of the robot

In the same way that the pypot robot class can automatically create a robot based on Dynamixel motors and generate an easy-to-use API, it can handle the variability of the I/O controllers. Indeed the desired I/O controllers can be indicated during the robot configuration then the robot will be generated in accordance with the specified controllers. The v-rep simulator is an example of such use.

Thus the pypot robot can handle multiple controllers both for motor control and for the sensors acquisition (e.g. IMU, tactile, camera).

7.3.2 Primitive paradigms

The high-level modularity of pypot is expressed by the primitive paradigms. We call a Primitive any simple or complex behaviour applied to a Robot. A primitive can access all sensors and effectors in the robot. A primitive is supposed to be independent[7] from other primitives. In particular, a primitive is not aware of the other primitives running on the robot at the same time. We imagine those primitives as elementary blocks that can be combined to create more complex blocks in a hierarchical manner.

To ensure this independence, the primitive is running in a sort of sandbox. More precisely, this means that the primitive does not have direct access to the robot. It can only request commands (e.g. set a new goal position of a motor) to a Primitive Manager, which transmits them to the "real" robot. As multiple primitives can run on the robot at the same time, their request orders are combined by the manager[8].

The manager uses a filter function to combine all orders sent by primitives. By default, this filter function is a simple mean but you can choose your own specific filter (e.g. add function).

[7] The independence of primitives is really important when one creates complex behaviours - such as balance - where many primitives are needed. Adding another primitive - such as walking - should be direct and not force the rewriting of everything. Furthermore, the balance primitive could also be combined with another behaviour - such as shooting a ball - without modifying it.

[8] The primitives all share the same manager. In further versions, we would like to move from this linear combination of all primitives toward a hierarchical structure and have different layer of managers.

```python
import time

import pypot.primitive

class DancePrimitive(pypot.primitive.Primitive):
    def run(self, amp=30, freq=0.5):
        # self.elapsed_time gives you the time (in s) since the
            primitive has been running
        while self.elapsed_time < 30:
            x = amp * numpy.sin(2 * numpy.pi * freq * self.elapsed_time)

            self.robot.base_pan.goal_position = x
            self.robot.head_pan.goal_position = -x

            time.sleep(0.02)
```

Code 7.2: Example of a script primitive, this kind of primitive is called by the start() method then runs its content and stops then reaching the end of the script.

```python
import time

import pypot.primitive

class LoopDancePrimitive(pypot.primitive.LoopPrimitive):
    # The update function is automatically called at the frequency given
        on the constructor
    def update(self, amp=30, freq=0.5):
        x = amp * numpy.sin(2 * numpy.pi * freq * self.elapsed_time)

        self.robot.base_pan.goal_position = x
        self.robot.head_pan.goal_position = -x
```

Code 7.3: Example of a Loop primitive, this primitive is also called by the start() method but, unlike the script primitive, loops until the stop() method is called.

To write a primitive, the user only needs to create a subclass of the Primitive class. It provides basic mechanisms (e.g. connection to the manager, setup of the thread) to allow the direct "plug" of novel primitives to a robot and run it.

Currently there are two kinds of primitives.

The primitive can be start(), stop(), pause() and resume(). Unlike a regular python thread, a primitive can be restarted by calling the start() method again.

When overriding the Primitive, you are responsible for correctly handling those events. For instance, the stop method will only trigger the *should_stop* event that you should watch in your run loop and break when the event is set. In particular, you should check the should_stop() and should_pause() in your run loop. You can also use the wait_to_stop() and wait_to_resume() to wait until the commands have actually been executed.

```
my_robot = pypot.robot.from_config(...)
my_robot.start_sync()

dance = LoopDancePrimitive(my_robot, 50)
# The robot will dance until you call dance.stop()
dance.start()
```

Code 7.4: Simple call of a primitive.

```
my_robot = pypot.robot.from_config(...)
my_robot.start_sync()

my_robot.attach_primitive(DancePrimitive(my_robot), 'dance')
my_robot.dance.start()
```

Code 7.5: Here the same primitive is first attached to the robot, by doing so, the robot is "aware" of this primitive. In particular, attached primitives can be called or stopped by the robot or another primitives

The move feature described in section 7.2.3 is actually based on the pypot primitive paradigm. More precisely, the *MoveRecorder* and *MovePlayer* are defined as a subclass of *LoopPrimitive*.

7.3.3 Extensible API

We added the possibility to remotely access and control your robot through the TCP network. This can be useful both to work with client/server architecture (e.g. to separate the low-level control running on an embedded computer and higher-level computation on a more powerful computer) and to allow you to plug your existing code, written in another language, into the pypot's API.

We defined a protocol that allows all the robot variables and methods (including motors and primitives) to be accessed via a JSON request. The protocol is entirely described in the section Protocol below. Two transport methods have been developed so far:

HTTP server

The *HTTPServer* is based on the bottle python framework (http://bottlepy.org/). We have developed a sort of REST API based on the protocol described above:

- GET /motor/list.json
- GET /primitive/list.json

```python
import urllib2
import json
import time

import pypot.robot
import pypot.server

robot = pypot.robot.from_config(...)
robot.start_sync()

server = pypot.server.HTTPServer(robot, host, port)
server.start()

time.sleep(1) # Make sure the server is really started

url = 'http://{}:{}/motor/list.json'.format(host, port)
print urllib2.urlopen(url).read()

url = 'http://{}:{}/motor/base_tilt_lower/goal_position'.format(host, port)
data = 20.0
r = urllib2.Request(url, data=json.dumps(data), headers={'Content-Type': 'application/json'})
print urllib2.urlopen(r).read()
```

Code 7.6: Example of the use of the HTTP server

- GET /motor/<name>/register.json (or GET /<name>/register.json)
- GET /motor/<name>/<register> (or GET /<name>/<register>)
- POST /motor/<name>/<register> (or POST /<name>/<register>)
- POST /primitive/<prim_name>/call/<meth_name> (or GET /<prim_name>/call/<meth_name>)
- POST /request.json

ZMQ server

The Zmq Server used a Zmq socket to send (resp. receive) JSON request (JSON answer). It is based on the REQ/REP pattern. So you should always alternate sending and receiving. It will probably be switched to PUB/SUB soon.

Zmq has been chosen as it has been bound to most languages[9] and can thus be used to connect code in other languages to pypot. For instance, we used it to connect RLPark[10] to pypot.

[9] `http://zeromq.org/bindings:_start`

[10] RLPark is a Java reinforcement learning library developed by Thomas Degris to experiment with online learning algorithms on robots and benchmarks, see `http://rlpark.github.io/`

```
import zmq

#As an example of what you can do, here is the code of getting the load
    of a motor and changing its position:

robot = pypot.robot.from_config(...)
robot.start_sync()

server = pypot.server.ZMQServer(robot, host, port)
server.start()

c = zmq.Context()
s = c.socket(zmq.REQ)
s.connect('tcp://{}:{}'.format(host, port))

req = {
    'get': {motor_name: ('present_load', )},
    'set': {motor_name: {'goal_position': 20.0}}
}

s.send_json(req)
answer = s.recv_json()
```

Code 7.7: Example of the use of the ZMQ server.

The Zmq server is faster than the HTTP version and should be preferred when working with high frequency control loops.

In particular, those for which the extension could be used to create a monitor interface using web technology.

7.4 Discussion

7.4.1 Why not using ROS instead ?

One famous robotics software is the Robot Operating System[11]. ROS is very widespread in the research community and one could quite rightly question the choice we made in creating a whole new architecture rather than using a well-known and efficient one.

Actually ROS has several aspects that do not fit in with the objectives we have. Indeed, using ROS is not a simple task. The installation is only recommended on one precise Ubuntu distribution[12], and the whole architecture is of course powerful but difficult to set and maintain. Also changing the low-level is complex and requires

[11]The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms.

[12]ROS Hydro only supports Precise, Quantal, and Raring for debian packages `http://wiki.ros.org/indigo/Installation`

hacks, sometimes not very elegant ones. Finally, ROS require high computational power, which make it difficult to embed on small computers.

In the Poppy project, we are trying to create a multidisciplinary robotic community, involving as a result, non-robotics experts. Thus, we endeavour to bring down the complexity of using Poppy. To reach this goal, we need a simple user interface/API and a lightweight library, easy to setup whichever system used. Finally, ROS is very modular for high-level but we desire to have modular low-level control.

For all the reasons mentioned above, we considered it to be more simple and effective to create a novel lightweight robot control library rather than adapting ROS to our needs.

7.4.2 Limitations

The pypot library is currently rather mature and robust, yet there remain some limitations that are challenging to a variable degree.

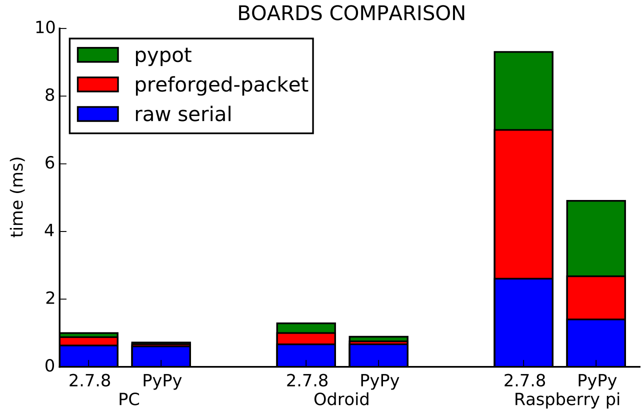
Performances

Pypot is written in python because it allows fast development and simple API for non-expert users. Also the main goal of pypot is to be an easy-to-use prototyping environment so users can run complete experiments with a custom robot in a couple of minutes to few hours. These choices imply a number of layers and multiple calls of functions, slowing the general execution of the code.

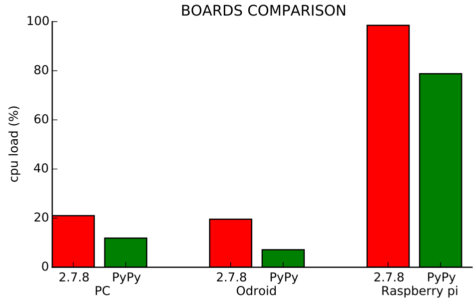
While it is not a problem on modern personal computers, it is more limiting on very light configurations (see Fig. 7.5). For example, running pypot on a Raspberry Pi takes all CPU resources and the sensorimotor acquisition of Poppy at $50Hz$ (i.e. $< 20ms$) is not achieved. The performance could be improved by splitting pypot with the low-level executed on

The primitive paradigms: The Good, the Bad and the Ugly

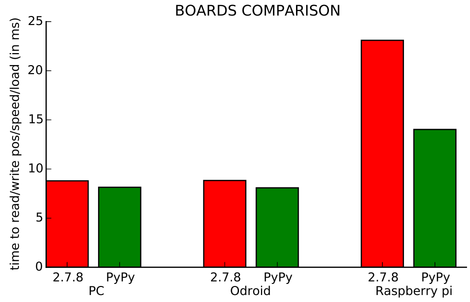
The use of primitives (see section 7.3.2) raises some complicated limitations. Indeed it is really convenient to run complex behaviours by splitting them into more simple ones but the interaction between primitives is really complex to manage and can lead to undesired behaviours. Indeed, while multiple primitives can request to change the same value, the final value is the result of the combination (e.g. mean, sum) of several primitives.



(a) Time spent to synchronise one motor



(b) CPU load when pypot run on Poppy to synchronize the 25 motors



(c) Time spent in each synchronization loop for 25 motors

Fig. 7.5.: These charts show a comparison between a normal computer, the odroid U3 board and raspberry pi board, concerning the time spent to synchronize data of Dynamixels motors with pypot when interpreted by Python 2.7 and Pypy. We can notice the Odroid U3 performances are equivalent to a standard computer while the raspberry pi is clearly slower.

It is a case of emergent behaviour[13] problem as explained in chapter 2, but at the control software level. It is really interesting because it forces us to design for emergence as explained by Steels (Steels, 1990), yet it is still challenging to explain to the end user and in particular to non-expert ones.

7.5 Conclusion

As explained in this chapter, Pypot is a modular robot control library, simple to use and extendable to the needs of users. Moreover its very modular low-level permits

[13]Finding the rules that can lead to a desired behaviour is more difficult than explaining the real, observable complex behaviour of an agent interacting with its environment. Since the behaviour itself cannot be preprogramed but is always the result of an agent-environment interaction, we must design for emergence rather than directly for a specific behaviour (Pfeifer and Bongard, 2006).

to easily manage morphological variability while its high-level primitive paradigm permits to quickly run more or less complex behaviour on the robot.

Pypot is open source and distributed under GPLv3 license. All sources are available on the GitHub repository of the project: `https://github.com/poppy-project/pypot`. Also the complete documentation can be accessed here: `https://poppy-project.github.io/pypot/`.

Part III

Applications

8 Changing Poppy's morphology

Poppy has been designed to be a new experimental platform opening up the possibility of systematically studying the role of morphology in sensorimotor control, in human-robot interaction and in cognitive development. Indeed, as we discussed in chapter REF, a suitable design of a robot's morphology can greatly simplify control problems, increase robustness, and open the way for new modes of interaction with the physical and social world. Thus, being able to study the body as an experimental variable, something which can be systematically changed and experimented with, is of paramount importance. Yet, until recently it was complicated because building a robot relied on heavy and costly manufacturing techniques, but 3D printing has changed the landscape of possibility.

We introduced a design methodology relying on the use off-the-shelf components and Arduino electronic architecture, for which 3D printing plays a central role in the production of mechanical parts (see chapter 5).

Poppy transposes this methodology to humanoid robotics, and it is now possible to explore new body shapes in just a few days. In addition, its size, weight and power actuation highly reduce the risk of self-damage if a programming error occurs, which means experimentation can be directly conducted in the real world without having to either use physical simulator or build a heavy experimental setup.

In this chapter, we present several experiments aiming to show through examples how the Poppy's morphology can be easily and quickly hacked to explore morphological variants in the real world. These experiments will be presented to show different aspects:

1. Experimenting the role of morphology (section 8.1)
2. Fast exploration of morphological variants (section 8.2)
3. Adding new sensors to Poppy (section 8.3)

8.1 Experimental evaluation of the role of the morphology: the thigh shape

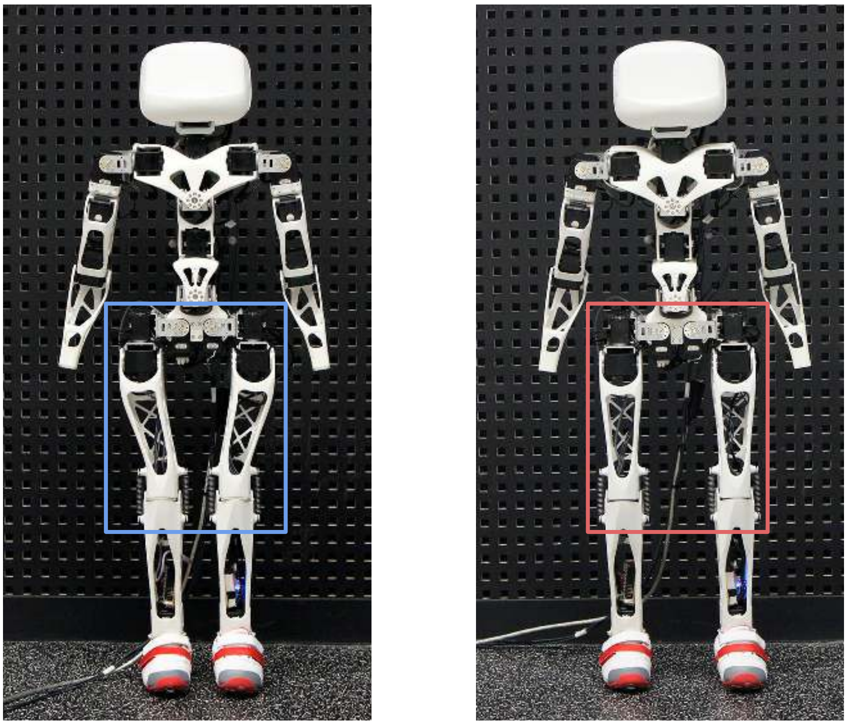
The role of morphology in robot bipedal locomotion has been particularly explored through the research on passive dynamic walkers (Wisse et al., 2007). The most famous example concerns Tad MacGeer's work (McGeer, 1990). Thanks to the understanding of the intrinsic dynamics of its structure, McGeer has managed to create a 2D biped robot capable of producing several steps without any controller or motor. The only control of this robot is obtained through the interaction between the intrinsic inertia of the structure and gravity. This work has been pursued with the appearance of semi-passive walkers combining both specific passive properties and low power actuation to increase their robustness (Anderson et al., 2005). We can note the work of Collins (Collins and Ruina, 2005) which explored the case of a semi-passive 3D biped robot. Its morphology is based on a particular mass distribution, knee locking, round feet and springs on the legs to generate an efficient walking gait while keeping its lateral and frontal balance. The concept of the 3D semi-passive robot has been pushed even further with the realization of a complete humanoid robot with torso, arms and head: the robot Denise (Wisse, 2005) and Flame presented in (Hobbelen et al., 2008).

The geometry and distribution of mass in the body has complex influences on bipedal locomotion. Several studies have for example, explored the role of the foot and ankle morphology for bipedal walking in both humans (Adamczyk et al., 2006) (Hughes et al., 1990) and robots (Hobbelen and Wisse, 2005) (Davis and Caldwell, 2010). However, to our best knowledge no research has focused on the role of the thigh for bipedal locomotion. A few robots like HRP-4C (Kaneko et al., 2009) and Kenshiro humanoid (Nakanishi et al., 2013) robots seem to visually have a morphological design close to the thigh shape of Poppy, but no comparative study of the role of this shape was presented so far.

Thanks to the mechanical design of Poppy, allowing easy, cheap and fast morphological modifications, we are able to experiment with various thigh shapes on the robot's dynamics and find out what impact those have. In particular, in this experiment, we will focus on its bio-inspired thigh shape, bended by an angle of 6°. We will investigate the impact of this thigh design on balance and bipedal locomotion using a comparison with a more traditional straight thigh (see Fig. 8.1).

8.1.1 Understanding the role of the thigh shape in humans

If we look closely at the morphology of the human femur, it appears that it is inclined by an angle of 6°. This makes the feet closer to the projection of the center of gravity



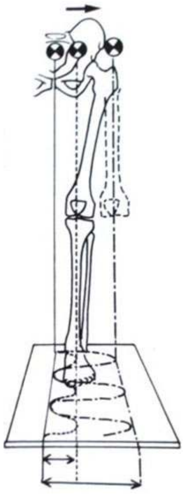
(a)bended thighs (b)straight thighs

Fig. 8.1.: We evaluate the effect of thigh morphology on the bipedal locomotion dynamic. Experiments are made using the Poppy humanoid platform. In this paper, we compare two thigh morphologies: (a) thigh bended by an angle of 6° and (b) a more classical approach with straight thighs.

(CoG) (see Fig. 8.2a) therefore it reduces the distance travelled by the CoG to move from foot to another.

The model presented in appendix A has been used during the conception of Poppy to decide the use of bended thigh rather than classic straight one. This simple model, based on an inverted pendulum, shows this particular shape may enhance the stability in two main ways during the walking gait:

- As the feet are closer to the center of gravity, the lateral translation of the CoG necessary to transfer the mass of the robot from one foot to another is reduced (see Fig 8.2a). In the case of Poppy's morphology, thanks to the 6^o bended thigh, the lateral motion of the CoG is reduced by about 30% ($5cm$ instead of $7.1cm$).
- During the stance phase, the CoG initial conditions are slightly modified. Therefore it reduces the CoG falling speed at the beginning. So if we consider the first 700 ms of the system behaviour simulation and compare the two systems, the mean of the CoG falling speed is reduced by around 56% in the bended thigh case.



(a) Effect of bended human femur on human bipedal locomotion

(b) Implementation of the bended thigh on the poppy platform

Fig. 8.2.: The human femur is inclined by an angle of 6°, this reduce the distance travelled by the CoG to be supported from one leg to the other during walking. We reproduced it on Poppy and suggest exploring its impact on the robot dynamics.

8.1.2 Experimenting with variable thigh properties on Poppy

The simple model described in appendix A showed that a slight inclination (6°) of the thigh can theoretically achieve a significant gain in the lateral stability of the robot during the two main phases of the walking gait (i.e. single stance phase and double stance phase). Yet this model is very simple and Poppy allows to experiments easily the role of morphology in the real world with all its complexity.

Therefore we modified the thigh shape and printed it. As shown on the Fig. 8.3, the only modified paramater is the thigh bending angle, two cases: 6° and 0°.

In this section, we describe representative experiments, which evaluate the actual gain of the thigh shape on the real Poppy platform. To do this, we used both a pair of straight thighs and the bended thighs presented above. We will compare Poppy's reactions with those different legs (see Fig. 8.1 and Fig. 8.3) on three experiments:

- Evaluate the falling speed during single support stance.
- Measure the lateral translation to move the CoG Form one foot to the other.

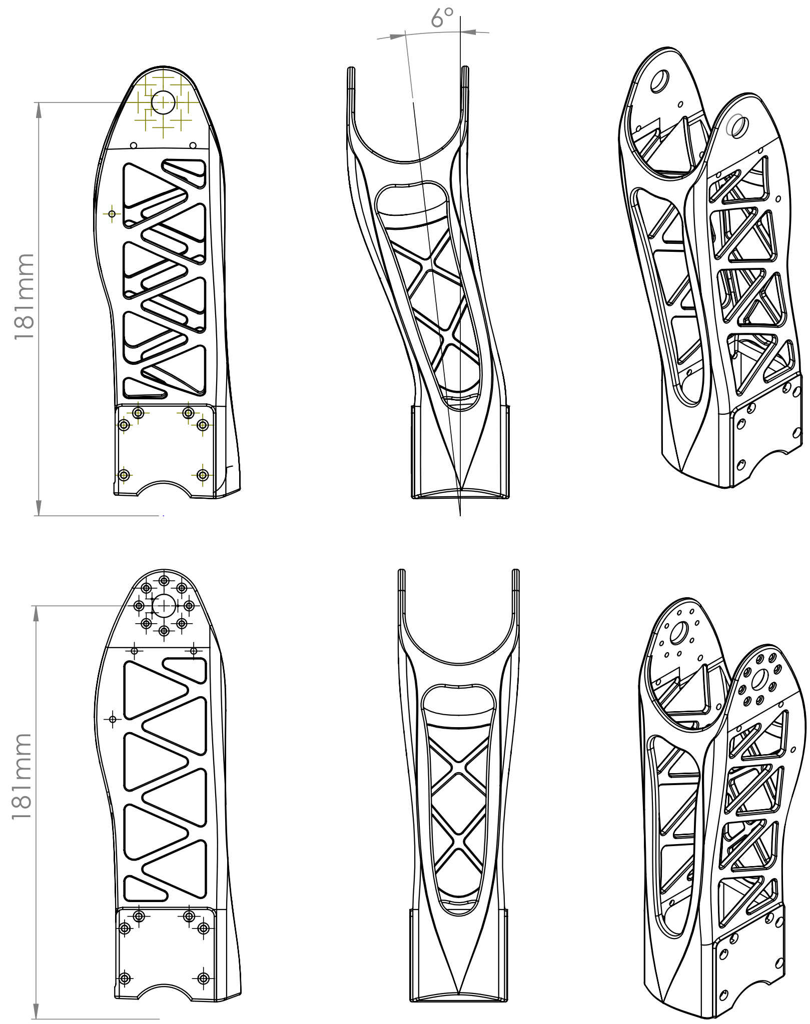
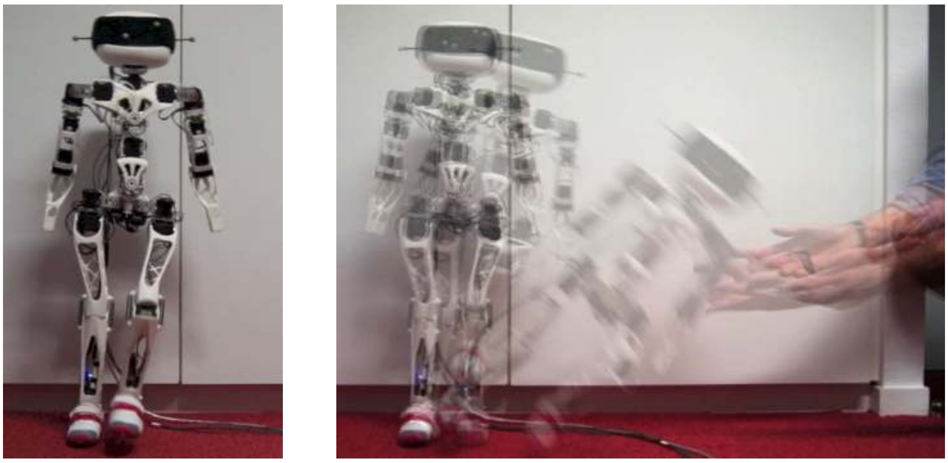


Fig. 8.3.: Blueprints of the two thighs tested in this experiment. The only parameter explored is the bending angle of the thigh: 6º and 0º

- Record the upper body motion during bipedal locomotion.

Single support falling velocity

The experiment evaluates the velocity of fall when Poppy is supported on only one foot and compare it with the theoretical results obtained in A.1. To do so, the robot's head is tracked by an Optitrack[1] device and markers are placed on the head. In postural balance on two feet, a motor order triggers the raise of a foot which unbalances the robot (see Fig. 8.4a) and causes its lateral fall (see Fig. 8.4b). This experiment was repeated about fifteen times for the two cases studied, i.e. with bended legs (Fig. 8.1a) and with straight legs (Fig. 8.1b).



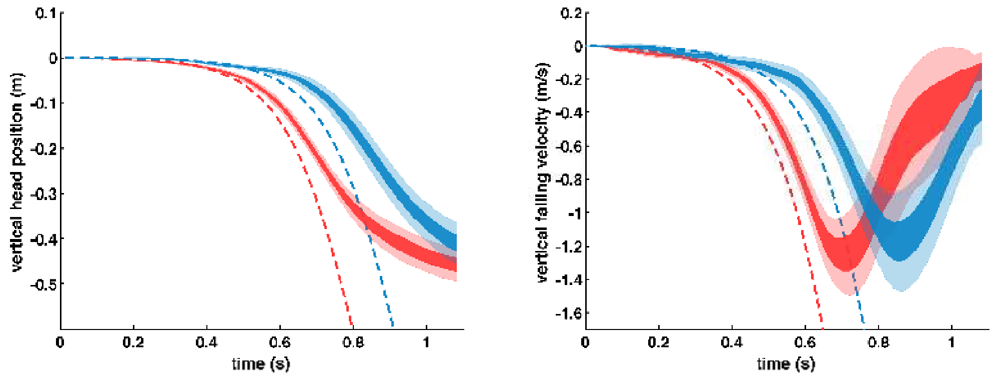
(a) Initial perturbation provoked by the sudden raise of one foot

(b) view of Poppy's lateral fall over time

Fig. 8.4.: Run of the single support falling experiment. The markers on Poppy's headband track its absolute position over time.

Experiments results are shown in Fig. 8.5. The blue colour is assigned to experiments with bended thighs while the red colour is assigned to straight thighs. For each case, the light colour corresponds to the standard deviation and the dark color to the 95% confidence interval of the mean value. The first figure (8.5a) refers to the head altitude position over time and the second (8.5b) to the falling velocity of the head. Dashed lines represent theoretical results obtained with the model presented in section A.1b. We notice the strong similarity both on the shape and on the difference between the two morphologies studied. Yet, there is a slight time shift between theoretical and experimental results. This can be explained by the inertia of the real robot which was not taken into account during the simulation.

[1] http://www.naturalpoint.com/optitrack/products/v120-trio/



(a)Vertical head position (b)Vertical head falling velocity

Fig. 8.5.: Results of the single support falling experiment. The blue colour is associated with experiments conducted with bended thighs while the red colour is assigned to straight thighs. For each case, the light colour corresponds to the standard deviation and the dark colour to the 95% confidence interval of the mean value while dashed lines represent theoretical results. These figures show the vertical position (a) and vertical falling velocity (b) of the head of Poppy over time for each case studied. The curve's behaviour change after 800 min is due to the fact that we catch the robot before it touches the ground.

These figures show a clear improvement for the version of Poppy with bended thighs (blue curves) with a 200 min time shift compared to the straight thighs (as illustrated on the attached video[2]). Thanks to this delay, the falling speed is reduced by about 56% during the first 700ms. Thus the robot remains almost stationary for 600 min (400 min in the case of straight thigh). Poppy's typical walking gait takes a period of one second so the mono-pedal stance phase lasts around 420 min (Lapeyre et al., 2013c). Considering that the robot remains stationary during more time than the single stance phase, we can imagine that the lateral balance control will be reduced during the walking gait.

Double support CoG transfer

In this experiment we evaluate the lateral movement of the robot necessary to cause a displacement of its center of gravity from one foot to the other and we verify the theoretical results obtained previously. For this, Poppy is placed on a force platform to measure the displacement of its center of pressure. The absolute movements of the robot are tracked with an OptiTrack device and markers placed at the head and lower back (approximately the position of the actual center of gravity). The robot is kept rigid in a neutral position and a human physically pushed it from left to right until it reached its lateral falling limit. As this operation is not very accurate, the experiment is repeated one hundred times.

[2] http://flowers.inria.fr/Humanoid2013/

	Straight tigh	Bended Tigh	diff(%)
CoP	74.6 ± 9.0 mm	49.8 ± 7.7 mm	33
Head	100.1± 14.4 mm	62.9± 22.0 mm	37
Lower Back	64.1± 11.5 mm	43.4± 15.0 mm	32

Tab. 8.1.: Summary of the results obtained during the experiment on the lateral motion needed to transfer the robot's mass from one foot to the other.

