ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

BIOLOGICALLY INSPIRED
ROBOTICS GROUP (BIRG)

Master Project Report

Design and simulation of locomotion of self-organising modular robots for adaptive furniture

Rafael Arco Arredondo

July 2006

Swiss Federal Institute of Technology Lausanne

Biologically Inspired Robotics Group

Supervisor: Prof. Auke Jan Ijspeert

Abstract

The Biologically Inspired Robotics Group (BIRG) at the Swiss Federal Institute of Technology Lausanne (EPFL) recently started a project on Modular Robotics for the future Learning Centre of the EPFL: *Roombots, Modular Robotics for adaptive and self-organising furniture*. It intends to design modular robots for furniture with ability for locomotion, self-assembling, self-organisation, self-reconfiguration and self-repairing.

This report covers the first stage of the project: the design of the external structure of the first prototypes of the modules as well as the simulation of locomotion of some multi-unit robots.

Acknowledgements

I knew very few things when I arrived in Lausanne in October. I did not know the city, the university, or even the project I would do. Everything started around June 2005, when from Granada I entered the web page of the laboratory and loved the research the group was doing. I did not hesitate and decided that I would do my Master Project at BIRG, coming to the EPFL with an Erasmus scholarship. I contacted Professor Ijspeert and he agreed to let me do it, but we did not decide the exact topic of the project, we set it off until I was at the EPFL.

We had a meeting a few days after I got settled and he talked to me about *Roombots*, a new project the laboratory was planning to start soon. After listening to the description of the project, I immediately decided that I would work on it for the year. It had many ingredients that were really attractive for me: Robotics, Artificial Intelligence, Agent Theory, combination of several disciplines, innovation, the beauty of building complex structures from very simple ones, utility... A few days ago I got to know the proposal of the project received an award from Microsoft Research in December 2005, which did not surprise me at all since the idea of the project was brilliant and the research it involved was interesting not only for the project itself, but also for many other developments.

By the beginning of December, Professor Ijspeert, Sébastien and I decided that Sébastien would study reconfiguration as the goal for his Semester Project while I would develop the control of locomotion. And here is the result of this particular pregnancy that lasted until now, the end of July 2006.

During these nine months I have learned a lot from people in the laboratory, including many things that I would have never learned in my university. I would like to thank everyone for all the help you have kindly given to this student who was completely lost the first day he crossed the door of the lab. Especially, one of most helpful people was Yvan, with whom I had a great deal of discussions about Webots and the course of Biomodels. Of course, all my thanks also go for Professor Ijspeert, who, in addition to giving me the possibility of doing this wonderful project, had innumerable meetings with me to check the progress of what I was doing as well as gave me many advices, explanations, clues, ideas, etc. and pointed me in the right direction always.

But this year was not only a new academic experience. Erasmus is much more, it is about discovering a new country, a new culture, meeting people from all Europe or, as a friend of mine said the other day, is about opening your mind and your heart. I never expected I would make here so good friends as I did: Mar, Pablo, Álvaro, Mercè, Juan, Javi, Alejandro, Ana, María, Ana, Anna, Andrea, Thomas, Jamal, Ervin, Daniel... I will feel a huge sadness in a few days when I will leave you. *Os llevaré siempre en el corazón.*

Last but absolutely not least are my family and friends in Spain. Your support from the distance during all the year was vital, both in the good times and in the bad ones. I am so happy of meeting you again shortly.

Lausanne, 28th July 2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modularity, since adopted for Computer Science by Edsger Dijkstra in the late 60s [Dij68], has become one of the key concepts of this field. It provides many desirable properties such as organisation, simplicity, flexibility or re-usability for a computing system (hardware or software). In Robotics, these ideas have also been applied. The objective of Modular Robotics is to design and build fairly complex robots from identical pieces (or very few kinds of them) that have few degrees of freedom (DOF) and simple controllers. Proper combination and coordination (emergent properties) of the modules can make this possible.

The Biologically Inspired Robotics Group (BIRG), headed by Professor Auke Jan Ijspeert, has carried out already quite a lot of research on this topic (see chapter 2) and it recently started another project: *Roombots,*[1] *Modular Robotics for adaptive and self-organising furniture*. The aim of this research is to develop simple, robust, autonomous building blocks (called *modules*) which, with an appropriate combination and coordination, can form pieces of furniture (e.g. a stool, a chair, a table, a sofa...) and can perform tasks such as locomotion, self-organisation, self-assembling, self-reconfiguration or self-repairing. A possible scenario could be, for instance, eight modules come together and assemble becoming a stool, then some more modules are added and they transform into a chair and finally the chair walks towards the place where the user is and she/he sits on it. The *roombots* should have a friendly and complete user interface, for example to set the possible configurations of the robots (stool, table, etc.) or to define the position of some robots in the room in order to create a particular "decoration". This last point is being addressed by the Learning Algorithms and Systems Laboratory (LASA) at the EPFL, headed by Professor Aude Billard.

1.1 Objective

Roombots is a large, very ambitious project. It requires research on many fields: modular design, emergent properties, autonomous control, self-organisation, adaptation, control of locomotion, reconfiguration, mechanical design, human-machine interaction... This master project only focuses on the earliest stage. In particular, on the structural design of the prototypes of the modules as well as the necessary mechanisms to provide the multi-unit robots with the ability to perform locomotion. Locomotion was only explored in simulation, because the prototypes have not been realised yet.

[1] From "roomware robots"

1.2 Background

1.2.1 Modular Robotics

Modular Robotics is the branch of Robotics that tries to develop complex robots made of several simple parts, although its foundations are in the field of Artificial Life, in particular in Von Neumann's cellular automata [Neu66]. The characteristics of Modular Robotics are mainly the following ones:

Flexibility: Despite being composed by parts with very little functionality, modular robots offer the possibility of combining these elements to perform very different tasks: locomotion, auto-organisation, auto-repairing... When combined, several modules can produce a great variety of structures and behaviours, like it happens with Lego blocks. This is known as emergent properties, complex patterns originated by simple agents when they act as a collective: the whole is greater than the sum of the parts.

Adaptability: Thanks to the ability to self-organise, a modular robot can adapt to many kinds of situations and be used for many different purposes, even for those ones the designer maybe did not consider.

Reliability: The fact of being composed by autonomous units makes a modular robot more reliable. When one of the parts fails, the rest will be still working (normally with less efficacy, but the whole system will not collapse). Besides, the broken module can be substituted by another one if available. Ideally, the robot should try, if possible, to adapt to the new situation, learning how to cope with the problem to minimise the damage.

Low cost: All the modules in a modular robot are generally identical, or at least the differences are very small. This means the design and realisation is relatively cheap (in comparison with the cost an heterogeneous robot would involve) because only one or very few types modules have to be considered.

One of the first works on Modular Robotics was carried out by Karl Sims [Sim94]. It is a study in simulation on how walking, swimming and jumping abilities can be incorporated to virtual *creatures* made of several modules (boxes of different dimensions). These creatures can be seen as modular robots, because each one has its own controller and sensory-motor system, and their structure and controller is evolved with a genetic algorithm in order to optimise the desired gait. The genotype of the creatures and the corresponding phenotype, as well as some of the resulting optimised structures are showed in Figure 1.1.

1.2.2 Central pattern generators

Locomotion is one of the main properties that characterises animals, making them different from the other living creatures. People have always wondered what the structures and processes responsible of locomotion are. However, despite the apparent simplicity of it (because it is a common and natural activity for human beings), no precise answer was provided until a few decades ago.

Previously, around the end of the 19th century, there were mainly two currents which tried to explain the mechanisms of locomotion: one said the rhythms that produced the patterns of locomotion were due to chains of reflexes, that is, involving the peripheral system with one reflex entraining another [She10], the other claimed that rhythms were caused by a central mechanism [GB11]. The evidences from the first explanation came from the observation that sensory stimulation could initiate a rhythmic sequence of steps. The second one was based

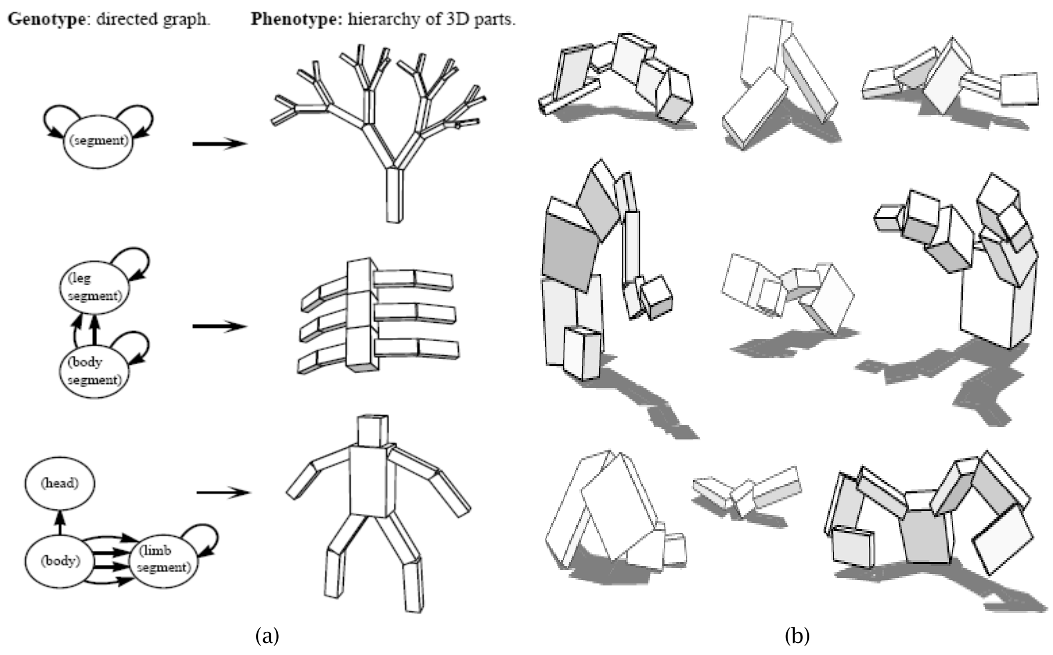


Figure 1.1: Karl Sims' creatures representation (genotype and corresponding phenotype (a)), and evolution (b) (from [Sim94])

on the observations that locomotion could take place despite lesions of the dorsal roots of the spinal cord.

In the seventies Sten Grillner and his colleagues concluded that locomotion was possible thanks to a combination of both mechanisms: central networks that generate essential features of the motor pattern and sensory feedback signals that are used to modulate locomotion depending on what happens in each step, for instance changes in the shape of the terrain [Gri85, Gri96]. By doing research on decerebrate cats, Grillner discovered that the spinal cord itself contains neural circuits that, when activated, can be made to coordinate the different muscles to produce locomotive movements. These circuits are so called central pattern generators (CPGs).

Grillner's research with cats was followed with the study of the central nervous system of the lamprey. He was able to isolate the spinal cord of the lamprey from the rest of the nervous system and proved that rhythmic patterns were generated if the spinal cord was properly stimulated. Figure 1.2 illustrates the structure of the spinal cord of the lamprey.

Since then, the concept of central pattern generator has been largely used in Robotics, in particular for the control of locomotion of multi-legged robots. Inspiration from biology provides very simple control mechanisms and allows to synchronise the movement of the different legs as well as to set the phase difference of one leg with respect to another. CPGs also permit to adapt the gait of the robot to the external environment by changing few input signals.

Basically, the approach to implement a CPG on a robot is the following one:

1. A model of the CPG is designed. Normally the building block for it is the representation of a neuron (implemented, for instance, as a leaky-integrator) or a group of neurons (e.g. a non-linear oscillator such as a Hopf oscillator). The building blocks are then connected

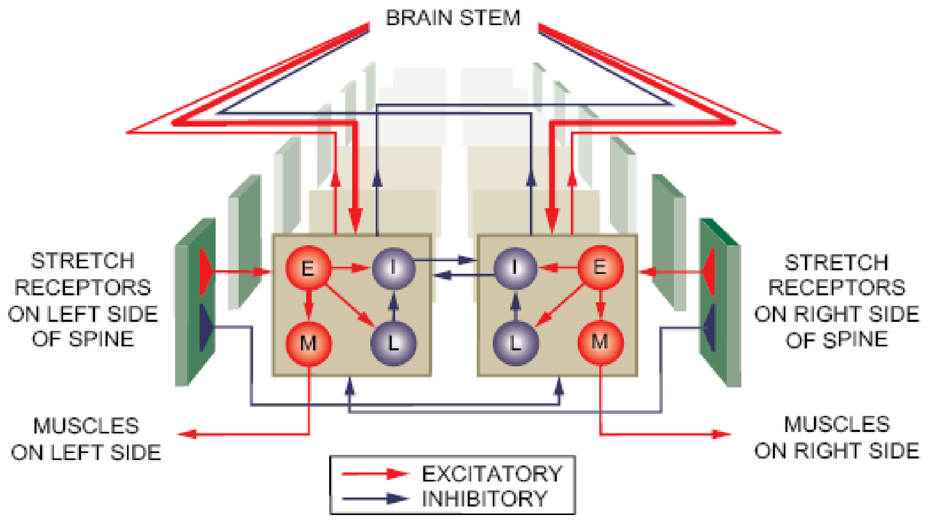

BRAIN STEM
STRETCH RECEPTORS ON LEFT SIDE OF SPINE
STRETCH RECEPTORS ON RIGHT SIDE OF SPINE
E
I
M
L
MUSCLES ON LEFT SIDE
MUSCLES ON RIGHT SIDE
EXCITATORY
INHIBITORY

Figure 1.2: A representation of the spinal cord of the lamprey (from [Gri96])

in such a way that they are mutually entrained and produce the necessary oscillations to reproduce the rhythmic behaviour of a real CPG.

2. The output of the CPG is used as input for the actuators of the robot (for example, the target position of a servomotor), so the trajectory of the actuators follows the CPG.

3. (Optional) Sensory information is taken as input for the CPG, so the CPG is modulated with external variables that can depend as well on the output of the CPG (that is, the control loop is closed with this feedback).

Successful cases of CPG-controlled robots are, among many others, Ijspeert's model of the salamander [Ijs01, ICC05], Fukuoka's quadruped robot [FKC03] or Conradt's worm robot [CV03].

1.3 Tools used

The design of structure of the modules, as well as their simulation, was done with Cyberbotics' Webots [Mic04, Web]. Webots is a rapid prototyping environment for modeling, programming and simulating mobile robots. It also allows to transfer the controllers of the virtual robots to some actual robots.

Webots provide quite an accurate simulation of real physics by using Open Dynamics Engine (ODE) [ODE] and offering the user an interface to directly call the ODE functions in order to modify or extend the features Webots directly provides for simulating physics. Customisation of the physics with ODE was used to attach several robots, which is a feature not implemented in Webots so far.

The controllers of the robots and the utilities to automatically generate worlds and controllers were programmed in C and C++ respectively and compiled with the GCC compiler. For the integration of the equations of the CPGs, the implementation of the Runge-Kutta-Fehlberg method provided by the GNU Scientific Library [GDT$^+$03] was used.

The simplex online optimisation method was implemented with the description in *Numerical Recipes in C* [PTVF92].

1.4 Structure of the report

This report is structured as follows. First, an overview of the most recent projects and developments on Modular Robotics and control of locomotion is done in chapter 2. Then, the design of the prototypes of the modules that will be used as building block for the furniture, as well as the creation of some multi-unit structures, is tackled in chapter 3. Control of locomotion is explored in chapter 4 in several ways: with a sine-based controller, with a CPG-based controller, adjusting the amplitude and frequency of the different gaits which are obtained, controlling the direction of the robots, adapting the locomotion gait to irregular surfaces and finally performing online optimisation of locomotion. The thesis ends in chapter 5 commenting the conclusions obtained during the project and proposing further research that would be useful to improve the knowledge of the topics this project deals with or is related to.

Chapter 2

State of the Art

This chapter introduces some of the newest developments in the fields involved with this project: self-organisation and control of locomotion in Modular Robotics. This research deals with several topics, problems and solutions which are useful for the *Roombots* project and, in consequence, some of their ideas have been adopted for it.

2.1 Self-organising modular robots

2.1.1 M-TRAN

M-TRAN [YMK+02], built by the Distributed System Design Research Group at the National Institute of Advanced Industrial Science and Technology (AIST) of Japan is a distributed, self-reconfigurable system composed of homogeneous robotic modules.[1] M-TRAN is able to perform both locomotion and self-reconfiguration by changing the position and connection of each module. The modules are made of two semi-cylindrical boxes which are linked together, in addition to a microprocessor, sensors, actuators, a connection mechanism and a power transmission system. Two servo motors (i.e. two DOFs) are placed in the link between the boxes. The connection mechanism is based on permanent magnets, with a polarity such that one of the boxes, named active, can be attached to the other box, called passive, of another module. When reconfiguring, the allowed angles between the modules are multiples of right angles ($0°$, $90°$, $180°$ and $270°$), although when performing locomotion any intermediate angle is valid.

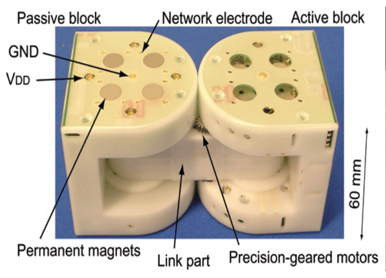
With this design, M-TRAN can move in several ways (crawl, quadruped locomotion, rolling and lattice-like locomotion[2]), climb obstacles, change from one shape to another or make a new bigger robot out of two smaller ones that come together. Figure 2.1 shows the module design and some structures of M-TRAN.

2.1.2 Conro project

The Conro project [CBW02], carried out by the Polymorphic Robotics Laboratory at the University of Southern California has developed non-lattice homogeneous modular robots with ability for self-reconfiguration. Their modules are composed of three parts: the module body,

[1] Homogeneous means all the modules are identical, unlike heterogeneous modules.

[2] Cluster-flow locomotion where the robot moves by continuously attaching and detaching over a lattice of modules.



(a)

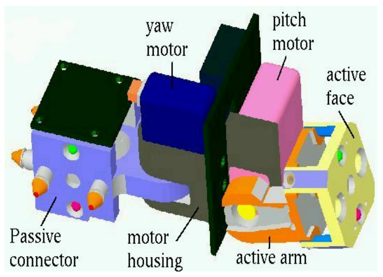
(b)

(c)

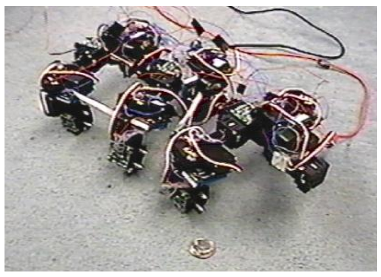
Figure 2.1: M-TRAN's module design (a), quadruped-like locomotion (b) and snake-like locomotion (c)

a passive connector and an active connector (cf. Figure 2.2). The module body is equipped with two servos that control the rotation of the passive connector and the active connector, respectively. One rotation is done over the yaw axis and one over the pitch axis. The connectors allow the module to attach to other modules. The passive connector has three lateral faces with protruding aluminum pins which fit into the sockets of the active connector. This connection mechanism only allows connections of modules in the same plane. Finally, the active connector performs the attachment and detachment of the modules. The attachment process is completely mechanical, that is, it does not require additional energy that the one used to rotate the servos: the aluminum pins are plugged into the sockets thanks to the dome-shaped head of the pins that forces an engagement latch to rotate and fix the connection. In order to detach, a shape-memory alloy wire is used to rotate the latch in the opposite direction.

Some of the robots which have been developed can be seen in Figure 2.2 too.



(a)


(b)

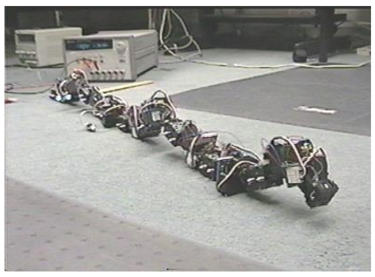

(c)

Figure 2.2: Conro's module design (a), hexapod robot (b) and snake robot (c)

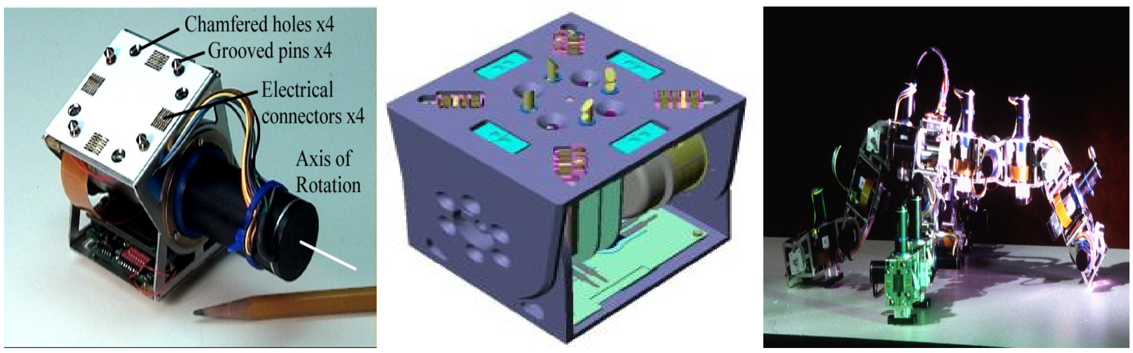
2.1.3 PolyBot

PolyBot [YDR00] is another modular reconfigurable robot system built by the Palo Alto Research Center (PARC) in California. It is composed of two types of modules (it is not, therefore, a completely homogeneous system), one called *segment*, which is active, an one called *node*, which is passive. The segment module has one DOF and two connection ports, while the node has no DOF but six connection ports. The shape of both modules is the same.

Polybot's modules are basically 5 cm cubes, weighing 416 g each, with two opposing lateral faces that have connection plates. The degree of freedom is used to rotate these faces so they are no longer parallel. The rotation of the servo that controls the DOF lies in the interval $[-90^\circ, +90^\circ]$, and can generate a torque of 4.5 Nm.

In order to attach two modules, connection plates in the above mentioned lateral faces are used. The connection plate consists of 4 grooved pins along with 4 chamfered holes, which makes a hermaphrodite connection mechanism. A shape-memory alloy actuator rotates a plate that catches the 4 grooves in the pins from a mating connection plate. Polybot allows two connection plates to attach in multiples of 90° with the possibility to act together in plane or out of plane.

Figure 2.3 shows Polybot's module design and an example of a multiple-unit robot.



(a) (b) (c)

Figure 2.3: Second generation Polybot's module design (a), third generation Polybot's design (b) and spider robot (c)

2.1.4 Crystalline

Crystalline [RV00] is a modular, self-reconfigurable robot developed by the Robot Lab at the Dartmouth College in Hanover (New Hampshire, USA). In simulation there are examples for instance of reconfiguration from a dog-like structure to a coach structure. Its physical 2D version is based on a mechanism that permits the robot to expand and contract its lateral faces, which have a connector (either active or passive, depending on the face) that allows the face to connect to other modules. The building blocks, called *atoms*, have a height of 7 inch ($\approx$ 18 cm), and are 2-inch ($\approx$ 5 cm) width when they are totally contracted and 4-inch ($\approx$ 10 cm) width when the faces are completely expanded, and its weight is 12 oz ($\approx$ 340 g). The faces are expanded (or contracted) all together with a rack-and-pinion mechanism, using only one DOF for this purpose. Crystalline is therefore a lattice modular robot, because locomotion can only be achieved by moving some modules with respect to others thanks to the expansion/contraction strategy. The other two degrees of freedom the modules have are used in the active connectors to perform the attachments and detachments. Figure 2.4 shows the design of the modules as well as the way expansion and contraction is done and how reconfiguration is achieved.

2.1.5 YaMoR

YaMoR [MJD+05] is a modular robot system developed by the Biologically Inspired Robotics Group (BIRG) at the Swiss Federal Institute of Technology Lausanne. YaMoR's modules are

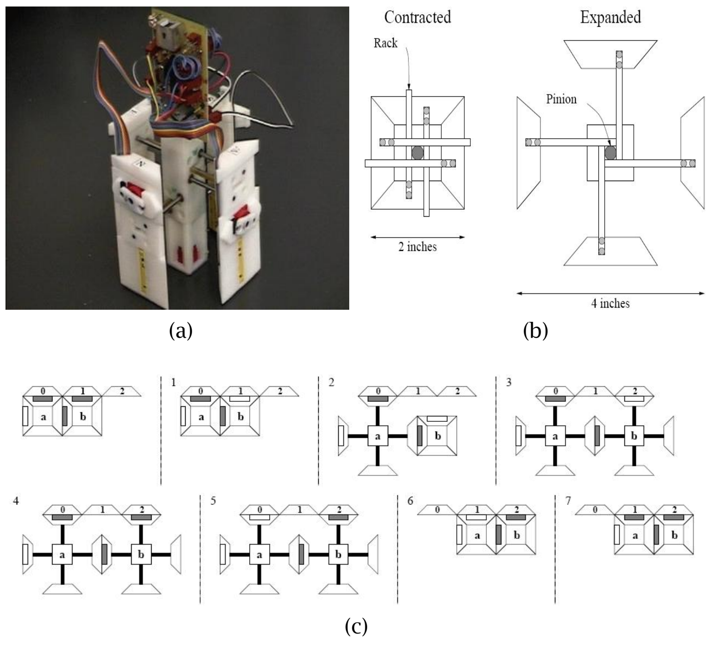


Figure 2.4: The Crystalline self-reconfigurable robot. Design of the module (a), expansion and contraction system (b) and reconfiguration example (c)

homogeneous, and are entirely constructed with off-the-shelf components. Each module has one DOF, controlled by a servo motor with 73 Ncm of maximum torque. One module can be attached to another thanks to strong Velcros, which allow to connect two modules with any angle between their surfaces. The attachments can only be made by hand; self-reconfiguration is not supported.

The modules are autonomous: every module has its own power supply (a Li-Ion battery), motor control and communication mechanisms, etc. They are flexible with respect to the device that controls their operation: an FPGA board can be used for this purpose, but also a micro-controller or a more sophisticated board with more specific sensors, for instance.

The communication among the modules and the high-level control of the robots (e.g., to define a new trajectory the robot has to follow) is done via a Bluetooth-ARM board. In Figure 2.5 some pictures of YaMor can be seen.

2.2 Control of locomotion in Robotics

2.2.1 ASIMO

ASIMO [HHHT98, SWA+02, CLC+05] is a 1.20-meter height, biped, humanoid robot developed by Honda which is intended to work as a household robot, and which, among other things, is able to perform locomotion using a ZMP-based mechanism.

ASIMO has 6 degrees of freedom per leg and 7 per arm. The movement of the legs is determined so that the robot remains with a stable posture. ASIMO's body angle is calculated with an inclination sensor and then the posture is corrected with a ZMP (Zero-Moment Point) controller and a strategy called *foot landing position control* in such a way that the robot is

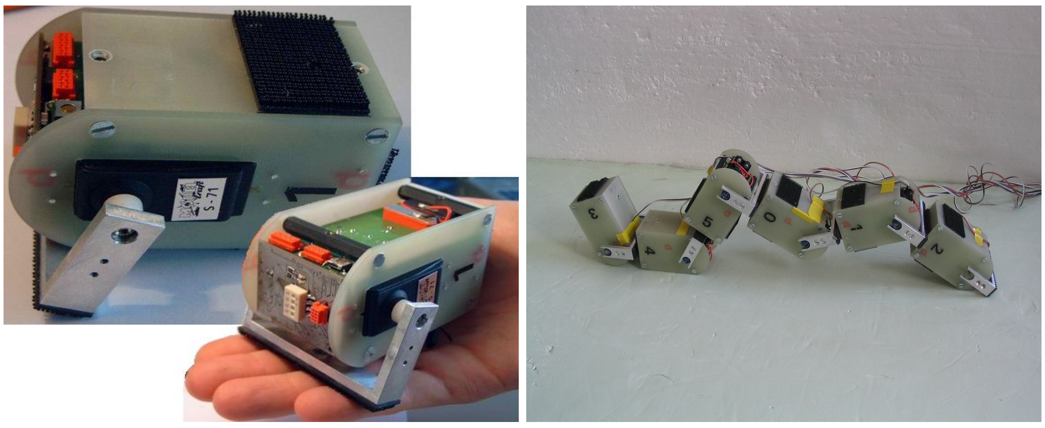

(a) (b)

Figure 2.5: A YaMoR's module (a) and a snake robot (b)

able to walk and run (up to 6 km/h both in even and irregular terrain, as well as deal with obstacles and stairs. Figure 2.6 shows a picture of ASIMO.

Figure 2.6: The ASIMO humanoid robot

2.2.2 QRIO

QRIO [Ish04] is a 0.58-meter height, biped, humanoid robot developed by Sony which also has locomotion abilities that are controlled by a ZMP-based mechanism. In a similar way to ASIMO, an inclination sensor system in the trunk and a force sensor system in the feet prevent the robot from falling down when it finds irregularities in the terrain or obstacles.

A different approach for biped locomotion with QRIO can be found in [ENMC05], where the robot is controlled by a central pattern generator. It uses the Matsuoka oscillator (cf. subsection 4.2.1) as building block for the oscillators of the CPG. The CPG is composed of two oscillators, one controlling the position of the legs with respect to the vertical plane and the other one controlling the position of the legs with respect to the sagittal plane. The output of the oscillator is sent to the legs in such a way that there is always a phase difference of π radians between the two legs. Control of equilibrium is done as well by adding three different sensory feedback pathways: *extensor response*, *vestibulo-spinal response* and *sagittal motion feedback*. In few words, they are models of reflexes that affect the normal oscillation of the CPG depending on the received sensory input. QRIO can manage 3-mm height unknown obstacles by autonomously adjusting its stepping period. Figure 2.7 shows a picture of QRIO.

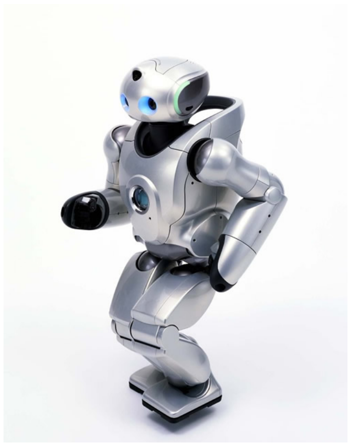


Figure 2.7: The QRIO humanoid robot

2.2.3 AIBO

AIBO [FK97] is a dog-like commercial robot for entertainment developed by Sony with locomotion, computer vision and speech recognition abilities. It uses and architecture named *OpenR*, which allows to customise the control mechanism of the robot. AIBO has a total of 20 degrees of freedom (to move the head, mouth, ears, tail and legs), from which 12 of them are associated with the legs (3 per leg: 2 on the hip/shoulder and 1 on the knee/elbow, the ankles are articulated but passive). In Figure 2.8 there is a picture of AIBO.

AIBO has been extensively used to do research on locomotion in Robotics. In [HTY$^+$00], a evolutionary algorithm with a centralised-gait-control-table controller is used. A total of 20 parameters (e.g. body pitch, posture center, step length or swing height) is evolved on two different surfaces (a flat surface and a surface with ridges). The best individual of the evolutionary process is able to walk at 387 cm/min on the flat surface and at 600 cm/min on the ridged one. In [BI00], an approach based on CPGs is adopted. The oscillators of the CPGs are made of leaky-integrator neurons, having one oscillator per articulation (i.e., one on each hip and one on each knee). Three quadrupedal gaits (walk, trot and gallop) are implemented by activating each oscillator asynchronously and the transition between gaits is obtained by changing the tonic input each oscillator receives.

Figure 2.8: The AIBO ERS-7 model

2.2.4 Tekken

Tekken [FKC03] is a quadruped robot built by the Graduate School of Information Systems at the University of Electro-Communications in Tokyo. Its second generation is 27.5 cm tall and 30 cm long, and weighs 4.3 kg including power batteries. Each leg has a hip pitch joint, a hip yaw joint, a knee pitch joint and an ankle pitch joint. The three first joints are activated by DC motors which are PD controlled, while the ankle joint is passively rotated when the toe contacts with an obstacle during the swing phase.

Tekken is controlled with a biologically inspired strategy. A CPG and a set of reflexes rules the movement of the legs. The CPG is made out of four Matsuoka oscillators, with one oscillator per leg. The oscillators are mutually entrained among them and also with a tonic input which is the sum of two reflexes. One of them is called *vestibulospinal reflex/response* and controls the inclination of the robot in the pitch plane, making Tekken flex the front legs and extend the hind legs when the first ones are above the last ones (e.g. when going up a slope) and vice versa, extending the front legs and flexing the hind legs when these are above the first ones. The other reflex is called *tonic labyrinthine response for rolling* and is responsible for maintaining the equilibrium when the rolling angle of the robot is not zero (that is, the right side is higher than the left side, or the opposite). The reflexes make Tekken to keep a stable position even when walking on irregular terrain. In Figure 2.9 two generations of Tekken are shown as well as the schema of the CPG and reflexes.

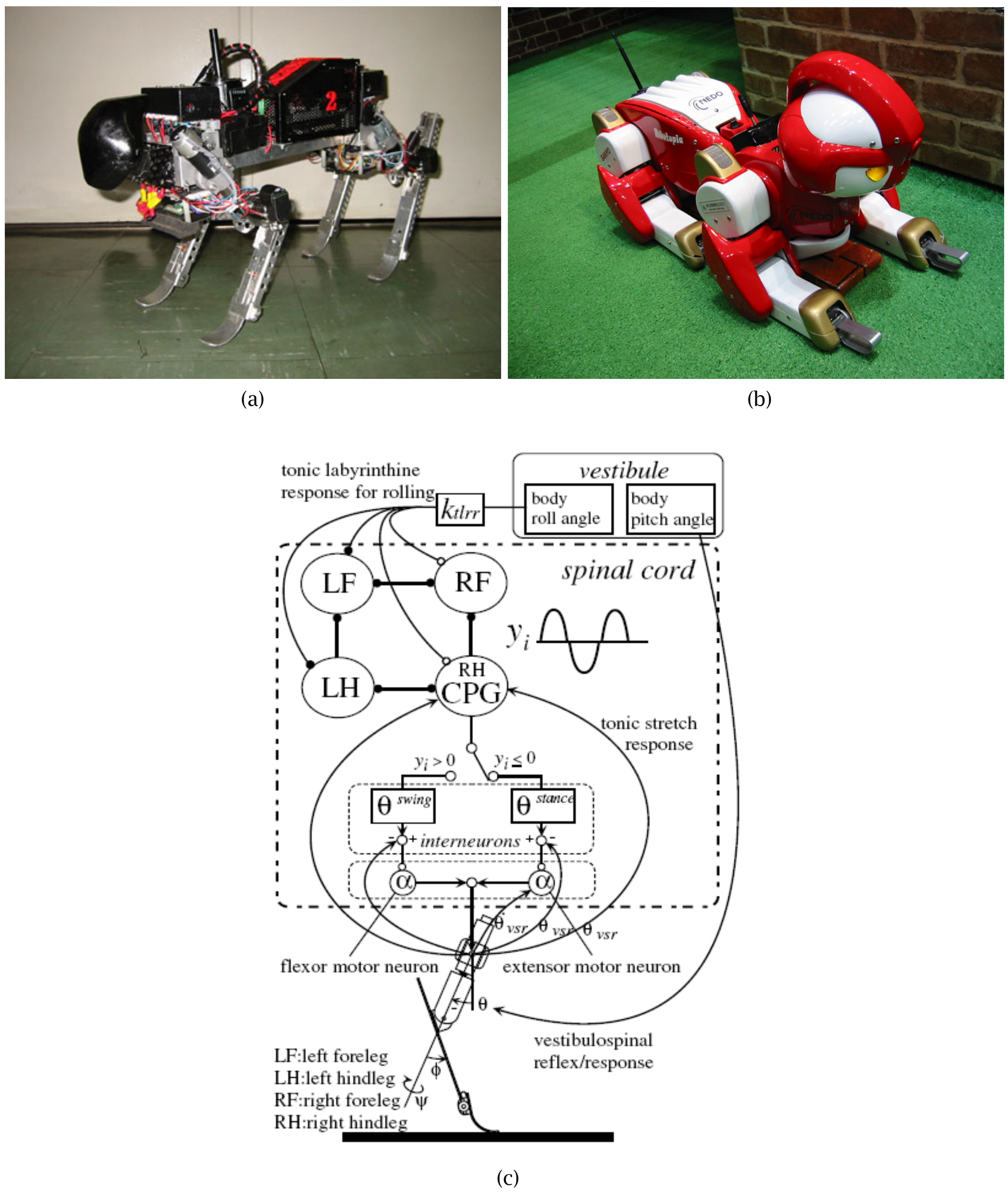


Figure 2.9: The Tekken quadruped robot. Second (a) and fourth (b) generations and control mechanism (c)

Chapter 3

Design of the Modules

The first part of this project was the design of the building blocks the multi-unit robots will be composed of, and the appropriate combination of the modules to make locomotion and self-organisation possible. The effort of this stage was shared with Sébastien Cevey, who did a semester project at BIRG in parallel with this master project [Cev06]. In addition to the structural design, he focused on the reconfiguration of the modules.

First of all, it is good to remember the main requirements the robots should fulfill:

1. The control of each robot should be as simple as possible.
2. The robots should be able to reconfigure (change their structure, dynamically add or remove new modules...).
3. The robots should have locomotion abilities.
4. The robots should be able to self-organise, i.e. depending on their specific structure, the external conditions, the user's requirements and so on, they should be able to adapt to the new conditions.
5. The robots should hold the weight of a person, since they will be used as furniture.

From item 1 it can be deduced that the number of degrees of freedom (DOFs) should be low and the controllers should have few input variables. One approach (the one that was chosen for this project) to do this is to modularise the robots so that the individual control of each module is quite simple and try to satisfy item 2, item 3 and item 4 by modifying the control parameters of each module and/or with appropriate attachments between robots and/or with interactions between the modules (an example of these interactions could be to use the state of one of the DOFs of one robot as the input of the controller of another robot). Finally, item 5 means the materials the modules are made of, but especially the interconnections between them, must be quite strong (for instance, a connection mechanism like the one M-TRAN has would not be suitable because normal magnets are too weak).

3.1 Exploration of some possible building blocks

3.1.1 Design decisions

The previous requirements lead to fixing some constraints about the properties the modules should have. These restrictions were carefully studied, because their importance is huge: a very small variation on the shape, the connection mechanism, the control strategy, the casing and so on can determine the success or failure of the whole project, due to the fact that locomotion and especially reconfiguration can drastically change because of a little change on these characteristics.

The main decisions that were adopted are:

Homogeneous modules: The use of homogeneous modules clearly simplifies the design, realisation and control of the robots. It also provides more flexibility to a multi-unit robot since every module can substitute any other one if the substituted module fails or for any other reason. Heterogeneous modules lack at this point. They might be useful if there were very defined, fixed, different roles in the multi-unit structures, but this is not the case is explored here, due to the fact that the need of reconfiguration abilities makes a module can play more than one role in principle.

Two degrees of freedom: With two DOFs, the control of each module is quite simple (only two actuators) as well as the range of movements the modules can show once they have been properly attached can be extremely wide. This versatility depends on the appropriate selection of the position and orientation of the actuators and the direction of the force it applies. For instance, it is convenient that the degrees of freedom apply orthogonal forces, because the region of the space a robot can reach is larger this way.

Dimensions suitable for furniture: An important property is the one referred to the size of the modules. Most of the modules will be used as part of the legs of the furniture, which will be the parts more likely to be active. The natural shape for the modules is therefore a prism. Since a chair is about 50 cm tall and a table around 75 cm, the best length for the modules could be around 25 cm (this way, the leg of a chair will be composed of two attached modules and the leg of a table will have three modules). The width and depth can be 10 cm roughly.

Strong attachment mechanisms: In addition to have an appropriate shape and dimension, the modules should be equipped with attachment mechanisms in order to create multi-unit structures and make reconfiguration possible. One important constraint for these connection devices is that the mechanism has to be strong enough to hold the weight of the furniture structures as well as the weight of a person sitting on them. Some other projects (e.g. M-TRAN) use magnets for the interconnection of the modules, but this solution is not valid here since the forces the attachments points will support will be much higher than the ones an ordinary magnet can provide. Therefore, another solution is needed. A mechanical connection device is likely to be suitable for the purposes required here.

Static pieces: Due to the relatively large dimensions of the pieces of furniture and the fact that it is not necessary that all the parts of a given item of furniture move (e.g. the top of a table would not), some static pieces are going to be used (such as the top of a table or the cushions of a sofa). These static pieces will have bindings to attach the active modules to them.

3.1.2 The wheeled module

The first module that was designed was the *wheeled module*. It is made of two rectangular prisms, one larger that is intended to contain all the circuitry of the module and one smaller with a wheel stuck on one of the faces. The two boxes are joined by a cylinder, which can rotate an angle $\alpha \in [-\pi/2 - \frac{\pi}{8}\text{ rad}, \pi/2 + \frac{\pi}{8}\text{ rad}]$ and also makes the small box rotate with respect to the big one. The wheel turns around its axis with no restrictions in the angle of rotation. Therefore, the modules have two DOFs. Figure 3.1 gives a better idea of the structure of the wheeled module, the position of the axes of rotation and the dimensions of its parts.[1]

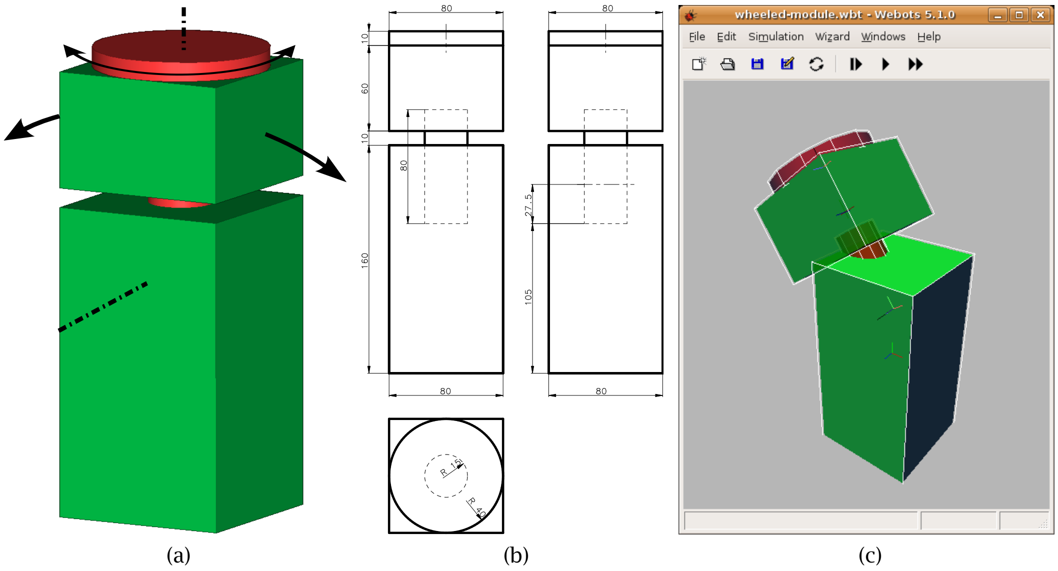

80
10
60
10
80
160
80
80
27.5
105
80
R 12
R 40
wheeled-module.wbt - Webots 5.1.0
File Edit Simulation Wizard Windows Help
(a)
(b)
(c)

Figure 3.1: Structure of the wheeled module: 3D view (a), 2D view with dimensions in mm (b) and screenshot of the representation of the module in Webots (c)

The modules fit into a box of $24 \times 8 \times 8$ cm, making the attachments simpler because the longest dimension is a multiple of the two others. The total weight of the module is intended to be of about 500 g.[2]

Three different versions of this module were studied: in the first one the wheel is inscribed in the 8×8 cm square of the top base of the small box (as shown in Figure 3.1); in the second version the wheel circumscribes the base of the small box (in this case, the dimensions of the small box are shortened so that it is inscribed in the wheel, whose dimensions do not change). This last alternative is depicted in Figure 3.2.

Both versions have pros and cons. The inscribed wheel allows a larger number of attachments, because nothing sticks out (for instance, the wheel could be attached to one of the lateral faces of another module, which is not possible with the circumscribed version because the wheel of one module would collide with the one of the other module). In addition to that, with the circumscribed version it is no longer possible to perform attachments where the small box participates because it becomes too small. On the other hand, wheels offer the possibility of rolling locomotion, but with inscribed wheels the small box would touch the ground and, in order to roll, much more energy would be needed due to the higher friction.

Finally, in the third version the axis responsible of the rotation of the small box lies in the small box itself, instead of in the big box as the two previous versions had. The main

[1]A large amount of details was skipped in the structural design of the modules. This level of detail simplifies the simulations of locomotion and is enough to emulate the behaviour of the real robots. The actual modules will have of course more pieces in order to achieve the functionality described above.

[2]The actual weight is not specified because for this project it was not necessary to do a whole mechanical study of the modules.