Table 8.1 presents for each area considered (i.e. center of pressure (under feet), lower back and head motion) the amplitude of the lateral motion (in millimeters) needed to translate the CoG of the robot from one foot to the other for the two versions of Poppy's thigh design. The last columns summarize the relative difference between the two conceptions (in percent). One can note that the results show a reduction of lateral movement of around 30%. Thanks to the shape of the thigh, the lateral displacement of the upper body required to move the CoG from one foot to the other can be reduced.

The results presented on the two first experiments show improvement for two main aspects needed during bipedal locomotion: lateral stability and mass transfer. In the next experiment, we will evaluate if there is a significant performance gain in a complex dynamic phase such as bipedal walking.

Walking dynamic

As explained in the introduction and description of the platform, Poppy has been specially designed to study bipedal walking and human-robot interaction.

Here the experiment consists of playing an open-loop walking pattern while the robot is guided through the physical interaction with a human. The user's role is to provide both balance and control of mass transfer. By producing small lateral motion on the upper-body they can help the robot to move its CoG from one foot to another.

The gait is based on the actual human sagittal joint kinematic (Nester et al., 2003): hip, knee, ankle (see Fig. 8.7.a). A direct transposition of the human joint kinematic on the Poppy's morphology results in a walking speed which is too fast to be handled by users (see Fig. 8.7.b). A simple reduction of a joint's amplitude leads to an unsuitable leg trajectory where toes bump into the ground during the swing phase (see Fig. 8.7.c). So to ensure enough clearance during the swing phase and suitable walking speed for the guidance with a user, we modified the trajectories of the

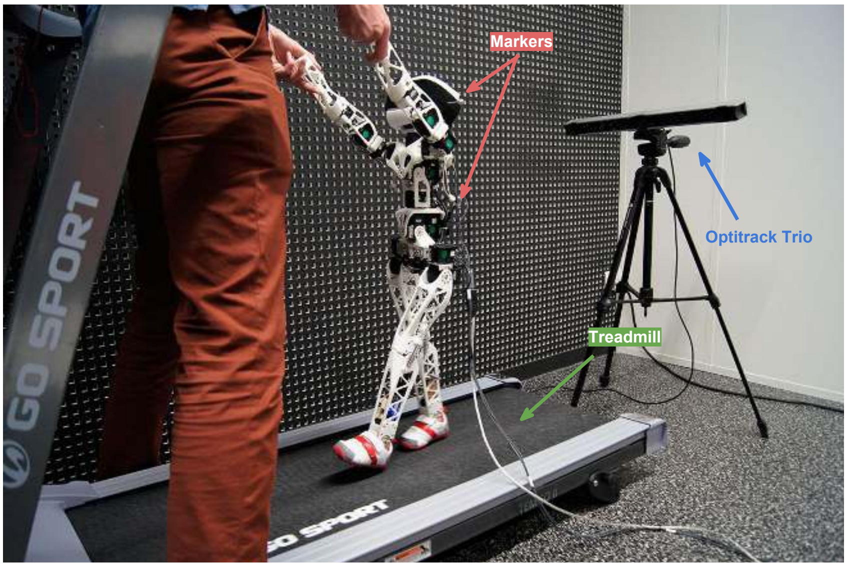


Fig. 8.6.: Proceeding of the walking experiment. Poppy is tracked by an Optitrack trio while it is walking on a treadmill set at 1.8km/h. An expert user provides the sagittal balance needed throughout the experiment.

joints by hand to both reduce the length step and increase the foot clearance (see Fig. 8.7.d). The actual gait on Poppy is shown in Fig. 8.8.

In this experiment we are interested in the dynamic of Poppy especially on the frontal plane and we will compare the effect of the thigh shape on this dynamic. Poppy walks on a treadmill following the walking gait described above. An expert user trained to keep the robot in the correct walking cycle provides guidance to the robot. This is done by keeping the robot in a vertical position and supporting, in a compliant manner, the lateral movement of the robot as illustrated in attached videos. The user is asked to do the best he can to minimize the movement/forces he applies in both morphologies to reduce the bias towards the two designs experimented. All proprioceptive sensors are recorded at 50hz while an Optitrack device associated with markers located at the head and lower back measure the absolute displacements of the robot (voir Fig. 8.6).

Poppy's movement is recorded for around 1800 walking gait cycles for each thigh design (around 90,000 data points for each case). Data are folded over to extract the gait behavior over a gait cycle. Results are presented in Fig. 8.9. As previously, the blue color is assigned to experiments with bended thighs, the red color is assigned to straight thighs. For each case, the light color corresponds to the standard deviation and the dark color to the 95% confidence interval of the mean value.

The two first figures (i.e. 8.9a and 8.9b) show the lateral motion of the upper body in millimeters over the gait cycle. We notice that for the two designs, the motion pattern shown by the upper body (head and lower back) is similar. However,

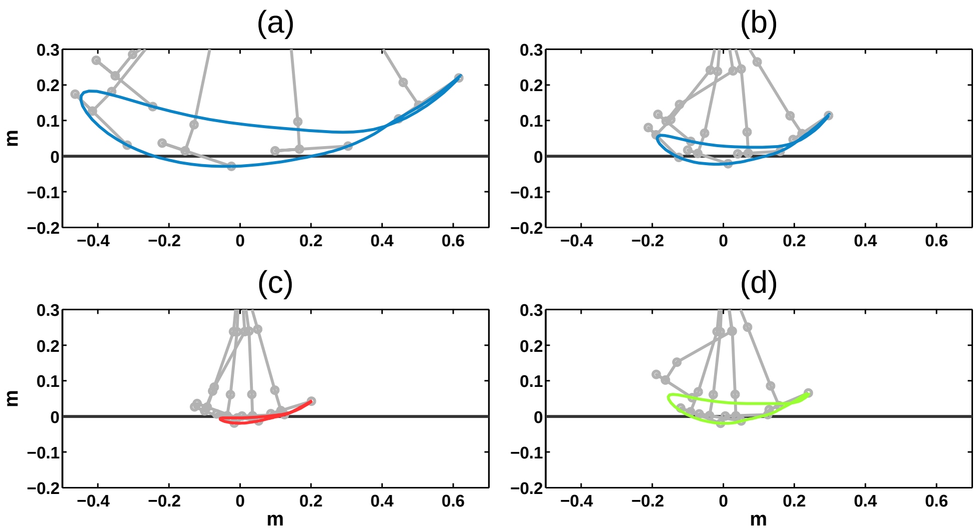


Fig. 8.7.: Trajectories of toes generated by the walking pattern a) Kinematics of human walking with human morphology b) Direct transposition of human kinematics onto Poppy's morphology c) Reducing amplitude of the human kinematics joints with Poppy's morphology d) Walking pattern used for the experiment with Poppy

in the case of the bended thigh (blue) the amplitude of the motion is reduced by about 45%. Another interesting effect concerns the head perturbations shown on figures 8.9c, and 8.9d. Here also, patterns are similar but in the case of the bended thigh the head is clearly less perturbed by the walking dynamic, with a reduction in amplitude of approximately 30%. Five pictures were taken while Poppy was walking and were stacked in Fig. 8.10. This shows a qualitative point of view of the walking dynamic for both studied cases. We notice that the lateral motion of the version of Poppy with bended thighs 8.10a is small compared to the version with straight thighs 8.10b.

8.1.3 Conclusion on the thigh shape role for bipedal locomotion

We focus on the shape of the Poppy thigh and its effect on the robot's dynamic. We studied the role of morphology in the reduction and simplification of the control needed to perform complex task such as bipedal walking. We have presented the simple theoretical model we used for the design of Poppy's thigh based on the inverse pendulum dynamic. We have conducted experiments to evaluate the improvements of the bended thigh on the real robot dynamic and compared it to the model. Since Poppy's structural design allows easy, cheap and fast morphological modifications, we were able to try another thigh design. We also used a pair of straight thighs which is a more classical approach in humanoid conception. The experimental comparison between the two thighs design confirmed the theoretical results, the bio-inspired thigh design improves Poppy's dynamic in two main ways useful for bipedal walking:

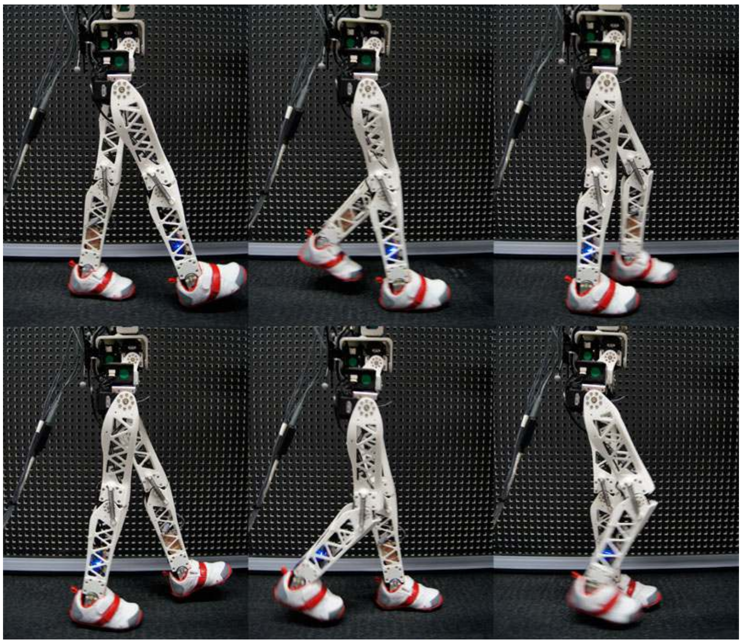
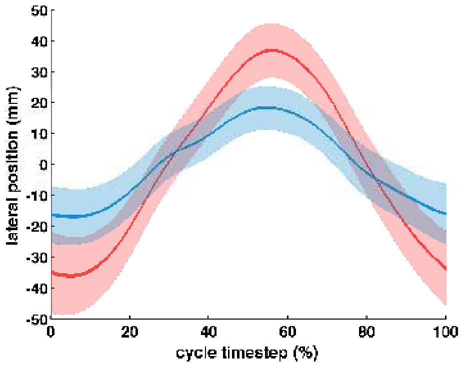


Fig. 8.8.: Walking gait CPG described on Fig. 8.7.d applied on the actual Poppy robot. The CPG generates a human-like walking gait allowing the robot to walk at 1.8km/h and involves straight legs during the stance phase. There is no balance control but stability is obtained through physical guidance with a trained expert user.

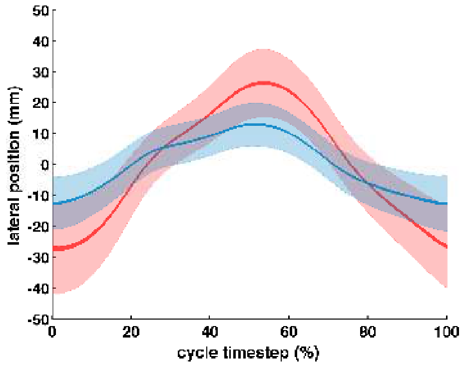
- It reduces the falling velocity by almost 60% when the robot is on one foot (single support phase).
- It reduces the lateral motion needed to transfer the mass of the robot from one foot to the other (double support phase) by 30%.

It is really interesting to note that such a small modification of the robot morphology has a very significant impact on the robot's behaviour.

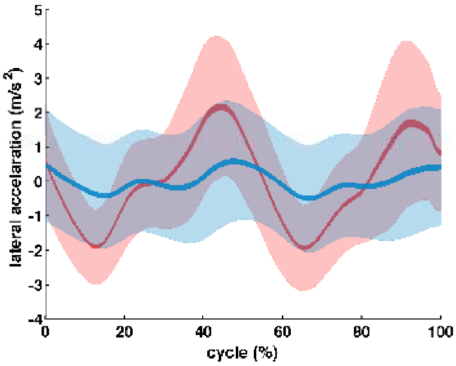
These results are interesting but they do not reflect the Poppy's real walking dynamic. To evaluate the effect of the bended thigh on bipedal locomotion, we conducted a third experiment where Poppy is walking on a treadmill. In this experiment, we show that the bended thigh has an effect on a complex dynamic task such as the biped locomotion: it reduces the motion amplitude on the upper body by 45% and increases the head stability by 30%. We choose these metrics due to our experimental constraints (fixed speed, social guidance) as a qualitative evaluation of the walking gait. Moreover it provides us with an intuitive, yet incomplete evaluation of the walking. Many other measures could have been chosen or combined such as speed, energy consumption or robustness to external perturbations. It is still complicated to understand which metric is the most adapted for robotic biped locomotion. As human beings are trained to recognize a bipedal gait, users can provide guidance to the robot for both safety of exploration and evaluation of the walking behaviour.



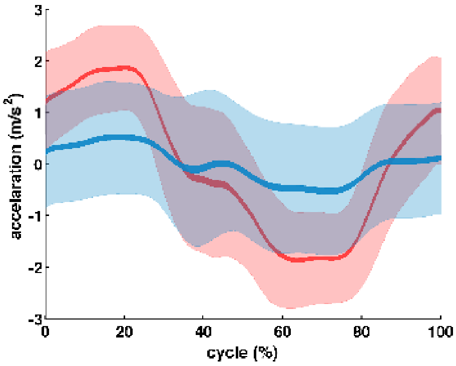
(a) Lateral head displacement



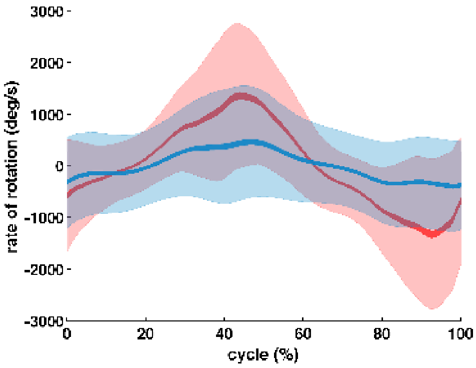
(b) Lateral lower back displacement



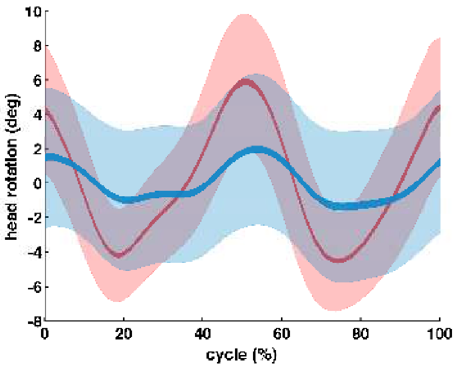
(c) Sagittal head acceleration



(d) Lateral head acceleration

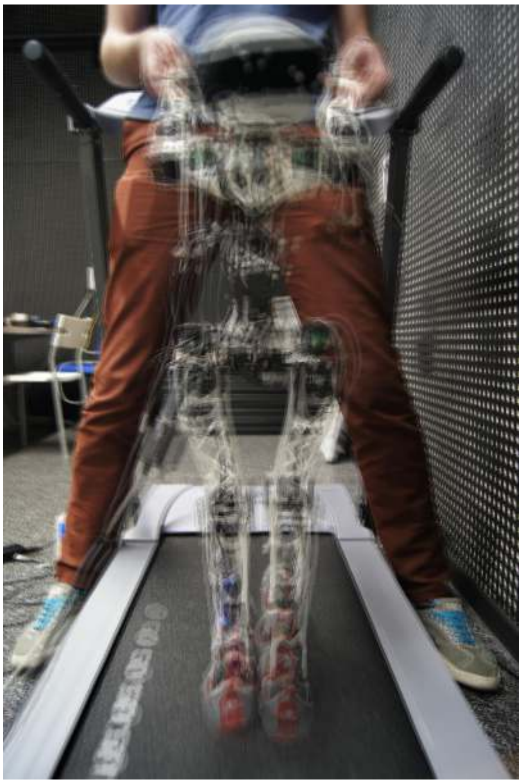


(e) Speed of rotation in the frontal plane

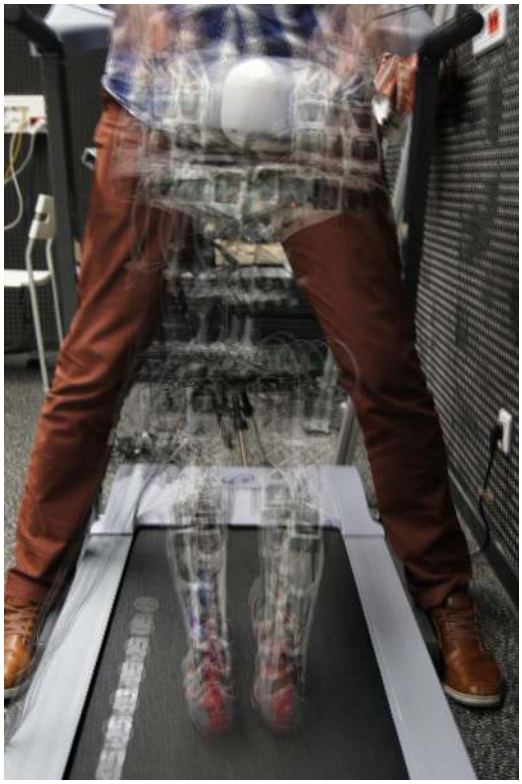


(f) Head inclinaison

Fig. 8.9.: Results obtained during the walking experiment. The blue color is associated with experiments conducted with bended thighs while the red color is assigned to straight thighs. For each case, the light color corresponds to the standard deviation and the dark color to the 95% confidence interval of the mean value. All data are folded over to extract the mean gait behavior and its standard deviation over a walking gait cycle expressed in percent.



(a)bended thigh



(b)straight thigh

Fig. 8.10.: Five pictures have been taken while Poppy was walking and were stacked to obtain a qualitative view of the difference in the walking behavior in relation to the morphology of the thigh.

8.2 Rapid morphological exploration

In the previous section we showed how Poppy can be used to explore the actual role of morphology for humanoid behaviours. However, Poppy is a prototyping platform designed to test and experiment quickly several technological solution, especially thanks to modular properties, but until now, we did not actually evaluate it.

Therefore while we were working on a new design for Poppy's feet and exploring a design similar to "foot 1" (see Fig. 8.11a), we decided to use this as a context to conduct an experiment into multiple variations of the foot morphology as an illustration of the methodology we have initiated with Poppy and presented in chapter 2.

The aim of this experiment is to quickly explore the effect of foot morphology on stability. Here, we are particularly interested in the stability of the head after a stepping impact. These impacts are quite challenging to simulate realistically and the natural compliance of the Poppy platform means it is even more important to test this on the real robot.

Type	Foot 1	Foot 2	Foot 3	Foot 4
Double rotation	Passive	No	Active	Passive
Human-like foot	Yes	Yes	No	Yes
Toes	Yes	No	No	Yes
Rotation axis height	75.70 mm	33 mm	39 mm	35.5 mm

Tab. 8.2.: Table summarizing the different types of feet used.
Passive double-rotation: one active rotation (motor: Dynamixel MX 28) for the sagittal plan and a passive rotation for the frontal plan with two springs.
Active double-rotation: A two motorized rotations (sagittal plan and frontal plan). No double-rotation: one motorized rotation (sagittal plan).
Human-like foot: a foot design resembling a human foot of a two year-old child (size: 130.7 mm shoes size: 23 EU). The feet were tested with and without shoes.
Rotation axis height: the height between the axis of rotation of the sagittal plan and the floor without shoes.
Toes: Indicates that the foot has toes.

For the sake of lightness, the initial design of Poppy's feet only had one degree of freedom (DoF): pitch rotation. This configuration carried the inconvenience of preventing a proper parallel foot/ground contact. Thus, we developed several different feet with two degrees of freedom. Along with a standard motorized 2 DoF flat foot design, we also wanted to explore passive joints with springs. The use of passive joints allows for both lightness and reactive torque for stability.

Moreover, it appeared that a proper foot/ground contact with convenient friction was difficult to obtain based only on 3D-printed material. One simple solution to this problem is to use a shoe which can provide a high friction and adapt slightly to imperfections on the ground. Furthermore, this solution also allows keeping the feet close to humans ones. Thus, the feet tested (with the exception of the flat foot) were designed from a moulding of the interior of a shoe. It is to be noted that we also included passive toes (with springs) on some of the feet tested for future work on locomotion. These toes should not have any significant impact on the criterion tested.

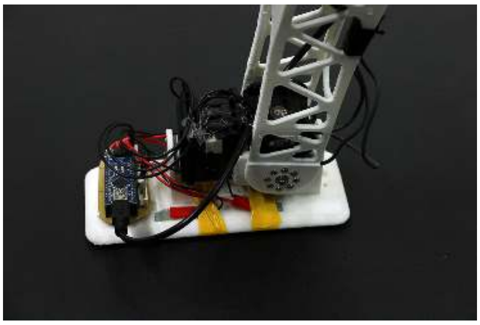
8.2.1 Experimental setup

For this experiment, the robot simply stands upright secured by a slack strap on a fixed gantry. Different markers on the robot are tracked by a motion capture system at 100Hz (Natural Point OptiTrack). See figure 8.12 for more details.

Four different feet were tested (cf. Table 8.2). Three out of the four feet were tested both with and without shoes.

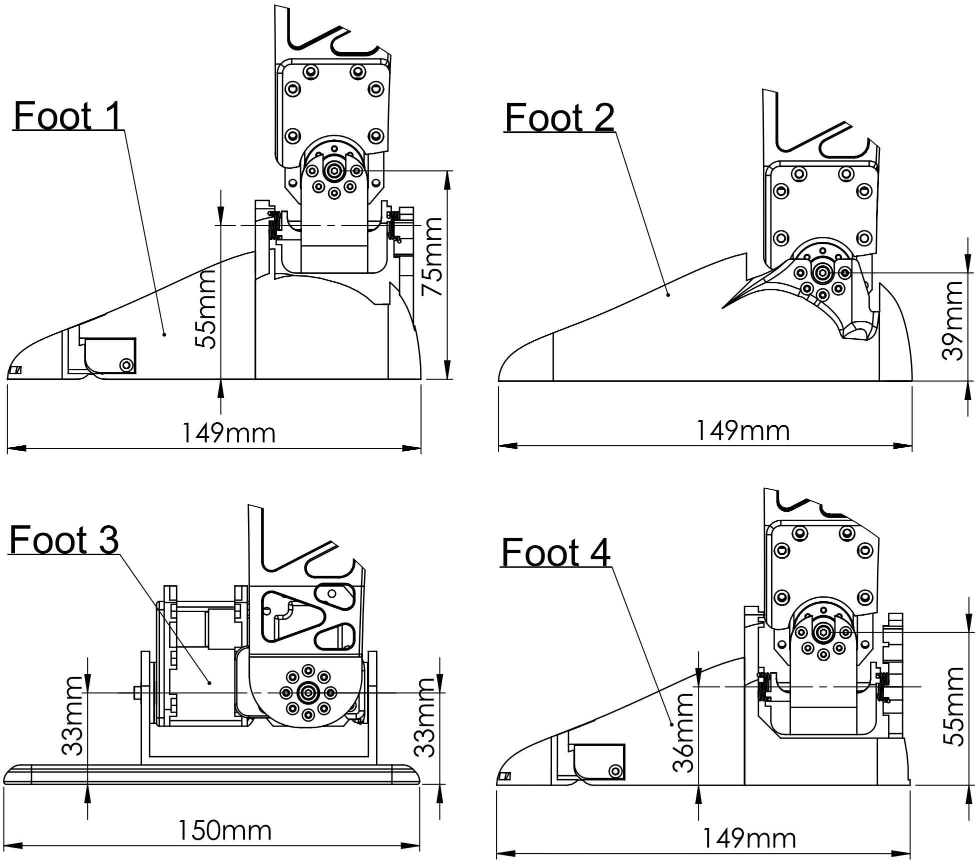
(a)Foot_1

(b)Foot_2



(c)Foot_3

(d)Foot_4



(e)Blueprints of the various foot designs studied in this experiments.

Fig. 8.11.: Visual and technical descriptions of the foot designs explored in this experiments.

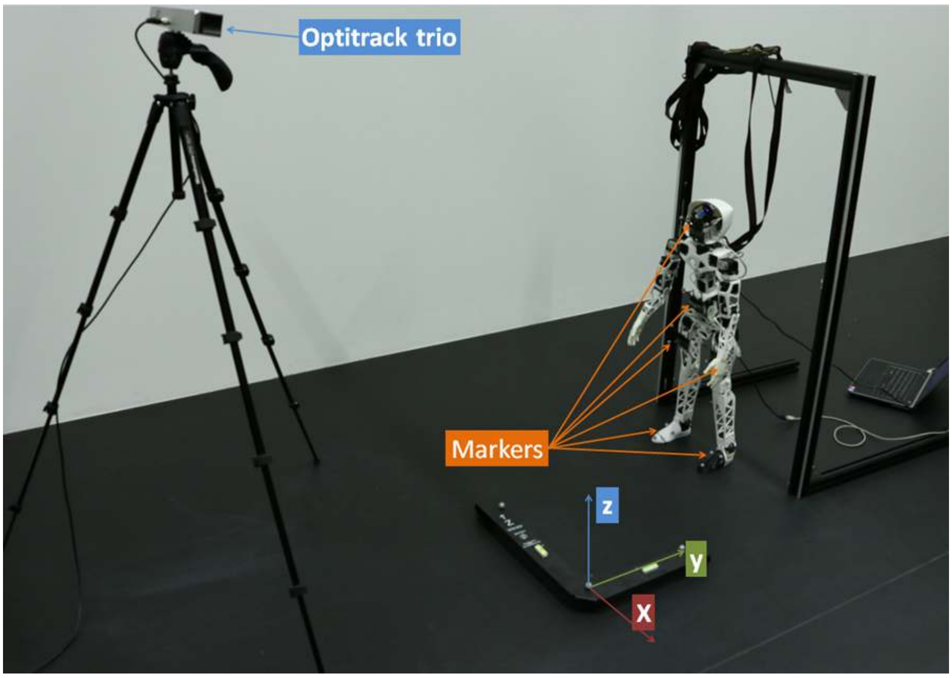


Fig. 8.12.: Experimental setup. The robot is secured by slack strap on a gantry and tracked by an OptiTrack trio device. Markers are placed on the feet, hips, abdomen, and on the head

8.2.2 Experiments

The feet were tested with a very simple discrete movement (see Code 8.1), representative of the kind of impacts that occur during walking. The robot performs a single step leftward with the left leg. The left foot is lifted (3cm) and then put back on the ground with a slight lateral displacement towards the exterior (5° at the level of the hip). The duration of the whole movement is about 0.4 s and repeated 20 times for each configuration.

8.2.3 Results

Figures 8.13, 8.14 and 8.15 respectively show the evolution of the position of the head marker in the x, y and z axis for each foot tested. Dotted vertical lines indicate the beginning and the end of the leg movement.

These figures show that the dynamics of the robot are not trivial, even for the simple movement we tested, the standard deviation is not negligible and shows how chaotic the reaction of such an impact can be. This particularity is another proof of the significance of the use of experimentation versus simulation.

```
# PID gains of legs' actuators
poppy.r_ankle_y.pid = (50, 50, 0)
poppy.l_ankle_y.pid = (50, 50, 0)

poppy.r_hip_x.pid = (50, 50, 0)
poppy.l_hip_x.pid = (50, 50, 0)

poppy.r_hip_y.pid = (30, 30, 0)
poppy.l_hip_y.pid = (30, 30, 0)

poppy.abs_y.pid = (20, 20, 0)
poppy.abs_x.pid = (20, 20, 0)


# Mouvement parameters
up_duration = 0.15 #duration of leg lift up in s
down_duration = 0.15 #duration of leg put down in s
up = 0.03 #height of leg lift up in m

#we create a 1D minimum jerk trajectory from 0 to up (m) with a duration of up_duration (s) with initial and final null velocities.
mj1 = min_jerk.MJTraj(0, up, up_duration)
mj2 = min_jerk.MJTraj(up, 0, down_duration)


#attache a primitive to "mjleftup" to lift the left leg according to inverse kinematics following the minimum jerk trajectory.
poppy.attach_primitive(min_jerk.MJLegs1D(poppy, mj1), 'mjleftup')
poppy.attach_primitive(min_jerk.MJLegs1D(poppy, mj2), 'mjleftdown')


# lift the left foot
poppy.mjleftup.start()
poppy.mjleftup.wait_to_stop()

# move the left leg to the left
poppy.l_hip_x.goal_position=15

#land the left foot
poppy.mjleftdown.start()
poppy.mjleftdown.wait_to_stop()
```

Code 8.1: Discrete mouvement executed on Poppy

We can clearly see that foot 3 (standard flat foot) behaves quite differently than the other feet tested. In particular in the x and y directions, we see that with this foot the head tends to move more towards the exterior (left of the robot) and towards the rear.

Regarding the effect of the shoes, results are less clear but most of the time (except for foot 1) differences occur between a given foot with and without shoes. The friction with the ground can explain these differences. Bare feet tend to slip more than those with shoes.