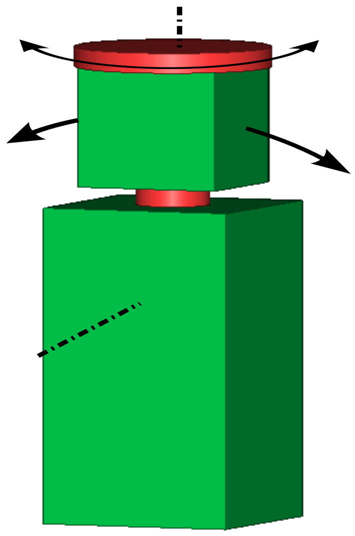


Figure 3.2: Circumscribed version of the wheeled module

advantage of this alternative is that the components of the module always stay inside the $24 \times 8 \times 8$ box, whatever the rotations of the actuators are. This property might lead to an easier exploration of reconfiguration. However, this design was discontinued because the realisation of this module would be slightly more complex than the two first versions.

Attachments of the modules

One of the critical features the robots must have is the ability for reconfiguration. In order to fulfill such property, the robots should be able to attach and detach to each other. The strategy that has been adopted is to attach and detach the modules by joining one of the 14 faces of one robot with one of the 14 faces of another. The angle of the attachment between two modules cannot be anyone: the allowed angles are $0°$, $90°$, $180°$ and $270°$, which is likely to offer enough flexibility for reconfiguration as well as it largely prunes the tree of possible configurations. Figure 3.3 shows the way the division in attachment faces was done as well as the number of each one.

One important aspect this thesis leaves as an open problem (because reconfiguration was not covered) is whether the attachments between the robots should be done with male/female pins or with hermaphrodite ones. If hermaphrodite connectors are used, then the possible range of configurations is the one just described. Otherwise, incorporating male pins to some faces and female ones to some others might reduce the versatility in terms of the attachments.

However, quite a lot of flexibility can still be achieved with a male/female schema. For instance, a distribution of connectors as shown in Figure 3.4, where male and female pins alternate in adjacent faces, would give a very large number of combinations because practically all attachments where a lateral face participates would be feasible. When an attachment of faces both with male (or female) pins is about to be done, it is just enough with rotating the module around its longer axis (the one that goes through the wheel and the big box). This action would restrict the available faces for future attachments, but not too much, because normally a rotation could be performed to solve these collision problems. It is important to notice that with this particular distribution, attachments wheel-wheel would not be allowed either.

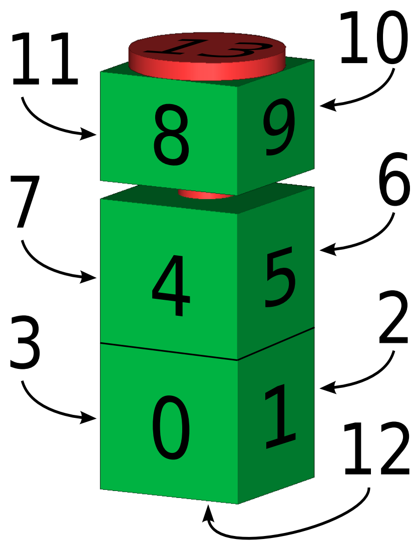


Figure 3.3: Numbering of the faces of the wheeled module

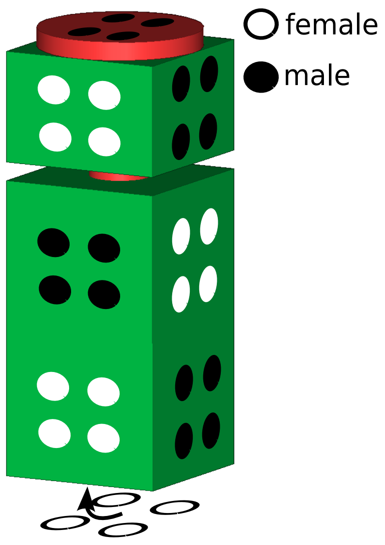


Figure 3.4: A schema for the distribution of male/female connectors in the wheeled module

The final choice will basically depend on the difficulties that might happen in the design of the hermaphrodite connectors. If these are easy to realise and strong enough to assure the attachments will not break when the robots hold a person, then they will be the best option. If not, male/female pins can probably have the a similar performance as hermaphrodite ones as commented above.

3.1.3 The universal joint module

The second module that was taken into account was a module with two DOFs too, but this time a universal joint is used (see Figure 3.5).

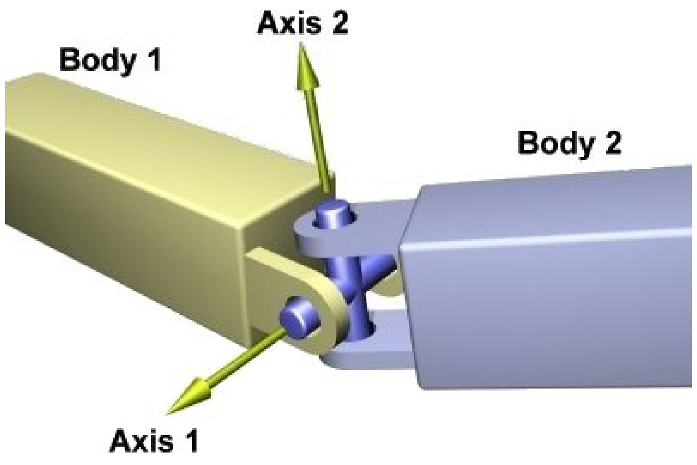


Figure 3.5: Representation of a universal joint between two bodies (from [ODE])

The design of this module is a composition of two boxes of the same dimensions linked by a universal joint. Each of these boxes is approximately 12-cm long and 8-cm wide and deep. Figure 3.6 depicts in a more detailed way the structure of the universal joint module.

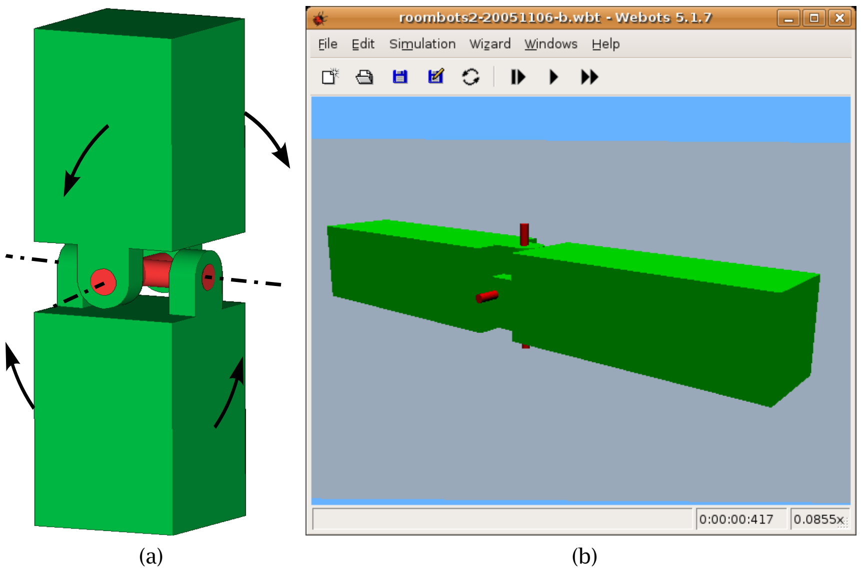

(a)
roombots2-20051106-b.wbt - Webots 5.1.7
File Edit Simulation Wizard Windows Help
0:00:00:417
0.0855x
(b)

Figure 3.6: Structure of the universal joint module: 3D view (a) and representation in Webots (b)

The detailed dimensions are not provided because finally it was decided to concentrate all the effort in the wheeled module. The reason was mainly the fact that, although the universal joint module is probably able to cover a larger 3D region, a proper combination of wheeled modules can be as good as the universal joint modules in this sense and, moreover, the wheeled module results in the possibility of rotations of modules (and, consequently, due to the attachments, of groups of modules) and also creates the option of rolling locomotion.

3.1.4 Conclusion

For all the reasons described in the last paragraph of subsection 3.1.3, the decision of using the wheeled module for all the subsequent designs of multi-unit structures, as well as to provide the robots with locomotion abilities, was adopted.

3.2 Pieces of furniture

Once the structure of the building blocks has been fixed, the next step is the design of several predefined pieces of furniture (such as a chair or a table). Then, locomotion mechanisms will be implemented on them and finally, in upcoming projects or PhD theses, the strategies for reconfiguring from one piece to another will be explored. It is convenient not to forget in principle the possible furniture-like structures depends also on the user. Ideally, the user should be able to specify new structures, by using an interface which will be developed at the Learning Algorithms and Systems Laboratory (LASA) at the EPFL. These structures should self-organise to be able to move, probably with the method commented in [MI05] or the one developed in this project (see section 4.4) and to reconfigure to the previously stored structures.

Some of the ideas for the different structures that were used as a source of inspiration can be seen in Figure 3.7, in addition to a preliminary design of the universal joint module (which finally was discarded).

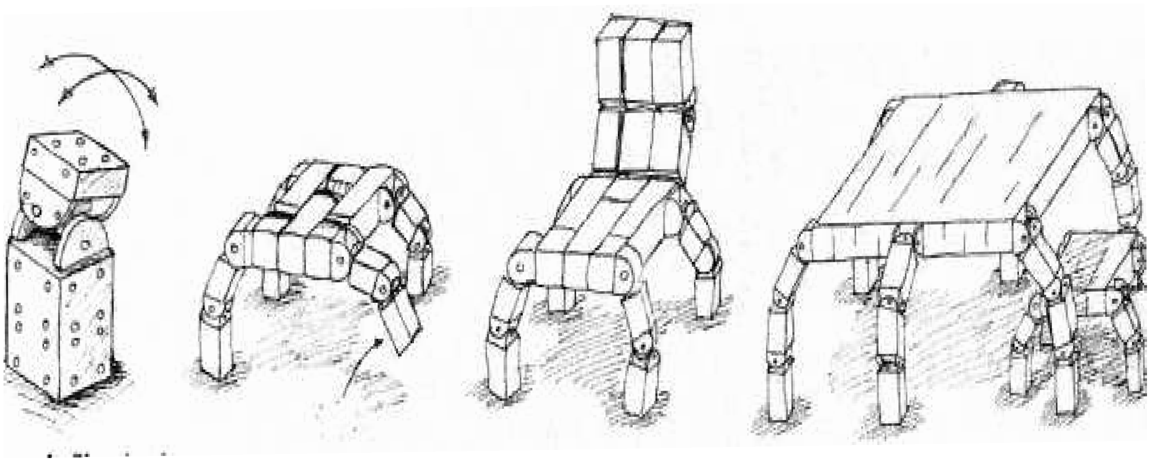


Figure 3.7: Some of Professor Ijspeert's drawings for the Roombots *project: universal joint module, stool, chair and table*

3.2.1 The stool

The stool is the first piece of furniture that was designed, since it is the simplest one (a chair is basically a stool with a back and a table would also require more modules because it is bigger). In Figure 3.8 the representation of the stool in Webots is shown.

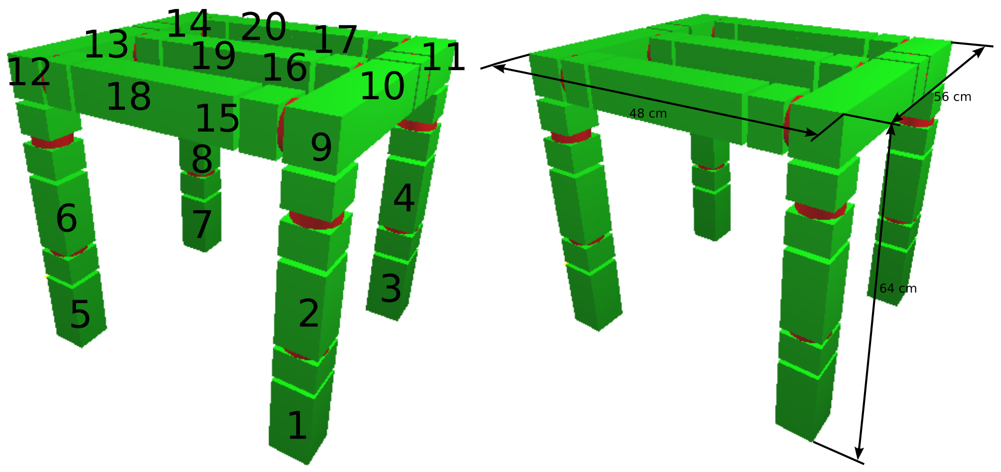


Figure 3.8: Design of the stool in Webots: detail of the component modules (a) and main dimensions (b)

The stool is made of 20 modules that distribute in such a way there is a top where people can sit and four legs. It is 64-cm tall, 48-cm width and 56-cm deep. The top is composed of 12 modules and is not supposed to move (i.e. this part will be static in order to guarantee people's equilibrium and comfort). It could be complemented with some additional objects such as cushions. The 8 resting modules are located in the 4 legs of the stool, that is, there are two modules per leg. These two modules are attached in such a way that the actuators that rotate the small box work as the joints of the hip and knee, being able then to produce the required movements for legged locomotion. Besides, the wheel of the upper module can be used to rotate the whole leg so that the stool can turn as explained in subsection 4.2.5. Figure 3.9 shows a snapshot of the stool while walking (videos are available as well on the web page of the project [Arc06]).

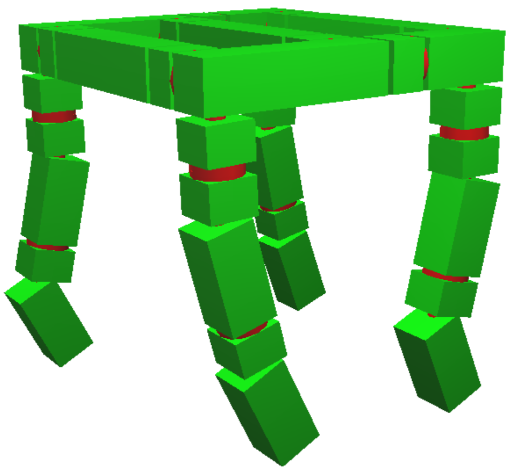


Figure 3.9: Snapshot of the stool walking

Alternative structure: the rolling stool

Rolling locomotion is an interesting possibility on flat surfaces or surfaces with little irregularity. In order to achieve such rolling gait, the design of the stool has to be slightly modified so that wheels are in contact with the ground. These changes are depicted in Figure 3.10.

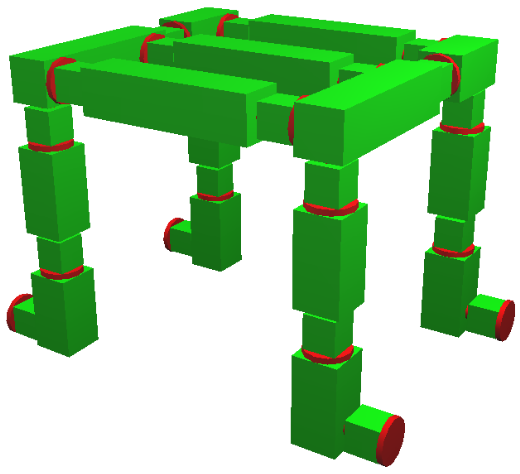


Figure 3.10: Structure of the rolling stool

As advantages and drawbacks, it is convenient to remark this configuration offers the option of rolling and an easy control of direction by rotating the wheels of the upper modules of the legs. With this design, legged locomotion is also possible. It simply needs that the actuators that control the rotation of the small box turn with an angle $\alpha > \pi$ rad, so that the small box and the wheel are not in contact with the ground and the mechanism to walk becomes the same as the one of the normal design of the stool.

On the other hand, locomotion by rolling requires the not-so-elegant design of circumscribed wheels that stick out. It also needs very powerful servomotors as actuators to hold the weight of the structure itself and possibly a person, because now only the wheels would support all the weight of the stool. Moreover, control of direction for legged locomotion with this configuration seemed to be more difficult to perform.

3.2.2 The chair

Another structure which was tested is the chair. It just has the same structure as the stool but with a back in one of the sides. The back is made of 9 modules distributed in 3 columns of 3 modules each. These modules could also be substituted by one or several static pieces. Figure 3.11 shows the representation of the chair in Webots. As well as for the stool, a rolling version of the chair can be implemented.

3.2.3 The table

The table is the third piece of furniture that was taken into account. Its design is different from the stool or the chair. Unlike them, the four legs of the table are composed of three modules so this structure is higher than the previously described ones (in particular, 72-cm tall). However, the control strategy for the table is basically the same as the stool or chair.

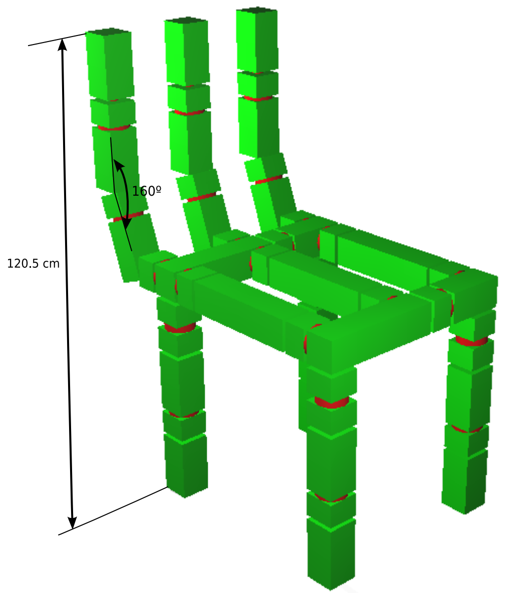


Figure 3.11: Design of the chair in Webots

Only two joints move: hip and knee, located in the middle and lower modules respectively. Control of direction can again be performed by rotating the wheel of the middle module. In addition to the 12 modules of the legs, 16 modules are used to interconnect the legs (namely, 4 modules between one leg and another) and hold the top of the table, for which a static panel is used.

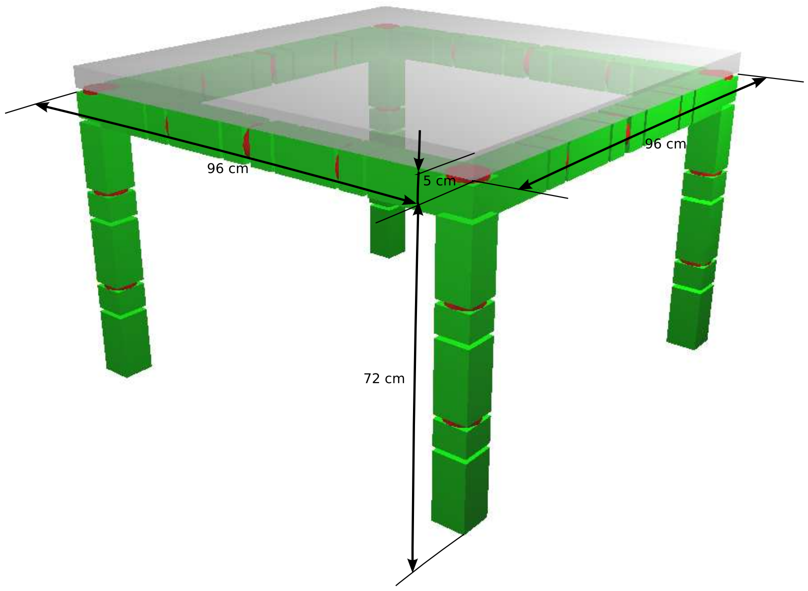


Figure 3.12: Design of the table in Webots

3.2.4 The sofa

The sofa is the last pre-designed piece of furniture that was studied. It is made of 56 modules divided into legs, seats, back and armrests. The sofa was conceived as the composition of two chairs that come together (excluding the armrests). A screenshot of the representation of the sofa in Webots can be seen in Figure 3.13.

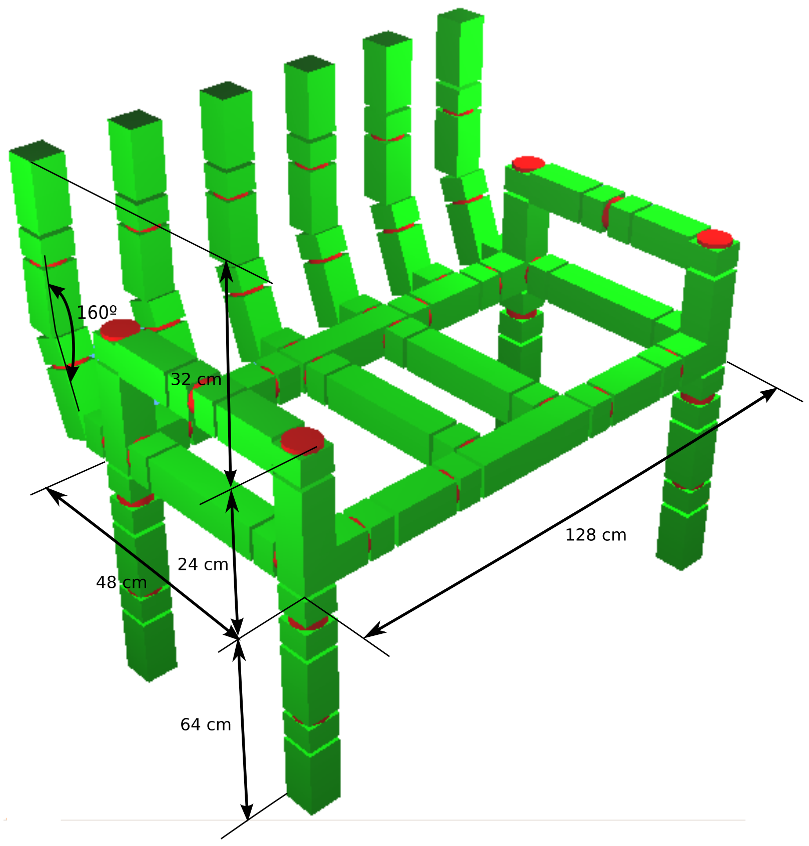


Figure 3.13: Design of the sofa in Webots

Locomotion was not explored with the sofa because the simulations were too slow due to the large number of modules and, especially, joints (the speed of the physics simulation largely depends on the number of joints). Anyway, the control strategy should be similar to the one used for the other structures because the layout of the legs is the same.

3.3 Other structures

Some other structures have also been built, mainly to be used as a mechanism to move some modules from one position to another (e.g. for reconfiguration purposes). Two examples of them are depicted in Figure 3.14.