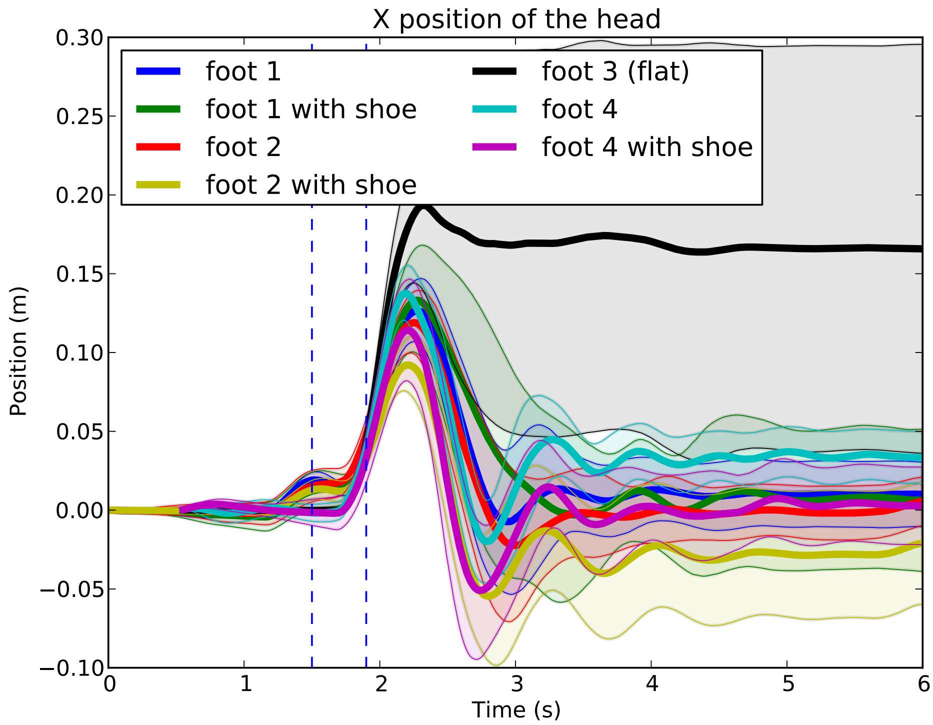


Fig. 8.13.: Evolution of the position of the head in the x axis for each foot tested (see Fig. 8.11 for illustration of each foot)

This first experiment allowed us to determine that the use of an active double rotation of the ankle may not be mandatory. Indeed, the behaviors observed with the passive feet were even better than with the flat feet with active rotation. Although a clear interpretation of this phenomenon is still difficult to propose, some hypotheses related to the weight (with one more motor feet are heavier) and the area of surface in contact with the ground (flat foot surface is bigger) have to be investigated.

Moreover, we observed that the shoes added extra friction in relation to the ground without really impairing the stability. Although rarely used in humanoid robotics, these early results encourage us to explore this possibility in more depth.

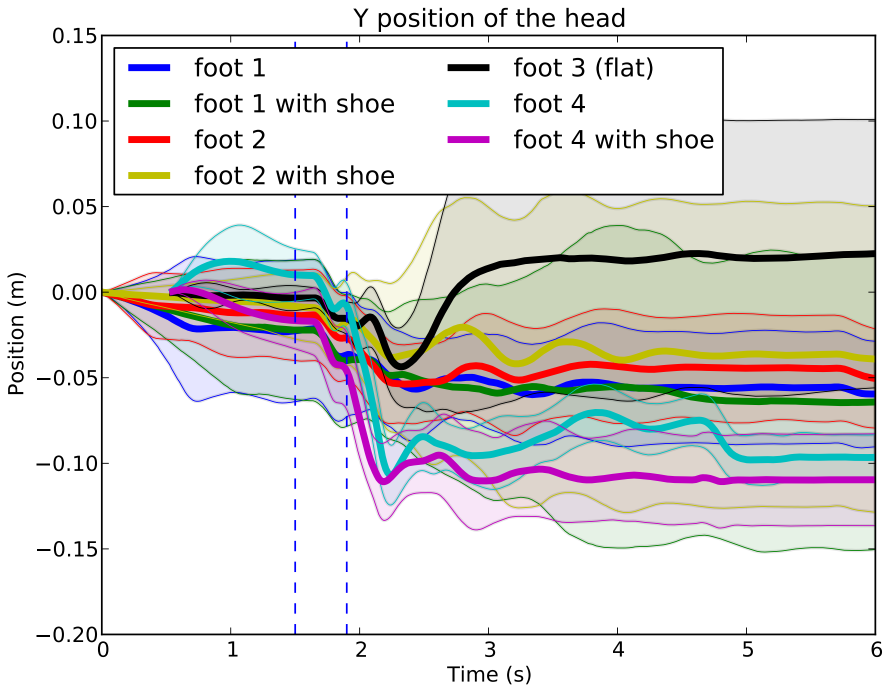


Fig. 8.14.: Evolution of the position of the head in the y axis for each foot tested (see Fig. 8.11 for illustration of each foot).

Finally the most important aspect for us was to actually evaluate the amount of time needed to conduct such experiments with Poppy. The starting point was "foot 1" as it was the work in progress. Thus morphological design modifications only concern foot 3 and 4:

- **Foot 3 (flat):** Modifying Poppy's initial foot design to permit the integration of two Dynamixel motors and the associated flat feet required 16 hours of CAD design. The printing of the whole required part (2 legs, 2 feet and 2 ankles) took approximately 30 hours on a low-cost FDM printer (Makerbot Replicator 2).

- **Foot 4:** While the difference with foot 1 concerns only one parameter (i.e. the joint position), the modification needed to produce foot 4 based on foot 1 was done in approximately 2 hours of CAD. Then the printing of the new part was achieved in 10 hours.

Then, conducting the whole experiment (i.e. design the leg motion, establishment of the experimental setup and data acquisition) was achieved in about one week with two people. **In particular, the actual experimentation involving changing Poppy's feet seven times and acquiring at least 20 trials for each took less than two days.**

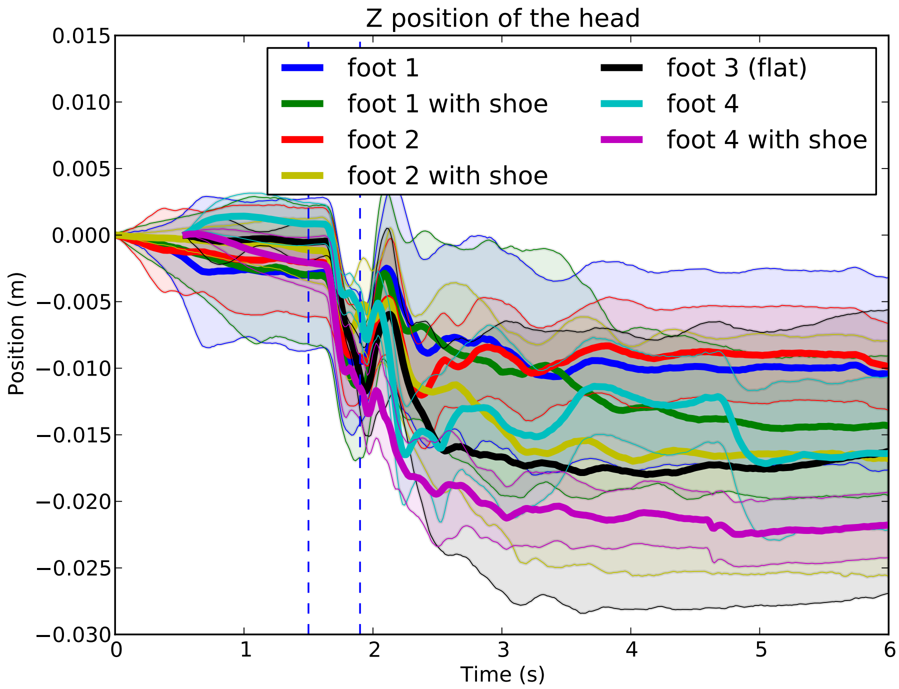


Fig. 8.15.: Evolution of the position of the head in the z axis for each foot tested (see Fig. 8.11 for illustration of each foot).

8.2.4 Reuse of this experiment

Everything necessary to obtain and use Poppy is available on our GitHub project page: `www.github.com/poppy_project`. Also, to complete the illustration of this Poppy use-case, we diffuse along with the present paper:

- the whole setup materials i.e. the code used for the experiment and the 3D files to reproduce/modify each foot,
- the raw data acquired that include for each trial: all markers position, head IMU measurement and the complete motors data (proprioceptive position evaluation overtime),
- the code used to extract and plot the results presented.

All these materials are available on the repository associated with this experiment: `https://github.com/matthieu-lapeyre/Humanoids2014` and can be freely used e.g. for further investigation with the acquired data, or to reproduce and extend the experiment.

8.3 Extending the sensor apparatus of Poppy

Poppy has been designed following a methodology (presented in chapter 2) which makes easy the hacking of the platform. The last experiments presented mechanical modifications of Poppy's morphology. Indeed thanks to its 3D printed structure, it is quite easy and straightforward to modify its mechanical parts, unfortunately we cannot (yet) print complex electronics circuits and components.

We therefore chose design a custom I/O board based on Arduino (detailed in chapter 6). As its name suggests, this board has for main purpose to ensure the several inputs/outputs of the robot and offers:

- 2 Dynamixel buses (TTL),
- 2 internal USB and 2 external USB ports,
- analog and digital pins available on a classic Arduino Due which can be use as direct input/output or for communication buses such as UART, I2C or SPI.

Thus there are many more I/Os than required for Poppy. These extra ports has been intended to let Poppy users extend its sensorimotor space and adapt it to their needs.

During our first trials to design walking a primitive with Poppy, we have been interested in the measurement of under feet pressures but the simple foot design Poppy had, does not involve such sensors. With a traditional robotic platform, we should have to either use the available sensors, here the load measurement in the ankle Dynamixel motor, or add an external device with its own power supply and communication system.

With the Poppy electronic modularity, we can hack the robot and integrate new sensors. Then they can be plugged on the I/O board for communication and power supply needs.

To provide an example of how we can actually hack the Poppy robot, we will explain here what we did to integrate force sensors under the feet and acquire the data with the pypot library.

8.3.1 Integration of foot pressure sensors on Poppy

To obtain measurement of the pressure variation under our Poppy's feet we used FSR sensors from Interlink Electronics (see Fig. C.1a). The FSR sensor will vary its resistance depending on how much pressure is being applied to the sensing area. The harder the force, the lower the resistance is. The acquisition of the FSR value require to create a simple voltage-divider those the design is explained in appendix C. These sensors are low-cost -6$ each- yet theirs behaviors are very non-linear (see Fig. C.3) and the calibration is quite variable depending on the production batch and the thermal conditions. So we cannot expect having precise results.

When we did the integration of foot sensors on the Poppy, its feet were still a really simple and flexible 3D printed part. The shoes did the actual force transmission. We therefore had to directly attach the sensors below the shoes. To avoid multiple wires (2 per sensor) going from the head to the feet, we decided to use additional arduino nano boards to acquire sensors values of each foot and stream the data through serial communication up to the IO board.

Because Arduino nano board has 8 analog inputs, we have added 8 sensors under each foot (see Fig. 8.16) but actually only used 5 (the big ones) and integrated the arduino nano in the leg. While it was a hack of a real shoe and not just a print of new part, the intervention was quite annoying but still achievable in one day. Here we have chosen to use USB cable to plug each Arduino nano in the Poppy's head but it could also have be done using UART, SPI or I2C communication.

Fig. 8.16.: Physical implementation of the sensors on Poppy. FSR have been placed under feer while Arduino nano board are placed on the leg acquire the data and stream them to the main IO board through USB.

As we explained in section 5.3.2, the Arduino programming language bring the low level programming accessible to anyone. The Code 8.2 shows that we actually uploaded on each Arduino nano board. With just 10 lines of code we can stream the values of 5 pressure sensors.

```
void setup() {
  // open serial communaction at 57600 baud
  Serial.begin(57600);
}

// the loop routine runs over and over again forever:
void loop() {

  // Read input voltage for the 5 FSR sensors
  int sen1 = analogRead(A3);
  int sen2 = analogRead(A4);
  int sen3 = analogRead(A5);
  int sen4 = analogRead(A6);
  int sen5 = analogRead(A7);

  // send data with the serial communication and use ',' as formater
  Serial.println(sen1 + ',' + sen2 + ',' + sen3 + ',' + sen4 + ',' +
      sen5);

  delay(20);        // delay in between reads for stability
}
```

Code 8.2: Arduino code to read force sensors data

Then we just have to create a novel sensor controller in pypot (see section 7.3.1 for details) which describes the I/O communication and get the desired values (see Code 8.3). Here again, the design of the pypot library makes this task easy, only 20 lines of code are required to get access to add a novel sensor and create variable to obtain its value.

8.3.2 Measured data

With our novel sensors, we conducted similar walking experiment as the one explains in section 8.1.2 and shows on the Fig. 8.8 and recorded at $50hz$ the measured force variations under Poppy's feet.

The sensors are not very precise but as we can see on Fig. 8.17, the variation of the ground reaction force (mean of the 5 force sensors) over the gait cycle has a similar M-shape as the one we can find in human gait (see Fig. 8.18). Also we can notice than the reaction is slightly different between the two foot (Fig. 8.17a Vs Fig. 8.17b).

The bad precision of the sensors prevents us from affirming conclusions but it can still give insights to understand the walking behaviour of Poppy:

1. The second peak of the M-shape corresponding to the toe impulsion is weak or inexistent on Poppy. Indeed, when we look at the video of the walking gait made by Poppy, we can notice it barely uses its toes.

```python
class FootIO(IO, StoppableThreadLoop):
    def __init__(self, port, baudrate):
        self.serial_com = serial.Serial(port, baudrate)
        self._measured_values = []

    def last_values(self):
        return self._measured_values

    def update(self):
        l = self.serial_com.readline()
        l = l.replace('\r\n', '')
        self._measured_values = map(float, l.split(','))


class FootPressure(SensorsController):
    def __init__(self, io):
        SensorsController.__init__(self, io)
        self._values = []

    def update(self):
        self._values = self.io.last_values()

    @property
    def pressure_values(self):
        return self._values
```

Code 8.3: Example of Python code written to add custom foot sensors in pypot. The *FootIO* class describes how we can read the data from the Arduino nano placed in the foot. The *FootPressure* class is the sensor controller which is called by the pypot to synchronize the sensorimotor space of Poppy.

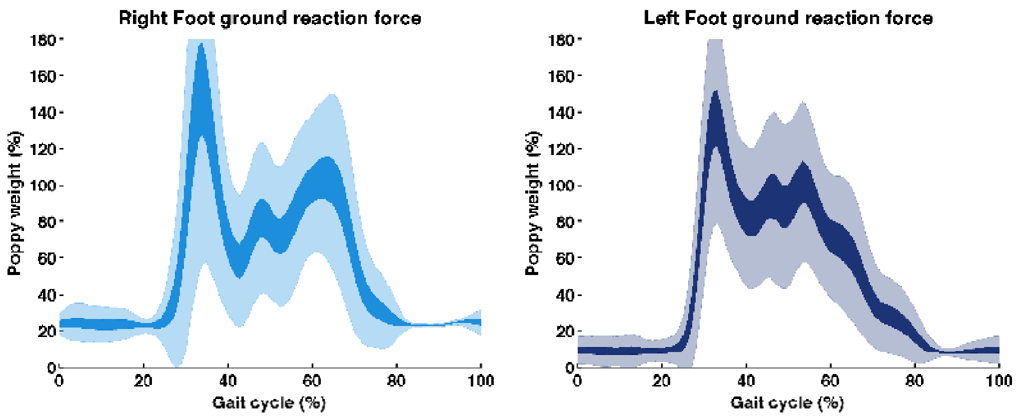


Fig. 8.17.: Ground reaction forces measured with the FSR sensors placed under Poppy's feet.

2. The first peak of the M-shape is very strong. Either the walking gait of Poppy was fast or the structure is too rigid and does not absorb correctly the initial impact.

We can therefore explore some improvements axes:

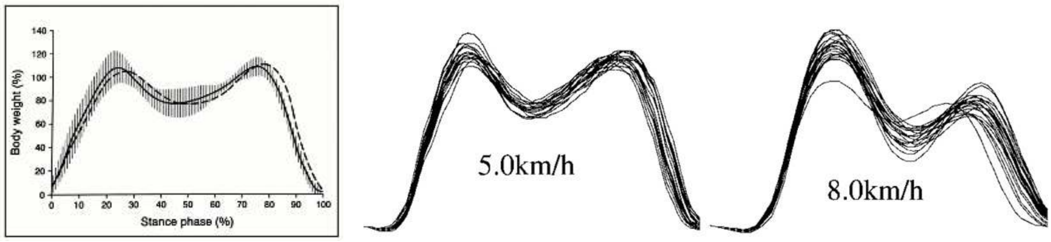


Fig. 8.18.: Typical ground reaction shape of the human gait.

1. Explore why the current walking behaviour does not involve clearly the passive toes. Is it cause of the walking primitive design or the mechanical design of the toes?

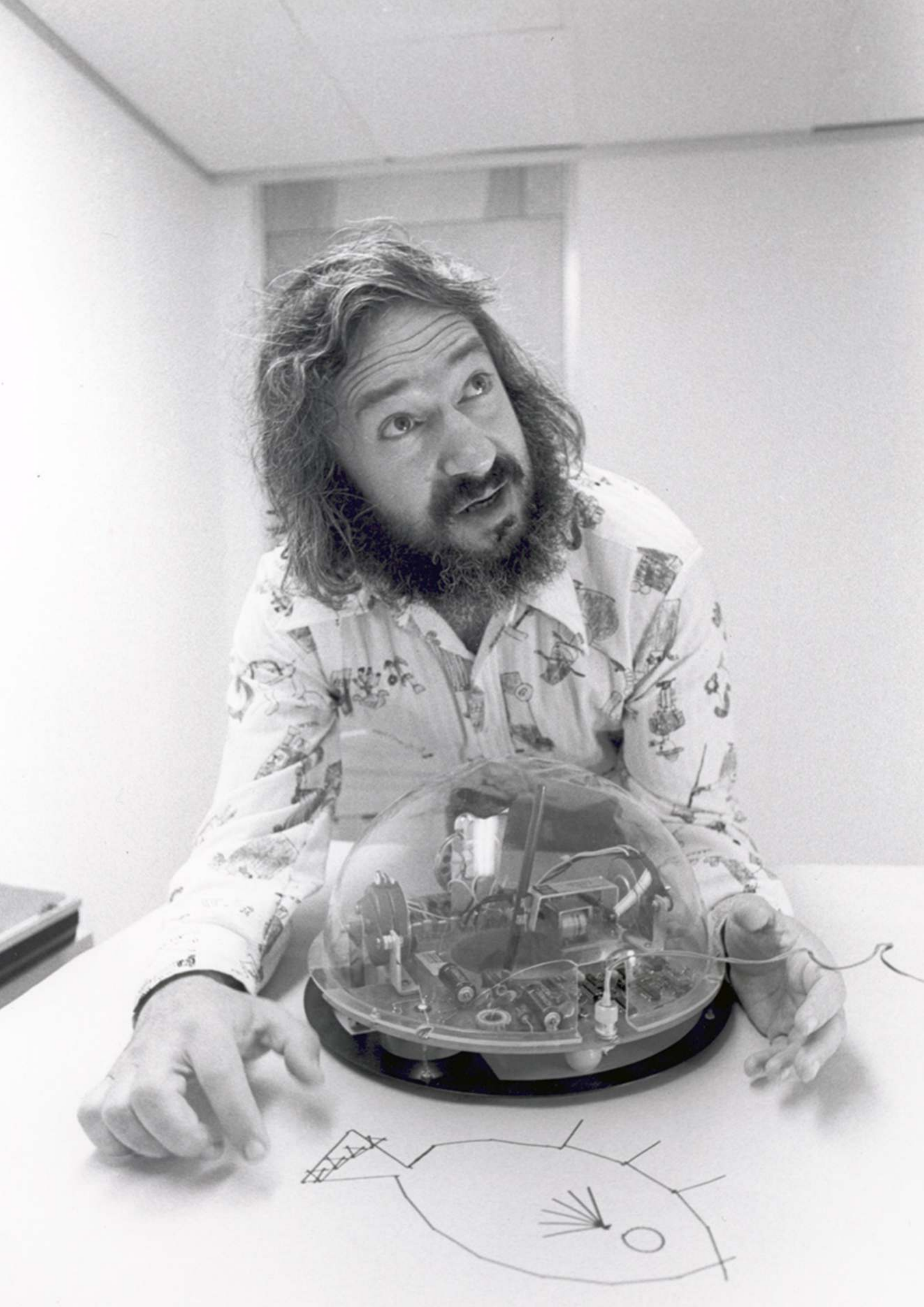
2. The initial impact is not desirable toward the achievement of a self-balanced walking behaviour. We should explore solutions to absorb it.

8.4 Discussion & Conclusion

The work presented in this chapter summarizes several experiments conducted during this thesis. Unlike the work presented in chapter 2, our experimental results can be considered as preliminary and incomplete to clearly show and demonstrate impact of the robot morphology over its dynamic and control.

Indeed, in this thesis, we are more interested by finding an appropriate methodology to explore easily in the real world the robot morphology impact rather than the actual exploration of the morphology. In this context, experiments conducted are more for illustration and evaluation of the actual way to change and hack Poppy. Toward this goal, confronting our methodology to real usage has been really instructive both to validate the design methods and highlight some non-optimized point.

After these experiments we focused ourselves on improving the platform to make it more easy-to-use, easy-to-hack and robust to the real world. In this way, we decided to go outside the laboratory and put Poppy in non-expert users' hands. Indeed, as we explained, we are trying to construct a multidisciplinary community involving therefore multiple profiles more or less experts in robotics. Among these profiles, two potential usages draw particularly our attention: the education and the art, and will be discussed in the next chapter.



Education

9

> *New technologies help students navigate the creative thinking spiral.*
>
> **— Mitchel Resnick**

9.1 Introduction

In our modern societies, in particular in France, education is a top priority and a concern for the government and the people. Thus education is central and has to be adapted and relevant to societal evolution. In particular, new technologies have a deep impact on our societies and raise legitimate question and concern (Plester et al., 2008). Also, they can become great vectors to make the education system more efficient and fair.

1- Improve access to and dissemination of education:

Knowledge is the wealth of humanity and thanks to the very large democratization of the internet, anyone with a connection can now access the whole of humanity's knowledge in just a couple of seconds. The most famous example is certainly Wikipedia[1], a community encyclopaedia with non-stop growing content both in term of quantity and quality. With such a tool, anyone can instantly access anything one could wish to know, from the history of French "crêpes"[2] to the explanation of the most famous quantum mechanics principles[3].

In addition to a global encyclopaedia referencing -almost- all of human knowledge, the use of the internet is also changing the way we learn. Indeed, there are more and more websites designed to share free online courses. One of the first successes is the Khan Academy. The Khan Academy[4] is an educational website founded by Salman Khan which discusses, along about 2400 videos, principles of math, science, and economics. It is aimed at helping people master the basics, the humble bread-and-butter equations they encounter in elementary and high school. In addition to

[1] http://www.wikipedia.org/
[2] http://en.wikipedia.org/wiki/Crêpe
[3] http://en.wikipedia.org/wiki/Schrödinger_equation
[4] https://www.khanacademy.org/

the video, the website offers software that tracks the evolution of learning, generates practice problems and uses gamefication to reward performance. Complementary to Khan's site, which is ruthlessly practical and oriented toward mastering the basics (Thompson, 2011), the most famous Universities are beginning to freely diffuse their advanced courses over the internet. We can cite as an example the MIT open course ware (OCW)[5] or the open Yale courses[6].

Finally, the launch of massive open online courses (MOOCs) is currently creating a growing buzz (Mackness et al., 2010).

> A MOOC integrates the connectivity of social networking, the facilitation of an acknowledged expert in a field of study, and a collection of freely accessible online resources. Perhaps most importantly, however, a MOOC builds on the active engagement of several hundred to several thousand "students" who self-organize their participation according to learning goals, prior knowledge and skills, and common interests.[...]A MOOC generally carries no fees, no prerequisites other than Internet access and interest, no predefined expectations for participation, and no formal accreditation.
>
> The MOOC model for digital practice (McAuley et al., 2010)

MOOCs are a recent development in distance education which began to emerge in 2012 (Pappano, 2012). In France, a first step was taken at the end of October 2013 with the creation of the website France Université Numérique[7], which tries to promote the development of MOOC teaching in France.

2- Teaching people to be comfortable with and aware of the digital world

A commonplace is to refer young people as "digital natives". Yet their apparent fluency with digital technologies (e.g. smartphone and computer) does not mean they actually understand the technology as well as the hidden implications, in particular about the privacy.

> Although young people interact with digital media all of the time, few of them can create their own games, animations, or simulations. It is as if they can "read" but not "write".
>
> Scratch: Programming for Everyone (Resnick et al., 2009)

[5] http://ocw.mit.edu/
[6] http://oyc.yale.edu/
[7] http://www.france-universite-numerique.fr/

To introduce computer concepts to children, MIT Media Lab developed Scratch (Resnick et al., 2009) forked by Berkeley into Snap, an extended reimplementation of Scratch that makes it suitable for a serious introduction to computer science for high school or college students. With its block interface, it allows to code by combining blocks like Lego.

> Digital fluency requires not just the ability to chat, browse, and interact, but also the ability to design, create, and invent with new media (Resnick, 2008). To do that, you need to learn some type of programming. The ability to program offers many important benefits: it greatly expands the range of what you can create (and how you can express yourself) with the computer, while also expanding the range of what you can learn. In particular, programming supports the development of “computational thinking,” helping you learn important problem-solving and design strategies (such as modularization and iterative design) that carry over to non-programming domains. And since programming involves the creation of external representations of your problem-solving processes, programming provides you with opportunities to reflect on your own thinking and even to think about thinking itself (DiSessa, 2001)
>
> Scratch: Programming for Everyone (Resnick et al., 2009)

3- Using robots as new pedagogical tools

Robots have a great potential to become ideal tools for teaching a wide range of engineering disciplines. Indeed, robots are intrinsically multidisciplinary objects embedding technology from diverse fields, among them computer science, mechanics, electronics or signal processing. The current low-cost projects emerging from the makers revolution(Anderson, 2012) such as Arduino[8] (Mellis et al., 2007), Raspberry Pi[9], and 3D printing bring the tools needed to create robots at a cost that is compatible with the funding available in education. Now the technology lets students create or modify actual robots, a great motivational tool because they permit instant application in the real world, giving a meaningful context favourable for constructivist teaching (Palincsar (1998), Papert et al. (1991)). There are already several great success stories such as the e-puck robot (Mondada et al., 2009) which specifically targets engineering education at university level or Thymio II adapted for teaching robotic and computer science in primary education (Riedo et al. (2012), Riedo et al. (2013)).

[8] Open source electronics boards
[9] Low cost micro Linux computer

9.1.1 Education and the Poppy project

The Poppy platform was initially designed for research purposes and even more specifically for studying biped locomotion and human-robot interaction. However, it has been designed with open science goals in mind, both to share our research and create tools for researchers. As we are convinced of the need for multidisciplinary contributions in order to improve the state of the art in the robotics field, we decided right from the beginning to use and create modern and easy-to-use tools. This choice has strongly affected the way we designed our platform. Indeed, being simple to use, easily reproducible and hackable, modular, 3D printable and as plug 'n play as possible lead to the development of hardware (Poppy) and software (pypot) tools that can be also used by non-expert people.

Thus Poppy meets a growing societal need: education and training in technologies combining computer science, electronics and mechanics, as well as a training tool for the emergent revolutionary 3D printing process. Since October 2013 (open source release), we have been contacted by several Fablabs, universities, engineering schools and even high schools. We have had the opportunity to meet with educational teams and it appears they are looking for new motivational tools for group projects.

In this context, the Poppy platform appears well suited. Indeed, it integrates advanced and yet easily accessible techniques (3D printing, Arduino, Python) in an embodiment that motivates students and the wider public. With its openness, design and rather low-cost, Poppy is highly hackable and provides a unique context for learning and experimenting with these technologies in a Do-It-Yourself (DIY) way.

Several experiments with Poppy in middle and high schools, science museums and Fablabs in France and abroad are already underway and will be discussed in the upcoming sections.

9.2 Educational exploration with Poppy in high school

After the artistic residency that took place in the chapel at the Saintonge Sainte Famille high school (see chapter 10), some teachers have become interested in the educational potential of the Poppy project and would like to integrate it as a common thread into the school year.

Poppy was initially designed for research purposes and seems to be also adapted for higher education. Yet using Poppy in secondary education seems excessive as it is expensive and the use of high quality servo-actuators is not really justified. However, the experience with high-school students is still interesting and we accepted this opportunity to do a pilot experiment.

For the teachers, the main goal was to gain experience of using such tools in a project context and evaluate the potential and limitations for educational purposes.

For us, we were interested in the reaction of young students to Poppy and in getting an opinion on the relevance of Poppy for education at this level. Also, it was a real crash test of our design (hardware and software) in non-experienced hands and outside the laboratory.

9.2.1 Proceedings

The experiment took place in the Saintonge Sainte Famille high school on May 26th & 27th, and involved near 40 *première STI2D* students (equivalent to UK Year 12) preparing a professional baccalaureate and three teachers ("*Energy and environment*", "*Architecture and construction*", and "*Digital information systems*"). It was organized as a workshop in three 4-hour sessions. The last two hours were dedicated to oral presentations in the lecture hall allowing students to share their experiences and work (see Fig. 9.3 and Fig. 9.4).

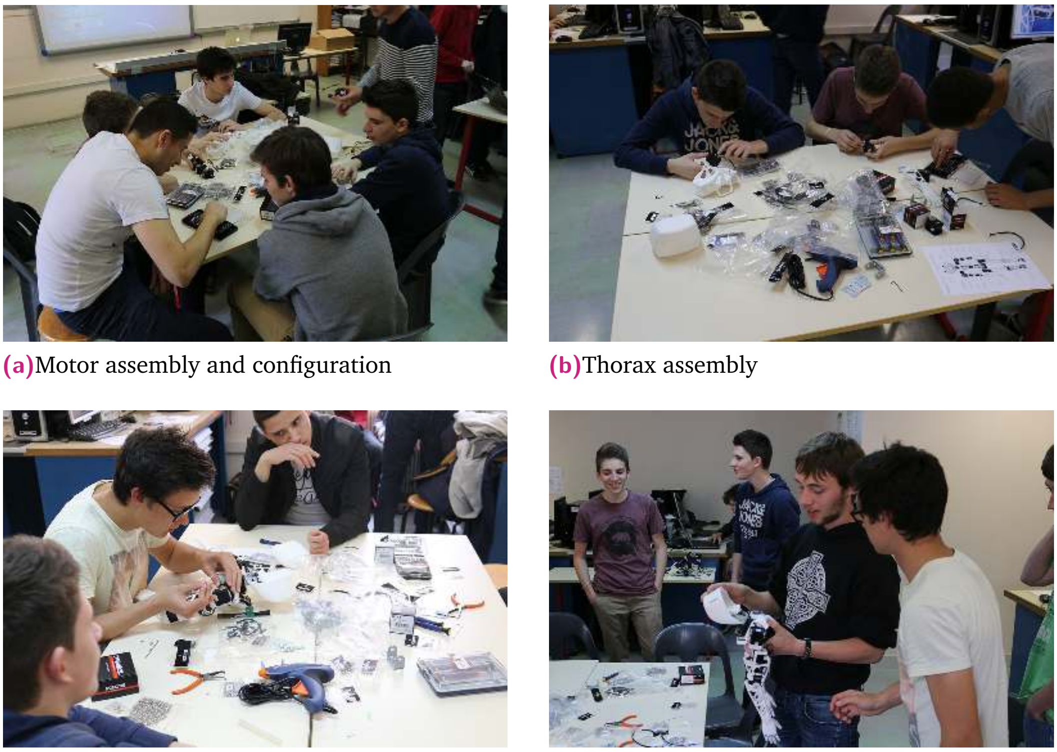
For this first pilot experiment, we decided to reduce the cost by using only a sub-part of the whole Poppy. For us the most relevant part for high-school students was the upper body (thorax, head and the two arms), because it avoids to work on complex sensory-motor behaviours such as balancing and walking while keeping the expressive potential of Poppy. The total cost of Robotis Dynamixel motors, electronics and 3D printing service was €2500 (20 % tax included), the BOM list is available in the appendix **??**.

Students were assigned several sub-tasks that we will describe in the following sections.

9.2.2 Assembling Poppy

The assembly of Poppy was divided in three groups: one was doing the assembly of the thorax and the head (4 motors) and two others for each arm (3 motors). At the end of the first day the half-Poppy was assembled (see Fig. 9.1) with little difficulty.

However, as we will discuss in more detail in the upcoming section 9.4, we experienced some difficulties with the high school internet connection and all documentation is online... So it was unfortunately rather difficult to evaluate if our



(a)Motor assembly and configuration

(b)Thorax assembly

(c)Arm assembly

(d)Poppy almost finished, only the face is missing

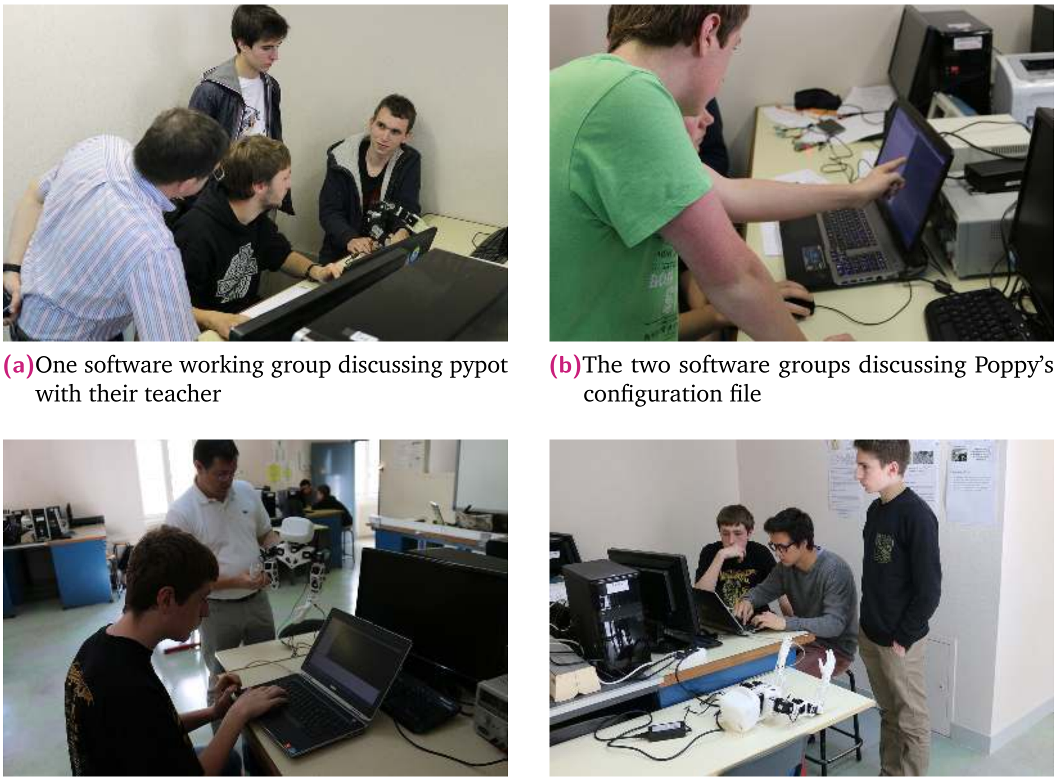
Fig. 9.1.: Poppy was assembled by 3 parallel working groups divided into thorax + head and the 2 arms. At the end of the first day, the half-Poppy was assembled and functional.

documentation was clear enough for high school students. Yet, from that we experienced explaining the "how-to" guide and it appears we need highly detailed documentation for the very first steps. General guidelines seem to be enough to achieve a well-assembled Poppy robot.

9.2.3 Python programming with pypot

Two working groups were dedicated to learning basic Python programming with pypot (see Fig. 9.2). The teacher had selected these students because of their enthusiasm for computer science. Indeed when they heard about the Poppy project several weeks earlier, they became interested in Python and began to look for further information on the internet.

The first hours of the software workshop were complicated. The school computers were not outdated but were running on Windows and used by a lot of different people, so configurations were not consistent between machines and sometimes rather exotic. The installation of the necessary tools (Python + packages + text editor) was particularly long and painful. We will come back to this point in the section 9.4.



(a) One software working group discussing pypot with their teacher

(b) The two software groups discussing Poppy's configuration file

(c) Experimentation through trial and error

(d) Transmission of knowledge between classmates

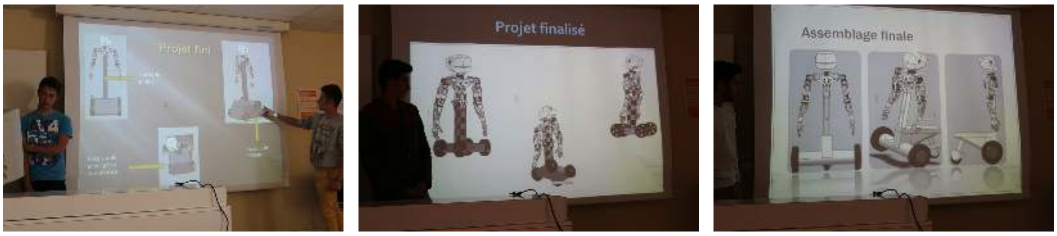
Fig. 9.2.: High-school students discovering programming with Python and the pypot library

Once the desktops were set up, students took a look at the basic pypot tutorials[10]. They first tried to control one Dynamixel motor (reading/setting positions). Then they used a group of motors (each with one Poppy arm) and we introduced the pypot robot configuration feature. At this point they were able, thanks to the basic pypot tutorial, to create a robot with specific motor configurations and make it move by setting direct goal positions or using sinus trajectories. This improvised complexity slope lead them to a good understanding of the very basic features of pypot and Dynamixel motor properties in just 2 hours and without any previous Python experience.

The hardware groups then assembled Poppy, the software group modified Poppy's configuration file and adapted it to their own Poppy. At the end of the first day, the two groups were able to make their Poppy move.

The software team continued their work the next day, trying to create more complex behaviour. One participant (see Fig. 9.2c), managed to create on his own (we did not

[10] `http://poppy-project.github.io/pypot/tutorial.html`



(a) Caterpillar design (b) Four independent wheel design (c) Two motorized and a free-wheel

Fig. 9.3.: Students of the design support work group explain their choices to others.

write a single line of code) the copy-arm behaviour we showed in the Poppy overview video[11]. Then he explained how he did it to his classmates (see Fig. 9.2d).

After learning how this behaviour was achieved, another group decided to make Poppy clap its hands. By trial and error they managed to make the basic movement by using sinus. This self-exploration required an understanding of what a variable is and how a sinus acts and they experimented on their own with the different sinus properties such as frequency, amplitude and even offset. As Mitchel Resnick explained in (Resnick et al., 2009), here the meaningful context of making a robot move arouses the students' curiosity in mathematics and computer science concepts.

9.2.4 Design of a Poppy's support in Solidworks

As we were only building the very upper part of the robot, several students worked in pairs to design a wheeled platform supporting their own Poppy version. The teacher's instructions were to create a mobile support for their Poppy with enough room to include batteries and a computer.

For most of them it was their first experience using parametric modelling tools. However, the final result (see Fig. 9.3) is quite impressive. All working pairs managed to finish the basic idea of their design choices and integrate the Poppy robot.

One really interesting point is that they all chose a different design to make their support structure move. Some used caterpillar tracks, others four or two wheels and one freewheel. The Poppy robot gives a pretext for creation and leaves students free to explore.

[11] http://vimeo.com/poppyproject/poppyoverview

(a)First Poppy's moves drew particular attention.

(b)Live demonstration of the robot behavior students created

Fig. 9.4.: After finishing the hardware assembly, students began to make Poppy move using the pypot library. They manage in just few hours to create impressive movement and behaviour despite no previous development experience with Python.

Of course, they did not manage to produce the prototype within the two days but it showed to the teaching staff what kind of project they could launch if they had a 3D printer to produce the student works.

9.2.5 Documentation

Finally, the other students were in working groups in charge of reporting the different workshop activities. They had to take pictures of the robot assembly and capture an overall view of the project to extract meaningful information for building SYSML diagrams with MagicDraw. They also had to practice English and report on the translation of technical words in a robotic context.

The students of these groups were far less enthusiastic than the others, some even discreetly sneaked into other workshops. Of course, building and programming a robot is far more fun than reporting it. This shows some limitation of a project such as Poppy: there is not enough work for lots of students, introducing inequalities in the repartition of tasks.

9.2.6 Results

The student team managed to assemble a fully functional Poppy. Groups working on control were able to make a live demo of Poppy moving at the end of the workshop (see Fig. 9.4).

This experience was very instructive on several aspects relative to the usage of Poppy for education purpose. In particular, it raises some problems we would have never thought about without a "real world" experimentation in a school environement. All these points will be largely discussed in the section 9.4.

9.3 Fablab workshop

On March 22nd & 23rd 2014, UniverSciences[12] organized a hackathon for the general public on the assembly of a Poppy robot (see Fig. 9.5). It involved 21 robotic enthusiasts (aged from 23 to 46) with various backgrounds (informatics, electronics, physics, sociology, mechanics, architecture).

Participants were divided into several working groups during the two days. While a group was dedicated to the actual assembly of Poppy, others were exploring how to program the robot in Python or working on designing and 3D printing hardware improvements. During the weekend, nearly 100 visitors came to visit the workshop and some of them participated in the hackathon.

This hackathon was really interesting for us because it was the first time a Poppy was assembled without any member of the team present, therefore we were able to see what happens when people are left alone with just online documentation, the forum and the robot parts. We were giving live support through the forum (see associated topic[13]) and at the same time completing the documentation[14] as problems occured.

The support was needed to help the team and showed us some points on which we have to improve in order to make Poppy easier to use and assemble:

Cable routing: if mounted too soon, some motors have to be dismounted in order to be connected, which can be frustrating.

Dynamixel motors: require several critical steps which can cause painful difficulties afterwards, among them:

- The horn of the motor has to be correctly oriented to set the initial position but can be only mounted once.
- The configuration of each motor needs to be set individually before being assembled. The motor has a unique ID for communication set to 1 by default.If all motors are plugged with the same ID, neither communication nor configuration is possible.

Screw size: This is something we had not considered but it is actually difficult for people to distinguish the differences between two and where they should go.

[12] Paris museum of sciences and technologies
[13] https://forum.poppy-project.org/t/poppy-project-at-la-cite-des-sciences-et-de-lindustrie
[14] https://forum.poppy-project.org/category/documentation

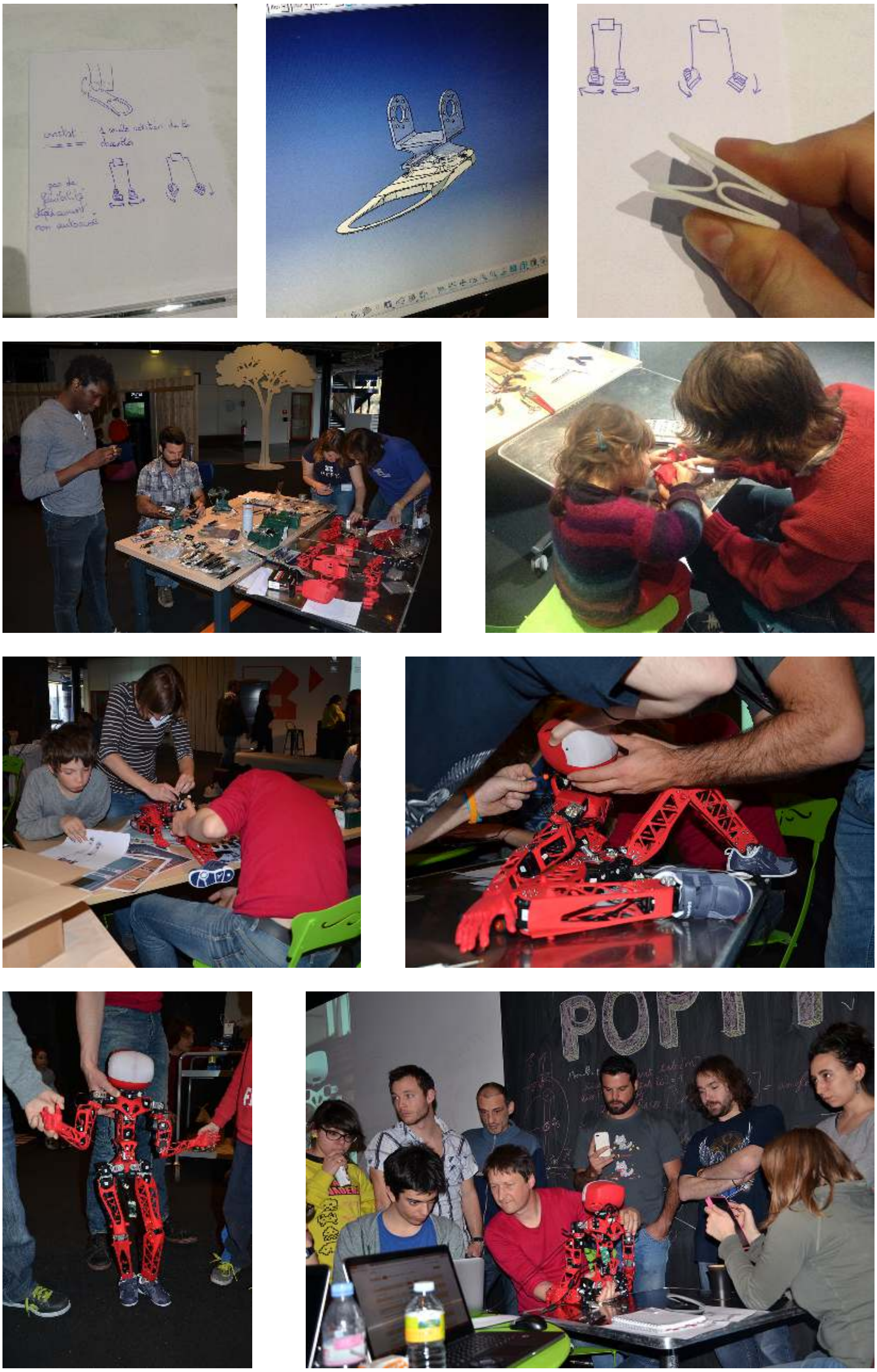


Fig. 9.5.: Several photos taken during Poppy's assembly for the UniverScience hackathon.

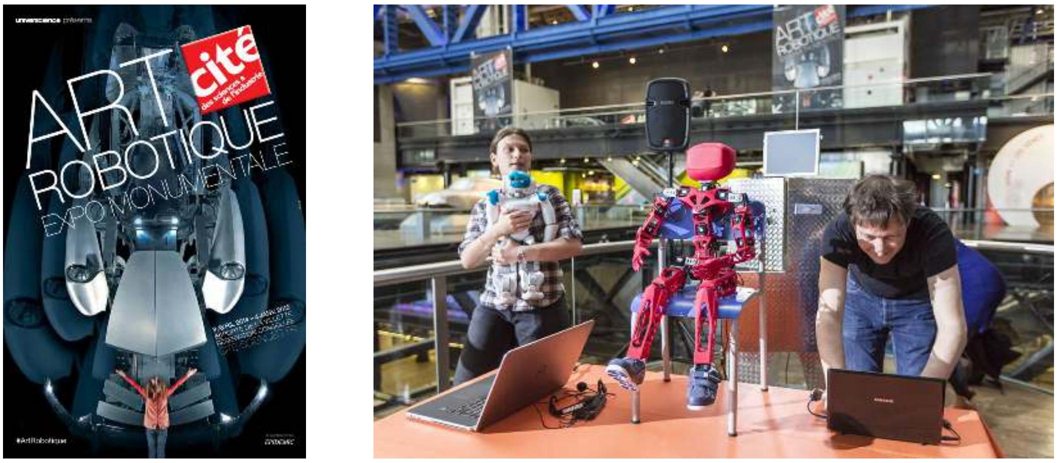


Fig. 9.6.: Currently Poppy is used by UniverScience for mediation acts along the "Art Robotique" exposition.

Despite these minor difficulties, this group of new users, self-trained using only online community tools were able to build the whole robot from scratch and make it move using the pypot library. Eventually, two young students had an idea: 3D printing an ankle that could give some articulated movement to Poppy's feet. They designed a new original semi-passive solution for the ankle joint as well as a robot helmet that was 3D-printed and assembled within the time of the workshop (see Fig. 9.5). Also the feedback we received from the mediation team was really good and they are enthusiastic to repeat this kind of event.

Since this hackathon, Poppy has been used by the UniverSciences Fablab, but also for activities and animation aside from the "Robotic Art" exhibition (08/04/2014 to 04/01/2015) Fig. 9.6. It formed part of the opening of the exhibition on April 8th, and is part of the science show "l'ère des Robots" (The Robots Era)[15].

9.4 Lessons learn

These experiments were really instructive as we had the chance to do a real crash-test in ecological conditions: a group of people left alone to build a complex robot in a Fablab style environment and real students in their high school and with their own equipment. As we personally took part in the workshop in the high school, this section resuming the educational interest will be more focused on this experiment.

For example, Mitchel Resnick cited the experience a teacher had while using Scratch for a school project (Resnick, 2008):

[15] http://www.cite-sciences.fr/fr/au-programme/expos-temporaires/art-robotique/activites-associees/

> There is a buzz in the room when the kids get going on Scratch projects. Students set design goals for their projects and problem-solve to fix program bugs. They collaborate, cooperate, and co-teach. They appreciate the power that Scratch gives them to create their own versions of games and animations.
>
> Karen Randall, teacher at the Expo Elementary School in (Resnick, 2008)

This description closely resembles my experience with the high school students. We were really surprised and pleased at how the students were able to switch from a pyramidal way of learning, where they are passive and expect the teacher to transmit knowledge, to a horizontal one or even a bottom-up one, where students are proactive and ask the teacher for explanations, use these to understand novel concepts then transmit them to their classmates.

This experiment turns out to be much more instructive than expected and taught us several lessons:

1- A robotic project is very motivating: the goal of having their own "cool" robot moving was a really impressive motivational fuel. Students were not discouraged by the difficulties that cropped up on the path toward achieving their goal. Also they had a positive way of trying to overcome them. Surprisingly, they did not seem bored by all the inconveniences that occurred such as installing tools on machines, Python syntax tricks or remounting an assembly because they had made a small mistake. Actually, the teacher seemed far more bothered than his students.

2- Robots provide a meaningful context, conducive learning mathematical and computer science concepts:

Problems that crop up when making and controlling a robot require the use of math and students see in this the tools needed to reach their goal. They are learning math because it is on the way to achieving a “cooler” and more rewarding goal. In this pilot experiment, the students were from a professional baccalaureate, typically the kind of students saying they are not "born mathematicians" and so will never succeed in doing it. However, while trying to control Poppy, especially making it clap, they were confronted with mathematical problems and used sinusoidal signal to resolve them. Using Poppy they were able to see, immediately and in real life, the usefulness of sinus and eventually began to explore on their own the role of each parameter (amplitude, frequency, phase, offset).

The same effect can be observed in teaching computer science and programming. Indeed, having to deal with strange syntax rules, complex commands and austere interfaces is discouraging. Using robots makes the programming experience much different. You are not studying computer science to print characters on a black system console, but you are using programming to make a robot move! During the experience with the high school students we saw two main advantages. Firstly, the motivation of making a robot move helps students to overcome the syntax tricks problems. Secondly, teaching Object-oriented programming (OOP) using actual object is far easier. Indeed, understanding that a motor is an object with properties and functions is really meaningful.

3- Poppy is open source, it can be hacked and adapted to specific needs: The fact Poppy is at the same time open source, 3D printed and "cool" definitely makes it an ideal application for education. Its design catches the students' attention while the free use of all its sources allows teachers to create educational content based on exploration and understanding of the state of the art and lets students express their creativity by hacking the platform.

Also, because Poppy is modular, it allows teachers to adapt its use to their needs. Here, it was possible to only take one part of the robot and use the student's creativity to create the missing elements. This kind of usage would not be possible if Poppy was a commercial and closed robot.

4- Do not rely on educational equipment: The internet access was really bad, so slow that our forum could not be displayed. Also, even if school computers are quite up-to-date, they are used by a lot of different users with various levels of expertise and potentially have incompatibilities or odd configurations. So we cannot expect to find a working environment close to the one we have in our labs. We have to either ensure before the workshop that every desktop is ready or use tools more robust to system configuration issues.

5- Well-designed and stand-alone IDE is needed.

The previous point showed us that the current way of working with Poppy was not adapted to its use in high school. Indeed, we lost almost all of the first 4-hour session just trying to set up a development environment allowing the use of the pypot library. We eventually gave up on certain high-school machines and we stopped when we got the minimum required tools. However, there are complementary tools such as ipython notebook that greatly improves the convenience when programming in Python.

Thus eventually, high-school students managed to develop complex behaviour with Poppy but the teacher had pre-selected them on the basis of their enthusiasm about computer science. Students without prior excitement would probably give up when faced with so many system setup difficulties.

This problem led us to look for improvements and we eventually found an interesting way to overcome these setup issues. The week after, we submitted a project proposal to the Aquitaine Region funding call on the design of a development environment adapted to the use of Poppy in educational context. We will describe this project more in future work (see section 9.6).

7- Content should be translated into the native language:

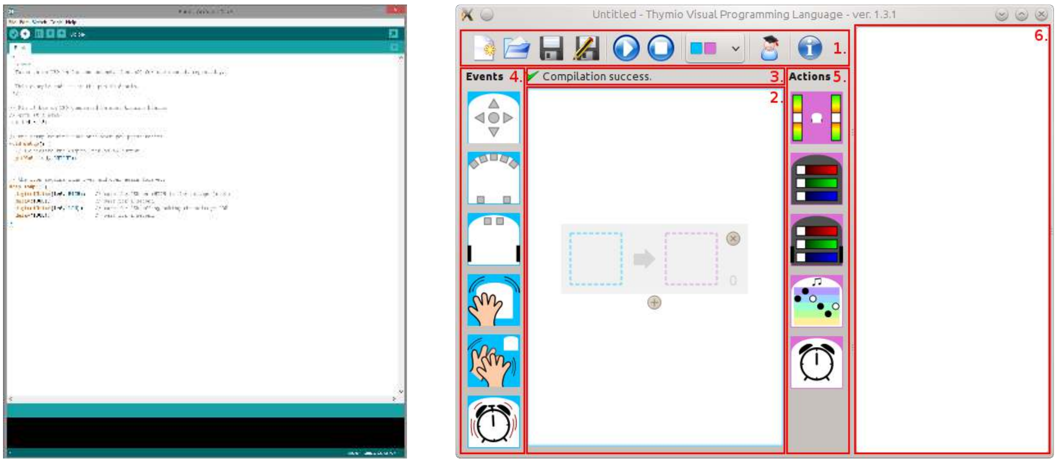
The Poppy project was initially a research project and English is the only language used for information and documentation. Yet this choice is a real problem in the educational world. In France, a large number of people are either not fluent or completely unable to speak and understand English. Of course, the younger the students, are, the less they are comfortable they are with foreign languages. In addition, teachers' English level is also pretty poor and so the fact that the information we give is in English is a stumbling block to the diffusion of Poppy in the French education system.

9.5 Discussion

Open source 3D-printed robots can help in the creation of meaningful contexts allowing people and students to explore several aspects of science, among them, computer science, mechanics and electronics. In particular, the use of Poppy can be a great gamification tool for scientific mediation and educational purposes.

As it allows people to explore by themselves the basics of robotics, Poppy confront users with the use of programming and mathematical concepts which can raise the awareness of the general public about complex scientific challenges and the usefulness of such technical sciences, often studied in a meaningless context. Poppy is not a "black box". It has to be assembled and even if it is quite simple to program, people need to understand the basics of programming and then find out by trial and error how to achieve a desired behaviour. Thus, using Poppy leads to the use of the scientific synthetic methodology "understanding by building" and a constructivist learning approach, which are great methods for both fostering critical thinking and creating deep reflexions on the understanding of nature.

Also for a small amount of time, students are scientific researchers lost in an open field of exploration. It is a really great way of introducing people to scientific thought, doing experiments by trial and error, trying to understand how to achieve a desired goal, creating models using math or algorithms.



(a)Arduino (b)Thymio 2 IDE

Fig. 9.7.: Arduino and Thymio are stand-alone IDE allowing to have plug 'n play devices, immediately useable right out of the box. Also, the user interface is done to make writing and debugging code easy. Creating great IDE seems essential for success in education.

However, even if the two experiments conducted showed a high potential for Poppy's societal impact on education, there are still several limitations:

Firstly, Poppy is low cost for research labs but it is still too expensive to have a real impact on the education system. Its current cost makes it accessible only to some privileged high schools. Also, even in the high schools that can afford such a robot, it is only possible to have one or two which leads to inequalities in the project course. A few students have the chance to build the robot or make it move while others are passive and cannot take part in the activity.

However, using such high quality actuators is not really relevant for educational uses, at least before baccalaureate. While the humanoid shape seems to really have an impact on the students' motivation, the fact Poppy is tall and compliant is not a real need. We could and should definitely propose a cheaper alternative more adapted to elementary and high schools. Poppy being still relevant in universities, engineering schools and of course research labs.

Secondly, the way the robot is controlled is not adapted to an educational context. Having to spend time installing each package needed and then figure out why the system crashes is not really interesting or relevant for students. They should instead spend time on programming and using the robot. Also, there are successful educational projects such as Arduino (Banzi, 2009) and Thymio (Siegwart et al., 2014) propose such plug 'n play IDE (see Fig. 9.7).

Finally, we should be careful that the activities associated with Poppy are not only playful but have a real educational impact. To do so, actual teachers need to create educational content. Of course it is easy to get the students attention for 2 days with a project as fun and playful as Poppy but creating relevant content running as projects along a whole school year is far more challenging.

Confronted with these three points, we found promising technical solutions and we set up a partnership with educational local actors to propose a project to the Aquitaine Region in order to get the resources needed for development.