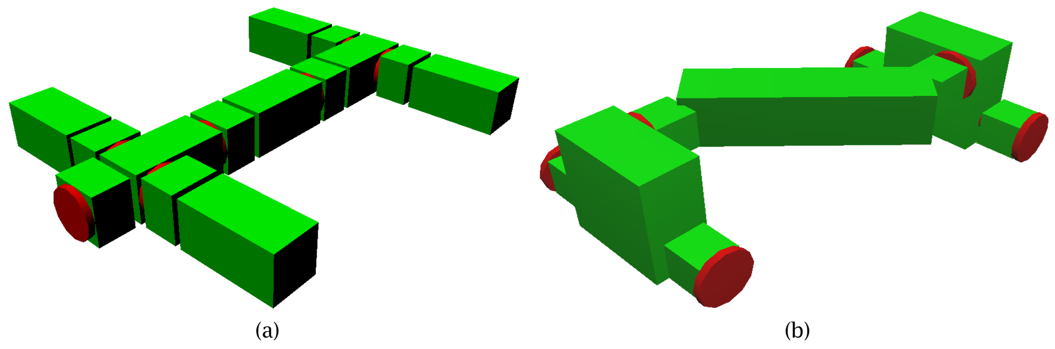


Figure 3.14: Two structures for moving modules from one location to another: salamander-like robot, which crawls (a), and moving bridge *structure by S. Cevey [Cev06], which rolls and can be driven as a car (b)*

Chapter 4

Control of Locomotion

The second objective of this project, after the design of the prototypes of the building blocks and the creation of some predefined pieces of furniture (chapter 3), was to provide the multi-unit robots with locomotion abilities. In this process, locomotion was explored using the stool, although for the other pieces of furniture the control would be the same since the distribution of the legs and the degrees of freedom and so forth is identical for all pieces of furniture. Probably some of the parameters of the controller have to be changed for the rest of the furniture, because their structure and weight are different. However, the methods described here for the stool should be still valid.

First, a simple sine-based controller was developed to test the ability of the modules to perform locomotion. Then, a more sophisticated control mechanism based on CPGs and reflexes is studied, which allows to easily switch between different locomotion gaits, perform control of direction and finally adapt to the irregularities of the terrain. Locomotion by rolling of wheels is also briefly introduced for the versions of the furniture which have wheels making contact with the ground.

Finally, a different approach is introduced. An algorithm of online optimisation of locomotion was explored in order to provide the modules with adaptive locomotion for new, unseen situations. The algorithm explored here allows the robots to adapt its gait to the external conditions (irregularities on the terrain, failure a module, change in the structure of the robot and so forth). This exploration is important because the arrangement of the structures changes during the reconfiguration processes, and the intermediate structures should be able to move, as well as the new possible structures the users create. Therefore it is appropriate to have a way to learn new locomotion strategies that suit the new pieces of furniture.

4.1 Sine-based controller

A very basic, open-loop, sine-based controller was developed at the beginning. The idea is to use a sinusoidal function to generate the rhythms that will control the movements of the legs of the furniture. The furniture will move in a quadruped-like fashion, as quadruped animals do. No control of direction or feedback is performed by this controller.

Particularly, the four legs of the furniture oscillate in the following way. Each leg has three degrees of freedom: hip, knee and one actuator that controls the rotation of the hip around the vertical axis.[1] The articulations are the rotational servos of the modules, which produce

[1] Actually the legs have more articulations (all the degrees of freedom of the modules they are composed of), but only these ones are used for locomotion.

the required torques to move the legs in an oscillatory, inverted pendulum-like fashion. Figure 4.1 shows the arrangement of the joints in a graphical way.

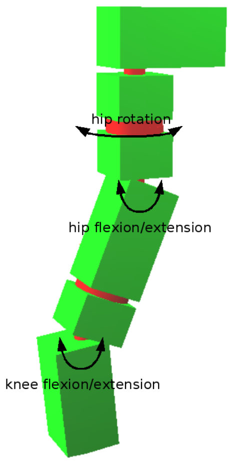


Figure 4.1: Articulations in the legs of the stool and the chair

Sinusoidal functions are sent to the modules, whose values are used as the target position of the actuators that control the hips and the knees. These functions have the same amplitude and frequency, which produces a more stable gait than when having different amplitudes or frequencies, and differ on the phase of the signal. In a more concrete way, the sinusoidal equations are these ones:

$$hip_i(t) = A\sin(\omega t + \theta_i) \tag{4.1}$$

$$knee_i(t) = A\sin(\omega t + \theta_i - \pi/2) + 2A \tag{4.2}$$

For $1 \le i \le 4$. Leg 1 is the front left leg, while leg 2 is the hind left one, leg 3 the front right leg and finally the hind right one is leg 4. A is the amplitude of the leg oscillation, ω the angular velocity, t is the time and θ_i is the phase for leg i. These equations are plotted in Figure 4.2, and the evolution of the trajectory of the legs can be seen as snapshots in Figure 4.3.

In every simulation step t_j, the controller calculates the values $hip_i(t_j)$ and $knee_i(t_j)$ and send them to the servos controlling the flexion of the respective articulations of the leg i. The servos use these values as their target position.[2]

A distinctive characteristic of quadrupeds is the capacity of switching from one locomotion gait to another, depending normally on the desired speed of locomotion. This way, an animal moving at low speeds can have a walking gait, if it is moving at medium speed it can adopt a trotting gait, while for high speeds the most common gaits are galloping and bounding.

Since the pieces of furniture are composed of four legs, an exploration of different gaits can be carried out (cf. subsection 4.2.3). This gaits are obtained by tuning the relative phase differences of the oscillators. The main ones are indicated in Table 4.1. In this project, only walk, trot and bound were tested.

4.1.1 Results and analysis

Locomotion with the sine-based controller was explored only for the stool. The goal of this exploration was simply to test if the modules were able to interact in such a way that locomo-

[2]Another approach is to use this value as the torque the servo produces, as in [FKC03].

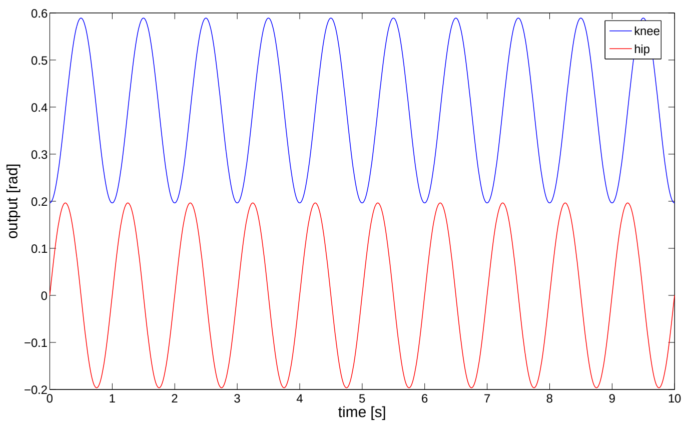


Figure 4.2: Output of the sine-based controller for the front left leg. The parameters ($A = \pi/16$ rad*,* $\omega = 2\pi$ rad/s *and* $\theta = 0$ rad*) were chosen by hand and they produced a stable gait*

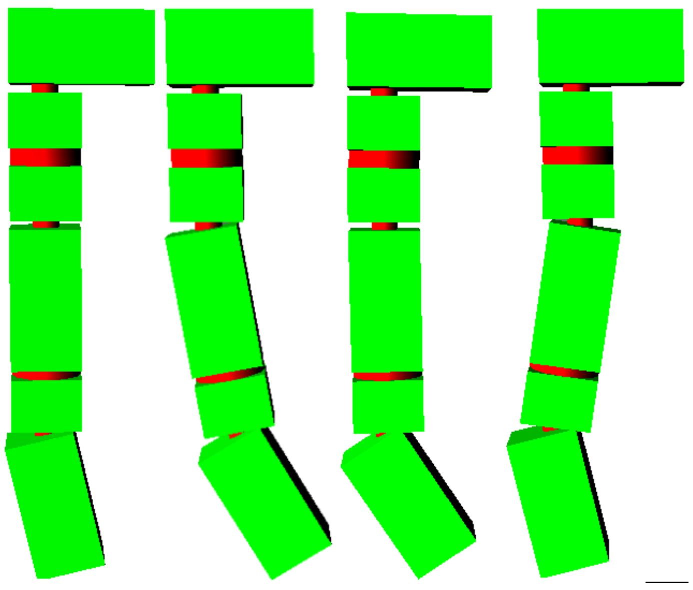


Figure 4.3: Trajectory followed by the leg with the sine-based controller for $t = 0$, $t = T/4$, $t = T/2$ *and* $t = 3T/4$ *(*$T = 1/f = 2\pi/\omega = 1$ s*),* $A = \pi/16$ rad *and* $\theta = 0$ rad

Gait	$\boldsymbol{\theta_{fl}}$	$\boldsymbol{\theta_{hl}}$	$\boldsymbol{\theta_{fr}}$	$\boldsymbol{\theta_{hr}}$
Walk	0	π	$3\pi/2$	$\pi/2$
Trot	0	π	π	0
Bound	0	π	0	π
Pace	0	0	π	π
Jump	0	$\pi/2$	0	$\pi/2$
Pronk	0	0	0	0

Table 4.1: Phase differences (in radians) for the most common gaits

tion was possible, and see how the speed of the stool varies with the changes on the amplitude and frequency of the oscillations.

The experiment that was designed consisted in trying a trotting gait (cf. Table 4.1) on a flat surface, with no control of direction or feedback. For every pair amplitude/angular velocity (A, ω), the locomotion of the stool was simulated for a time $t = 10$ s, with all the hips and knees having the same pair (A, ω). After that, the linear velocity of the stool for the pair (A, ω) was measured as

$$|\vec{v}| = \frac{|\Delta\vec{s}|}{\Delta t} = \frac{|\Delta\vec{x} + \Delta\vec{z}|}{\Delta t} \tag{4.3}$$

The amplitude took values in $[\pi/32, \pi/8]$, with increments of $\pi/32$ units, while the angular velocity ranged in $[\pi/4 \text{ rad/s}, 2\pi \text{ rad/s}]$, with increments of $\pi/4$ rad/s. The simulation was run in Webots with a time step of 8 ms for the integration of the equations that rule the physics of the world. Figure 4.4 summarises the results of such an experiment.

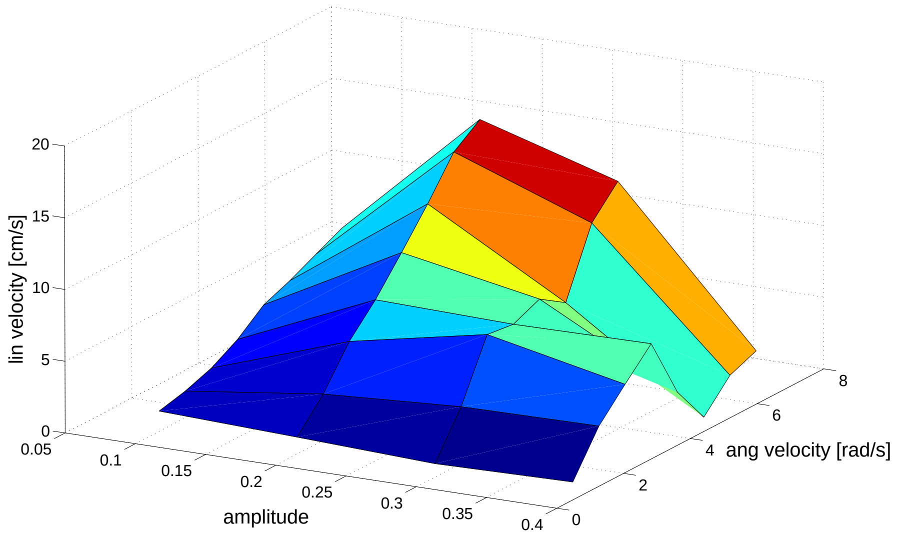


Figure 4.4: Results of the exploration of amplitude and angular velocity with the sine-based controller

The reader can notice how the linear velocity increases when the angular velocity does the same. However, with the amplitude, the best configuration was for $A = \pi/16$, which provided a linear velocity of 16.41 cm/s for an angular velocity of 2π rad/s. The reason of this is that with very high amplitudes the movements of the legs of the stool are very wide and lead to a loss of equilibrium and the stool collapses.

Due to these results, a deeper exploration of the influence of the angular velocity was carried out, with the aim of checking if the linear velocity always increased with the angular velocity or, on the contrary, there was a point where it began to decrease. This time, the angular velocity ranged in $[\pi/4 \text{ rad/s}, 4\pi \text{ rad/s}]$, with increments of $\pi/4$ rad/s. The amplitude was fixed at the best value that was found, that is, $\pi/16$ units. Figure 4.5 shows the results of this exploration.

It turns out that the linear velocity keeps on increasing until $\omega = 5\pi/2$ rad/s, for which the linear velocity is $v = 2.04$ cm/s. However, after that it remains roughly constant. This happens because, despite the more-rapidly-changing values which are sent to the servos with higher angular velocities, the servos are not able to *follow* the rhythm of the signal they

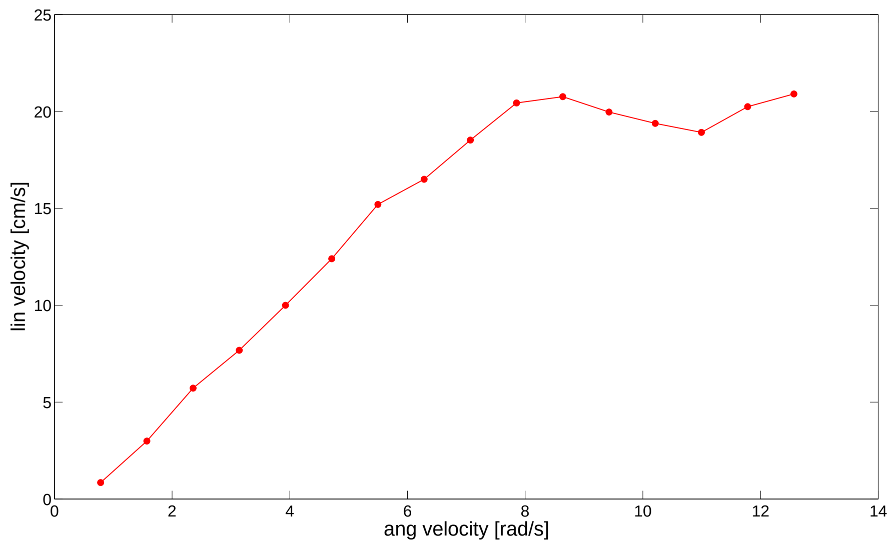


Figure 4.5: Results of the exploration of the angular velocity with the sine-based controller for an amplitude of $\pi/16$ *units*

receive, because they need some time to reach the position they are commanded to move to, and sometimes, when they are in their way to reach some position, the sinusoidal signal goes in the opposite direction they go and they stop their movement and continue in the direction of the signal. Therefore, they never get the maximum or the minimum of the target position with high-frequency signals.[3]

In conclusion, this controller is extremely simple and locomotion can be performed effectively. However, it presents a great drawback: it is really complex to modulate the signal sent to the servos in a good way. For instance, when switching from one gait (e.g. walk) to another (e.g. trot), it is not very appropriate to simply change the relative phase among the legs in a particular instant, because that entails an abrupt jump in the output of the controller.[4] A similar problem occurs when changing the amplitude or the frequency. Furthermore, within this controller it is also rather difficult to include a mechanism to modulate locomotion depending on sensory input. For all these reasons it is much more appropriate to take advantage of the benefits of central pattern generators.

Anyway, the exploration of the linear velocity depending on the amplitude and angular velocity is not useless since it gives a clue to the amplitudes and frequencies the CPG-based controller should have in order to optimise the speed of locomotion.

[3]In particular, the movement of the servos in Webots is ruled by a PID controller, whose equation is $T(t) = K_p \cdot e(t) + K_i \cdot \int e(t)\mathrm{d}t + K_d \cdot \frac{\mathrm{d}}{\mathrm{d}t} \cdot e(t)$, where $T(t)$ is the torque applied by the servo at time t, $e(t) = \theta_{comm}(t) - \theta_{real}(t)$, that is, the difference between the commanded position and the real one, and K_p, K_i and K_d are constants representing the gain of the signal.

[4]In order to solve this problem, a method to smooth this transition could be implemented.

4.2 CPG-based controller

After the sine-based controller, a CPG-based controller was designed to control locomotion in the stool. It uses a set of coupled Matsuoka oscillators (subsection 4.2.1) to produce the rhythms of the legs. This strategy provides the legs with coordinated oscillations in a natural and easy way (because of the couplings), and also allows to modulate the output of the controller by modifying simple input signals (normally the parameters of the oscillator) to adapt to the environment conditions (irregular terrain, perturbations, obstacles) or to change the characteristics of locomotion (increasing the amplitude and/or the frequency, changing the phase differences to switch from one gait to another, changing the amplitude of some oscillators with respect to others to turn and so on).

4.2.1 The Matsuoka oscillator

The building block for the CPGs that control most of the features of locomotion in the robots is the Matsuoka oscillator [Mat87]:

$$\tau \dot{u}_e = u_0 - u_e + w_{fe} y_f - \beta v_e \tag{4.4}$$

$$\tau' \dot{v}_e = y_e - v_e \tag{4.5}$$

$$\tau \dot{u}_f = u_0 - u_f + w_{fe} y_e - \beta v_f \tag{4.6}$$

$$\tau' \dot{v}_f = y_f - v_f \tag{4.7}$$

$$y_{\{e,f\}} = \max(u_{\{e,f\}}, 0) \tag{4.8}$$

This oscillator models two coupled neurons, e and f (representing, respectively, an extensor motoneuron and a flexor motoneuron), with two state variables each, u and v. The variable u represents the membrane potential of the neuron body, while v is related to the degree of fatigue or adaptation in the neuron. The parameter u_0 models the impulse rate of the tonic or slowly varying input, while w_{fe} refers to the strength of the bidirectional coupling between the neurons. y is the firing rate or output, and β is a parameter that determines the steady-state firing rate for a constant input. Finally, τ is a time constant that represents the rise time of the neuron and τ' refers to the adaptation time. The final output of the oscillator is computed as $y_{osc} = y_f - y_e$. A graphical explanation is provided in Figure 4.6.

The choice of the Matsuoka oscillator was done on the fact that it is a well-known oscillator, widely used to control locomotion of robots and whose behaviour has been precisely studied as mentioned just below. In addition to this, the oscillator provides good robustness against perturbations [Wil99], which is an important point here since the pieces of furniture will be tested on irregular surfaces and with obstacles such as steps. In other words, the detection of an irregularity in the terrain or an obstacle will result in a modification of the normal oscillations of the CPG. When the exceptional event finishes, then the Matsuoka oscillator is able to quickly recover the oscillatory pattern it just had before such event. In fact, the oscillator has been successfully used to deal with irregularities on the terrain [FKC03].

Oscillatory behaviour

The neurons of the Matsuoka oscillator are able to produce stable oscillations that can be used to control locomotion. The conditions for this oscillation are the following ones [Ars04]:

$$\beta > -w_{fe} - 1, \quad w_{fe} < -1 - \frac{\tau}{\tau'} \tag{4.9}$$

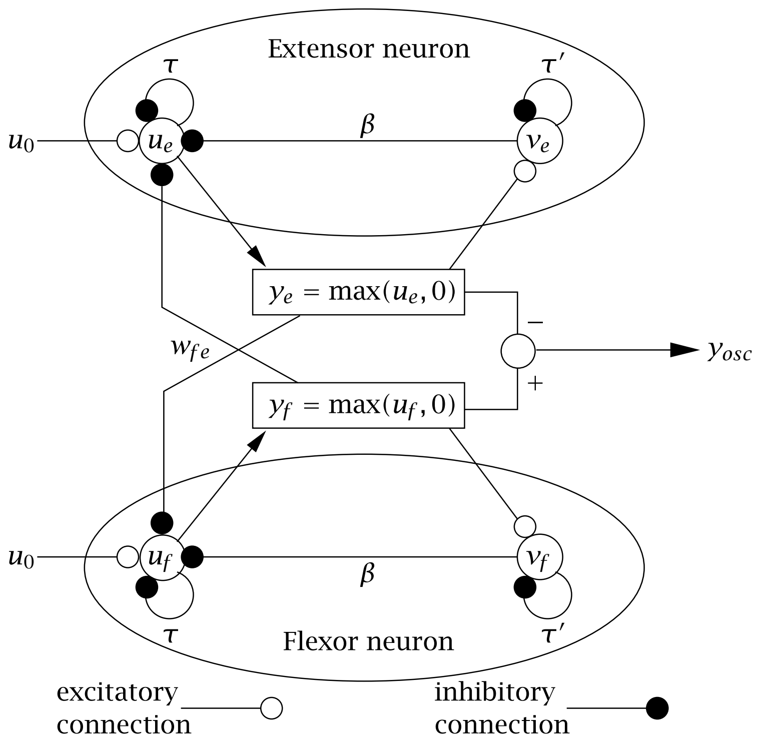


Figure 4.6: The Matsuoka neural oscillator (from [FKC03])

The amplitude of the oscillations depends linearly on the parameter u_0, while the frequency is proportional to $1/\sqrt{\tau\tau'}$. These relations allow an easy tuning of the parameters to achieve the desired amplitude and frequency. These and other properties of the Matsuoka oscillator (equilibrium points, stability, limit cycles and so forth) can be found in [Ars00, Ars04].

This oscillator shows a characteristic ellipsoidal-like limit cycle, whose exact shape can be controlled by modifying the parameters β, w_{fe} and the ratio τ/τ'. Figure 4.7 shows two plots with the oscillations of the Matsuoka oscillator.

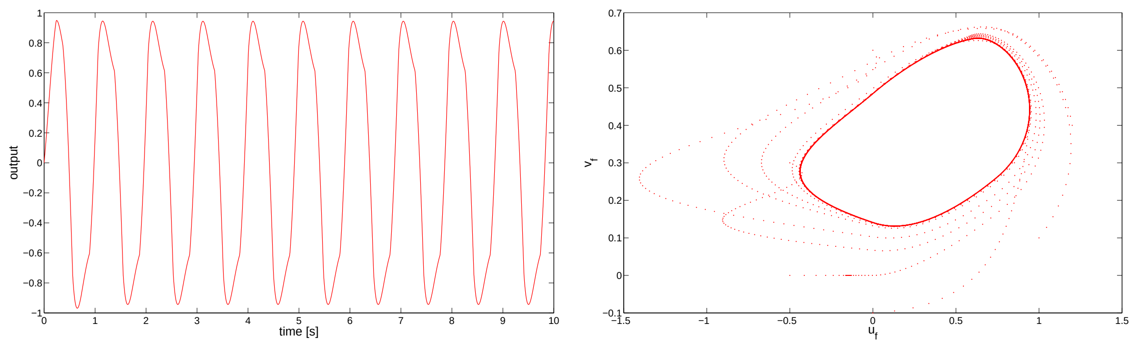
Coupling between oscillators

In order to have stable gaits and synchronisation among the legs of the different pieces of furniture, control of the relative phase between the legs must be done. The best way to do such control is to first study how the mutual influence of two coupled oscillators is.

Within the set of equations of an oscillator, a coupling between oscillator i and oscillator j (in this way) is represented as two terms, $w_{ij}y_f$ and $w_{ij}y_e$, which are incorporated into the equations of oscillator i by changing Equation 4.4 and Equation 4.6 by these ones, respectively:

$$\tau\dot{u}_{ei} = u_0 - u_{ei} + w_{fe}y_{fi} - \beta v_{ei} + \sum_{j=1}^{n} w_{ji}y_{ej} \tag{4.10}$$

$$\tau\dot{u}_{fi} = u_0 - u_{fi} + w_{fe}y_{ei} - \beta v_{fi} + \sum_{j=1}^{n} w_{ji}y_{fj} \tag{4.11}$$



(a) (b)

Figure 4.7: Output of the Matsuoka oscillator. Time evolution (a) and phase plot for different initial conditions (b). The parameters for both plots are $u_0 = 2.05$, $\beta = 2.5$, $\tau = 0.13$, $\tau' = 0.26$, and $w_{fe} = -2.0$. In the centre of the phase plot the ellipsoidal limit cycle can be seen

The decision that was taken to perform this exploration was to vary the weights (in both directions) between a pair of bidirectionally coupled Matsuoka oscillators and look at the phase difference the weights produced. The rest of the parameters of the oscillators were kept constant, in particular $u_0 = 2.05$, $\beta = 2.5$, $w_{fe} = -2.0$, $\tau = 0.13$, $\tau' = 0.26$. Both parameters, w_{12} and w_{21}, were set to values in $[-1, 1]$, with increments of 0.1 units. For every pair (w_{12}, w_{21}), the phase difference of the oscillators is registered after a few cycles needed to phase-lock. Figure 4.8 shows graphically these values for all the combinations of the parameters.

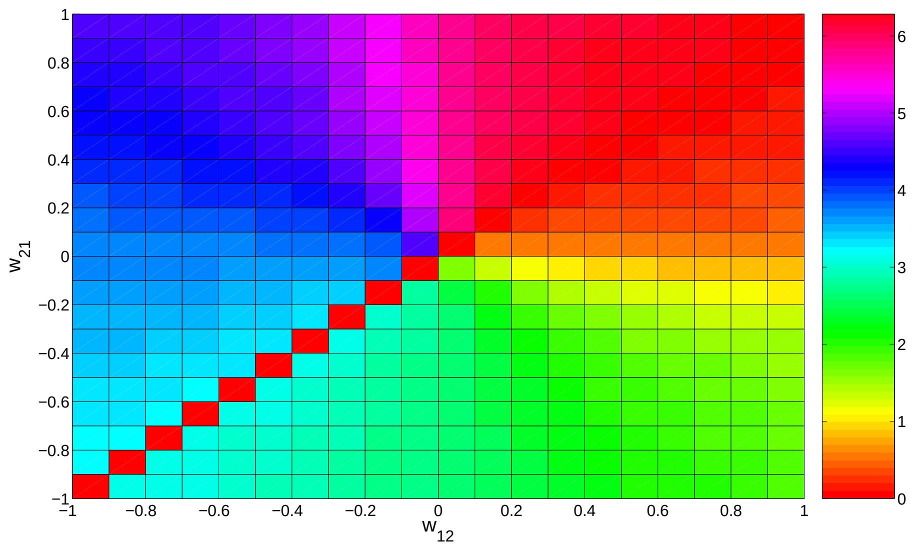


Figure 4.8: Phase difference between two coupled Matsuoka oscillators. The colour of each cell represents the phase difference measured in radians

The plot clearly shows that two bidirectionally coupled oscillators can mutually entrain with any phase difference depending on the exact value of the weights of the couplings. This is useful when developing more complex structures (like the CPG described next) in order to have a deeper idea about the values of the different coupling weights. However, this is not sufficient since not only do these structures have groups of two coupled oscillators, but each oscillator is usually coupled to more than one oscillator. An analytical solution is required at this point. Unfortunately a mathematical study of the phase differences among the oscillators is difficult to perform for the Matsuoka oscillator (as far as we know, no previous studies exist up to now for the Matsuoka oscillator as, for instance, the one described in [BI04] for the Hopf oscillator).

4.2.2 CPG structure

The design that was adopted to build the CPG consists of a network of coupled oscillators, where each oscillator directly controls the output of one articulation and only one (and of course, it affects the output of the other oscillators because of the couplings). There is one oscillator for each hip and one for each knee. This gives a total number of 8 oscillators for every piece of furniture. The couplings of the network are arranged in the following way: every oscillator on the hips is bidirectionally coupled with the rest of the oscillators of the hips, and is also bidirectionally coupled with the oscillator of the knee of the same leg.
Figure 4.9 illustrates this last description more clearly.

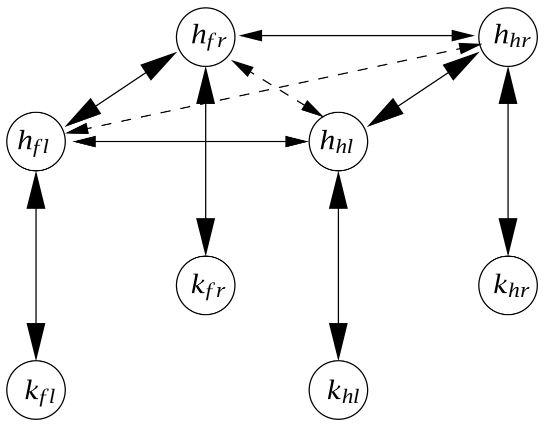


Figure 4.9: CPG network of the stool

The reasons for choosing this structure are quite simple:

- Bidirectional couplings allow to easily have a mutual entrainment between the oscillators in the coupling.[5]
- The couplings of the hips make phase locking possible for the corresponding oscillators, which is needed to reproduce the different gaits.
- The couplings between the hip and the knee in every leg are useful to mutually entrain these two articulations, which is required to get a natural movement (it is logical to think that knee and hip have to synchronise despite the perturbations one articulation can suffer).

[5]Bidirectional couplings have the advantage that can also be used like unidirectional couplings just by setting the weight of the coupling to zero, as it was done sometimes indeed.

- The arrangement of the oscillators corresponds to the mechanical structure of the robots as well. Matching the CPG with the physical configuration will normally produce better results.

It is convenient here to point out here the role of crossed couplings (i.e. couplings between front-right-hip and hind-left-hip, or front-left-hip and hind-right-hip). These couplings, which were configured as excitatory connections (with a positive weight), are not needed to get a particular rhythm between the legs (for instance, a trot or a walk gait), but they make synchronisation between the legs faster. In other words, the transient phases of the oscillators are reduced and the limit cycle is reached more quickly. However, since these couplings are excitatory, the activity that one leg presents affect to its opposite leg by inducing this leg oscillate in the same way the first one. (Obviously this influence does not have to lead to a zero phase difference, but the trend is to *push* the contrary leg to the same oscillatory pattern) And this effect can be inappropriate in some situations where it is better that the contrary leg preserves its natural oscillation (e.g. when overcoming an obstacle (cf. subsection 4.2.6), where the leg that finds the obstacle is lifted to try to avoid it while the other 3 legs should keep oscillating in the same way they did before the obstacle was found, in order to maintain the equilibrium). In conclusion, crossed couplings present a trade-off that must be resolved depending on the desired objective (either faster synchronisation or a better equilibrium in a scenario with obstacles).

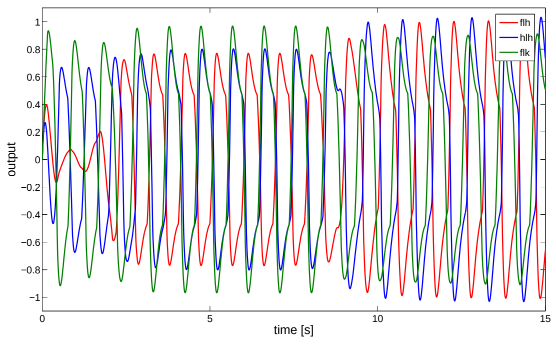
The implementation of the CPG for the simulations was done with a master-slave hierarchy. The output of every oscillator in the CPG is calculated in the master module (which, in particular, is the one that is just above the front left hip) and then the values for the different joints are sent to the slaves at every simulation. The slaves finally send to their actuators the signal received from the master.

Figure 4.10 shows a plot of the oscillations of the described CPG for the trotting configuration explained in subsection 4.2.3 (with no crossed couplings), as well as the results of the simulation in Webots, as a series of snapshots, of the CPG controlling the stool on a flat surface. Videos of this simulation and many others are available at [Arc06].

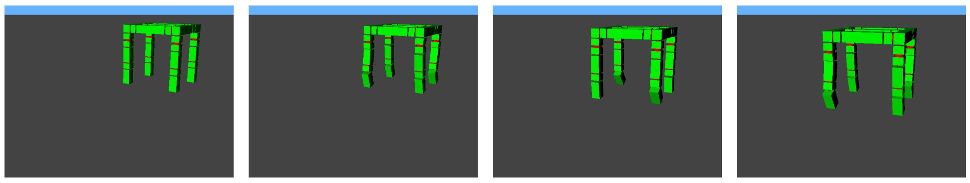
The plot shows that for this particular case (a trotting gait) the front left hip and the front right hip oscillate in anti-phase (i.e. with a phase difference of π radians). Besides, each knee, once the transient phase of the oscillations ends, is *delayed* about $\pi/2$ rad in its phase with respect to the hip on the same leg, so that there is an appropriate synchronisation hip-knee as the one explained in section 4.1. The plot also show the oscillators need quite a time (around 9 seconds) to reach phase-lock behaviour. This transition can be accelerated by increasing the frequency of the oscillator (i.e. the values of τ and τ'), but this is not very convenient for the simulations since there is a point in the frequency spectrum, as commented before, where the mechanics can no longer follow the output of the CPG so the commanded signals are very different from the actual movement of the actuators. Finally, the output of the oscillators is scaled and shifted in order to set the maximum and the minimum of the values sent to the actuators to the interval described for the sine-based controller. In other words, the value for the actuators of the hips is calculated as $y_{osc} \cdot \frac{\pi}{16}$ so that it is within the interval $[-\pi/16, \pi/16]$, while the value for the knees is set at $y_{osc} \cdot \frac{\pi}{16} + \pi/8$ so that it is in $[\pi/16, 3\pi/16]$ (the output y_{osc} of the oscillators is between -1 and 1 approximately).

4.2.3 Exploration of different gaits

All designed pieces of furniture have four legs, so an exploration of how to tune the phase differences of the oscillators in the CPG (and therefore obtain several locomotion gaits) can be performed. One of the ways to change such phase differences is to modify the weights of



(a)



(b) (c) (d) (e)

Figure 4.10: Output of the CPG (front left hip, front right hip and front left knee) with a trotting configuration (no crossed couplings) (a), and snapshots of the simulation of such a controller in Webots (b) (c) (d)(e). The parameters common to all oscillators are $u_0 = 2.05$, $\beta = 3.0$, $w_{fe} = -2.0$, $\tau = 0.1$, $\tau' = 0.3$

the couplings of the oscillators. In particular, three locomotion gaits are proposed here: walk, trot and bound. Unlike the values of the couplings weights, which change depending on the gait, the parameters of the Matsuoka oscillator were the same and kept for all the oscillators and all the gaits: $u_0 = 2.05$, $\beta = 3.0$, $w_{fe} = -2.0$, $\tau = 0.1$, $\tau' = 0.3$.

Table 4.2 and Figure 4.11 illustrate possible combinations of values for such gaits. Additionally, the output of some oscillators of the CPG for the different configurations is depicted in Figure 4.12.

Trot	FLH	HLH	FRH	HRH	→K	K→
FLH	-	−0.4	−0.5	(+0.4)	−0.3	−0.3
HLH	−0.8	-	(−0.4)	−0.5	−0.3	−0.3
FRH	−0.5	(+0.4)	-	−0.4	−0.3	−0.3
HRH	(−0.4)	−0.5	−0.8	-	−0.3	−0.3

Walk	FLH	HLH	FRH	HRH	→K	K→
FLH	-	−0.4	−0.5	-	−0.2	+0.2
HLH	+0.8	-	(−0.8)	−0.5	−0.2	+0.2
FRH	+0.5	-	-	−0.4	−0.2	+0.2
HRH	(−0.8)	+0.5	+0.8	-	−0.2	+0.2

Bound	FLH	HLH	FRH	HRH	→K	K→
FLH	-	+0.5	+0.1	(−0.5)	−0.3	−0.3
HLH	−0.5	-	(−0.5)	+0.1	−0.3	−0.3
FRH	+0.1	(−0.5)	-	+0.5	−0.3	−0.3
HRH	(−0.5)	+0.1	−0.5	-	−0.3	−0.3

Table 4.2: Weights for the trotting, walking and bounding gaits

Improving the gaits

Once the relations of the couplings to produce different gaits were found, it was interesting to improve the gaits in order to maximise the speed of locomotion. This was done for the stool by tuning the amplitude and frequency of the CPG for each gait. Figure 4.13 summarises such exploration.

The plots show that the fastest gait is the bounding gait, for which the stool travels a maximum distance of 10.56 m (which gives a speed of 35.2 cm/s), followed by the walking gait (7.35 m, 24.5 cm/s) and the trotting gait (2.43 m, 8.11 cm/s). It is surprising that the

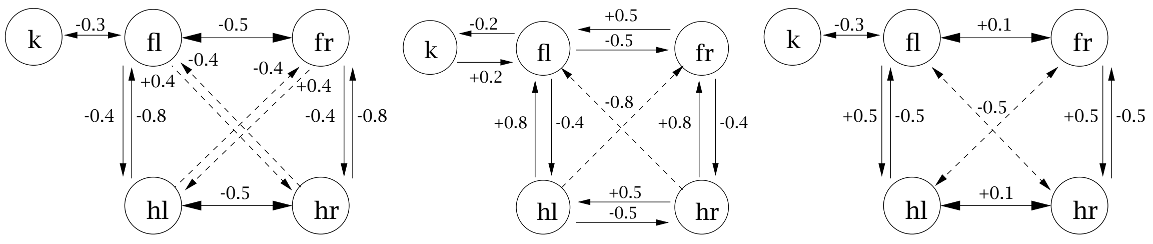


Figure 4.11: Graphical representation of the couplings for trotting, walking and bounding

speed is much higher for walk than for trot, when normally what happens is the opposite. The reason is that the behaviour of the robots in simulation is very sensitive to the arrangement of contacts of the bodies the physics simulation is composed of, and very small changes can lead to rather a different behaviour eventually. For instance, for the walking gait the stool normally move backwards, but for some different values of u_0 or τ the stool walks forwards, despite the parameters are very close to the ones that produce a backward gait. These small variations in the parameters can turn into very different contact points between the bodies so the forces are completely different and so is the locomotion gait.

This problem can be corrected with a smaller step in the algorithm that performs the integration of the equations that describe the physics. A time step of 8 ms was used for this purpose. With smaller values the regularity of the results would be higher, but the drawback is that the simulation becomes really slow (normally, the speed of the simulation with a time step of 8 ms is around 0.150 times real time speed).

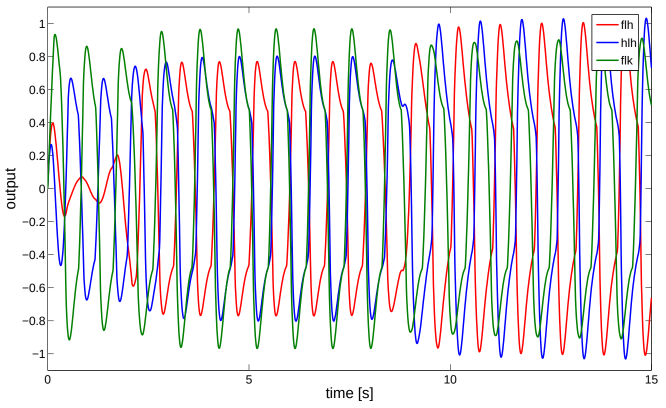
Another problem is that these results may be not exactly reproducible, due to the fact that the simulations with the different parameters were done with a Supervisor node of Webots that sends every set of parameters that define the simulation, sets the furniture to move, collect the results of the case and finally resets the simulation to the initial state and prepares other parameters. It was found that Webots did not reset the simulation correctly, so some variables (such as velocities, acceleration and so forth) might not correctly reinitialise, and therefore there would be some variations during the simulation that would lead to different results. Unfortunately this problem has not been solved yet (in principle, versions 5.1.6 and higher should correct it, but they introduced a new problem in the communication mechanism between robots —Emitter and Receiver nodes— that made simulations impossible to be done).

Concerning the best values for the amplitude and frequency of the oscillations, they are high values for u_0 (around 3) and low values for τ (0.025 and 0.05). However, there are also some exceptions, some points with higher u_0 and/or lower τ whose speed is smaller. This happens mainly because of two reasons: the first one, as commented before, is that for very low values of τ the frequency of the oscillations is so big that the servos cannot complete all the movement they would do with smaller frequencies. The second one is the imprecisions in the integration of the equations and the problems Webots has with regard to the reset of the simulations.

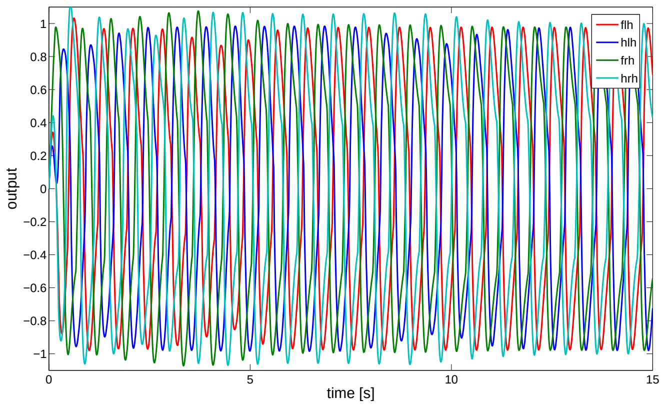
It is also necessary to remark that the results are rather different from one gait to another. While the walking and bounding gait present quite a monotonic increment of the speed as u_0 and τ grow, for the trotting gait there are quite a lot of areas with *valleys* or *peaks*.

4.2.4 Gait transitions

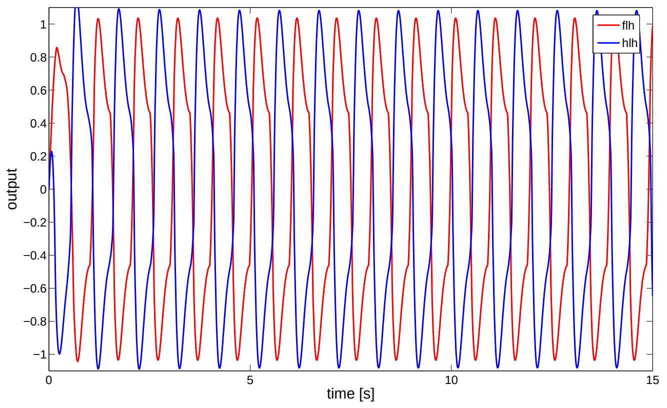
An interesting feature about locomotion that can be implemented is the transition from one gait to another in the middle of the locomotion process. The strategy that was adopted was



(a)

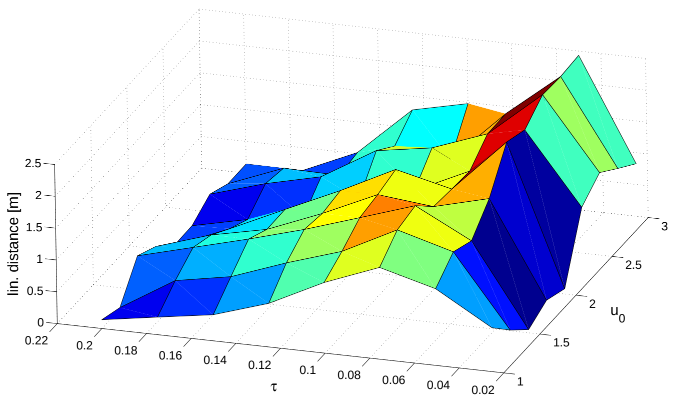


(b)

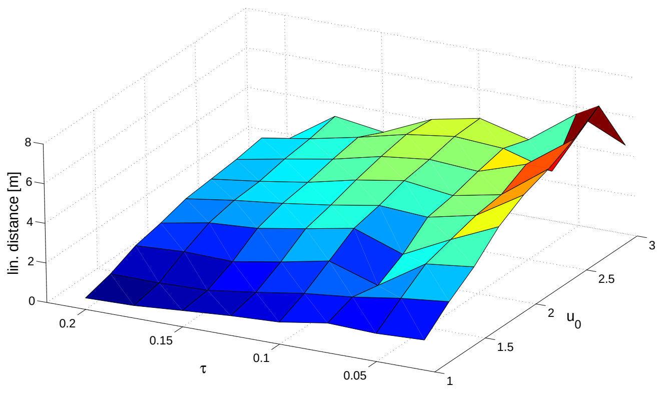


(c)

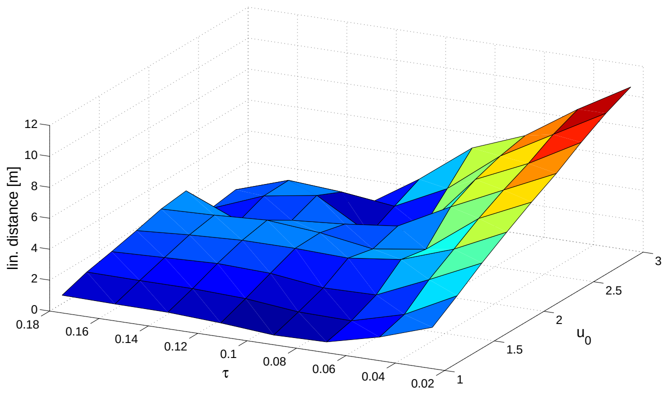
Figure 4.12: Output of some oscillators of the CPG for the trotting (a), walking (b) and bounding (c) gaits (with no crossed couplings)



(a)



(b)



(c)

Figure 4.13: Linear speed vs. amplitude and frequency for different gaits: trot (a), walk (b) and bound (c)

to simply change at a moment of the simulation the weights of the couplings from the values that characterise one gait to the ones of the other gait. The parameters of the oscillators stay the same: $u_0 = 2.05$, $\beta = 3.0$, $w_{fe} = -2.0$, $\tau = 0.1$, $\tau' = 0.3$.[6]

Figure 4.14 shows four plots of the output of the oscillators for two cases: the transition from a trotting gait to a walking gait and the transition from a trotting gait to a bounding gait, and for each one, two different results of doing the transition: with and without crossed couplings.