9.6 Future work

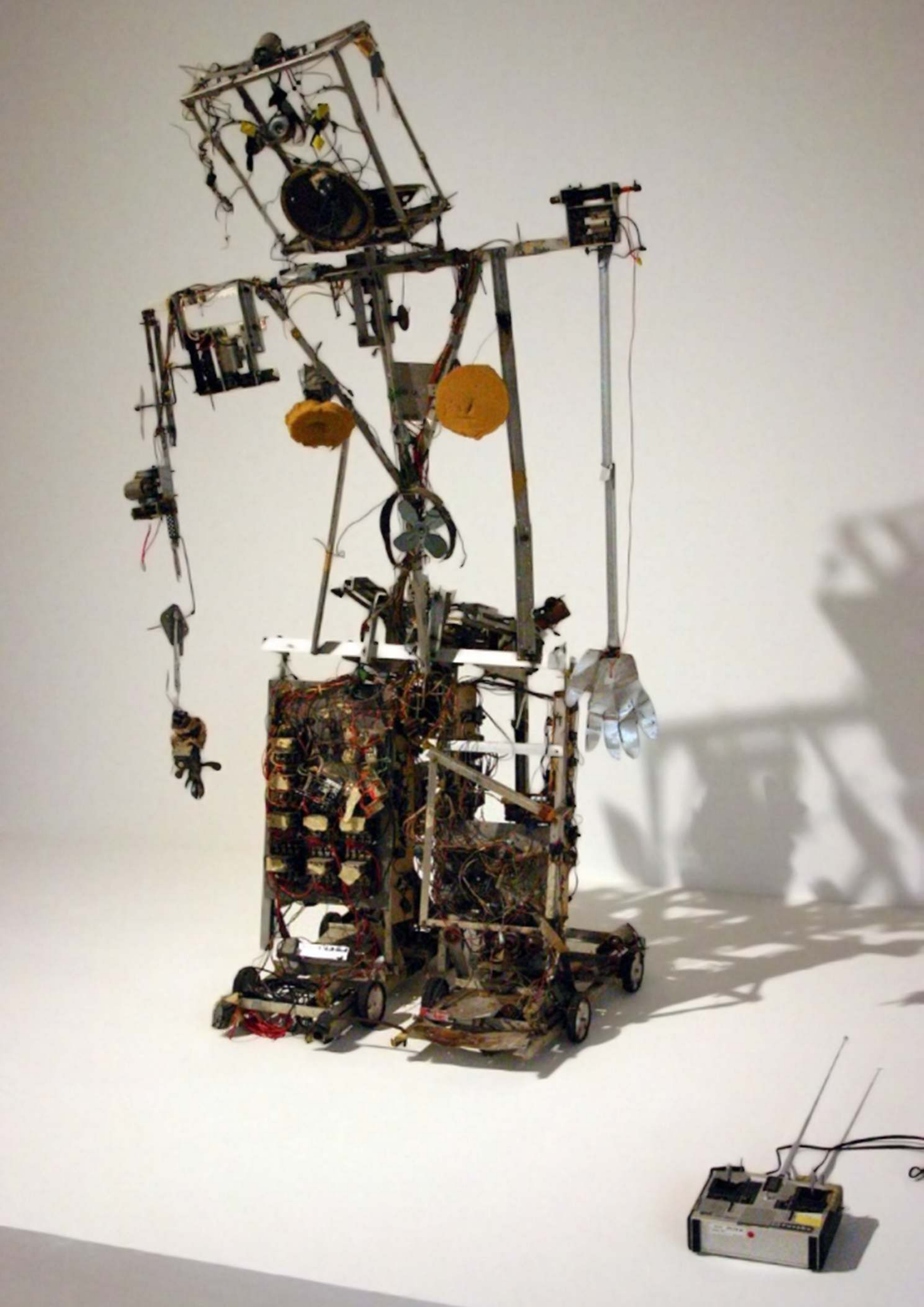
As we explained previously, experiments involving Poppy have shown potential for educational application. However, given the cost of Poppy, the complex setup to run pypot and the lack of educational content, the relevance of Poppy in high schools and first years of universities is still questionable.

So we decided to address these needs and we launched a partnership with a famous engineering school (ENSAM) and a high school (Saintonge Sainte Famille) working towards the creation of educational content and user experimentation in real contexts.

The Poppy Education project aims to develop, utilize and disseminate a teaching platform based on the use of Poppy for integrated learning of computer sciences and engineering. In particular, it aims to provide:

- An open source integrated development environment (IDE) based on web technologies with an embedded Python interpreter. This application can be stand-alone and will allow both the sharing and deployment of educational resources based around the use of the robot, and a friendly interface to program and monitor it.
- Design a mini version of the current Poppy robot. Using $20 Robotis Dynamixel XL320, it would be possible to build a 30cm tall 3D-printed Poppy for around $500.
- educational content using this environment, Poppy, and Poppy mini in real situations with high school and bachelor courses (thanks to our partnership with ENSAM and Aerocampus).
- the dissemination of these educational tools under open source and Creative Commons licenses.

- the translation of the software, the documentation and educational content in French and English. Other language translations will be done with external contributions.



Art

10

> *Artists and mathematicians have a lot of points in common in their way of working in the end,. There is this important role for the mathematician and for scientists generally, of inspiration, culture, exchange of ideas, schools of thought which change over time or which vary from one country to another, and at the same time, scientific universality is the same as that found in the arts.*
>
> — **Cédric Villani**

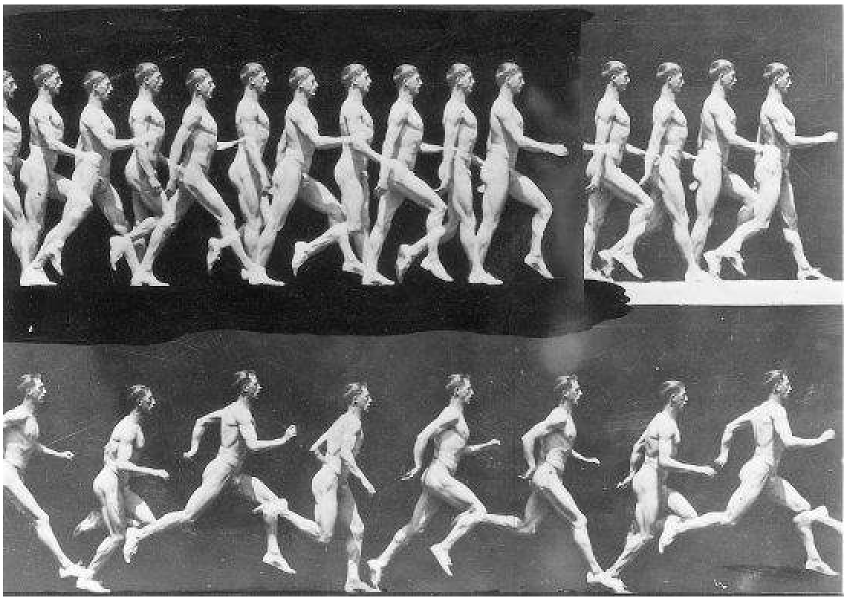
10.1 Introduction

Science and Art have been intermingled for ages. Leonard da Vinci[1] is a meaningful example of the very existence of such a scientist-artist status. Contrary to popular opinion, the two worlds are more in agreement with each other than they are in opposition. Indeed, in both areas, although the methods of investigation and processes implemented to experiment the world may differ, both artists and scientists are motivated toward the same goal: understanding the world around them to share and exchange knowledge with others.

Several results and technologies coming from scientific applications have been transformed into material resources, instrumental, or technical processes for Art. For example, chronophotography, invented by Etienne Jules Marey and Eadweard Muybridge to study human and animal locomotion (see Fig. 10.1a), has been a source of inspiration for Futurists[2] who reproduce in their work the decomposition of movement visible in chronophotographies (see Fig. 10.1b and Fig. 10.1c). Futurists are proponents of the fusion of art with technology and the natural sciences: "*The*

[1] Humanist artist who lived during the Renaissance, Leonardo da Vinci had many different hats. He was a painter, sculptor and musician but also an engineer, mathematician, physicist, biologist, astrologer, architect and urban planner. Famous today for his painting, he also left behind visionary flying and war machines.

[2] At the time of the Futurists, Art sought to express the dynamism of modern life and the representation of contemporary society: they consider movement and speed as the most significant emerging phenomena of the twentieth century.



(a) Example of a chronophotography of a man walking and running.

(b) Marcel Duchamp, *Nu descendant un escalier*, oil on canvas, 146 x 89 cm (1912) - Philadelphia Museum of Art, Philadelphie (États-Unis).

(c) Luigi Russolo, *Dynamisme d'une automobile*, 1912 – Centre Georges Pompidou.

Fig. 10.1.: The chronophotography invented by Etienne Jules Marey and Eadweard Muybridge to study human and animal locomotion has been a source of inspiration for artist such as Marcel Duchamp and Luigi Russolo.

method of constructing a machine is similar to the method of undertaking a work of art" (Severini, 1917).

If Science proves to be a source of inspiration for Art, in return, Art is also involved in Science and plays a major role in providing the general public with an understandable representation of complex scientific discoveries or concepts. Indeed when mixed with science and technical applications, artistic creation also serves as an original vector

Fig. 10.2.: In a big egg that has just opened, a tribe of young robotic creatures evolves and explores its environment, wreathed by a large zero that symbolizes the "origin". Beyond their innate capabilities, they are outfitted with mechanisms that allow them to learn new skills and invent their own language. Endowed with artificial curiosity, they explore objects around them, as well as the effect their vocalizations produce on humans. Human, also curious to see what these creatures can do, react with their own gestures, creating a loop of interaction which progressively self-organizes into a new communication system established between man and ergo-robots. "Ergo-Robots: Artificial Curiosity and Language" is an installation and experiment presented in the exhibition "Mathematics: A Beautiful Elsewhere", from 21st october 2011 to 18th march 2012, in Fondation Cartier pour l'Art Contemporain, Paris, France.

of mediation to reach the uninitiated. The sensory experience brought by a work of art favors the appropriation of innovative technology or research results, especially because it demystifies the complexity of the mechanisms involved with the sensory approach and concrete production is more easily comprehensible than theoretical explanations. In doing so, Art often helps to understand issues or potential uses, defuse fears of novelty and expand distribution. Astrophysics is a great example of this. None of us will ever see a giant black hole or the sun becoming a red giant. In this context, artistic contributions are vital as it would be difficult to arouse the attention and interest needed for funding of such expansive research without appropriate mediation.

In this direction, the Flowers team in collaboration with the artist and movie-maker David Lynch lead a project called "ergo robots" (see Fig. 10.2) as part of the "Mathematics: A Beautiful Elsewhere"[3] exhibition in 2011 at the Cartier Foundation for Contemporary Art. The experiment of the Flowers team addressed artificial curiosity, the embodiment and discovery of language in robots and was aimed at, among other goals, interaction with a non-science-enthusiast audience. In this

[3]Mathematics: A Beautiful Elsewhere is a unique exhibition created by the Fondation Cartier pour l'art contemporain with the aim of offering visitors, to use the mathematician Alexandre Grothendieck's expression, "a sudden change of scenery." The Fondation Cartier opened its doors to the mathematics community and invited a number of artists to accompany them. They are the artisans and thinkers, the explorers and builders of this exhibition. More info on `http://fondation.cartier.com/en/art-contemporain/26/exhibitions/294/all-the-exhibitions/89/mathematics-a-beautiful-elsewhere/`

experiment the collaboration between science and art has revealed art as a medium and tool for scientific mediation.

Yet this kind of collaboration between a scientific laboratory and artists is rare. We indeed notice the recent increase in complexity and in the level of culture/education needed to understand the latest work coming out of these two worlds[4]. Without knowledge of art history, it is sometimes difficult for a scientist to understand contemporary works of art. Likewise, it is somewhat difficult for artists[5] to appreciate the scientific work because of the technical complexity involved. Interaction is indeed sometimes difficult because we do not speak the same language, but robots raise countless issues and challenges and even if some of them are very technical and require the use of mathematical formalisms to be explored, a wide range of problems can be addressed with the relevant collaboration of artists.

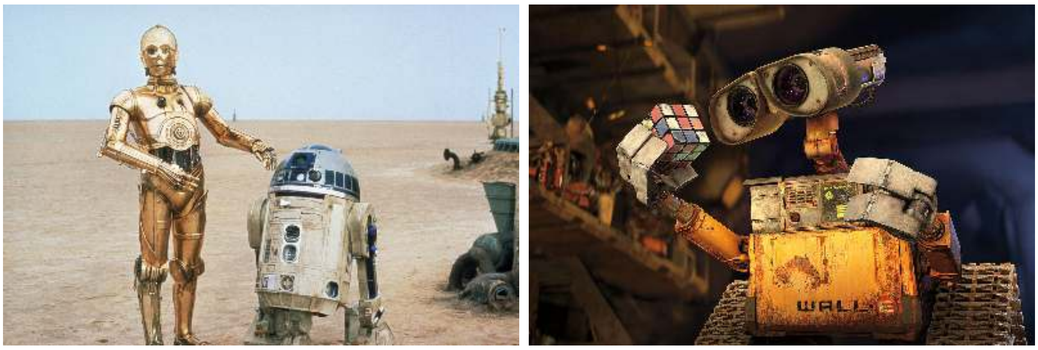
What interests us particularly, and relates to this research, is the alternative perspectives artists bring to the dialogue on the societal impact of robots or human-robot interaction, opening new research axes or new ways of exploring them. One emblematic loan from art is robotic emotion. The expression of emotions in a virtual machine or humanoid is a dual representation: firstly, it depends on how the human emotional functions have been understood; secondly, it is based on a representation of the iconography of emotions. These two representations are well mastered by artists. Brilliant examples can be found in cinema (see Fig. 10.3). Only using basic beep-based sounds, R2D2 is able to communicate and is actually more appealing than C3PO, a classic humanoid robot speaking hundreds of languages. Another impressive example is Wall-E, without an actual human face, the animators have been able to create a wide range of intense emotions. These two examples show how roboticists could benefit from artists' expertise in the field of sensitivity in order to design more expressive robots.

10.2 Motivation

Poppy is fully hackable, so artists could be interested by the freedom for exploration they have to change the morphology or the design of the robot, add new features or sensors, change its behaviour and so on. Also, Poppy is designed to be experimental-proof (see section 6.1.1), it is quite robust and easily reparable so it can be used in rather difficult conditions.

[4]C. P. Snow explained in "two culture" theory (Snow, 2012) that people in humanities and art, and those in science had developed sufficiently different languages and world views that they did not understand each other.

[5]It should be noted that the arts community hosts diverse profiles of artists. Curiously (in many cases), works combining art and science are creations of artists who are none other than converted scientists.



(a) The C3-PO and R2D2 robots in Star Wars: A new hope (b) Wall-E being curious about a Rubik's cube

Fig. 10.3.: Movie and animation are sources of inspiration for roboticists, in particular for the expression of emotions, on this point, R2D2 and Wall-E are two great examples.

As we discussed previously, the artistic community is a rich source of inspiration and can provide new perspectives on scientific and technological questions. The work of artists is complementary to that of scientists. In the open source robotic community we are trying to set up, this complementarity is a great opportunity that we want to encourage by making the robot accessible for non-robotic-expert users.

While it is a real desire to make Poppy accessible and useful for the artistic community, we needed to gain experience of such uses in an actual artistic project to evaluate how relevant Poppy is for artists and what artists can bring into the development of the Poppy project.

10.3 The *Êtres et numérique* project

The first artistic project in which Poppy is involved is entitled "Êtres et Numérique". This contemporary art project focuses on ways of representing and interacting with movement digitally. It is led by the artists[6] Amandine Braconnier (mixed media artist) and Marie-Aline Villard (dancer-researcher), and supervised by Thomas Desmaison (Point barre[7]) from the Fabrik Pola[8].

A video trailer is available here: `https://vimeo.com/92281019`.

For these artists, the use of a hackable humanoid robot is a whole expressive tool that opens up new horizons. Indeed, a robot permits to dissect and analyse movements. It allows them to play around with its body and model gestures as one could

[6] Comacina Capsule Creative, `http://www.comacina.org/`
[7] `http://www.pointbarre.biz/`
[8] `http://www.pola.fr/`

Fig. 10.4.: The "Êtres et numérique" residency and performances took place in the gorgeous chapel of the *lycée des metiers Sainte Famille Saintonge*

sculpt shapes in clay. Also, the use of non-rigid actuation allows the emergence of unpredictable and unexpected movement, while Poppy under-actuation ensures safe physical interaction.

The first "Êtres et Numérique" work took the form of a ten-day art-science-mediation residency involving members of the Poppy project, the artists with the participation of Jean Marc Weber (music composer) and was supported by the Aquitaine Region. It took place in a Bordeaux (Fr) high school (Lycée Saintonge Sainte Famille[9]), which made its gorgeous chapel available (see Fig. 10.4) for the artistic performances.

This residency was really important for the poppy project: it was the first trial of a real artistic application of Poppy, bringing to light new potential applications and bugs.

The main objective of this first residency was the preparation of a dance performance using the passive properties of Poppy and a physical human-robot interaction with the dancer, but several experiments has been conducted aside from this common thread.

10.3.1 Artistic exploration

The residency week was a playground of exploration for both Poppy and us. The artists were especially interested in how to represent movement and interaction between humans and robots.

[9] http://www.lyceesaintefamille.com/

(a) Poppy dressed to protect it against dirt

(b) Poppy and the dancer playing on canvas covered with pigments

(c) Result of this human-robot interaction

(d) The whole canvas collection was exhibited in the chapel

Fig. 10.5.: Exploration of combined movement with the dancer and Poppy over canvas covered by pigments.

A first experiment consisted in visually tracing over a canvas the combined movement of the dancer and the robot (see Fig. 10.5). To do so, we dressed Poppy to protect it against dirt (see Fig. 10.5a). Then M. A. Villard danced with Poppy on canvas covered with pigments (see Fig. 10.5b). These dancing movements were captured by pigment traces and created paintings of sorts (see Fig. 10.5c). The whole collection was exhibited in the chapel (see Fig. 10.5d.

An alternative way of representing movement is to transform motion into sounds. This is the playground of Jean Marc Weber, an electro-acoustician and music-composer, creating music using the interaction between probabilistic composer and body space motions. To do so, he created plugins allowing the use of webcam,

(a) Poppy playing music with Leap motion device

(b) Poppy playing music with Kinect

Fig. 10.6.: Exploration of Poppy producing music from its movements, tracked by a Leap motion and a kinect.

Kinect or Leap motion with the music-software Usine[10]. For this purpose we explored both the use of Kinect and Leap motion with Poppy.

When Poppy is dressed it can be tracked by Kinect sensors[11], also thanks to the human shape of the hands, it can also play with the Leap motion (see Fig. 10.6a).

Finally, Amandine Braconier (Plastic artist) wanted to make an experiment involving a small child (3 years). In this experiment, the child could modify the robot's appearance by adding clay on it (see Fig. 10.7). To protect Poppy, we wrapped it with cellophane.

This experiment lasted about 2 hours and involved very heavy clay. It turned out to be playful for the little boy to put on as much clay as possible. Eventually he managed to put almost 10kg on the poor Poppy...

On this point, Amandine Braconier says:

> How to use Poppy? The robot is put to the test. I do not know in advance what will happen. Each experiment is photographed and filmed. Some sequences are edited. I try to make other movements emerge that have not been calculated by the researchers. For example, I suggested an interaction with the robot: a child, through game, evolves with Poppy. It is in a continuous back and forth between the two actions, the child and the robot meet, separate and detach from one another. There is singularity and confusion in the exchange

[10] http://www.sensomusic.org/usine/

[11] A video was taken during tests and is accessible here: https://www.youtube.com/watch?v=VVjBVTtPkFE

(a) Poppy is covered with cellophane to protect him.

(b) Little boy playing while Poppy is silently suffering from all the extra weight put on it.

Fig. 10.7.: Modification of Poppy's body by a child using clay.

> that occurs. In the video, we see the child cover the robot with clay. The child transforms it, gives it a monstrous appearance. Also, the child causes the robot's downfall through his actions. The robot is as though exhausted. We do not know which of the two is the monster. The effect goes far beyond the game. The aim here is to choose what we will do with Poppy and how we will show it.
>
> Amandine Braconier (06/18/2014)

10.3.2 The dance performance

The common thread of this residency was the contemporary dance performance involving poetic choreography, alternating phases of autonomous robot movements and passive robot movements provoked by the dancer. Marie Aline Villard describes her work as:

> As a dancer, sharing this experience in movement with Poppy was very interesting both artistically and functionally/mechanically. On one hand, I liked working with the idea of who is leading who, trying to give the illusion of a duet. But on the other hand, it also amused me to show that Poppy remained an object, by playing on active/passive and by genuinely manipulating it. This contrast is a great discovery and artistic research should continue in this direction. Also sharing movement with robotic objects remains fascinating, since we project our own patterns of movement onto them. I was able to verify during a workshop, the extent to which we were projecting our own movement onto the structure of the robot. When we asked students to make a movement between posture A and posture B, in spite of themselves they began moving, dancing, just to record a movement for the robot which leads

> me to say that Poppy has disinhibiting properties to take advantage of in the interaction that dance can greatly contribute to.
>
> Mari-Aline Villard (Dancer) on the use of Poppy for artistic project

The choreography was "sculpted" by Marie Aline Villard using the pypot feature that records motion trajectories directly from physical demonstration (see section 7.2.3). Rather than coding, this "direct programming" feature allowed the dancer to express her idea in her own language i.e. movement. Thus she managed to create an entire floor choreography in which Poppy slowly moves with elegance and sensitivity. Other scenes were improvised dance with physical interaction and guidance where Poppy alternated between active and passive.

The whole performance with several scenes (see Fig. 10.8) lasted about 20 minutes and was shown in front of a live audience on the last day of the residency (with about one hundred people). Despite the problems we experienced during the residency, Poppy acted really great and no technical problems occurred during the whole performance (repeated 4 times in a row without any interruption).

10.3.3 Feedback

Unfortunately or fortunately, the use of Poppy by artists in these workshops was not without problems.(Fig. 10.9).

Firstly, it appears to be difficult for non-expert users to evaluate the real resistance of Poppy. After initially acting overly cautiously with it, they become overconfident about its robustness and no longer pay attention to signs indicating that the robot will break. The experiment with the little boy adding clay was quite destructive. Poppy spent almost 2 hours wrapped in cellophane withstanding kilos of clay, eventually it overheated and two motors in the hip and abs melted. Thus adding a security system that alerts users when the robot is in a dangerous state appears to be a real necessity. Otherwise non-expert users will regularly break motors without understanding why.

Secondly, the close interaction between Poppy and the dancer showed it is really complicated, when the robot is compliant, to avoid motors reaching dead-band. In addition, wires tend to tangle around motors and eventually unplug some of them. To avoid this recurrent problem, we built a primitive system on pypot to check the state of each motor and stop it when it reached a given amplitude. Also, we are working on mechanical ways of avoiding the full-rotation of certain Poppy links.

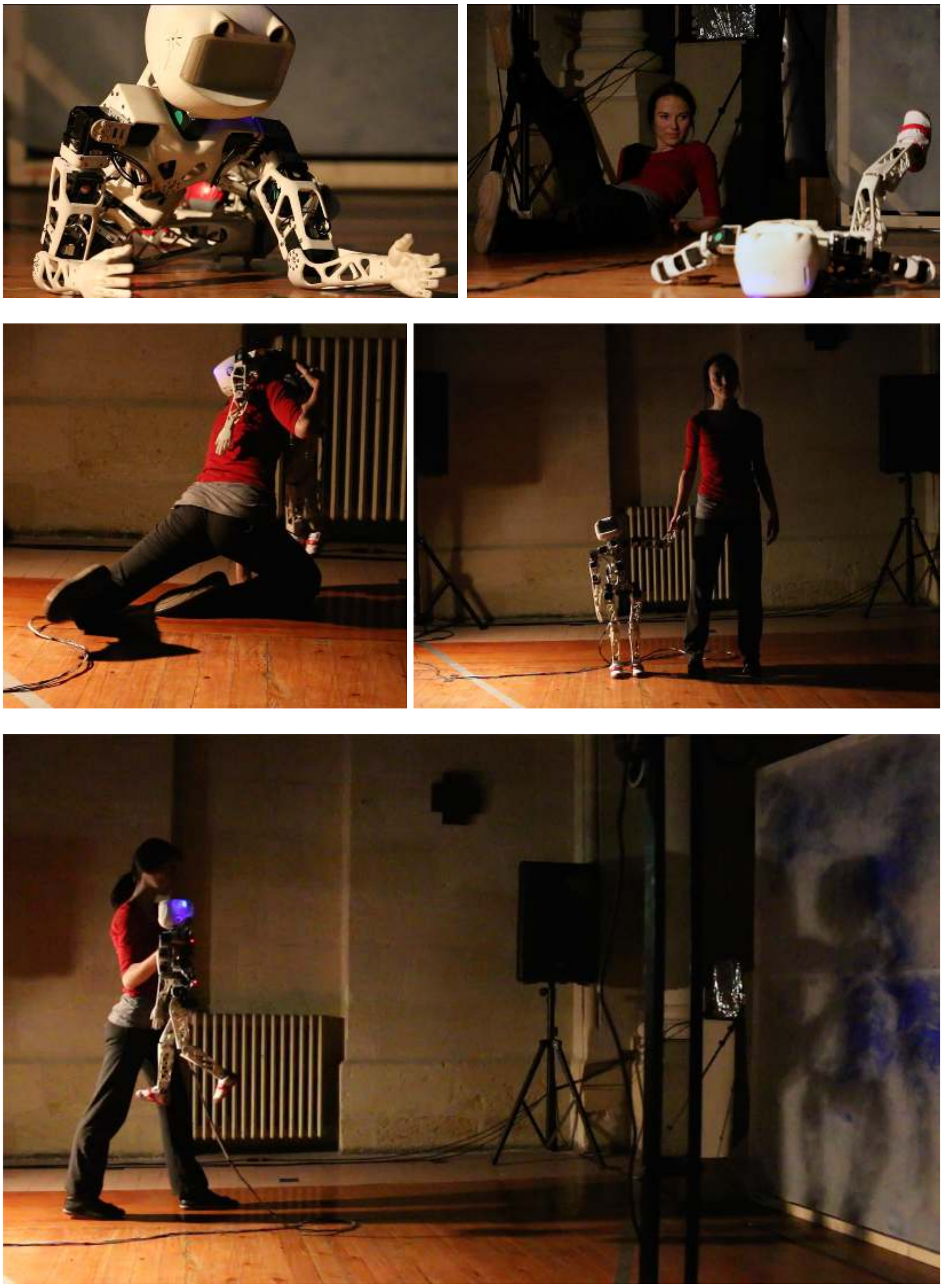


Fig. 10.8.: Extract of the dance performance involving Poppy and the dancer. The complete performance set can be seen here: `http://youtu.be/zp-vsVQcAvs`

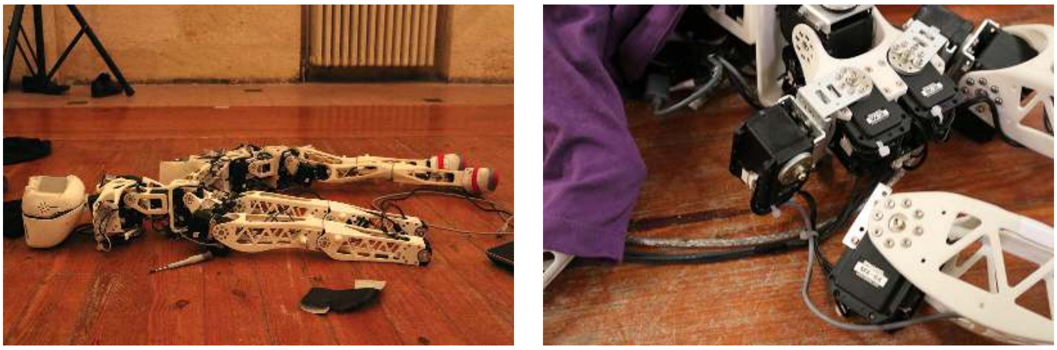


Fig. 10.9.: Due to intensive and unpredicted use, we broke Poppy several times. Yet we always managed to repair it quickly.

Thirdly, the wires are really problematic; they caused a lot of problems throughout the residency. In the context of intensive use of the robot performing large amplitude motion, wires were solicited and eventually some of them became damaged, which provoked short-circuit and so destroyed some motors and micro-controllers.

During the residency, the ease of programming through the Pypot library permitted to design a simple interface allowing the dancer to physically sculpt novel movements, the softness of which could be dynamically controlled.

Pypot programing by demonstration is really basic: positions are recorded at 50hz and played. It is not at all optimized and even a bit bothersome to use because you cannot edit your moves. But this way of "programming" the robot has also a great advantage. The artist really appreciated sculptinggestures in this way rather than editing curves on a nice interface because she could really feel the weight of the robot and work on how to move its mass from one support to another. Finally, because Poppy is not over-actuated, it constrains motion to whom requiring the less power actuation and therefore leads to the creation of more natural motion.

10.4 Discussion

The work the artists did was really amazing and they found unexpected potential in Poppy. Among them, the choreography Marie Aline Villard did was very elegant and sensitive. These movements put Poppy in the domain of sensitivity, rarely seen in humanoid robotics. This choreography is now often used for demonstration of the Poppy platform and closes the communication video showing Poppy being assembled in time-lapse https://vimeo.com/96262428. Also, from a community impact point of view, the topic[12] related to this experiment is by far the most followed subject of our forum.

[12] https://forum.poppy-project.org/t/artist-residency-etres-et-numerique/

But this work also showed the limits of Poppy. As for the Ergo-robots experience[13], the artist residency we did with Poppy was really instructive. Firstly, Poppy has been used in totally unknown and unexpected/able situations, playing in pigments, dancing for 2 hours and so on were not on the development to-do list. It was a real crash test for a novel experimental platform. Again, like in the ergo-robot experience, we learned the hard way how problematic the wires are. Most of the problems we had were due to damaged wires, which provoked short-circuit and eventually destroyed some motors and micro-controllers. It is a real problem in robotics right now, and it is really complicated to find a way to avoid it while keeping a modular and highly hackable robot.

We took note of each hardware problem to be solved for the release of the 1.0 version of Poppy and they are in progress. However, it would maybe not be enough (at the beginning) because it appears we also need to add specific software to monitor and protect the robot motors against overheating and overload. These protections are often task-dependent and the way we should handle it remains unclear. Therefore, the development will be done with the community in an iterative way until an efficient and robust solution is found.

Another problem for diffusion in the artistic community is the cost of the robot. Obtaining the €7000-8000 required to build a Poppy is really difficult and Artists are already having difficulty being paid so they have scarce funding for material.