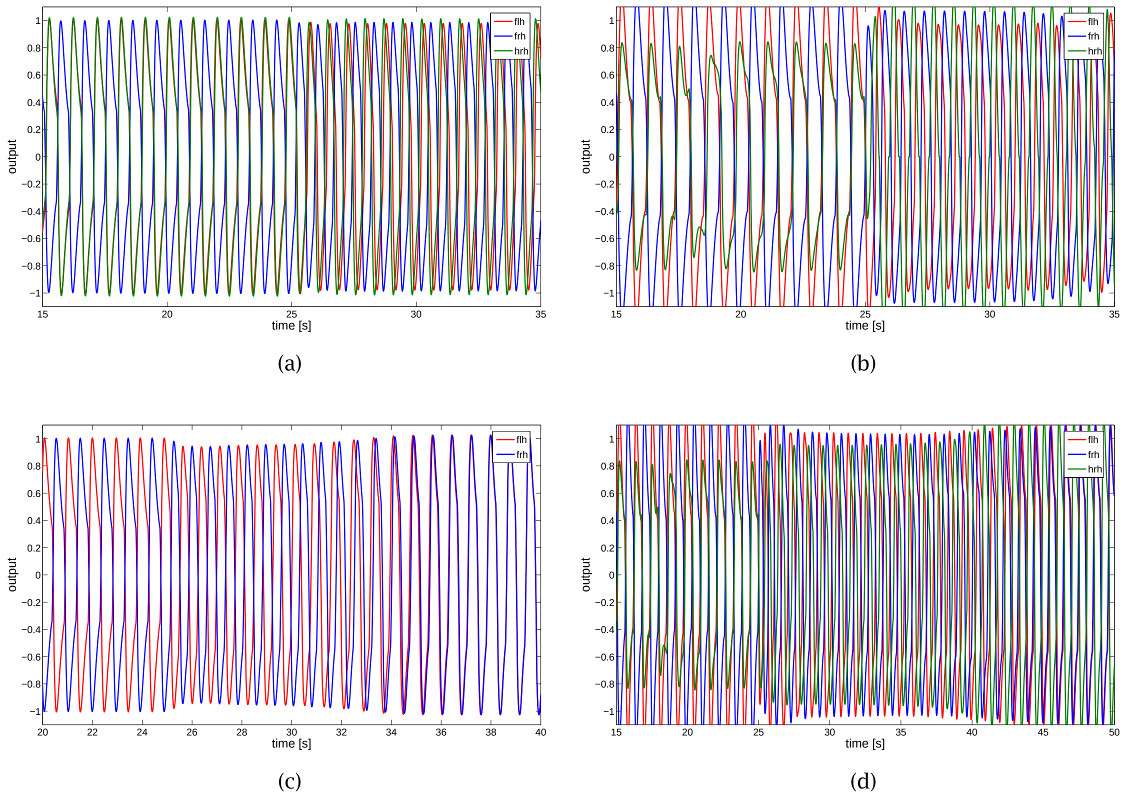


Figure 4.14: Transitions from one gait to another. Output of some oscillators for a trot to walk transition without (a) and with crossed couplings (b) and for a trot to a bound transition, without crossed couplings (c) and with crossed couplings (d)

The plots show that the results are better without crossed couplings. Not only is the output of the oscillators more regular (nearly between -1 and $+1$ always, while for the version with crossed couplings the output of some oscillators decreases to ± 0.8), but the time of the transition is also smaller without crossed couplings. The crossed couplings introduced here are good at the beginning of the evolution of the oscillator, before the limit cycle is reached. However, once the phase-lock happens, crossed couplings make the transitions to another gait slower. Another conclusion is that some transitions have an intermediate phase were the CPG oscillates with the configuration of a gait which is not the one there was before the modification of the couplings or the one that is related to the new couplings. For instance, the transition between the trotting gait and the bounding gait passes for a walking-like gait (not

[6] And also for the rest of experiments unless another thing is indicated.

exactly a walking gait). Therefore the transition from trot to bound is in fact done via kind of a walking gait.

The simulation of the transitions shows that the transient behaviour of the mechanics corresponds as expected with the change in the output pattern of the CPG. In other words, the transition of the output of the CPG matches the transition of the behaviour of the actuators, there is no longer transition for the mechanics.

4.2.5 Control of direction

One of the final objectives of the *Roombots* project is to incorporate into the robots navigation and obstacle avoidance capabilities. Although these features are not studied in this project, a basic ability which is needed to perform such tasks is explored: control of direction. Once control of direction is robustly done, navigation and obstacle avoidance are much simpler.

Two methods to do so are proposed here: modulation of amplitude and rotation of legs. This exploration was done with a trotting gait and $u_0 = 2.05$, $\beta = 3.0$, $w_{fe} = -2.0$, $\tau = 0.1$, $\tau' = 0.3$.

Modulation of amplitude

Control of locomotion by modulating the amplitude of the oscillations is based in the following principle: for legged locomotion (either bipedal or quadrupedal), if the legs on one side, while they oscillate in order for the robot (or person or animal) to move, spend more time touching the ground than the legs on the other side, the robot will eventually turn to the side whose legs are more time in contact with the ground. The reason is quite simple: the force applied on the side the robot turns to is higher than the force applied on the other side. And since the centre of mass of the robot is normally in the middle (such as for the developed pieces of furniture), the effect is a rotation to the side of the strong force.

In particular, what is done is the following: a new variable $turn$ is introduced into the equations of the CPG in such a way that the parameter u_0, which is responsible of the amplitude of the oscillations is affected. With the terms from the couplings not indicated here for simplicity, Equation 4.10 and Equation 4.11 become:

$$\tau \dot{u}_{\{e,f\},i} = \begin{cases} u_0 \cdot |turn| - u_{\{e,f\},i} + w_{fe} y_{\{f,e\},i} - \beta v_{\{e,f\},i} & \text{if } turn < 0 \text{ and } i \in \{FL, HL\} \\ u_0 \cdot |turn| - u_{\{e,f\},i} + w_{fe} y_{\{f,e\},i} - \beta v_{\{e,f\},i} & \text{if } turn > 0 \text{ and } i \in \{FR, HR\} \\ u_0 - u_{\{e,f\},i} + w_{fe} y_{\{f,e\},i} - \beta v_{\{e,f\},i} & \text{otherwise} \end{cases} \tag{4.12}$$

The interpretation of the equations is simple:

1. If the value of $turn$ is within the interval $(-\infty, -1)$, the amplitude of the legs on the left side will be increased and the robot will turn to the right side.

2. If the value of $turn$ is within the interval $(-1, 0)$, the amplitude of the legs on the left side will be decreased and the robot will turn to the left side.

3. If the value of $turn$ is within the interval $(0, 1)$, the amplitude of the legs on the right side will be decreased and the robot will turn to the right side.

4. If the value of $turn$ is within the interval $(1, +\infty)$, the amplitude of the legs on the right side will be increased and the robot will turn to the left side.

5. If $turn \in \{-1, 0, 1\}$, the original amplitude of the robot will not be changed.

Although item 1 and item 3 (or item 2 and item 4) in theory produce the same effect, that is, turning to the left side (or the right side) there may be some differences in the way turning is done in practice, so an experiment that tries both approaches (decreasing and increasing the amplitude) is suitable here.

The experiment that was designed in order to try control of direction by modulation of amplitude consisted on performing several simulations with different values for *turn*. Particularly, *turn* ranged in the interval $[-1.5, 1.5]$, and it was incremented with steps of 0.2 units. Then, the radius with which the stool turns is registered. The radius of the rotation is calculated in the following way: when the stool rotates an angle of $\pi/8$ radians, either clockwise or counterclockwise, the radius is computed as $r = \Delta z/\sin(\pi/8)$, where z is the coordinate of the middle of the segment that joins the front legs along the z axis of the Webots world. Figure 4.15 provides a graphical, clearer explanation on the calculus of the radius.

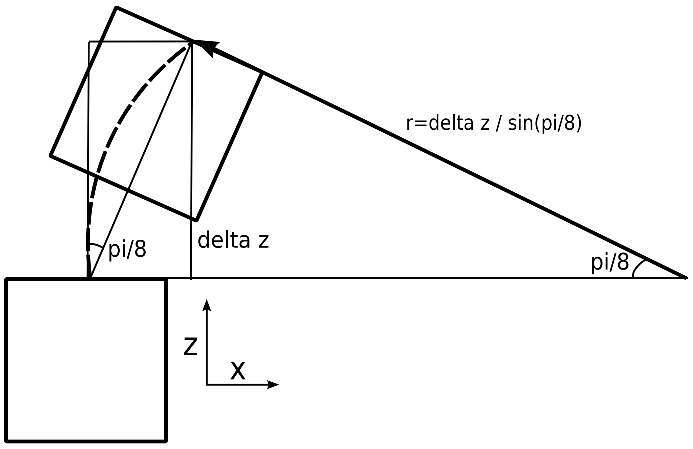


Figure 4.15: Computation of the radius of rotation

The results of the experiment are shown in Figure 4.16.

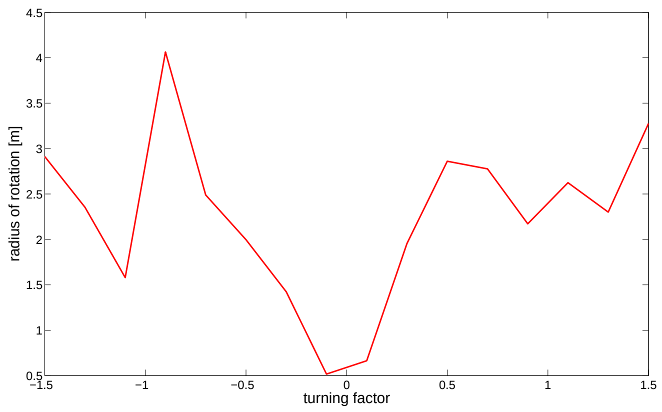


Figure 4.16: Radius of rotation vs. turning factor

Rotation of legs

The other strategy to perform control of direction is to rotate the wheels of the modules located on the hips so that the legs turn as well and oscillate in another plane, which makes the rotation of the whole piece of furniture possible. It is the same mechanism a car uses to turn, for instance. The front wheels rotate (the hind ones stay at the initial position) to the same side the robot will turn to eventually. A snapshot of the way the wheels are turned can be seen in Figure 4.17.

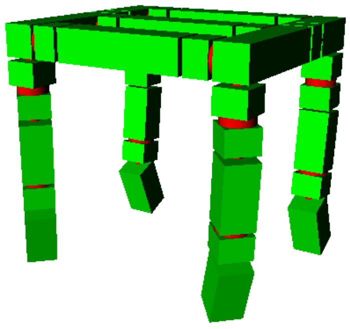


Figure 4.17: Rotation of the wheels to control direction

The same experiment as for the case of modulation of amplitude was carried out, although this time the legs were rotated instead of changing the amplitude of the legs on one of the sides. The angle of rotation of the wheels took values in $[-\pi/4 \text{ rad}, \pi/4 \text{ rad}]$, with increments of $\pi/16$ radians. Finally, the radius of rotation is registered as before. Figure 4.18 shows the results.

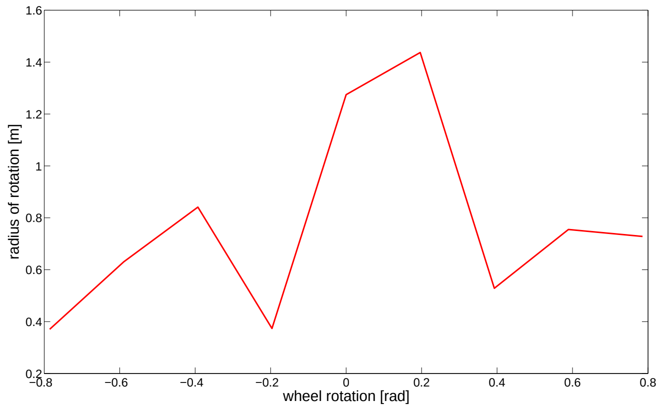


Figure 4.18: Radius of rotation vs. rotation angle of the wheels

Analysis

From the results of the experiment two main conclusions arise. First of all, both mechanisms are suitable to control the direction of locomotion. If the parameters that are used for this purpose (*turn* and the angle of rotation of the wheels) are given proper values, the furniture is able to turn with a larger or smaller angle of rotation. Secondly, the rotation of the wheels is a better method than the modulation of amplitude, since it gives much smaller radii of rotation and with a more regular behaviour (lower radii for values close to $\pm\pi/4$ and higher for values close to 0).

The results for modulation of amplitude are not so regular, especially in the intervals below -1 and above $+1$ (therefore the strategy of increasing the amplitude is not very good; the best results are obtained when the amplitude of the oscillations on one of the sides is decreased). For modulation of amplitude, the range $[-0.5, 0.5]$ turns to be good for achieving a small radius of rotation, with results similar to the rotation of the legs.

Another important point to notice the slight asymmetry for values of the parameters with different sign but same absolute value. This is due again to the particularities of the contacts of the robot, and also to the bug of Webots concerning the resetting of the simulations, because in theory there should be a symmetry in this point.

4.2.6 Locomotion on irregular surfaces

The last feature that was implemented within the framework of the CPG-based controller was the ability of locomotion on surfaces which are not even. The floor of Learning Centre at the EPFL, where the *roombots* will be located, is not completely flat. Therefore, in order to produce stable locomotion gaits for such a scenario it is necessary to study how this adaptation to the terrain can be done.

Animals are able to modulate their locomotion gaits in order to adapt to the terrain with an incredible effectiveness. They are able to walk not only on slopes or rugged surfaces, but also on surfaces with very different heights and a very small area (e.g. when climbing a tree), and of course easier surfaces such as stairs and so on. The mechanism that allows animals to modulate the locomotion gait depending on the external conditions is sensory feedback [Gri85]. The neural networks in the spinal cord are modulated with external signals coming from the senses (for control of locomotion the most important one is equilibrium). Sensory feedback is an essential complement to the intrinsic rhythms produced by central pattern generators in order to have robust locomotion.

Robotics is still far from such an extremely precise modulation of locomotion, but there are some examples which successfully adapt to the properties of the terrain. One of the most sophisticated robots is Tekken (cf. subsection 2.2.4), which is able to control the equilibrium on uneven surfaces thanks to two reflexes called *vestibulospinal reflex/response* and *tonic labyrinthine response for rolling* respectively.

The approach Tekken uses is in fact the one implemented here, with some modifications. Three reflexes are incorporated into the equations of the CPG. One is in charge of quickly flexing a leg when that leg hits an obstacle, with the goal of overcoming it. One controls the flexion and extension of the legs depending on the inclination of the furniture around the horizontal axis which is parallel to the direction of the movement (roll angle), and the last one does the same as the second one but with respect to the horizontal axis which is perpendicular to the direction of the movement (pitch angle). Figure 4.19 gives a graphical explanation on these two reflexes, as well as shows the distribution of the touch sensors that make the detection of obstacles possible.

The principles the reflexes are based on are the following ones:

- When an obstacle is found, it is necessary to quickly lift the leg that hit the obstacle and pull it down quickly too, because the obstacle is an irregularity with an abrupt change of the height of the terrain, so this strategy would provide in principle a good response to these events.

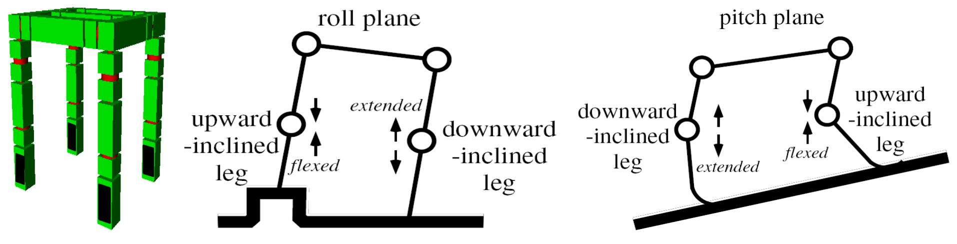


Figure 4.19: Arrangement of touch sensors for obstacle detection (a) and description of rolling (b) and pitch (c) angles (last two drawings from [FKC03])

- Concerning the two reflexes that depend on the inclination of the piece of furniture, the idea is that a leg must be flexed (or extended) if the height of the surface where that leg lands is higher (or lower) than the height of the centre of mass of the robot. This clearly provides a better equilibrium.

The new equations of the CPG introduce three feedback terms. Equation 4.10 and Equation 4.11 are replaced by:

$$\tau \dot{u}_{i,\{e,f\}} = u_0 - u_{i,\{e,f\}} + w_{fe} y_{i,\{f,e\}} - \beta v_{i,\{e,f\}} + \sum_j w_{ij} y_{j,\{e,f\}} + fb_{i,\{e,f\}} \tag{4.13}$$

$$fb_{i,f} = -fb_{i,e} \tag{4.14}$$

$$fb_{i,e} = fb_{i,obs} + fb_{i,roll} + fb_{i,pitch} \tag{4.15}$$

Where

$$fb_{i,obs} = \begin{cases} -g_{obs} & \text{if obst. found in } [t - t_{obs}, t] \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

$$fb_{i,roll} = \begin{cases} -g_{roll} \cdot \alpha & \text{if } \alpha > 0 \text{ and } i \in \{FL, HL\} \\ g_{roll} \cdot \alpha & \text{if } \alpha < 0 \text{ and } i \in \{FR, HR\} \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

$$fb_{i,pitch} = \begin{cases} -g_{pitch} \cdot \beta & \text{if } \beta > 0 \text{ and } i \in \{FL, FR\} \\ g_{pitch} \cdot \beta & \text{if } \beta < 0 \text{ and } i \in \{HL, HR\} \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

The values α and β are the roll and pitch angles respectively. The variables g_{obs}, g_{roll} and g_{pitch} are, respectively, gains for the obstacle feedback, roll feedback and pitch feedback. t_{obs} is the time the obstacle feedback is applied after the leg hits the obstacle. In other words, when a leg reaches an obstacle it needs the feedback to be present for some time in order to be lifted. With this parameter this is achieved for at least t_{obs} milliseconds, even if the obstacle is no longer detected.

This feedback is not applied to all oscillators. Only the oscillators in the hips receive the input caused by the detection of an obstacle or the inclination of the robot. Empirically this was confirmed enough to properly modify the oscillations of the legs, while the application of the feedback to both hips and knees led to an important loss of stability due to the fact that the rotation of the knees was too large and sometimes the lateral faces of the lower modules landed on the ground instead of the base of the modules, and then it was impossible to lift the leg again.

Although normally a leg that lands below the centre of mass should be extended, Equation 4.17 and Equation 4.18 only cause the flexion of the legs (for instance, if the robot is inclined to the left, the legs on the right will flex more, but the legs on the left will keep their ordinary oscillation). The reason this was done this way is that in the simulations the extension of the legs did not work very well, and moreover it was enough to flex the legs to get a stable gait, so there was no need to extend them.

Modulation of locomotion on slopes

Two experiments were performed to find the best values for g_{obs}, g_{roll}, g_{pitch} and t_{obs}. The first one consisted on setting the stool to walk (with a trotting gait actually) on slopes with an inclination of 0.1 radians ($\approx 5.73°$) with four different orientations (Figure 4.20 shows a screenshot of one of the orientations of the slopes):

- Inclination in the roll plane with the right side of the ground higher than the left side
- Inclination in the roll plane with the right side of the ground lower than the left side
- Inclination in the pitch plane due to a slope that goes up
- Inclination in the pitch plane due to a slope that goes down

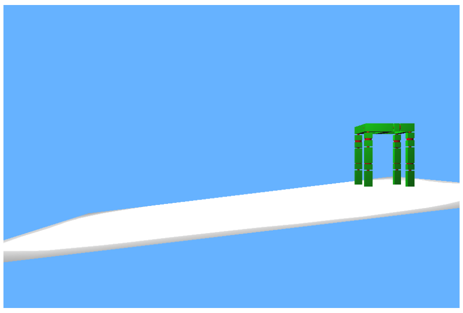


Figure 4.20: Screenshot of one the Webots worlds for the experiment of the slope

The first two surfaces are intended to find the best value for g_{roll}, the experiment with the surfaces that go up and down tries to find the best value for g_{pitch}. One experiment is done for each surface. For the surfaces inclined in the roll plane, several simulations are performed with values of g_{roll} ranging in the interval $[0, 20]$, with increments of 0.4 units, and $g_{obs} = g_{pitch} = 0$. The same is done for the surfaces inclined in the pitch plane, but varying g_{pitch} and setting g_{roll} and g_{obs} to zero.

After 30 seconds of simulation of locomotion on the slope, the following information is registered:

- Total distance the stool travels in a straight line from the starting position to the final position
- Distance along the z axis (the stool begins the simulation *looking* at the positive part of the z axis, the direction it would follow if it did not turn)

- Orientation around the vertical axis at the end of the simulation (0 if the stool finished the simulation in the same orientation it had at the beginning of it)

In addition to this information, the position of the stool (actually the position of the module above the front left hip) on the horizontal plane is stored every 296 milliseconds, so that the trajectory the stool followed can be reconstructed afterwards.

Figure 4.21 shows the three plots that summarise the experiment for the slope inclined to the left side, while Figure 4.22 shows the results for the slope inclined to the right side, Figure 4.23 illustrates the results for the ascending slope and Figure 4.24 for the descending slope.

The experiments show this kind of sensory feedback is useful to control the trajectory on slopes, in any of the four studied inclinations. Good values for both g_{roll} and g_{pitch} are around 10.0, where there is an interval where the stool turned very little and covered quite a distance when walking on the slope.

Another conclusion is that the influence of the gains to the final trajectory is not very clear, since often for close values of one of the gains the resulting trajectories are rather different, and even have opposite directions (this last point is frequent for low values of the gains, with high values the trend is to have roughly the same direction for different values of the gains which are close). However, within the interval $[8, 10]$ (approximately) the resulting trajectory is often close to a straight line.

One last point that has to be commented is that the feedback pathways make the oscillators entrain faster, the limit cycles are more quickly reached (tipically in 4 seconds approximately). This property suggests they should be used even on flat surfaces in order to get a better performance.