An alternative could be to rely on the growing Fablab community (Anderson, 2012). The collaboration between an artist and a Fablabs could at the same time, ensure technical support and assistance for using Poppy, avoid the funding problem by using Fablab's Poppy and promote local collaboration between complementary actors.

The interaction with Fablab is a very important point for the Poppy project, we will discuss it in chapter 11.

[13]Ergo-robots had to be functional for 5 months 8 hours per day. Our technologies have really participated in long-term experiments in the real world. In particular, the feedback we got from this experience greatly contributed to the methodology we built for the design of Poppy.

Part IV

Discussions

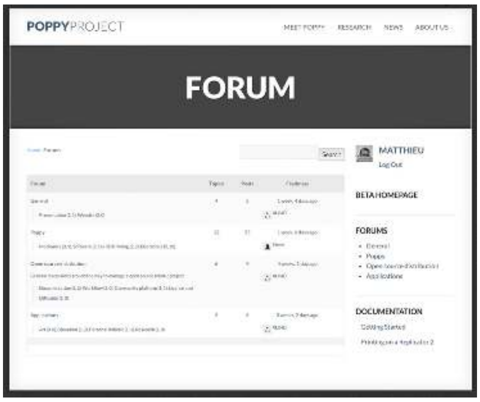
11 Open diffusion of Poppy: perspectives and challenges

11.1 Introduction

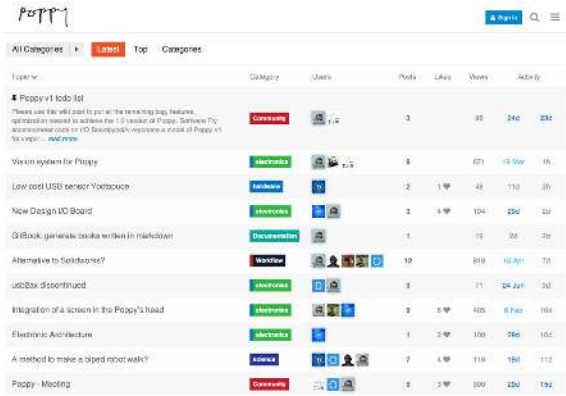
Poppy was initially designed with a scientific objective, aiming to be a new experimental platform opening the possibility to systematically study the role of morphology in sensorimotor control, in human-robot interaction and in cognitive development. Yet, until recently it was complicated because building a robot relied on heavy and costly manufacturing techniques. 3D printing has changed the landscape of what is possible: the design of Poppy transposed it to humanoid robotics and we developed complementary open source electronic and software. As we saw in chapter 8, it allows to explore new body shapes in just a few days. In addition, it enables and simplifies the experimentation, the reproduction and the modification of the morphology in research laboratories. It also allows collaborative working, sharing and replication of the results between laboratories.

On the scientific aspect, the ambition is to become a reference platform for benchmarking and dissemination of scientific results. Also the simple design of Poppy also targets non-engineering scientists so humanities, biologist and so on can contribute to the robotics field by using Poppy as experimental tool. Moreover, thanks to the fact that it integrates advanced and yet easily accessible techniques in an embodiment that motivates students and the wider public, this platform also meets a growing societal need: education and training in technologies combining computer science, electronics and mechanics, as well as a training tool to the emergent revolutionary 3D printing process. With its openness, its design and its rather low-cost, Poppy provides a unique context for experimentation and learning of these technologies in a Do-It-Yourself (DIY) approach. Finally, the possibility to easily modify both the hardware and the software also makes Poppy a useful tool for artistic projects working with interactive computerized installations.

The major challenge is now to succeed in the diffusion of the platform in all these fields. We think this challenge relies on several aspects: create a playful and motivating multidisciplinary community, create educationnal content, improve the hardware modularity, integrate Poppy in the maker revolution by using Fablab



(a)Old BBpress forum



(b)Current Discourse forum

Fig. 11.1.: Evolution of the forum for the Poppy project

for production and distribution, and continue the development of open hardware robotics tools.

11.2 Creating a multidisciplinary community

Toward the creation of such a community, some people are really interesting and have been a great source of inspiration for our work. Among them we can cite Massimo Banzi one of the co-founders of Arduino who spent nearly 10 years working on the Arduino community. We can also cite Jeff Atwood, co-founder of stack-exchange, a software developer who become a specialist in creating tools for the web community. Along his work he developed a real insight and pragmatic vision of the management of people into a community. The feedback he shares on his blog [1] is really worth reading for anyone with a desire to create a community.

The creation of Poppy's community began when we released Poppy under open source license (October 2013). The first interaction people have with the project is via the website. We managed to create a nice and simple website on `www.poppy-project.org` so experts can see potential of the platform while newbies are not frightened off by too much technical information. Yet this website and the open source release drew a lot of attention and we eventually got submerged by emails and community management. Indeed overall we failed to absorb all this enthusiasm, as we were not equipped with adapted community tools such as a wiki, a forum and so on.

Several months ago, we set up a novel forum technology called Discourse[2] and created by Jeff Atwood. This forum, available on `forum.poppy-project.org`

[1] Jeff Atwood's blog: `http://blog.codinghorror.com/`
[2] `www.discourse.com`

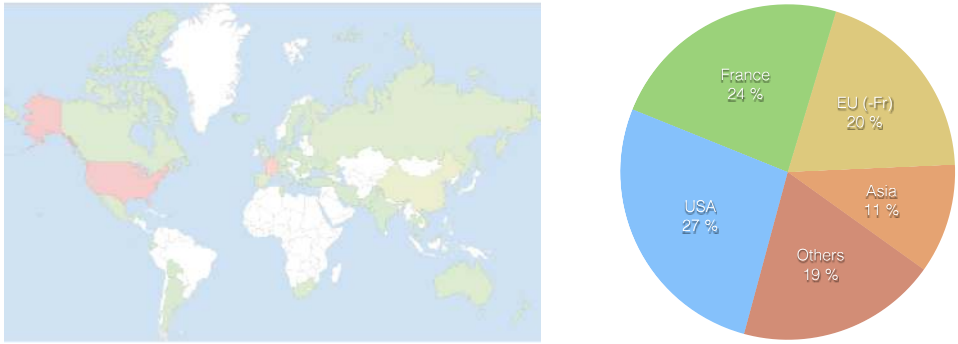


Fig. 11.2.: Activity tracked during October 2013 to February 2014

greatly helps us to manage the community, as it is playful and simple (see Fig. 11.1). It is so efficient that we began to use it also for our internal discussions about Poppy's development. These discussions are made public so we can merge time between internal and external communication, extending at the same time the openness of the project.

While this technology solves one of our problems, there are still several others. Firstly, we did not find a wiki technology as simple and playful as Discourse, therefore currently the project clearly lacked documentation (except from pypot whose documentation[3] is complete).

Secondly, as a science project, we naturally decided to use English as the main language for all communication associated with the project. This means the main website for presenting the project, as well as support on the forum and the current documentation. However, as we can see in the Fig. 11.2, English native speakers only represent one third of the Poppy community. While English is usually not a problem for Germanic countries, we have been confronted with a lack of contribution from Latin countries on our forum. It is especially the case with French educational and artistic communities who have never contributed to the forum even after the successful experiments we have done (see chapter 9 and chapter 10).

Following the Arduino example, we now made our forum multilingual[4]see this topic `https://forum.poppy-project.org/t/multi-language-enabled-on-this-forum/304`) by allowing and creating the associated categories for French, German, Italian, Spanish and Portuguese. We are also currently translating the main website. This work has not yet produced results, but we missed the opportunity during the experiments we did, and we certainly have to wait until new experiments in education and art to see if we can actually draw more contribution from non-English speaking participants.

[3] pypot documentation: `poppy-project.github.io/pypot/`
[4] (

Third, creating a multidisciplinary community implies that participants do not speak the same language and do not have the same technical level. Therefore we will have to improve the accessibility of the project to create a motivational learning curve. This can be achieved thanks to a well-designed interface, playful content and gamification (Deterding et al. (2011) Groh (2012)).

Also we have to take into account the fact that people do not read documentation, described as the paradox of the active user in (Carroll and Rosson, 1987):

> Users never read manuals but start using the software immediately. They are motivated to get started and to get their immediate task done: they do not care about the system as such, and do not want to spend time up front on getting established, set up, or going through learning packages. The "paradox of the active user" is a paradox because users would save time in the long term by learning more about the system. But that is not how people behave in the real world, so we cannot allow engineers to build products for an idealized rational user when real humans are irrational. We must design for the way users actually behave.

Jeff Atwood also explained this behaviour in his post on the "just in time theory[5]" and addressed the problem in Discourse by putting a reminder when a user is going to do something that may be wrong, by summarizing relative topics or forum rules for example, just when a user is about to post something.

Understanding users' needs is very complicated, especially as Poppy users can be young students, artists or robotics experts. Therefore a major part of the community is not even aware of basic robotic problems such as motor orientations, the concept of communication buses, the fact motors can burn if too loaded. Thus in addition to complete documentation (for the good students), we will have to work on the "just in time" reminders so other people can easily assemble and use Poppy. We already explored some solutions for the assembly. For example, motors can be oriented with 16 potential configurations, we need to use one so everyone can share the same robot configuration code (see section 7.2.2). We put a directly visible indication on the 3D-printed parts so users can see how the motor has to be oriented (see Fig. 11.3).

Yet this work should be propagated on all elements. On electronics devices so users have important information (e.g. max voltage, ground/ VCC orientation) directly printed on the board, like Arduino did on their boards. On the software so users can

[5] `http://blog.codinghorror.com/the-just-in-time-theory/`

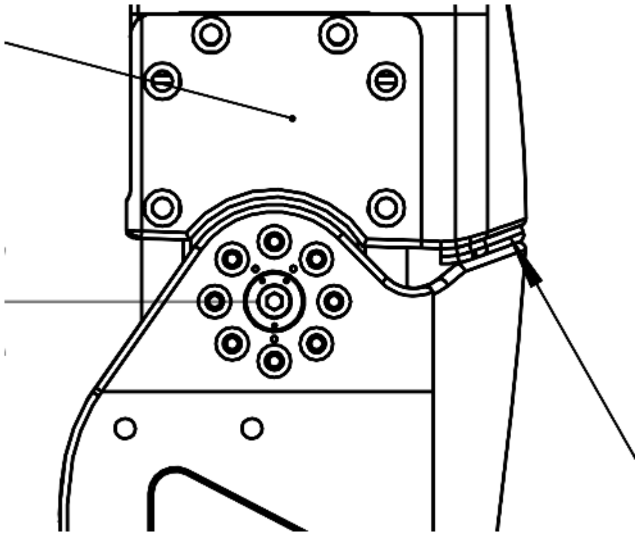


Fig. 11.3.: Image will be updated to be more explicit. Yet we can see the 3 points on the motor axis are also present on the part, it is only required to align these points to be sure that the motor is correctly assembled. It is kind of mechanical “just in time reminder".

be reminded that the primitive paradigms can create undesired behaviours[6] as well as being informed when the robot is suffering from too much load.

11.3 Create relevant educational content

The first experiments we did in education appeared promising (see chapter 9). We also received numerous requests from education structures in France as well as around the world. Indeed robotics is an atypical "science" intrinsically multi-disciplinary, merging engineering, computer science, biology, cognition and even humanities. Beyond teaching robotics, it provides a basis for creating original multi-disciplinary course. Also robots are very motivating tools as they can be incarnated in tangible objects.

However this impact cannot rely only on a technological platform, relevant content needs to be created for it to be a real education vector. For this reason, we are setting up a partnership with ENSAM[7] and Aérocampus[8] to create and evaluate educational content associated with the use of Poppy on the students from baccalaureate -3 to +3 i.e. high-school and bachelor level. All content will be produced and disseminated under Creative Commons and will be composed of: ready-to-use sheets of practical experiments with robots, specifying objectives and concrete progress in the classroom; their organization into a coherent and integrated curriculum; and

[6]The primitive paradigms: The Good, the Bad and the Ugly, see section7.4.2

[7]Arts et Métiers ParisTech is a French engineering and research graduate school in the fields of mechanics and industrialization.

[8]REF

tutorials in the form of videos or multimedia web pages. They may take the form of a MOOC that can be disseminated on the national platform FUN[9]. Also being open source, anyone can contribute to its improvement. One particular extension of this work could be evaluation for self-training through experiments in Fablab.

11.4 Improving hardware modularity

As we discussed in detail in this thesis, Poppy has a modular morphology. This modularity is expressed with all the technologies involved. For the mechanics, we use 3D printing techniques allowing producing quick and low-cost parts. For the electronics, we designed a board based on Arduino allowing to easily plug new sensors in. Finally for the software, we built a library using a modular architecture both for the low-level thanks to I/O controllers and for the high-level with primitive paradigms.

As we saw in the chapter 8, this modularity allows for quick experimentation with morphological variants. This functional modularity seems to be enough to allow a wide range of scientific experiments with Poppy and hopefully have a real scientific impact. However, to have an actual impact in the open source community, technological modularity is essential.

In software, modularity refers to the manner in which a design is decomposed into different "modules". It is based on the notion of interdependence within modules and independence between modules (Baldwin and Clark, 2000). This concept involves two related ideas: the need to allow work on a given module to be carried out without affecting other modules in the design, a concept known as "loose-coupling", and the need for well-designed "interfaces" between these modules (MacCormack et al., 2006).

The concept of modularity appears as a fundamental property in open source software collaboration. Indeed, code modularity allows the overall project to be divided into much smaller and well-defined tasks that individuals can tackle independently from other tasks and without affecting other aspects of the program (Narduzzo and Rossi, 2008).

Thus Linus Torvalds, emphasized modularity as a design criterion early in the development of Linux (DiBona and Ockman, 1999). Indeed without modularity, it would be improbable that contributors could understand the whole design architecture enough to make a relevant contribution. It would be difficult to add new features or fix bugs without affecting other parts of the design. Linux needed to be modular to

[9] REF

(a)Arduino form factor (b)Google Ara project

Fig. 11.4.: Examples of hardware modularity acheived so far.

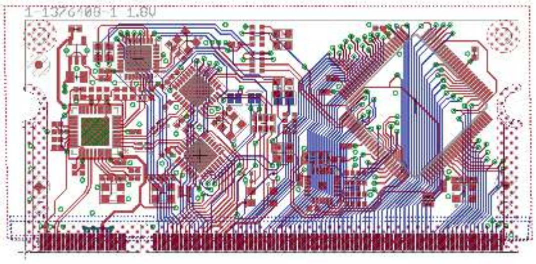
attract and facilitate a developer community. Code modularity allows partitioning of work among a global pool of developers and facilitates the recruitment of new contributors, as it reduces their learning curve to a subset of modules rather than the entire project (Fitzgerald, 2004).

Various efforts by corporations selling proprietary software products to develop additional products through an open source approach have been undertaken. One of the most visible of these efforts was Netscape's 1998 decision to make 'Mozilla', a portion of its browser source code, freely available. This effort encountered severe difficulties in its first year, only receiving approximately two dozen postings by outside developers. Much of the problems appeared to stem from the insufficiently modular nature of the software: reflecting its origins as a proprietary commercial product, the different portions of the program were highly interdependent and interwoven.

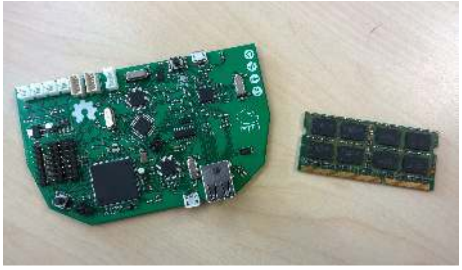
Over twenty years, open source software development has managed to find an efficient workflow. There are now tools, rules and guidelines allowing people to develop new software fluently together.

In hardware, modularity is not as developed as in software because it is not possible to abstract the interface. Hardware components have an overall shape, connector type and position, and so on, which makes it difficult to design an efficient interface.

Some projects have already addressed these challenges. For example, the Arduino boards (Uno, Leonardo, Yun, Mega, Due) share the same footprint (i.e. where the main I/O pins are placed), in this way any shield developed for one of these Arduino boards can also be used on other one. This modularity is a lever arm for the community as potential contributors know that the shield they develop and possibly sell will be compatible with several boards and future ones. Another example is the Google ARA project, it is a smart-phone with modular elements. It is possible to change only the processor or the camera without having to change the whole phone.



(a)IO Module schematics (under production)



(b)Size comparison between the IO board and SO-DIMM format

Fig. 11.5.: A novel IO module is under development. This IO module only include the core components for controlling robots such as Poppy. Its tiny size and standard interface make it easy to embed.

To achieve such a design, they created an interface block (see REF). Therefore any modules following the interface rules can be plugged on the phone.

To improve the potential impact of Poppy for technology we need to follow such ideas. Indeed the methodology used to build Poppy allows for modularity but the actual hardware design is interdependent. The functioning of each mechanical part depends also on elements connected to it, for example, the way the pelvis is designed changes the mobility of the legs. Also the particular shape of the IO board and the way connectors are placed, are designed to fit in the current design of Poppy's head. It does not prevent the reuse of such components in other projects but it reduces their relevance.

Thus it would be very useful for a robot platform such as Poppy, made to explore/hack/develop robotics, to easily switch between different technological solutions. On one hand we could test in minutes completely different morphologies (a new pair of legs for example). On the other hand, and it is maybe the most important point, it would permit the community to develop their own version of any of Poppy's mechatronics systems without having to reconsider the whole robot structure. By Poppy's systems, we mean legs, feet, arms, hands, torso or head.

Our future technological development will be more oriented toward hardware modularity. This work is already under process for the two main aspects of the robot.

We are also exploring splitting the electronics into modules. We are working on a new design of the IO board, which will involve 2 boards, one IO module with the core of the technology we need (Arduino, motor control) and a shield with connectors. While the shield is customized to one robot, depending on the needs and motors used, the IO module is very small and versatile so it can be integrated both in small and big robots. The design of this IO module has already been done (see Fig. 11.5) and is based on a DDR2 SO-DIMM format (67x30 mm) with up to 200 pins. This board involves an Arduino Mega with all its GPIO, two modified motor buses compatible for TTL and RS485 communication and an inertial unit MPU6000.

For the mechanics, a first step toward more modularity could be to split the robot into interchangeable sub-modules with defined interfaces. In this way, contributors could create new designs while only having to ensure that the interface is compatible. Therefore anyone could create a whole new head or legs system and distribute it so people could test it on their Poppy. We would like to suggest these sub-modules: Head (neck + head), Torso (abs, spine, chest), Arms (both from shoulder to hands) and Legs (both, include pelvis, legs and feet).

Thus it enables a large range of reconfigurations on Poppy and people are not limited to an evolution of the current design but can create completely new morphology while being compatible with Poppy's other modules. For example we could have several locomotive systems such as two legs, one jumping leg, wheels, 4 legs (centaur) or just a static base.

While the overview of the module system seems clear for us, the details are still rather muddled. Typical examples are the arms, where the functional distinction is not clear. In the current version of Poppy, the motor for one of the arms is in the torso. From a functional point of view, it should be integrated into the "arms" sub-modules, but on a practical level it means you cut out a huge part of the torso and will greatly complicate the interface design.

Another point is the pelvis/torso interface, on one hand, it would be simpler to consider the abs motor as the interface, on the other hand, we do not want to enforce having an abs motor for other submodules.

These kinds of submodules still raise a limitation because the interface has a fixed size, which will constrain the overall size of the robot elements. It should be possible to create a bigger robot, maybe up to 120cm high but it will be difficult to have a robot smaller than 60cm while keeping the same interface.

11.5 Production/Distribution: an alternative approach

Poppy includes three main parts: its mechatronic structure (skeleton and motors); its electronics; its software.

Reproducing and rebuilding the mechatronic structure is easy: the open-source skeleton can be printed on personal 3D printers (or using online services for higher

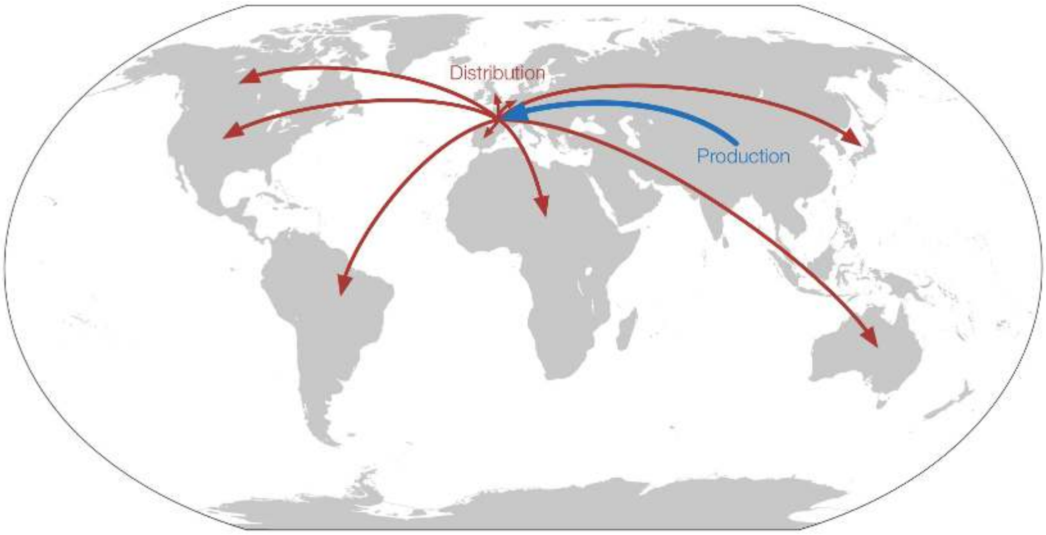


Fig. 11.6.: Classical approach for technological production and distribution

quality printing), and motors are bought off-the-shelf (motors are currently not open-source, but very standard). Obtaining and using the software is very easy: just download on the Poppy web site.

But manufacturing electronics is a bit more challenging. It is not yet possible to produce electronics components at home, and many institutional users do not have the skills or motivation to do so. There are some kick-starter projects on the way to facilitating the process, yet they are not ready and will not be ready for a couple of years. The current classical approach to building and distributing these electronics boards is to raise funding allowing for hundreds of boards to be manufactured, which can then be sold by a distribution company. Thanks to new online platforms such as CircuitHub it is easy to produce anything from a single model to thousands of boards. Yet the cost is exponentially decreasing and unlike 3D printing. But a French research institute like Inria is not a distributor, it cannot raise funding to "mass" produce electronic components before reselling them. It is not even legally allowed to do so.

Indeed, the mission of a research team at Inria is to do research, and find ways to apply and transfer the results of this research, but not directly to produce and sell a commercial product. If a commercial products emerge from our research, one way to take advantage of it is to create a start-up company which will set up a business plan around it, probably based on production in Asia and then worldwide distribution to research laboratories, universities and fablabs (see Fig. 11.6)

But Poppy is not designed to be a standard commercial product. While it might foster the creation of an economic ecosystem and jobs, its main purpose is to become an educational tool that remains open, as well as rather low cost and easily reproducible. If the goal had been to make it profitable, it would be necessary to sell it at a much higher price. The robot would not be as accessible at it needs to be to ensure the achievement of its scientific diffusion and educational missions... We would lose the intrinsic purpose of Poppy.

That being said, some users (e.g. artists) might want to obtain and use an already fully assembled Poppy robot and the sale of already prepared kits can be a lever arm for the diffusion of the platform in the research community. Also, one of the main purposes of Poppy is to be hacked and modified. While some people have all the tooling needed, others may find useful to have external support, even if they are charged for the service.

On one hand, the use of classic distributors is of course the most direct solution and such a process is already underway to permit the commercialization of Poppy kits by the end of this year. On the other hand, there are novel emerging actors who could add more sense to the distribution of Poppy.

11.5.1 Toward local open factories

Meanwhile, the "makers revolution" is gaining momentum (Anderson, 2012) and more and more Fablabs are created around the world. As one of the main missions of Poppy is to be a novel educational platform, Poppy could become a popular platform used, hacked, and transformed within the natural FabLab activities. But also, and this is the direction explored below, it would make sense that Poppy, as a whole or subsets of its components, be produced and distributed by Fablabs, and thus becoming a tool used by Fab Lab to develop and solidify the economic ecosystem in which they live.

An original and constructive organizational process would be to take advantage of the production phase for educational purposes. In this context, each fablab would have the possibility to produce, assemble and sell Poppy to local actors (see Figure 11.7), while the production phase could become a training resource for using 3D printing techniques and manufacturing electronic circuits, and later on the constructed platforms would be sold.

Also in a context where fablabs need to find an economic model, several sources of income may be found thanks to the distribution of platforms such as Poppy. The first and most obvious one is the sale to local actors of fully-assembled and functional Poppy robots produced by the Fablab. But a more advanced model could emerge. Poppy is a development robotic platform: it means that it can and will be broken,

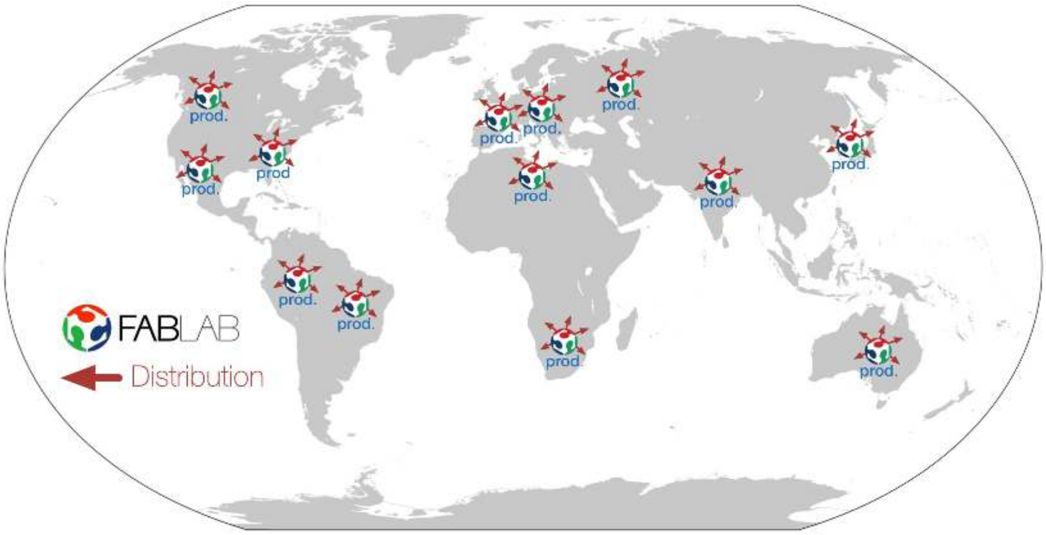


Fig. 11.7.: Local production and distribution by Fablabs

meaning that Fablabs may extend their commercial offers. Among them we can cite:

- Ensure technical support (repairs, upgrades, ...) and maybe sell maintenance contracts to labs/school/university and even other third party FabLabs.
- Provide a customization service to adapt Poppy to specific needs (e.g. a university or high school that would like to have a Poppy on wheels rather than legs)
- For an event or artist residency: the FabLab could rent a robot and provide a technician,
- Propose professional training for 3D printing to companies

From these kinds of interactions, links and collaboration between local actors and Fablabs may emerge, leading to other potentially funded projects.

11.5.2 Promote local collaboration

Beyond the act of production and sales, Poppy could become a pretext for promoting the linkage and exchange between local actors from multiple backgrounds. At the scale of a city or region, we can easily imagine a distribution of roles where several FabLabs could collaborate to build and distribute different parts of Poppy depending on their motivations, skills and equipment. Also, it helps to connect the fablabs with

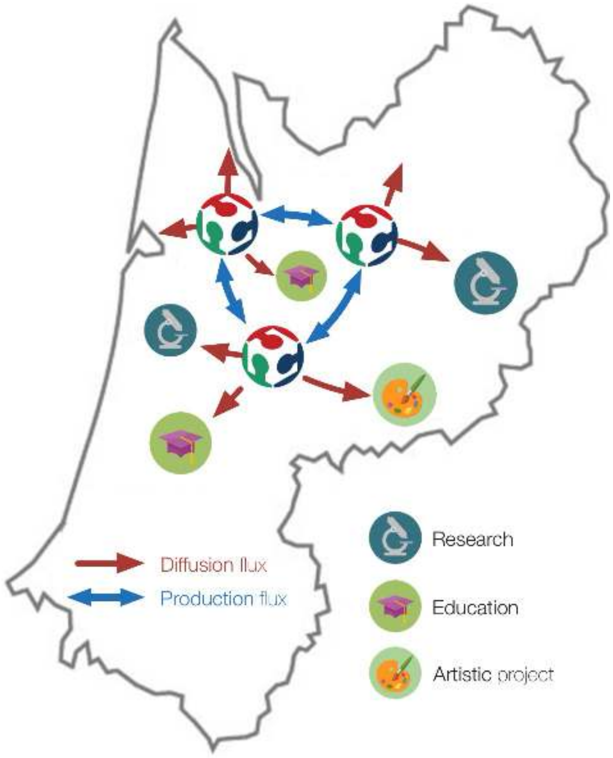


Fig. 11.8.: A synergy can emerge between Fablabs and local actors

local actors, public/private research labs, companies, schools/universities or artists (see Figure 11.8)

11.5.3 What is the role of the Flowers research team in such a process?

The Flowers research team's role remains essential. As the founders, designers and leaders of both the technological platform and its surrounding philosophy of openness and innovation, the Flowers team continues to improve the platform, take a central role in animating the community of users, and design new uses with scientists, educators, "geeks" and artists. Within this process, the Flowers team also coordinates the growth of the community of contributors and users, and designs strategies to ensure both the quality and sustainable development of the platform and its uses.

Among the tools used by the Flowers team to ensure such quality and sustainable development is the control of the "Poppy" brand, and through policies/charters:

- The "Poppy" brand is owned by Inria, and the use of the brand by third parties like FabLabs will only be possible through agreements ensuring that the Poppy project's policies and philosophy are implemented;

- Agreements take the form of charters/policies between Inria and FabLabs specifying guidelines to follow to ensure quality and protect the interests of each party (Inria, FabLab, users).

On the Inria side, the creation of a spin-off association whose role would be to oversee the technological development, community management and quality control, is under consideration.

Therefore, Poppy can be one of the first projects launching this new kind of production and distribution process. Over recent months we met several of the main French FabLabs. While they are quite enthusiastic about this idea, the organization is not completely ready to go in this way and we will certainly have to use both ways byrelying also on classical robots distributors.

11.6 Open source force control actuator

All the development we have done for Poppy has been released under open source licenses; the mechanical structure and electronics boards are under creative commons licenses and the pypot control library is distributed under GPL license. These various technological bricks allow for new experiments on the role of morphology to be explored, and foster a creative environment for students and artists. Yet a very important technology brick remains closed: robot actuation.

The Robotis Actuators are expensive (more than 60% of the cost of the robot), proprietary and no modification is permitted either to the low-level control or the hardware. Furthermore, these actuators are limited as they use a classical position control based on PID and do not permit force control.

The technical limits are firstly problematic for scientific reasons. Indeed, the actuator properties are elements that it must be possible to experimentally control. This is not possible with the Robotis ones, beyond tuning the PID controller values and estimating the maximum output torque. In addition, these actuators act as a black box, we do not have any way of guaranteeing or controlling how the tuning of some parameters actually changes the actuator behaviour. For example, parameters such as the delay and the internal loop frequency are very important when we explore reinforcement learning. Indeed, the algorithm needs to have a controlled synchronization between action and observation, and we were confronted with this issue while exploring biped learning with Poppy.

Secondly, the latest scientific work in the robotics field seems to show the efficiency of force control actuators over position control in creating robots able to act robustly in the real world (e.g. Boston Dynamics robots, MIT Cheetah). Hydraulic and pneumatic actuators are too complicated to be integrated in small and safe robots, but technology based on series elastic actuators[10] or impedance controls seem very promising.

Although there are numerous robots based on control impedance or SEA actuators, to our best knowledge, there is no project currently trying to create open source force-controlled actuators. Nevertheless, there are some open source projects to create open source servomotors called open servo and supermodified, yet they only address the electronic parts by offering open source electronic boards compatible with low cost RC servomotors.

Thus an important challenge we have already committed to is to create novel open source actuators allowing force control. Inria is supporting this 2-year project and two engineers have been hired for its development. The plan is to first create open source motors equivalent to the Robotis ones, then create a force control module. All the work, mechanics, electronics and software control will be released with open source licenses. In addition, special attention will be focused on creating 3D printable element so users can hack the actuator's mechanics.

On a scientific level, this actuator will permit free exploration of robot control with an open source control library, allowing to use classical PID control or even to implement a more exotic one.

Finally, we will try to keep as low a cost as possible to increase the potential impact in the robotics field. In particular, using these actuators, we hope to create either a cheaper version of Poppy or a more advanced one, implementing force control for a similar cost.

[10] The series elastic actuation (Pratt and Williamson, 1995) is a technology based on adding a spring between the motor output and the actuator output. The offset between the motor position and the actuator output is proportional to the external force applied (Hooke law). It is a simple, yet effective way of obtaining direct force measurements at the actuator output. Then a controller can drive the motors following the force feedback to create the desired output torque.

12 Conclusion

12.1 Contributions of this thesis

This thesis started with the desire to study mechanisms that allow open-ended learning and development in robots and humans. In particular, this thesis aimed to study how the morphological properties of the body could impact the acquisition of motor or social skills. We realized that if one wanted to really study the role of the body in cognition, one needed to be able to consider the body as an experimental variable: something that can be easily changed and experimented with.

Eventually we shifted from this first goal to focus our attention on more pragmatic ones. If we want to study the role of morphology in the real world, we need a real robotic platform whose morphology allows the exploration of morphological variants. Furthermore, a key aspect of Science is the reproducibility of results, so we need also to find appropriate methods to facilitate the evaluation of our work in other laboratories. Yet, this was impossible at the time because robot platforms were developed using classical machining techniques requiring a lot of time, resources, energy and funding. Also, classical machining techniques did not allow certain shapes to be built. In this thesis, we decided to take advantage of the 3D printing revolution by transposing it to humanoid robotics and this led us to design the Poppy platform.

All aspects of the platform were designed to be highly modular, modifiable, robust, and easily replicable in other labs for cumulative science. In just a few days, we can now systematically study how various shapes of the legs or feet influence balance in biped locomotion, or how various head morphologies will provoke different reactions when socially interacting with humans.

Poppy was firstly presented as an easily hackable platforms in the AMAM2013 conference (Lapeyre et al., 2013b) while its design was explained in more detail for IROS 2013 (Lapeyre et al., 2013d) as well as its potential relevance for exploring interaction (Lapeyre et al., 2013a). Then we conducted several experiments to demonstrate its unique properties. On one hand, by exploring the role of thigh morphology over bipedal dynamics, presented at Humanoids2013 („Poppy Humanoid Platform : Experimental Evaluation of the Role of a Bio-inspired Thigh Shape“). On the other hand, by demonstrating it is indeed possible to make morphology an experimental variable. Experiments have been done to test various feet designs and will be presented at Humanoids2014 (Lapeyre et al., 2014b).

Secondly, we used open source release, which drew particular attention, to start the creation of a multidisciplinary community. Thanks to the high level of interest we received, we found educational actors and artists eager to explore novel applications with robots. This work led us to conduct several instructive experiments opening perspectives for the use of Poppy for Art and Education; an associated paper was published for DI2014 (Lapeyre et al., 2014a).

Finally, the main contribution has certainly been the open source release of the first complete 3D-printed humanoid robot that can be freely used by anyone as an experimental platform. It compensates for the lack of open science tools available, so robotics researchers, even those working with robot morphology, can share their work with the scientific community. Moreover, Poppy offers an alternative to laboratories desiring to experiment in the real world. They are no longer constrained to either buying a closed and limited robot, or investing resources in the development of a new experimental platform, they can choose to use the work already done with Poppy and adapt it to their needs. Several research labs in Europe have already begun to use the Poppy platform for their own projects (e.g. Collège de France, Bristol Robotics Lab., Inria Nancy. . .).

12.2 Limits

Numerous limitations have already been discussed during this thesis. In this final conclusion, we would like to focus particular attention on the one associated with the initial motivations that led to the design of Poppy: namely creating an open source experimental robot for exploring the role of morphology.

Firstly, For us, it appears to be even more difficult than designing a whole humanoid robot because we are not in our field of expertise. Firstly, we did not have the time yet to explore real case of open science where two laboratories are evolved on the same scientific experiment to evaluate how fluent the reproducibility is. Also as we saw in the previous discussion chapter, the creation of a community is really challenging and they are still mediation needed to promote open collaboration in the research community.

Secondly, because we spend many time on the "Poppy open environment", the scientific contributions on the role of morphology were limited to preliminary results. We conducted experiments on the role of the thigh morphology that showed great improvements of a bio-inspired design over a classical straight thigh, yet these results are very limited to a specific case where the robot's balance is ensured by physical guidance. Also other labs already using Poppy did not yet publish results associated with the use of Poppy as main tool. Yet, Poppy was released open source one year ago, so it requires probably more time to be both adopted and used for research.

Finally, we designed Poppy to explore the role of morphology, especially in the scope of biped locomotion. The design of the platform has been directed toward new ways of achieving biped locomotion. Poppy has small feet, under-actuated legs, a multi-articulated torso and so on. While we are convinced these solutions are more interesting for research purposes, they are still less efficient than classical approaches using big feet and powerful actuators. Therefore currently, Poppy cannot walk by itself and it is clearly a limitation of the platform in particular for educational applications. To overcome this problem, we will use the modularity of Poppy to offer an alternative version of the leg design with a more traditional configuration, more suitable to achieve quickly a limited yet working biped locomotion.

12.3 What is Poppy ?

In this thesis, we suggest a novel approach to creating experimental robot platforms. To implement this methodology we created a set of tools based on open source environment and emergent technologies involving mechanics, electronics, software and a community (under construction). As a first instance, we used these tools to create a humanoid robot but actually, these tools could have been used for any kind of robot that has to move and act in the real world.

In this context, "Poppy" is more a meta-robot than an actual humanoid robot. The Poppy humanoid is an instance of this meta-robot, and thanks to the modularization of our hardware technology in progress, it will be more and more clear than we can easily reshape and reconfigure Poppy into any kind of robot, with various, sizes, number of DoF, limbs and so on. The community tools we are organizing will allow a multidisciplinary community of robotics enthusiasts to create and share new creatures based on the same technological bricks. Of course the challenges of creating a multidisciplinary community and hardware modularity presented in section 11.2 and section 11.6 will have to be addressed. In this way, the open source actuators we will develop will represent an essential technological brick to foster the creation of these creatures. In addition, special attention will be focused on the educational impact of such creative environment with the construction of open source educational content and ready-to-use projects.

Appendices

A Theoritical model of the human thigh expected impact on biped locomotion

If we look closely at the morphology of the human femur, it appears that it is inclined by an angle of 6°. This makes the feet closer to the projection of the center of gravity (CoG) (see Fig. A.1a) and enhances stability in two main ways during the walking gait:

- As the feet are closer to the center of gravity, the lateral translation of the CoG necessary to transfer the mass of the robot from one foot to another is reduced (see Fig A.1a). In the case of Poppy's morphology, thanks to the 6^o bended thigh, the lateral motion of the CoG is reduced by about 30% ($5cm$ instead of $7.1cm$).
- During the stance phase, the CoG initial conditions are slightly modified. As we will explain with a simple theoretical model in the next section, the bended thigh can reduce the falling rate.

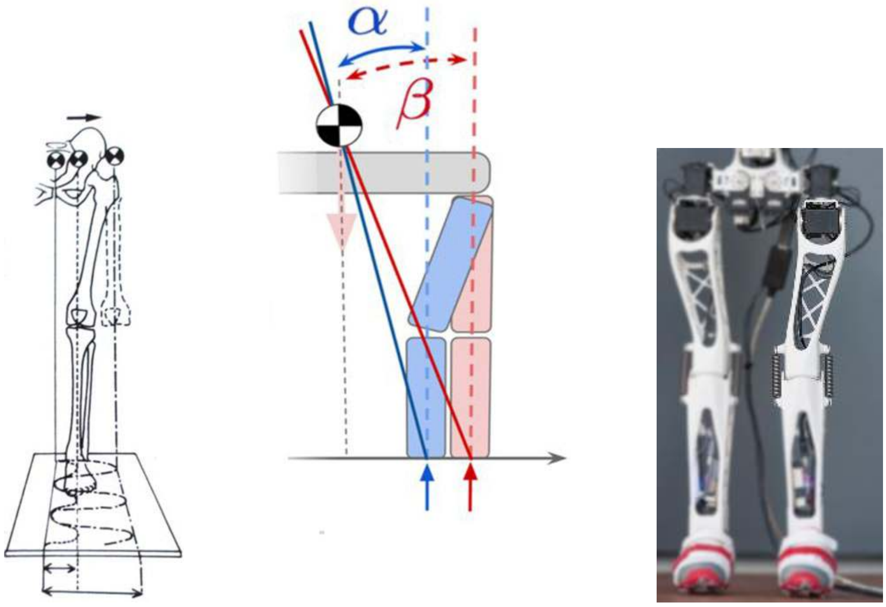
A.1 Theoretical model

We can model the situation where the robot is on one foot by an inverted pendulum with a mass point centered on the center of gravity (CoG) of the robot and the axis of rotation located at the foot position (see Fig. A.1c). With such a model, the dynamic of the whole structure depends on:

- the length l of the segment extending from the foot to the center of gravity,
- the angle θ of the segment relative to the vertical axis,
- the force of gravity g.

And the system follows this physical law:

$$\ddot{\theta}(t) + w_0 \cdot sin(\theta(t)) = 0 \tag{A.1}$$



(a) (b) (c)

Fig. A.1.: a) Effect of bended human femur on human bipedal locomotion. b) Model used for the comparison of the two thighs morphology. c) Implementation of the bended thigh on the poppy platform

with:

$$w_0 = \sqrt{\frac{g}{l}} \tag{A.2}$$

A.2 Intuitive expectation

To get an initial insight into this behavior, we can linearize the system for small disturbances such as:

$$\theta(t) = \theta_0 \cdot cos(w_0 \cdot t) \tag{A.3}$$

and

$$\dot{\theta}(t) = -\theta_0 \cdot w_0 \cdot sin(w_0 \cdot t) \tag{A.4}$$

One can see that the position and velocity of the pendulum varies linearly with the initial condition i.e θ angle. Therefore, reducing this initial angle θ_0 involves a direct reduction of the falling speed $\dot{\theta}(t)$ of the robot.

In the case of Poppy's geometry, the thigh bending allows a 40% reduction of the initial angle θ_0 ($\alpha = 3.8^o$ against $\beta = 6.4^o$ on Fig. A.1b).

A.3 Simulation

In the case of a fall, it is not possible to assume small perturbations, that is the reason why we have simulated the model in Matlab with a non-linear system and obtain the behavior represented in Fig. A.2.

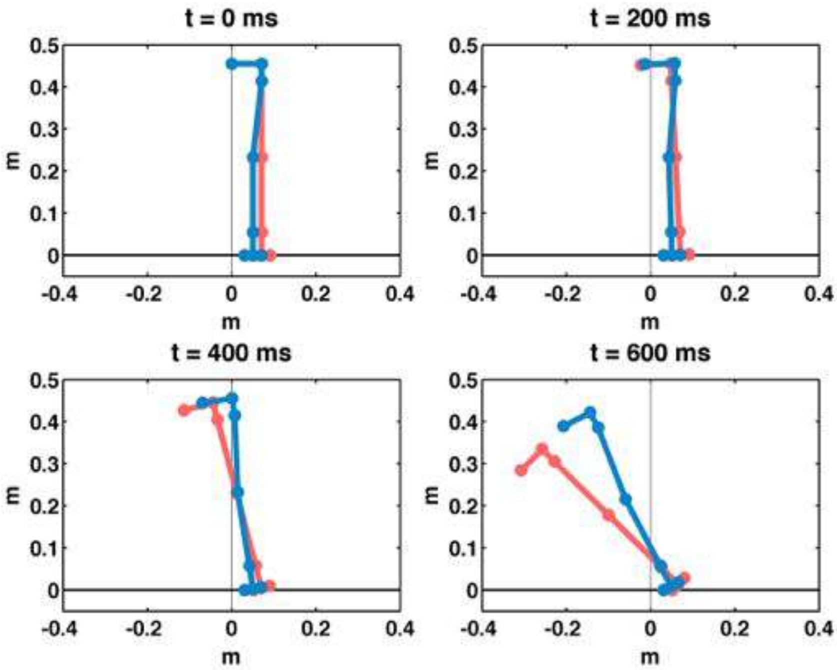


Fig. A.2.: Comparison of the falling dynamic over time when Poppy is standing on one foot, depending on its thigh morphology: with a bended thigh of 6° (blue) and with a straight thigh (red).

If we define the center of gravity altitude as:

$$z_{CoG} = l \cdot cos(\theta(t)) \tag{A.5}$$

We can express its falling speed over time as:

$$\dot{z}_{CoG} = -\dot{\theta}(t) \cdot l \cdot sin(\theta(t)) \tag{A.6}$$

In this condition, if we consider the first 700 ms of the system behavior simulation and compare the two systems, the mean of the CoG falling speed is reduced by around 56% in the bended thigh case.

Semi-passive knee design

B

B.1 Introduction

We performed a parametric optimization both on the position of the spring ties (M_T and M_L) and on its characteristic (K, L_0, D_i, F_{max}, L_{max}) (see Fig. B.1) to try to match the mentioned criterion above. These criterion are modelled as condition on the resultant torque:

- $C(\theta = 0) < -0.4$: Locking of the knee, where $0.4N.m$ is the necessary torque to keep the leg straight.
- $C(\theta = 25^o) = 0$: Transition between the two behaviors
- $C > 0$ if $\theta > 25deg$: Helps the motor to lift the leg.
- $max(|C(\theta)|) < \frac{C_{MX-28}}{2}$: The actuator $MX-28$ should always be powerful enough to control the joint motion.

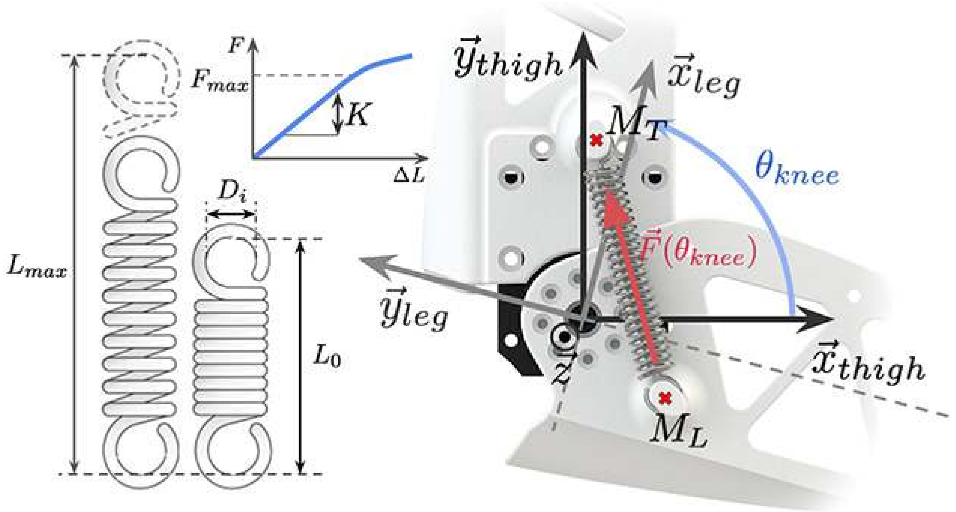


Fig. B.1.: Spring parameters to optimize

The resultant torque C generated by springs in function of the knee flexion θ ($n_{spring} = 2$) is computed as follow:

$$C(\theta) = n_{spring} \cdot \overrightarrow{OM_L}|_{R_{thigh}} \wedge \overrightarrow{F}(\theta) \cdot \overrightarrow{z} \tag{B.1}$$

with:

$$\|F(\theta)\| = K \cdot (L(\theta) - L_0) \tag{B.2}$$

$$L(\theta) = \left\| \overrightarrow{M_T M_L}|_{R_{thigh}} \right\| \tag{B.3}$$

$$\overrightarrow{OM_L}|_{R_{thigh}} = \overrightarrow{OM_L}|_{R_{leg}} \cdot \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.4}$$

We use an iterative selection on these criteria to determine the appropriate characteristics for the spring.

B.2 Minimizing stresses on the structure

The length of the lever arm is constrained by the dimensions of the legs, resulting in an increase of the force generated by the spring to produce the desired torque on the knee.

By maximizing the following criterion with the constraint $C(\theta = 25^o) = 0$:

$$c_1 = \frac{C_{max}}{F_{max}^2} \tag{B.5}$$

We were able to determine the ties' specific location (M_T and M_L), for both minimizing mechanical stress and changing the torque direction for $\theta = 25^o$,

$$M_T = \{2, 39, 0\}_{Rthigh} \qquad M_L = \{-12, 23, 0\}_{Rleg} \tag{B.6}$$

and constraints concerning the springs characteristics:

$$L_{min} < 42.6mm \qquad L_{max} > 65.12mm \tag{B.7}$$

B.3 Ties strength

We calculated the minimum diameter of the ties required to withstand the constraints imposed by the spring with a beam theory model:

$$D_{min} = \sqrt[3]{\frac{32 \times C_s \times F_{max} \times l_{tie}}{2\pi \times \sigma_{MaxPolyamide}}} \tag{B.8}$$

By considering Poppy's parameters and a coefficient of safety $C_s = 5$, we have found that the spring must respect the criterion $D_{min} > 6.5mm$.

B.4 Obtained behavior

Considering the desired spring behavior and geometrical conditions, an automatic selection out of 720 different springs[1] was performed. Only 5 springs satisfied all criteria. For the Poppy platform we chose a spring with the following characteristics: $\{D_i = 9.6mm,\ L_0 = 42mm,\ K = 1620N.m^{-1},\ F_{max} = 81.7N,\ L_{max} = 72.8mm\}$ inducing a resultant behavior shown in Fig. B.2. As we can see, even if the torque applied by the spring is quite low ($C_{max} = 0.74N.m$), the force subjected to spring ties is up to $40N$. The shape of the ties has been optimized using FEA (Finite Element Analysis) in order to handle the stress.

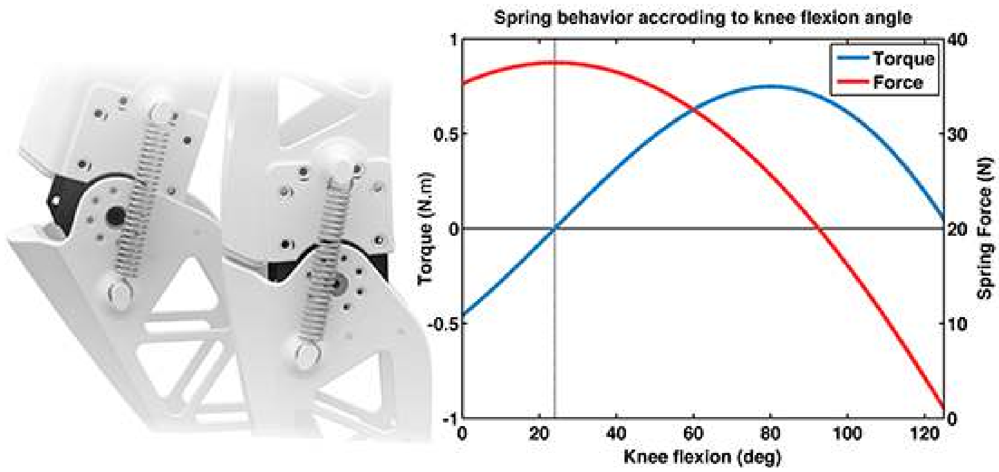


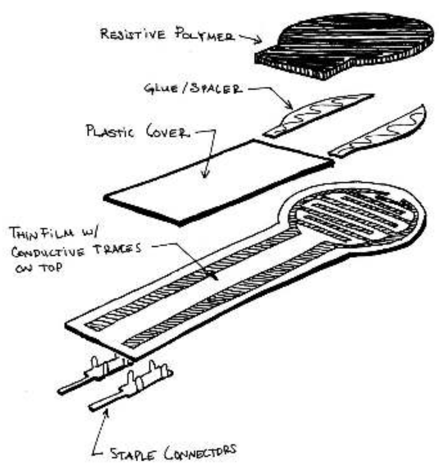
Fig. B.2.: Theoretical semi-passive mechanism behavior. The blue line corresponds to the torque applied by the springs on the leg according to the flexion angle of the knee. The red line corresponds to the force that the springs applied on ties.

[1] pre-selection of springs in the vanel.com catalogue

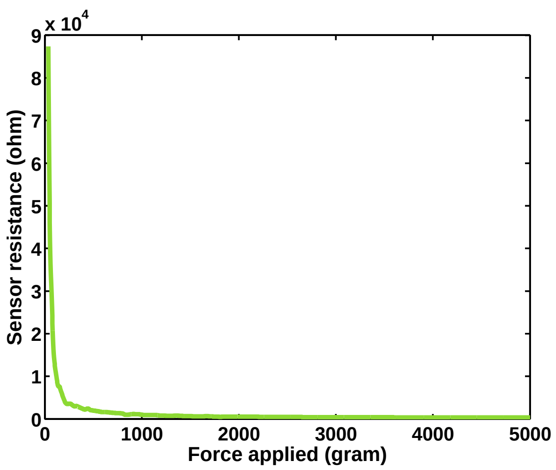
C Design of foot sensors pressure acquisition

C.1 Principles

To obtain measurement of the pressure variation under our Poppy's feet we used FSR sensors from Interlink Electronics (see Fig. C.1a). The FSR sensor will vary its resistance depending on how much pressure is being applied to the sensing area. The harder the force, the lower the resistance is. These sensors are low-cost -6$ each- yet theirs behaviors are very non-linear (see Fig. C.3) and the calibration is quite variable depending on the production batch and the thermal conditions. So we cannot expect having precise results.



(a)Exploded view[1] of a FSR sensor



(b)Measured FSR sensor resistance depending on the applied force.

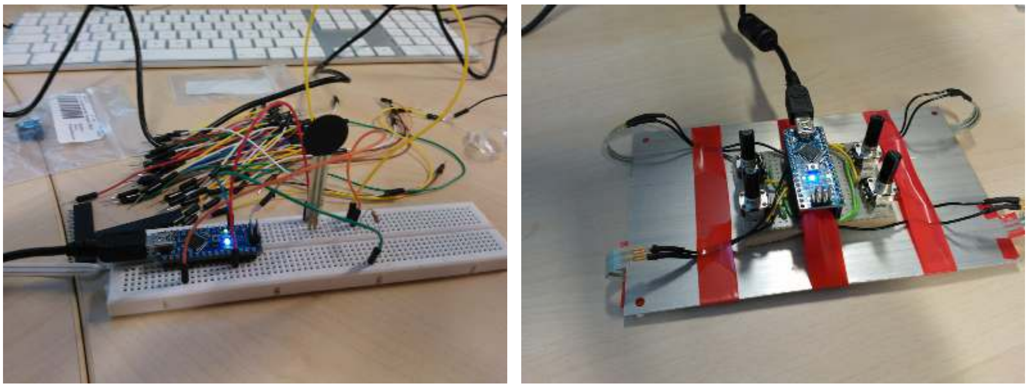
Fig. C.1.: The FSR force sensors are cheap but they have a really non-linear behavior and are not very precise.

The acquisition of the resistance can be done indirectly by designing a voltage-divider[2] and using the FSR resistance variations to make the voltage output varies

[2]A voltage divider is a linear circuit that produces an output voltage V_{out} that is a fraction of its input voltage V_{in}. It often consists of 2 resistors in series.

(see Equation C.1). This voltage (V_{out}) can be then measured by an analog input of an Arduino board (see Fig. C.2a).

$$V_{\text{out}} = \frac{R}{R + R_{\text{FSR}}} \cdot V_{\text{in}} \tag{C.1}$$



(a) A FSR sensor connected with Arduino nano board.

(b) Simple testing assembly with 4 voltage dividers with FSR sensors and potentiometers plugged on an Arduino nano board

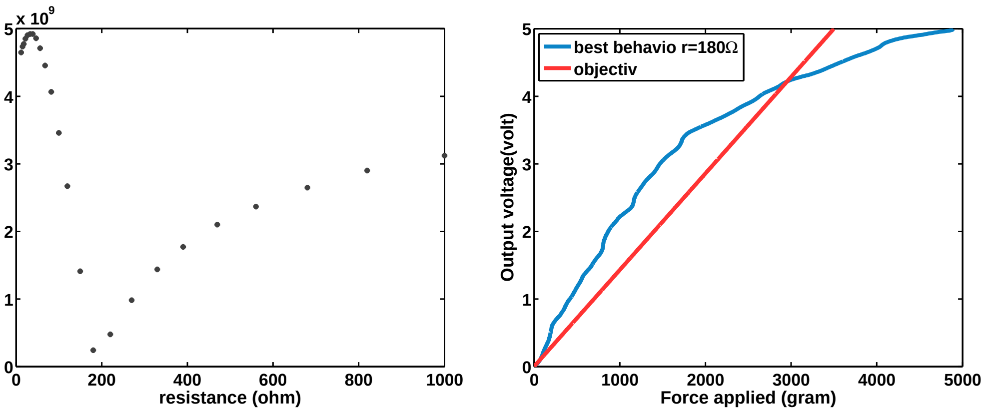
Fig. C.2.: We can easily measure the resistance variation of a FSR sensor using a voltage divider with the V_{out} connected to an Arduino analog port.

C.2 Design of the voltage divider to reduce the sensor non-linearity

A well tuned voltage-divider can help to reduce the non-linearity of the FSR sensors. Thus we conducted an optimization on the constant resistor choice depending on:

- the Arduino analog precision: 1024 values for a $5V$ input range,
- the use of the Dynamixel tension as voltage input i.e. $14V$,
- the standard resistor E12 precision series,

With an objective function sets to minimize the difference between the actual voltage-divider behavior and the perfect linear behavior $V_{\text{out}}(F) = \alpha \cdot F$ with $V_{\text{out}}(3.5kg) = 5V$ (see red curve on Fig. C.3b), we obtained the choice of a 180Ω resistor for the constant resistance of the voltage divider (see Fig. C.3a). The best behavior found is plotted in blue on the Fig. C.3b.



(a) Optimization of the voltage divider constant resistor: distance between the output behavior with respect to a linear behavior.

(b) Theoretical behavior (blue) of the output voltage with respect to the applied force with a $14v$ input and a 180Ω compared with the objective linear behavior.

Fig. C.3.: The design of the voltage divider for each FSR sensor is done by optimizing the output behavior toward a ideal linear behavior.

Bibliography

Adamczyk, Peter G, Collins, Steven H, and Kuo, Arthur D (2006). „The advantages of a rolling foot in human walking.“ *The Journal of experimental biology*, 209(Pt 20), pp. 3953–63 (cited on p. 130).

Alain, Cottereau (1997). „The fate of collective manufactures in the industrial world. The silk industries of Lyons and London, 1780-1850“. *World of Possibilities. Flexibility and Mass Production in Western Industrialization*, pp. 75–152 (cited on p. 40).