After the experiment was done, it was decided that the best combination of values was $g_{roll} = g_{pitch} = 9.2$. These gains were tried together on the four slopes and in some more with more inclination and the results were also good (even better than when using only one gain). Therefore the conclusion is that the implemented feedback pathways that control the inclination of the robots on the roll and pitch planes are appropriate to walk on slopes. Perhaps some different relation (e.g. quadratic, logarithmic or polynomial, the one used here is lineal) between the angles of inclination and the feedback applied is better than a lineal dependence, but the alternative that was implemented seemed to be good enough.

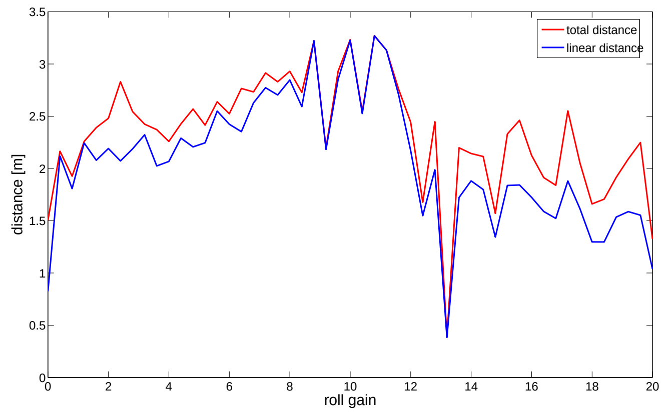
It is also interesting to compare the output of the oscillators when no feedback is used and when g_{roll} and g_{pitch} are given values such as $g_{roll} = g_{pitch} = 9.2$. Figure 4.25 shows the differences of these two cases. The world where the output was taken was the one with the descending slope.

Basically, as shown in the plot, the effect of the feedback is to *shift* the output of the oscillators so that the legs flex more and also to decrease a little the amplitude of the oscillations, in order for the stability of the robot to be higher. The plot also shows that the transient phase of the CPG is shortened to 4 seconds roughly (this phase correspond to the first oscillations with very low amplitude).

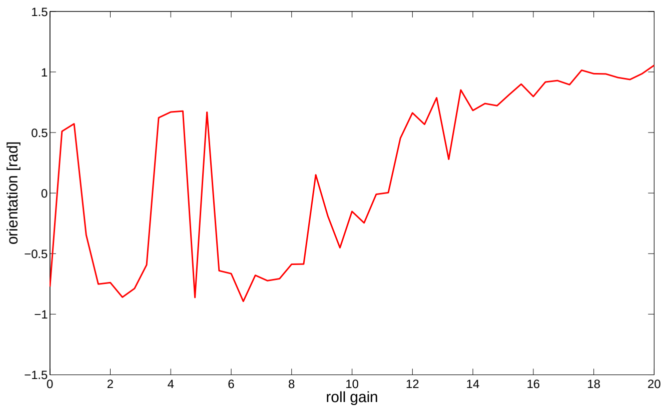
Overcoming obstacles

The other experiment that was performed consisted on the study of the best values for the gain g_{obs}. The scenario shown in Figure 4.26 was the world used for such an exploration.

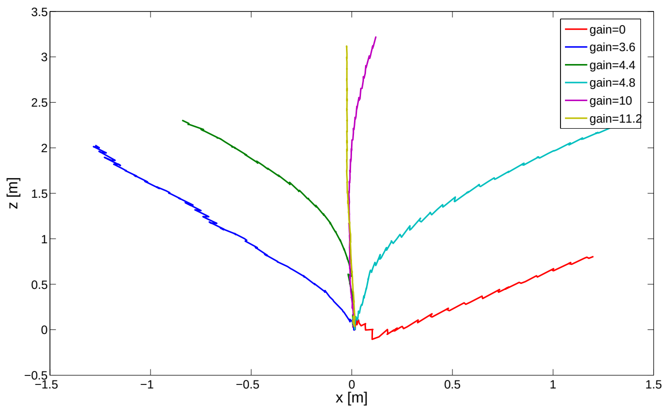
This world is a 5 m $\times$ 5 m grid which is divided in tiles of 1 m $\times$ 1 m each. Each tile is a flat square, but the heights of the different tiles are not the same from one tile to another. The purpose of this grid of tiles with different tiles is to have abrupt changes in the height of



(a)

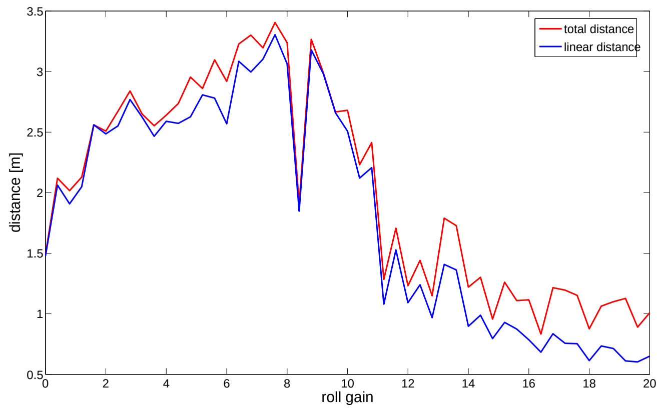


(b)

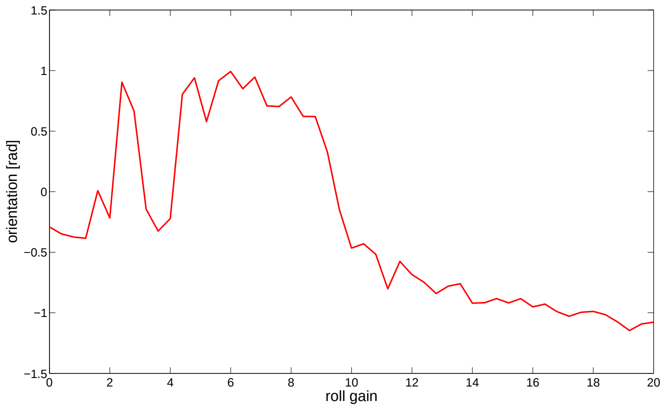


(c)

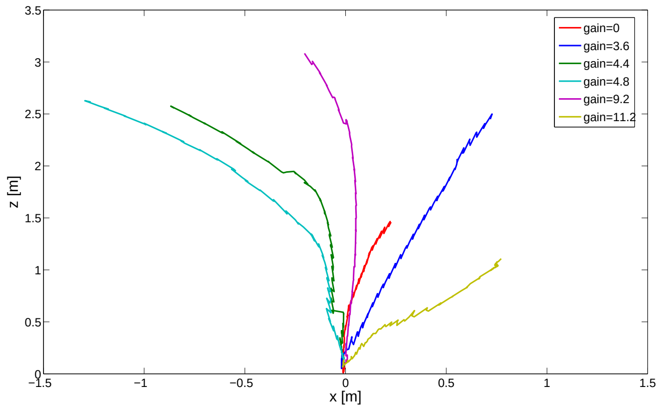
Figure 4.21: Results for the experiment with the slope inclined to the left: distance vs. roll gain (a), rotation vs. roll gain (b) and trajectory for some roll gains (c)



(a)

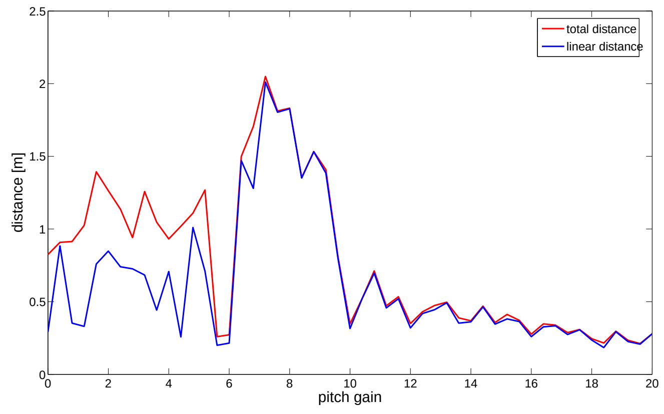


(b)

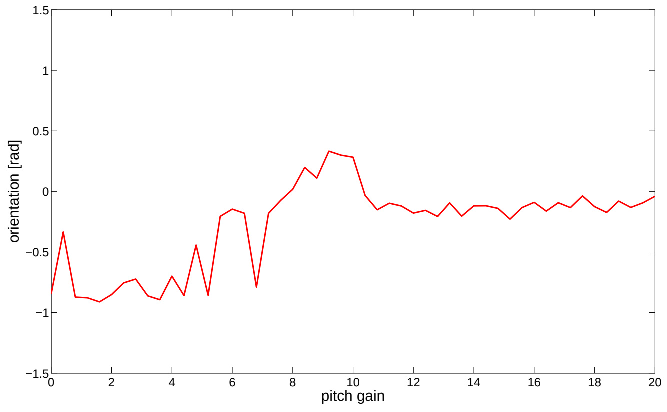


(c)

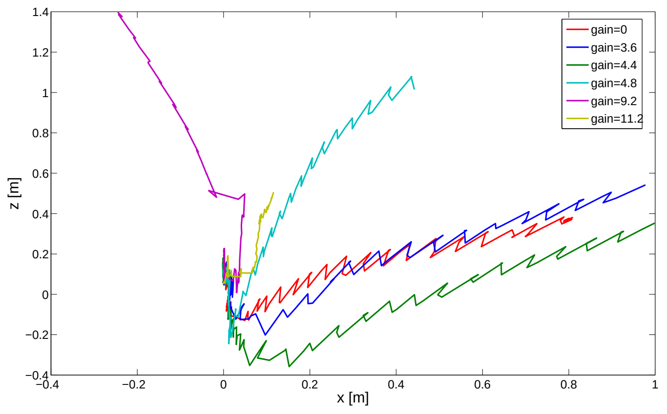
Figure 4.22: Results for the experiment with the slope inclined to the right: distance vs. roll gain (a), rotation vs. roll gain (b) and trajectory for some roll gains (c)



(a)

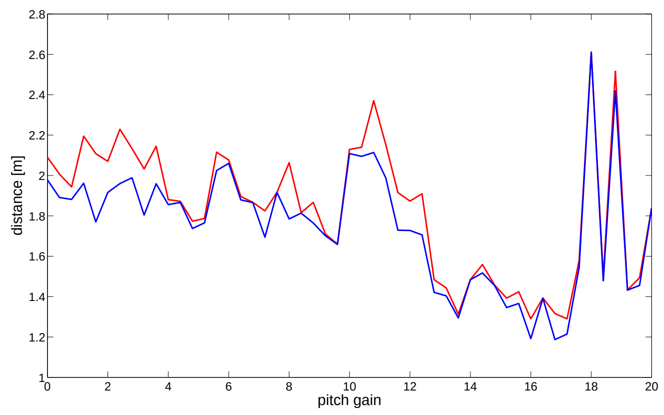


(b)

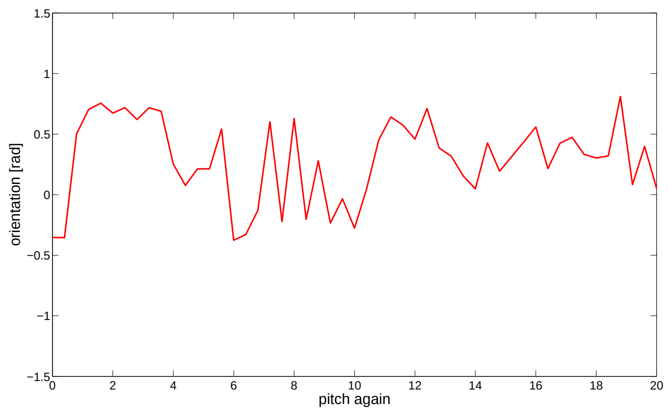


(c)

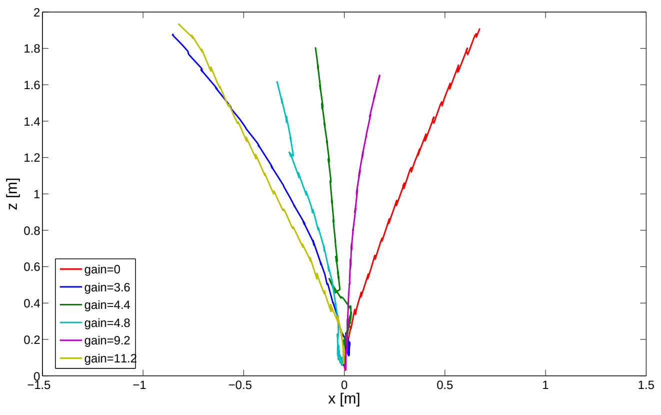
Figure 4.23: Results for the experiment with the ascending slope: distance vs. pitch gain (a), rotation vs. pitch gain (b) and trajectory for some pitch gains (c)



(a)



(b)



(c)

Figure 4.24: Results for the experiment with the descending slope: distance vs. pitch gain (a), rotation vs. pitch gain (b) and trajectory for some pitch gains (c)

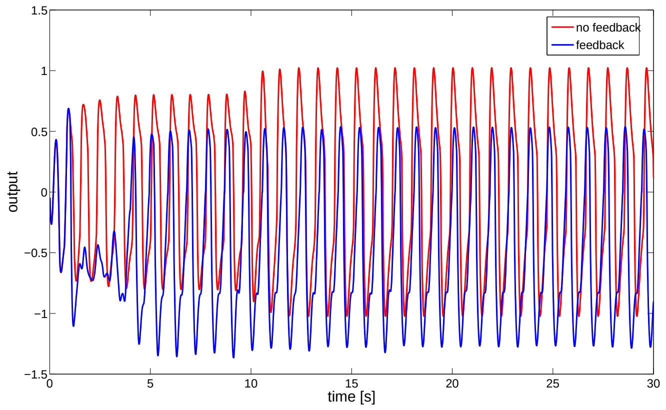


Figure 4.25: Output of the oscillator in the hind left hip on the descending slope, with and without feedback. The feedback is 0 during the first second in order to avoid setting any feedback this time where there are too strong oscillations of the CPG

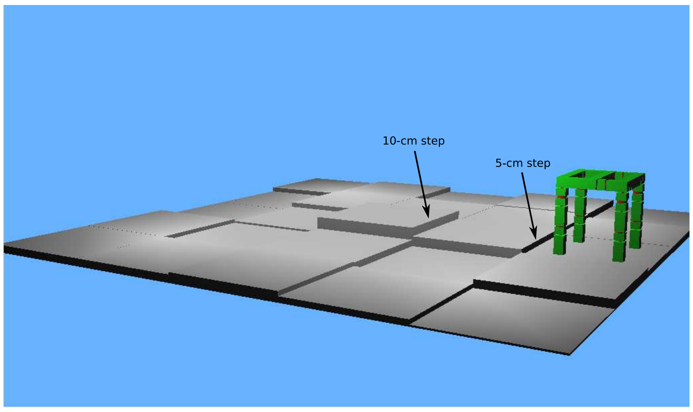


Figure 4.26: Steps world to overcome obstacles with sensory feedback

the terrain at some points, so that if an obstacle was found the obstacle feedback would be applied.

In particular, for the experiment the stool was placed in the middle of one of the tiles (cf. Figure 4.26) and the trotting gait was started. The robot, if it keeps a straight locomotion, faces first one step of 5-cm height and then, if this step is successfully overcome, a step of 10 cm. If the robot, for any reason, turned, it would find soon or later a change in the height of the terrain, either a positive increment (a step) or a negative one (a hole). The additional tiles are also interesting to check the response of the robot to these variations of the terrain.

The variable g_{obs} took values in the range $[0, 20]$, with increments of 0.4 units. One simulation is run for every possible value of g_{obs}. The parameter t_{obs} was set to 50 ms, because in preliminary simulations it turned to be a good value, which, for high values of g_{obs}, was low enough to cause the legs to quickly go forth and back when an obstacle was detected, and high enough to produce the effective overcoming of the obstacle.

At the end of the 25 seconds each simulation lasts, the same information that was taken into account for the experiment of the slope is registered (total distance, distance along the z axis, orientation of the robot at the end of the simulation and position of the robot every 296 ms.

Since the stool is going to pull up (or down) its legs to try to overcome the steps (or holes), its inclination in the roll plane and pitch plane may be modified when a change of height occurs, therefore it would be better for the equilibrium to apply the reflexes to control the inclination as well. These two refllexes are indeed applied, and in particular with the best values found in the previous exploration ($g_{roll} = g_{pitch} = 9.2$).

Figure 4.27 contains the plots that summarise the result of the exploration of the values of the obstacle gain.

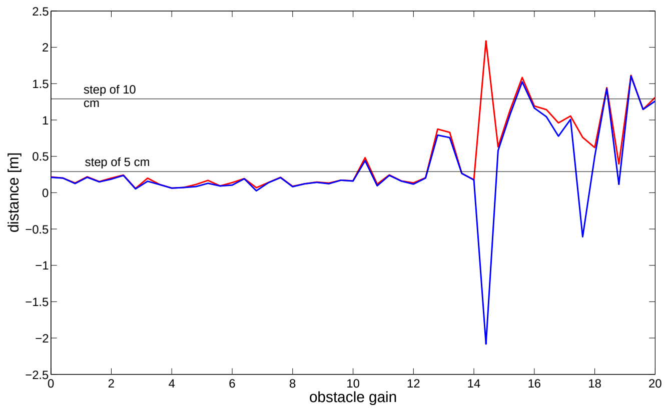
The conclusions after the experiment are the following ones. First of all, in order to overcome a step a very high gain is needed. The best value found for g_{roll} and g_{pitch} was 9.2, but in order to reach a step of 5 cm the obstacle gain must be above 10.

For higher values of g_{obs} the first step is overcome most of the times, although the stool turns quite a lot after going over it, in one or the other direction (this can be deducted from the saw teeth of the plot of the orientation for high values of the gain). The second step is much more difficult to manage. It is 10-cm height (5 cm higher than the first step) and is impossible to overcome for low gains. However, for $g_{obs} > 15$, the possibility of overcoming the second step becomes feasible, and in fact it is overcome for $g_{obs} = 15.6$, $g_{obs} = 18.0$ and $g_{obs} = 19.2$. For these cases the robot keeps an orientation which is close to 0 (for most of the cases, in the interval $[-0.2\text{ rad}, 0.2\text{ rad}]$).

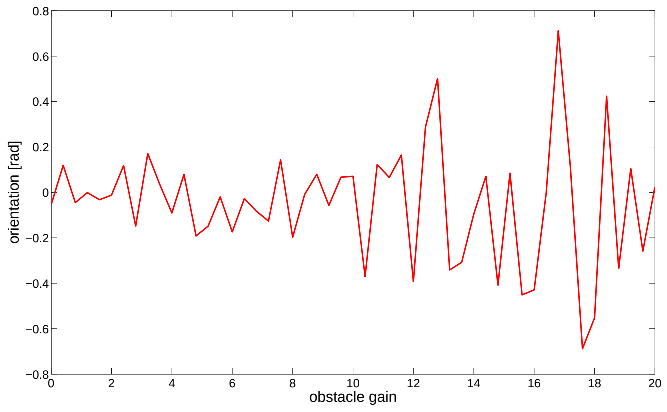
The second step is anyway difficult to deal with always, because it takes the stool to the limit of its equilibrium. When the robot lifts a leg in order to overcome the step, even with high obstacle gains and the reflexes to control the inclination, the equilibrium of the robot is normally lost when the lifted leg lands on the surface after avoiding the step and the result is that it falls down (one extreme case is the result for $g_{obs} = 14.4$, for which the stool ends 2 meters behind the initial position after falling).

As well as for the feedback associated with the roll and pitch angles, it is illustrative to show how the output of the oscillators is affected when an obstacle is detected. Figure 4.28

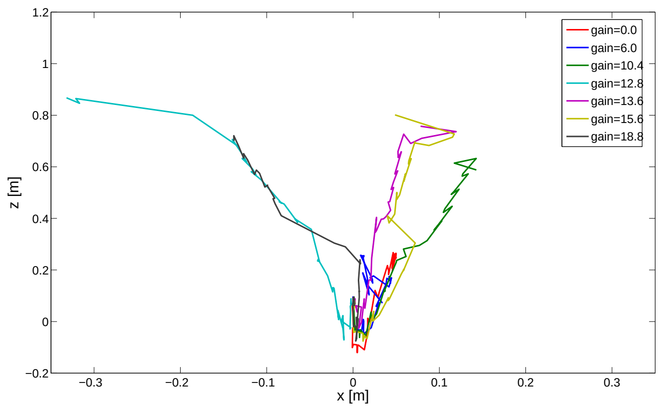
The plot shows how the CPG oscillates with the normal pattern until an obstacle is detected, which involves such an abrupt change in the feedback that the oscillator to which it is applied immediately *learns* by adapting its output to the input signal and then, once the feedback disappears, the output of the oscillator gets back to the ordinary values. The activations of the feedback in the plot correspond to the moments where the step is detected: first by the front right leg, then by the front left leg and finally by the hind right leg and the hind left leg (the step is overcome therefore).



(a)



(b)



(c)

Figure 4.27: Results for the experiment with the steps: distance vs. obstacle gain (a), rotation vs. obstacle gain (b) and trajectory for some obstacle gains (c)

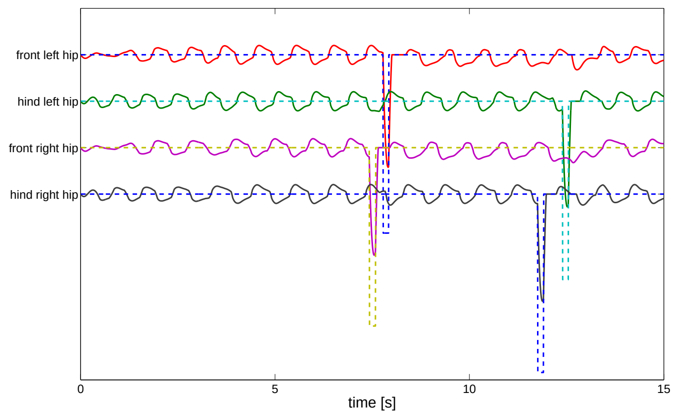


Figure 4.28: Output of the oscillators of the hips for the experiment of the steps (g_{obs} = 19.2, g_{roll} = 9.2, g_{pitch} = 9.2). With solid lines is the ouput of the oscillators, and with dashed lines the obstacle feedback which is applied when an obstacle is detected

The main conclusion of control of locomotion on irregular surfaces is that the implemented reflexes are good enough to control the inclination of the robot and also to deal with not-too-high obstacles (from 10 cm onwards the task of maintaining the equilibrium is really complicated). Some different feedback pathways should be studied in order to improve the response to higher obstacles, if possible (or perhaps it has more to do with the mechanical structure of the robot).