Alfayad, Samer, Ouezdou, Fethi B, Namoun, Faycal, and Gheng, Gordon (2011). „High performance integrated electro-hydraulic actuator for robotics–Part I: Principle, prototype design and first experiments“. *Sensors and Actuators A: Physical*, 169(1), pp. 115–123 (cited on p. 20).

Anders, Christoph, Wagner, Heiko, Puta, Christian, et al. (2007). „Trunk muscle activation patterns during walking at different speeds“. *Journal of Electromyography and Kinesiology*, 17(2), pp. 245–252 (cited on p. 87).

Anderson, Chris (2012). „Makers: the new industrial revolution“ (cited on pp. 2, 33, 63, 157, 187, 201).

Anderson, S.O., Wisse, M., Atkeson, C.G., et al. (2005). „Powered bipeds based on passive dynamic principles“. *5th IEEE-RAS International Conference on Humanoid Robots, 2005.* Pp. 110–116 (cited on pp. 17, 130).

Asada, Minoru, MacDorman, Karl F, Ishiguro, Hiroshi, and Kuniyoshi, Yasuo (2001). „Cognitive developmental robotics as a new paradigm for the design of humanoid robots“. *Robotics and Autonomous Systems*, 37(2), pp. 185–193 (cited on p. 1).

Atkeson, Christopher G, Hale, Joshua G, Pollick, Frank E, et al. (2000). „Using humanoid robots to study human behavior“. *IEEE Intelligent Systems and their applications*, 15(4), pp. 46–56 (cited on p. 1).

Baldwin, Carliss Young and Clark, Kim B (2000). *Design rules: The power of modularity*. Vol. 1. MIT press (cited on p. 196).

Banzi, Massimo (2009). *Getting Started with arduino*. " O’Reilly Media, Inc." (cited on p. 170).

Barragán, Hernando (2004). „Wiring: Prototyping physical interaction design“. *Interaction Design Institute, Ivrea, Italy*, (June) (cited on p. 44).

Bogin, Barry and Varela-Silva, Maria Inês (2010). „Leg length, body proportion, and health: a review with a note on beauty“. *International journal of environmental research and public health*, 7(3), pp. 1047–1075 (cited on p. 80).

Breazeal, Cynthia (2003). „Emotion and sociable humanoid robots“. *International Journal of Human-Computer Studies*, 59(1), pp. 119–155 (cited on p. 1).

Breazeal, Cynthia and Scassellati, Brian (2002). „Robots that imitate humans“. *Trends in cognitive sciences*, 6(11), pp. 481–487 (cited on p. 1).

Brinkman, William F, Haggan, Douglas E, and Troutman, William W (1997). „A history of the invention of the transistor and where it will lead us“. *Solid-State Circuits, IEEE Journal of*, 32(12), pp. 1858–1865 (cited on p. 11).

Brooks, Rodney A (1986). „Achieving artificial intelligence through building robots“ (cited on pp. 1, 13, 51).

– (1991). „Intelligence without representation“. *Artificial intelligence*, 47(1), pp. 139–159 (cited on pp. 2, 14, 51).

– (1995). „Intelligence without reason“. *The artificial life route to artificial intelligence: Building embodied, situated agents*, pp. 25–81 (cited on p. 14).

– (1996). „Prospects for human level intelligence for humanoid robots“. In: *Proceedings of the First International Symposium on Humanoid Robots (HURO-96)*, pp. 17–24 (cited on p. 14).

Brooks, Rodney A and Stein, Lynn Andrea (1994). „Building brains for bodies“. *Autonomous Robots*, 1(1), pp. 7–25 (cited on p. 1).

Brooks, Rodney A, Breazeal, Cynthia, Marjanović, Matthew, Scassellati, Brian, and Williamson, Matthew M (1999). „The Cog project: Building a humanoid robot“. In: *Computation for metaphors, analogy, and agents*. Springer, pp. 52–87 (cited on p. 1).

Bruneau, Olivier, Ouezdou, Fethi Ben, and Fontaine, J (2001). „Dynamic walk of a bipedal robot having flexible feet“. In: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 512–517 (cited on p. 81).

Carroll, John M and Rosson, Mary Beth (1987). *Paradox of the active user.* The MIT Press (cited on p. 194).

Ceccato, J C (2009). „Le tronc, de la locomotion à la commande“. PhD thesis. Montpellier II (cited on p. 87).

Cheng, Gordon, Hyon, Sang-Ho, Morimoto, Jun, et al. (2007). „CB: A humanoid research platform for exploring neuroscience“. *Advanced Robotics*, 21(10), pp. 1097–1114 (cited on p. 1).

Chestnutt, Joel, Lau, Manfred, Cheung, German, et al. (2005). „Footstep planning for the Honda ASIMO humanoid“. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, pp. 629–634 (cited on p. 1).

Collins, SH Steven Hartley and Ruina, Andy (2005). „A bipedal walking robot with efficient and human-like gait“. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, pp. 1983–1988 (cited on pp. 1, 17, 18, 130).

Collins, Steven H, Wisse, Martijn, and Ruina, Andy (2001). „A three-dimensional passive-dynamic walking robot with two legs and knees“. *The International Journal of Robotics Research*, 20(7), pp. 607–615 (cited on p. 26).

Davis, S and Caldwell, D G (2010). „The design of an anthropomorphic dexterous humanoid foot“. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2200–2205 (cited on p. 130).

Deterding, Sebastian, Dixon, Dan, Khaled, Rilla, and Nacke, Lennart (2011). „From game design elements to gamefulness: defining gamification“. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. ACM, pp. 9–15 (cited on p. 194).

DiBona, Chris and Ockman, Sam (1999). *Open sources: Voices from the open source revolution*. " O'Reilly Media, Inc." (cited on p. 196).

DiSessa, Andrea A (2001). *Changing minds: Computers, learning, and literacy*. Mit Press (cited on p. 157).

Dufour, Michel and Pillu, M (2005). *Biomécanique fonctionnelle: Membres-Tête-Tronc*. Paris: Masson (cited on pp. 79, 80).

Feipel, V, De Mesmaeker, T, Klein, P, and Rooze, M (2001). „Three-dimensional kinematics of the lumbar spine during treadmill walking at different speeds“. *European Spine Journal*, 10(1), pp. 16–22 (cited on p. 87).

Fitzgerald, Brian (2004). „A critical look at open source“. *Computer*, 37(7), pp. 92–94 (cited on p. 197).

Fodor, Jerry A (1987). *Psychosemantics: The problem of meaning in the philosophy of mind.* The MIT Press (cited on p. 12).

Garcia, M, Chatterjee, a, Ruina, a, and Coleman, M (1998). „The simplest walking model: stability, complexity, and scaling.“ *Journal of biomechanical engineering*, 120(2), pp. 281–8 (cited on p. 17).

Geyer, Hartmut, Seyfarth, Andre, and Blickhan, Reinhard (2006). „Compliant leg behaviour explains basic dynamics of walking and running.“ *Proceedings. Biological sciences / The Royal Society*, 273(1603), pp. 2861–7 (cited on p. 20).

Gouaillier, D, Hugel, V, Blazevic, P, et al. (2008). „The nao humanoid: a combination of performance and affordability“. *CoRR, vol. abs/0807.3223* (cited on pp. 3, 29).

Groh, Fabian (2012). „Gamification: State of the art definition and utilization“. *Institute of Media Informatics Ulm University*, 39 (cited on p. 194).

Ha, I, Tamura, Y, Asama, H, Han, J, and Hong, D W (2011). „Development of open humanoid platform DARwIn-OP“. In: *SICE Annual Conference (SICE), 2011 Proceedings of*. IEEE, pp. 2178–2181 (cited on pp. 3, 29, 59).

Haugeland, John (1989). *Artificial intelligence: The very idea*. MIT press (cited on p. 12).

Hild, Manfred, Siedel, Torsten, Benckendorff, Christian, Thiele, Christian, and Spranger, Michael (2012). „Myon, a new humanoid“. In: *Language Grounding in Robots*. Springer, pp. 25–44 (cited on p. 59).

Hirai, Kazuo, Hirose, Masato, Haikawa, Yuji, and Takenaka, Toru (1998). „The development of Honda humanoid robot“. In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 2. IEEE, pp. 1321–1326 (cited on p. 1).

Hobbelen, D., Boer, T. de, and Wisse, M. (2008). „System overview of bipedal robots Flame and TUlip: Tailor-made for Limit Cycle Walking“. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2486–2491 (cited on pp. 18, 26, 130).

Hobbelen, Daan G E DGE and Wisse, Martijn (2005). „Ankle joints and flat feet in dynamic walking“. In: *Climbing and Walking Robots*. Springer, pp. 787–800 (cited on p. 130).

Hughes, J, Clark, P, and Klenerman, L (1990). „The importance of the toes in walking.“ *The Journal of bone and joint surgery. British volume*, 72(2), pp. 245–51 (cited on pp. 81, 130).

Hyon, Sang-Ho and Mita, Tsutomu (2002). „Development of a biologically inspired hopping robot-" Kenken"“. In: *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*. Vol. 4. IEEE, pp. 3984–3991 (cited on p. 81).

Iida, Fumiya, Rummel, Jürgen, and Seyfarth, André (2007). „Bipedal walking and running with compliant legs“. In: *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, pp. 3970–3975 (cited on p. 20).

Ijspeert, A J, Crespi, A, Ryczko, D, and Cabelguen, J M (2007). „From swimming to walking with a salamander robot driven by a spinal cord model“. *Science*, 315(5817), p. 1416 (cited on pp. 21, 87).

Jansen, Theo (2007). *Theo Jansen: The Great Pretender* (cited on p. 18).

Jones, Rhys, Haufe, Patrick, Sells, Edward, et al. (2011). „RepRap–the replicating rapid prototyper“. *Robotica*, 29(01), pp. 177–191 (cited on p. 46).

Jorgensen, Morten Winkler, Ostergaard, Esben Hallundbk, and Lund, Henrik Hautop (2004). „Modular ATRON: Modules for a self-reconfigurable robot“. In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 2. IEEE, pp. 2068–2073 (cited on p. 27).

Kaneko, Kenji, Harada, Kensuke, Kanehiro, Fumio, Miyamori, Gou, and Akachi, Kazuhiko (2008). „Humanoid robot HRP-3“. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, pp. 2471–2478 (cited on p. 1).

Kaneko, Kenji, Kanehiro, Fumio, Morisawa, Mitsuharu, et al. (2009). „Cybernetic human HRP-4C“. In: *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*. IEEE, pp. 7–14 (cited on p. 130).

Kaplan, Frédéric and Oudeyer, Pierre-Yves (2008). „Le corps comme variable exp{é}rimentale“. *Revue philosophique de la France et de l'{é}tranger*, (3), pp. 287–298 (cited on pp. 2, 3, 12, 51).

Kaufmann, K and McCorduck, P (1979). *Machines who think* (cited on p. 12).

Lakoff, George and Núñez, Rafael E (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being*. Basic books (cited on p. 13).

Lamoth, Claudine J C, Meijer, Onno G, Daffertshofer, Andreas, Wuisman, Paul I J M, and Beek, Peter J (2006). „Effects of chronic low back pain on trunk coordination and back muscle activity during walking: changes in motor control“. *European Spine Journal*, 15(1), pp. 23–40 (cited on p. 87).

Lapeyre, Matthieu, Rouanet, Pierre, and Oudeyer, Pierre-yves. „Poppy Humanoid Platform : Experimental Evaluation of the Role of a Bio-inspired Thigh Shape“ (cited on p. 207).

Lapeyre, Matthieu, Rouanet, Pierre, and Oudeyer, Pierre-Yves (2013a). „Physical Human-like Interaction with the Acroban and Poppy Humanoid Platforms.“ In: *IROS Workshop*. Tokyo, Japon (cited on p. 207).

– (2013b). „Poppy: a New Bio-Inspired Humanoid Robot Platform for Biped Locomotion and Physical Human-Robot Interaction“. In: *Proceedings of the 6th International Symposium on Adaptive Motion in Animals and Machines (AMAM)*. Darmstadt, Allemagne (cited on p. 207).

Lapeyre, Matthieu, Rouanet, Pierre, Oudeyer, Pierre-Yves, and Others (2013c). „Poppy Humanoid Platform: Experimental Evaluation of the Role of a Bio-inspired Thigh Shape“. In: *Humanoids 2013* (cited on p. 135).

Lapeyre, Matthieu, Rouanet, Pierre, and Oudeyer, Pierre-Yves (2013d). „The Poppy Humanoid Robot: Leg Design for Biped Locomotion“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo, Japon (cited on p. 207).

Lapeyre, Matthieu, Rouanet, Pierre, Grizou, Jonathan, et al. (2014a). „Poppy Project: Open-Source Fabrication of 3D Printed Humanoid Robot for Science, Education and Art“. In: *Digital Intelligence 2014*. Nantes, France (cited on p. 208).

Lapeyre, Matthieu, N'Gyuen, Steve, Le Falher, Alexandre, and Oudeyer, Pierre-Yves (2014b). „Rapid morphological exploration with the Poppy humanoid platform“. In: *Humanoids 2014*. Madrid, Espana (cited on p. 208).

Larsen, Jørgen Christian, Brandt, David, and Stoy, Kasper (2012). „Locokit: A robot construction kit for studying and developing functional morphologies“. In: *From Animals to Animats 12*. Springer, pp. 12–22 (cited on p. 27).

Ly, Olivier, Lapeyre, Matthieu, and Oudeyer, Pierre-yves (2011). „Bio-inspired vertebral column, compliance and semi-passive dynamics in a lightweight humanoid robot“. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1465–1472 (cited on pp. 3, 20, 59, 65, 67, 88).

MacCormack, Alan, Rusnak, John, and Baldwin, Carliss Y (2006). „Exploring the structure of complex software designs: An empirical study of open source and proprietary code“. *Management Science*, 52(7), pp. 1015–1030 (cited on p. 196).

Mackness, Jenny, Mak, Sui, and Williams, Roy (2010). „The ideals and reality of participating in a MOOC“. In: *Networked Learing Conference*. University of Lancaster, pp. 266–275 (cited on p. 156).

Maris, Marinus and Boeckhorst, René (1996). „Exploiting physical constraints: heap formation through behavioral error in a group of robots“. In: *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*. Vol. 3. IEEE, pp. 1655–1660 (cited on p. 57).

Marques, Hugo Gravato, Jantsch, Michael, Wittmeier, Steffen, et al. (2010). „ECCE1: the first of a series of anthropomimetic musculoskeletal upper torsos“. In: *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, pp. 391–396 (cited on p. 25).

McAuley, Alexander, Stewart, Bonnie, Siemens, George, and Cormier, Dave (2010). *The MOOC model for digital practice* (cited on p. 156).

McCarthy, John (1978). „History of LISP“. In: *History of programming languages I*. ACM, pp. 173–185 (cited on p. 12).

McGeer, Tad (1990). „Passive dynamic walking“. *The International Journal of Robotics Research*, 9(2), pp. 62–82 (cited on pp. 16, 26, 130).

– (1992). „Principles of walking and running“. *Advances in comparative and environmental physiology*, 11, pp. 113–139 (cited on p. 16).

Mellis, D, Banzi, Massimo, Cuartielles, David, and Igoe, Tom (2007). „Arduino: An open electronic prototyping platform“. In: *Proc. CHI*. Vol. 2007 (cited on pp. 45, 157).

Metta, Giorgio, Sandini, Giulio, Vernon, David, Natale, Lorenzo, and Nori, Francesco (2008). „The iCub humanoid robot: an open platform for research in embodied cognition“. In: *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, pp. 50–56 (cited on pp. 3, 28, 29).

Mizuuchi, Ikuo, Waita, Hironori, Nakanishi, Yuto, et al. (2004). „Design and implementation of reinforceable muscle humanoid“. In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 828–833 (cited on p. 25).

Mizuuchi, Ikuo, Nakanishi, Yuto, Sodeyama, Yoshinao, et al. (2007). „An advanced musculoskeletal humanoid kojiro“. In: *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, pp. 294–299 (cited on p. 25).

Moeckel, R, Pouya, S, Maufroy, C, et al. (2013). „Locomorph robot system - towards legged modular robots with model-inspired morphology“. In: *Proceedings of the 6th International Symposium on Adaptive Motion in Animals and Machines (AMAM)* (cited on p. 27).

Mondada, Francesco, Bonani, Michael, Raemy, Xavier, et al. (2009). „The e-puck, a robot designed for education in engineering“. In: *Proceedings of the 9th conference on autonomous robot systems and competitions*. Vol. 1. 1, pp. 59–65 (cited on p. 157).

Moulin-Frier, Clément and Oudeyer, Pierre-Yves (2013). „Exploration strategies in developmental robotics: a unified probabilistic framework“. In: *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*. IEEE, pp. 1–6 (cited on p. 67).

Murata, Satoshi and Kurokawa, Haruhisa (2007). „Self-reconfigurable robots“. *Robotics & Automation Magazine, IEEE*, 14(1), pp. 71–78 (cited on p. 27).

Murata, Satoshi, Yoshida, Eiichi, Kamimura, Akiya, et al. (2002). „M-TRAN: Self-reconfigurable modular robotic system“. *Mechatronics, IEEE/ASME Transactions on*, 7(4), pp. 431–441 (cited on p. 27).

Nakanishi, Yuto, Ohta, Shigeki, Shirai, Takuma, et al. (2013). „Design Approach of Biologically-Inspired Musculoskeletal Humanoids“. *International Journal of Advanced Robotic Systems*, 10(216), p. 1 (cited on pp. 3, 54, 130).

Narduzzo, Alessandro and Rossi, Alessandro (2008). *Modularity in action: GNU/Linux and free/open source software development model unleashed*. Tech. rep. Department of Computer and Management Sciences, University of Trento, Italy (cited on p. 196).

Natale, Lorenzo, Nori, Francesco, Metta, Giorgio, et al. (2013). „The iCub platform: a tool for studying intrinsically motivated learning“. In: *Intrinsically motivated learning in natural and artificial systems*. Springer, pp. 433–458 (cited on p. 28).

Nester, C J, Linden, M L van der, and Bowker, P (2003). „Effect of foot orthoses on the kinematics and kinetics of normal walking gait.“ *Gait & posture*, 17(2), pp. 180–7 (cited on p. 136).

Niiyama, Ryuma, Nishikawa, Satoshi, and Kuniyoshi, Yasuo (2010). „Athlete robot with applied human muscle activation patterns for bipedal running“. In: *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, pp. 498–503 (cited on p. 3).

Oudeyer, P-Y (2010). „On the impact of robotics in behavioral and cognitive sciences: from insect navigation to human cognitive development“. *Autonomous Mental Development, IEEE Transactions on*, 2(1), pp. 2–16 (cited on p. 1).

Oudeyer, Pierre-yves, Kaplan, Frederic, and Hafner, V (2007). „Intrinsic Motivation Systems for Autonomous Mental Development“. *IEEE Transactions on Evolutionary Computation*, 11(2), pp. 265–286 (cited on p. 67).

Oudeyer, Pierre-Yves, Ly, Olivier, and Rouanet, Pierre (2011). „Exploring robust, intuitive and emergent physical human-robot interaction with the humanoid robot Acroban“. *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 120–127 (cited on p. 20).

Oztop, Erhan, Franklin, David W, Chaminade, Thierry, and Cheng, Gordon (2005). „Human–humanoid interaction: is a humanoid robot perceived as a human?“ *International Journal of Humanoid Robotics*, 2(04), pp. 537–559 (cited on p. 1).

Palincsar, A (1998). *Social constructivist perspectives on teaching and learning* (cited on p. 157).

Papert, Seymour, Harel, Idit, and Papert, By Seymour (1991). „Situating constructionism“. *Constructionism*, 36, pp. 1–11 (cited on p. 157).

Pappano, Laura (2012). „The Year of the MOOC“. *The New York Times*, 2(12), p. 2012 (cited on p. 156).

Park, Jong Hyeon (2001). „Impedance control for biped robot locomotion“. *Robotics and Automation, IEEE Transactions on*, 17(6), pp. 870–882 (cited on p. 20).

Paul, Chandana (2006). „Morphological computation: A basis for the analysis of morphology and control requirements“. *Robotics and Autonomous Systems*, 54(8), pp. 619–630 (cited on pp. 3, 15, 51).

Paul, Chandana, Pfeifer, Rolf, Lund, Henrik, and Seyfarth, Andre (2004). „Investigation of Morphology and Control in Biped Locomotion“. PhD thesis (cited on p. 15).

Peeling, Nic and Satchell, Julian (2001). „Analysis of the impact of open source software“. *QinetiQ Ltd. QINETIQ/KI/SEB/CR010223. Available at http://www. govtalk. gov. uk/interoperability/egif_document. asp* (cited on p. 33).

Pfeifer, Rolf and Bongard, JC Josh C (2006). *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press (cited on pp. 2, 13, 18, 51, 123).

Pfeifer, Rolf and Iida, Fumiya (2005). „Morphological computation: Connecting body, brain and environment“. *Japanese Scientific Monthly*, 58(2), pp. 48–54 (cited on pp. 2, 15).

Pfeifer, Rolf and Scheier, Christian (2001). *Understanding intelligence*. MIT press (cited on pp. 13, 15, 51).

Pfeifer, Rolf, Iida, Fumiya, and Bongard, Josh (2005). „New robotics: Design principles for intelligent systems“. *Artificial Life*, 11(1-2), pp. 99–120 (cited on p. 21).

Pfeifer, Rolf, Lungarella, Max, and Iida, Fumiya (2007). „Self-organization, embodiment, and biologically inspired robotics“. *Science*, 318(5853), pp. 1088–1093 (cited on pp. 21, 23).

Plester, Beverly, Wood, Clare, and Bell, Victoria (2008). „Txt msg n school literacy: does texting and knowledge of text abbreviations adversely affect children's literacy attainment?“ *Literacy*, 42(3), pp. 137–144 (cited on p. 155).

Pratt, Gill A and Williamson, Matthew M (1995). „Series elastic actuators“. In: *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 399–406 (cited on pp. 20, 205).

Prince, F, Winter, D A, Stergiou, P, and Walt, S E (1994). „Anticipatory control of upper body balance during human locomotion“. *Gait & Posture*, 2(1), pp. 19–25 (cited on p. 87).

Radkhah, Katayon, Maufroy, Christophe, Maus, Moritz, et al. (2011). „Concept and design of the biobiped1 robot for human-like walking and running“. *International Journal of Humanoid Robotics*, 8(03), pp. 439–458 (cited on p. 3).

Raibert, Marc, Blankespoor, Kevin, Nelson, Gabriel, Playter, Rob, and Others (2008). „Bigdog, the rough-terrain quadruped robot“. In: *Proceedings of the 17th World Congress*, pp. 10823–10825 (cited on p. 20).

Resnick, Mitchel (2008). „Sowing the Seeds for a More Creative Society.“ *Learning & Leading with Technology*, 35(4), pp. 18–22 (cited on pp. 157, 166, 167).

Resnick, Mitchel, Maloney, John, Monroy-Hernández, Andrés, et al. (2009). „Scratch: programming for all“. *Communications of the ACM*, 52(11), pp. 60–67 (cited on pp. 156, 157, 162).

Riedo, Fanny, Philippe, R, Bergeron, Luc, et al. (2012). „A two years informal learning experience using the thymio robot“. In: *Advances in Autonomous Mini Robots*. Springer, pp. 37–48 (cited on p. 157).

Riedo, Fanny, Chevalier, Morgane, Magnenat, Stephane, and Mondada, Francesco (2013). „Thymio II, a robot that grows wiser with children“. In: *Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on*. IEEE. Ieee, pp. 187–193 (cited on p. 157).

Rouanet, Pierre (2012). „Apprendre {à} un robot {à} reconna{î}tre des objets visuels nouveaux et {à} les associer {à} des mots nouveaux: le r{ô}le de l'interface“. PhD thesis. Universit{é} Sciences et Technologies-Bordeaux I (cited on p. 65).

Rouanet, Pierre, Oudeyer, P-Y, and Filliat, David (2009). „An integrated system for teaching new visually grounded words to a robot for non-expert users using a mobile device“. In: *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*. IEEE, pp. 391–398 (cited on p. 65).

Salemi, Behnam, Moll, Mark, and Shen, Wei-Min (2006). „SUPERBOT: A deployable, multifunctional, and modular self-reconfigurable robotic system“. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, pp. 3636–3641 (cited on p. 27).

Saranli, Uluc, Buehler, Martin, and Koditschek, Daniel E (2001). „Rhex: A simple and highly mobile hexapod robot“. *The International Journal of Robotics Research*, 20(7), pp. 616–631 (cited on p. 20).

Scarfogliero, Umberto, Stefanini, Cesare, and Dario, Paolo (2009). „The use of compliant joints and elastic energy storage in bio-inspired legged robots“. *Mechanism and Machine Theory*, 44(3), pp. 580–590 (cited on p. 21).

Schwarz, Max, Schreiber, Michael, Schueller, S, Missura, Marcell, and Behnke, Sven (2012). „NimbRo-OP Humanoid TeenSize Open Platform“. In: *Proceedings of 7th Workshop on Humanoid Soccer Robots*. IEEE-RAS International Conference on Humanoid Robots (cited on pp. 3, 29, 59).

Sellaouti, Ramzi, Stasse, Olivier, Kajita, Shuuji, Yokoi, Kazuhito, and Kheddar, Abderrahmane (2006). „Faster and Smoother Walking of Humanoid HRP-2 with Passive Toe Joints“. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4909–4914 (cited on p. 81).

Severini, Gino (1917). „Le Machinisme et l'art“. *La Peinture d'avant-garde, Le Mercure de France*, 1, p. 216 (cited on p. 176).

Sèze, Mathieu de, Falgairolle, Mélanie, Viel, Sébastien, Assaiante, Christine, and Cazalets, Jean-René (2008). „Sequential activation of axial muscles during different forms of rhythmic behavior in man“. *Experimental brain research*, 185(2), pp. 237–247 (cited on p. 87).

Siegwart, Roland, Shin, Jiwon, and Magnenat, Stéphane (2014). „Visual Programming Language for Thymio II Robot“ (cited on p. 170).

Snow, Charles Percy (2012). *The two cultures*. Cambridge University Press (cited on p. 178).

Sproewitz, Alexander, Billard, Aude, Dillenbourg, Pierre, and Ijspeert, Auke Jan (2009). „Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture“. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, pp. 4259–4264 (cited on p. 27).

Steels, Luc (1990). „Towards a Theory of Emergent Functionality“. In: *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*. Cambridge, MA, USA: MIT Press, pp. 451–461 (cited on pp. 2, 18, 23, 67, 123).

Steels, Luc and Brooks, Rodney Allen (1995). *The artificial life route to artificial intelligence: Building embodied, situated agents*. Psychology Press (cited on pp. 2, 13, 51).

Steingrube, Silke, Timme, Marc, Wörgötter, Florentin, and Manoonpong, Poramate (2010). „Self-organized adaptation of a simple neural circuit enables complex robot behaviour“. *Nature Physics*, 6(3), pp. 224–230. arXiv: `arXiv:1105.1386v1` (cited on p. 21).

Stiefelhagen, Rainer, Fugen, C, Gieselmann, R, et al. (2004). „Natural human-robot interaction using speech, head pose and gestures“. In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 3. IEEE, pp. 2422–2427 (cited on p. 1).

Tapus, Adriana, Mataric, Maja J, and Scassellati, Brian (2007). „Socially assistive robotics“. *IEEE Robotics and Automation Magazine*, 14(1), p. 35 (cited on p. 1).

Tedrake, Russ, Zhang, Teresa Weirui, and Seung, H Sebastian (2005). „Learning to walk in 20 minutes“. In: *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems* (cited on p. 26).

Thompson, Clive (2011). „How Khan Academy is changing the rules of education“. *Wired Magazine*, 126 (cited on p. 156).

Thorstensson, A, CARLSON, H, ZOMLEFER, M R, and Nilsson, J (1982). „Lumbar back muscle activity in relation to trunk movements during locomotion in man“. *Acta Physiologica Scandinavica*, 116(1), pp. 13–20 (cited on p. 85).

Ude, Aleš, Atkeson, Christopher G, and Riley, Marcia (2004). „Programming full-body movements for humanoid robots by observation“. *Robotics and autonomous systems*, 47(2), pp. 93–108 (cited on p. 1).

Walter, W Grey (1950). „An imitation of Life“ (cited on p. 11).

Waterman, Norman A and Dickens, Philip (1994). „Rapid product development in the USA, Europe and Japan“. *World Class Design to Manufacture*, 1(3), pp. 27–36 (cited on p. 34).

Wisse, Martijn (2005). „Three additions to passive dynamic walking: actuation, an upper body, and 3D stability“. *International Journal of Humanoid Robotics*, 2(04), pp. 459–478 (cited on pp. 18, 24, 26, 54, 130).

Wisse, Martijn, Feliksdal, Guillaume, Van Frankkenhuyzen, J, and Moyer, Brian (2007). „Passive-based walking robot“. *Robotics & Automation Magazine, IEEE*, 14(2), pp. 52–62 (cited on pp. 3, 15, 130).

Yamaguchi, Jin'ichi, Soga, Eiji, Inoue, Sadatoshi, and Takanishi, Atsuo (1999). „Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking“. In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. Vol. 1. IEEE, pp. 368–374 (cited on p. 1).

Zykov, Victor, Chan, Andrew, and Lipson, Hod (2007). „Molecubes: An open-source modular robotics kit“. In: *IROS-2007 Self-Reconfigurable Robotics Workshop* (cited on p. 27).