4.3 Rolling locomotion

The wheels the modules have make locomotion by rolling possible for the pieces of furniture. Of course, these pieces of furniture must have a structure with wheels on the feet, making contact with the ground. Then, it is just necessary to rotate the wheels and the furniture will move like a wheeled vehicle. Control of direction is also very simple. The wheels on the top of every leg can be rotated a particular angle and the robot will turn with this angle too.

From the point of view of research this approach is not very interesting since it is straightforward to implement, but *Roombots* is also a project that aims to build useful robots, and moreover rolling locomotion, when possible, requires much less energy to achieve the same results (e.g. moving from one point to another) that legged locomotion, so the use of rolling locomotion is absolutely justified.

The drawback rolling locomotion has is that it does not work as well as legged locomotion on irregular surfaces (for instance, dealing with steps is practically impossible only by using wheels). Besides, it needs really strong servomotors in order for the small box/wheel part of the modules to hold the weight of the whole structure.

A screenshot of the rolling locomotion for the stool is shown in Figure 4.29. Additionally, a video of the simulation can be found at [Arc06].

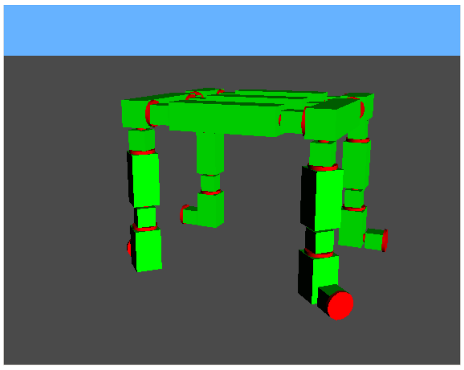


Figure 4.29: Screenshot of the simulation of rolling locomotion in Webots on a flat surface

4.4 Online optimisation of locomotion

The last topic that was addressed in this project was online optimisation of locomotion. Its goal is to improve the locomotion gait of a robot (or a modular robot in this case) dynamically at the same time the robot moves. In other words, there is no preexisting fixed control algorithm for the locomotion of the robot, but it has to learn how to move online. The learning process will be influenced by the characteristics of the robot and the external environment at the moment the optimisation is done, so the gait the robot learns is adapted to these particularities.

Online optimisation is useful for the *Roombots* project because the users could define new pieces of furniture (or, generally, multi-unit structures) for which there would be no previous control strategy, and can also be required for the locomotion of small multi-unit structures used, for instance, to transport modules from one point to another as part of a reconfiguration sequence. Online optimisation can be a mechanism to perform locomotion on irregular surfaces as well, since it is intrinsically an adaptive method.

As far as we know, the only research on online optimisation of locomotion has been done at BIRG by Daniel Marbach [MI05]. His research consisted on the optimisation of the phase differences between the oscillators composing the CPG that controls the locomotion of a modular robot, with the objective of getting a stable and fast locomotion gait. For this purpose he used the Powell's method for multidimensional optimisation. This algorithm performed very effectively since, for instance, it was able to find a fast locomotion gait for a modular robot composed of six blocks in less than 30 minutes in real time.

The method chosen for the online optimisation of the roombots was not the Powell's method, but the downhill simplex method in multidimensions, also known as the Nelder-Mead method [NM65]. Both methods try to minimise (or maximise) a multidimensional objective function, that is, a function with more than one independent variable, with evaluations of such a function. In theory, Powell's method requires a lower number of evaluations than the simplex method, but normally (also in theory) the simplex method gets better results in terms of the value of the objective function.

4.4.1 Description of the method

The simplex method, for n dimensions, uses a *simplex*, which is a set of $n + 1$ n-dimensional vectors, each of them defining a set of parameters for the objective function. Geometrically, a simplex of one dimension is a segment of a straight line, while for two dimensions is a triangle and for three dimensions is a tetrahedron.

At the beginning the method evaluates the objective function at these $n + 1$ points at then it performs a series of transformations of the points until it gets to a local (possibly global) optimum, where the algorithm can be reinitialised in order to *get out* of the local optimum or stopped if the value of the optimum is good enough. There are four transformations that are carried out in order to optimise the objective function. The simplex is transformed step by step with one of these transformations:

Reflection : The point with the worst value is *pushed* to the opposite side where it is in a factor of -1 (being the centre of the transformation the geometrical made of the simplex).

Reflection and expansion: If after the reflection of the worst point, the value of the new resulting point is better than the best value previously found, an extrapolation of factor 2 is performed along the same direction the reflection was done.

Contraction: If the reflected point is worse than the second best value, then the reflected point is discarded and the original one (the worst point) is contracted a factor of 0.5 to the centre of the simplex.

Multiple contraction: If after the three previous transformations the value of the worst point does not improve, then a multiple contraction of factor 2 for all the points is performed towards the best point (the best point is not changed).

Figure 4.30 illustrates graphically the four possible transformations.

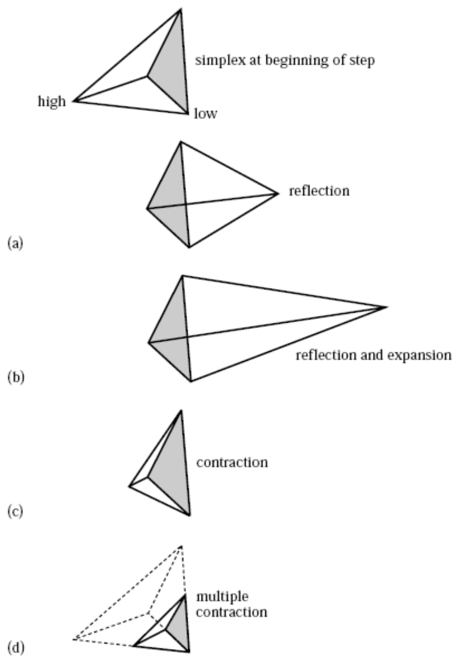
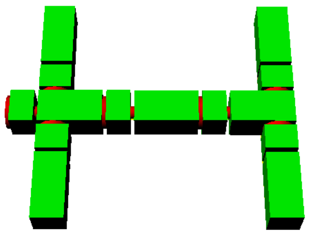
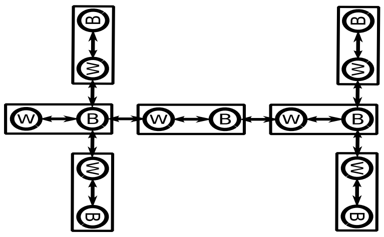


Figure 4.30: Graphical representation of the transformations of the simplex method (from [PTVF92])

4.4.2 Design of the experiment

The experiment that was designed tried to optimise the locomotion speed of arbitrary robots (not only the pre-designed pieces of furniture) made out of several attached wheeled modules. The robots are controlled by a distributed CPG (i.e., a CPG where every module integrates the equations of the CPG corresponding to their oscillators). Each module implements two Matsuoka oscillators, one per DOF, with parameters $u_0 = 1.2$, $\beta = 6.0$, $w_{fe} = -2.0$, $\tau = 0.3$, $\tau' = 0.3$, and which are bidirectionally coupled with weights ranging in $[-1, 1]$. In addition to the intra-module couplings, there are also inter-module couplings, between those modules that are attached. Figure 4.31 shows the arrangement of the couplings for a salamander-like robot.


(a)



(b)

Figure 4.31: Distribution of the couplings for a salamander-like robot. External structure (a) and coupling arrangement (b)

The algorithm, therefore, uses the couplings of the CPG ($4n-2$ values for a n-module robot) as parameters to optimise. First, these couplings are initialised with random values in $[-1, 1]$ and then, they are evolved depending on the progress of the algorithm. Each evaluation of the simplex method performs a simulation for 10 seconds after which the average distance traveled by the modules (which is equivalent to the average speed since the duration of the simulation is constant) is registered.

4.4.3 Results and analysis

Three different robots were optimised with the simplex method: a salamander-like robot (cf. Figure 4.31), a worm-like robot and a starfish-like robot (cf. Figure 4.32).[7]

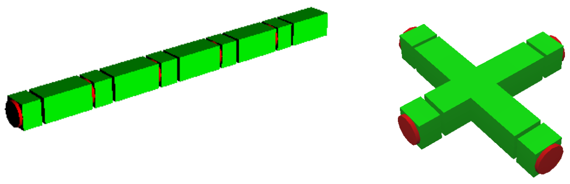


Figure 4.32: Structure of the worm and the starfish robots

For each robot, three different runnings of the algorithm were launched, each one with random, different initial weights for the couplings. The performance of the optimisation

[7]A little imagination is required to see such animals in the robots, especially for the starfish.

algorithm largely depends on the position of the search space were the algorithm begins. If the initial point is near a good combination of parameters, then the simplex method will probably get to this combination in short time. On the other hand, if the initial conditions place the algorithm in a bad region of the space, it will take a large amount of time to find it or, worse, a good set of parameters may not be found.

Three screenshots showing the characteristic gaits learned by the robots in the simulations can be seen in Figure 4.33, while the results of such simulations are summarised in Figure 4.34, Figure 4.35 and Figure 4.36.

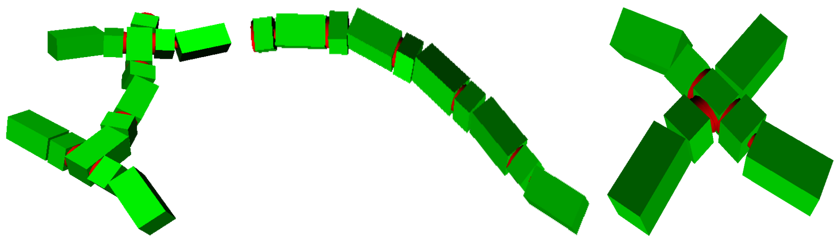
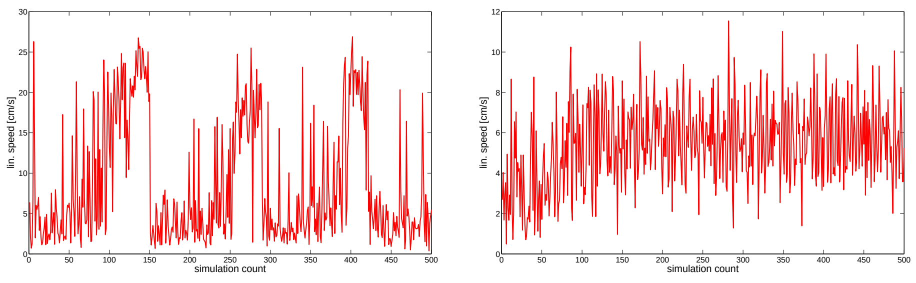
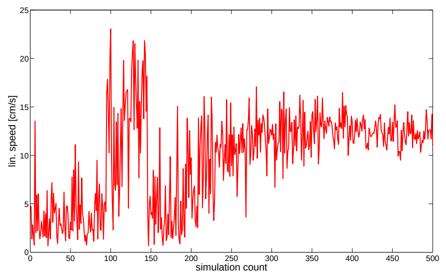

(a) (b) (c)

Figure 4.33: Screenshots of the simulations of the online optimisation method: salamander-like robot (a), worm-like robot (b) and starfish-like robot (c)



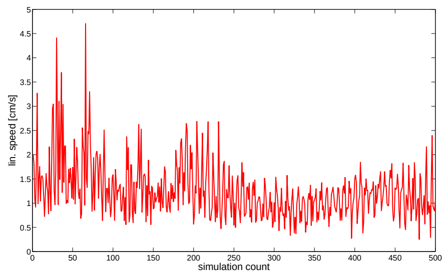
(a) (b)



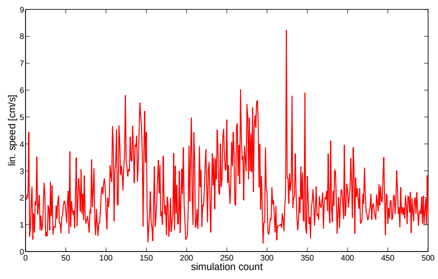
(c)

Figure 4.34: Results of the simulation of the salamander-like robot. First simulation (a), second simulation (b) and third simulation (c)

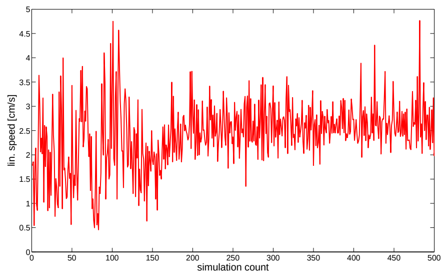
The plots show that the algorithm usually finds good gaits in very little time. In particular, for the salamander robot gaits over 20 cm/s are found with less than 100 evaluations or, in



(a)



(b)



(c)

Figure 4.35: Results of the simulation of the worm-like robot. First simulation (a), second simulation (b) and third simulation (c)

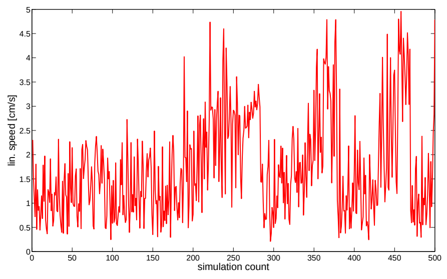
other words, less than 17 minutes of real time. Then the locomotion gait is improved up to 25 cm/s approximately. The most remarkable property is that the salamander robot moves in the same way a real salamander does, with the same coordination of trunk and limbs. The big depressions of the plots after a fast gait is found correspond to reinitialisations of the algorithm after a local maximum is reached (all the points but the best, which is kept, are reassigned new random values).

There are at least two more details to notice in the plots of the salamander robot. First, once a local maximum is found, later reinitialisations are no longer able to improve the gait, which probably means the optima found are not only local but global. Secondly, the efficacy of the algorithm clearly depends on the initial conditions, because for the second simulation the best gaits are not faster than 12 cm/s, half of the speed found in the first and third simulations. Possibly better gaits could be still found with more time for the simulations (the interval shown in the plots correspond to 500 evaluations, or 1h 23min).

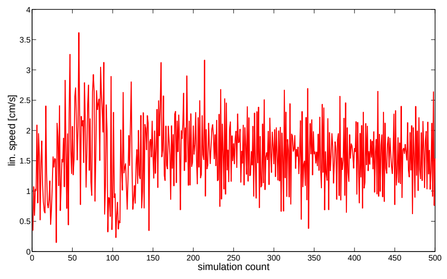
For the worm robot, the best gait found achieves 8 cm/s, although in the three simulations gaits of at least 4.5 cm/s are found.

For the starfish robot, the simplex method finds several configurations that provide a speed of 4.5 to 5 cm/s.

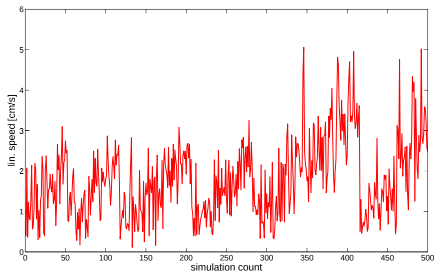
Online optimisation of locomotion with the simplex method turns to be a very effective mechanism to quickly get acceptable locomotion gaits for in principle any robot. The results presented here are roughly as good as the ones in [MI05] (both on the speed of the robot and convergence time). A drawback of the method is the dependence on the initial conditions for the final result, but this is typical for all the local search methods and can be corrected by reinitialising the algorithm if the best gait the algorithm is not good enough. Other methods



(a)



(b)



(c)

Figure 4.36: Results of the simulation of the starfish-like robot. First simulation (a), second simulation (b) and third simulation (c)

such as simulated annealing, tabu search or genetic algorithms, for instance, provide better mechanisms for exploring the search space, but these methods normally need much more time and therefore are not appropriate for online optimisation.

Chapter 5

Conclusion and Future Work

This Master Project covers the first stage of the *Roombots* project, in particular, the structural design of the prototypes of the modules and the control of locomotion in simulation. The designed modules are appropriate to create multi-unit structures that can be used as pieces of furniture, and that satisfy the requirements of homogeneity, simplicity, flexibility and proper dimensions.

The multi-unit structures can effectively move with different control mechanisms that go from simple strategies such as a sine-based controller to more sophisticated ones such as modulation of locomotion on irregular surfaces or online optimisation of locomotion. The choice of one of these strategies or another will eventually depend on the particular situations the robots will have to deal with.

CPGs are successfully applied as the foundations of all the control strategies for locomotion. Different quadrupedal locomotion gaits are explored, as well as the transition between them. The basis for navigation and obstacle avoidance are introduced with control of direction for legged locomotion, for which two different approaches are explored. Modulation of locomotion on irregular terrain is tackled by using sensory feedback in order to adapt the oscillations of the legs of the robots to the external conditions. Rolling locomotion is briefly explored on flat surfaces, with the possibility to control the direction of the robot in a very simple manner. Finally, online optimisation of locomotion is applied to find stable gaits for new types robots that could be created by the users or used to move the modules in processes like reconfiguration. Stable gaits are quickly found with the simplex method for multidimensional optimisation.

Since this is the beginning of a new project, there is still a great amount of work to be done. The design of the modules has to be more precisely characterised: exact shape, dimensions and weight, mechanical properties and attachments mechanisms. Reconfiguration has to be better studied in simulation, because the extisting results so far are not conclusive about how a good reconfiguration algorithm could be. A deeper exploration of reconfiguration might lead to the conclusion that the structure of the modules is not suitable for it, and eventually to a redesign of the modules. The development of self-repairing mechanisms is another interesting research topic that has not been addressed yet. And, of course, the modules must be realised and assembled to make multi-unit structures and locomotion, reconfiguration and the rest of the features the robots will have to be verified on the real robots, in order to have a working, useful environment of robotic furniture.

Concerning the particular improvements that can be performed on the topics related to this Master Project, the following ones can be indicated, among others:

- Exploration of additional building blocks
- Prediction of the relative phases of the oscillators that compose the CPG
- Exploration of different models for controlling the equilibrium and the response to obstacles
- Online optimisation of locomotion on irregular surfaces, in the middle of a reconfiguration process or, more generally, on dynamic environments.

References

[Arc06] Rafael Arco, *Design and simulation of locomotion of self-organising modular robots for adaptive furniture* (http://birg.epfl.ch/page60499.html), July 2006.

[Ars00] Artur M. Arsenio, *Tuning of neural oscillators for the design of rhythmic motions*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, April 2000, pp. 1888–1893.

[Ars04] ______, *On stability and tuning of neural oscillators to rhythmic control of a humanoid robot*, Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, vol. 1, IEEE, July 2004, pp. 99–104.

[BI00] Aude Billard and Auke Jan Ijspeert, *Biologically inspired neural controllers for motor control in a quadruped robot*, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks – IJCNN2000, vol. VI, IEEE Computer Society, 2000, pp. 637–641.

[BI04] Jonas Buchli and Auke Jan Ijspeert, *Distributed central pattern generator model for robotics application based on phase sensitivity analysis*, Biologically Inspired Approaches to Advanced Information Technology: First International Workshop, BioADIT 2004 (A.J. Ijspeert, M. Murata, and N. Wakamiya, eds.), Lecture Notes in Computer Science, vol. 3141, Springer Verlag Berlin Heidelberg, 2004, pp. 333–349.

[CBW02] Andrés Castaño, Alberto Behar, and Peter Will, *The Conro modules for reconfigurable robots*, IEEE/ASME Transactions on Mechatronics **7** (2002), no. 4, 403–409.

[Cev06] Sébastien Cevey, *Exploration of module design and reconfiguration*, Semester project at the Biologically Inspired Robotics Group, Swiss Federal Institute of Technology Lausanne, February 2006.

[CLC+05] Joel Chestnutt, Manfred Lau, German Cheung, James Kuffner, Jessica Hodgins, and Takeo Kanade, *Footstep planning for the Honda ASIMO humanoid*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, IEEE, April 2005, pp. 629–634.

[CV03] Jörg Conradt and Paulina Varshavskaya, *Distributed central pattern generator control for a serpentine robot*, Proceedings of the Joint International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP), 2003.

[Dij68] Edsger W. Dijkstra, *Letters to the editor: go to statement considered harmful*, Communications of the ACM **11** (1968), no. 3, 147–148.

[ENMC05] Gen Endo, Jun Nakanishi, Jun Morimoto, and Gordon Cheng, *Experimental studies of a neural oscillator for biped locomotion with QRIO*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, IEEE, April 2005, pp. 596–602.

[FK97] Masahiro Fujita and Koji Kageyama, *An open architecture for robot entertainment*, AGENTS '97: Proceedings of the first international conference on Autonomous agents (New York, NY, USA), ACM Press, 1997, pp. 435–442.

[FKC03] Yasuhiro Fukuoka, Hiroshi Kimura, and Avis H. Cohen, *Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts*, The International Journal of Robotics Research **22** (2003), no. 3-4, 187–202.

[GB11] T. Graham Brown, *The intrinsic factors in the act of progression in the mammal*, Proceedings of the Royal Society of London, vol. 84, 1911, pp. 308–319.

[GDT+03] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi, *GNU Scientific Library reference manual - second edition*, Network Theory Ltd., 2003.

[Gri85] Sten Grillner, *Neurobiological bases of rhythmic motor acts in vertebrates*, Science **228** (1985), no. 4696, 143–149.

[Gri96] ———, *Neural networks for vertebrate locomotion*, Scientific American (1996), 48–53.

[HHHT98] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka, *The development of the Honda humanoid robot*, Proceedings of the 1998 IEEE International Conference on Robotics and Automation, IEEE, May 1998, pp. 1321–1326.

[HTY+00] G.S. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita, *Evolving robust gaits with AIBO*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, vol. 3, IEEE, April 2000, pp. 3040–3045.

[ICC05] Auke Jan Ijspeert, Alessandro Crespi, and Jean-Marie Cabelguen, *Simulation and robotics studies of salamander locomotion. Applying neurobiological principles to the control of locomotion in robots*, Neuroinformatics **3** (2005), no. 3, 171–196.

[Ijs01] Auke Jan Ijspeert, *A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander*, Biological Cybernetics **85** (2001), no. 5, 331–348.

[Ish04] Tatsuzo Ishida, *Development of a small biped entertainment robot QRIO*, Proceedings of the 2004 International Symposium on Micro-Nanomechatronics for information-based Society, IEEE, October 2004, pp. 23–28.

[Mat87] Kiyotoshi Matsuoka, *Mechanisms of frequency and pattern control in the neural rhythm generators*, Biological Cybernetics **56** (1987), no. 5-6, 345–353.

[MI05] Daniel Marbach and Auke Jan Ijspeert, *Online optimization of modular robot locomotion*, Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA 2005), 2005, pp. 248–253.

[Mic04] Olivier Michel, *Webots: Professional mobile robot simulation*, Journal of Advanced Robotics Systems **1** (2004), no. 1, 39–42.

[MJD+05] Rico Möckel, Cyril Jaquier, Kevin Drapel, Elmar Dittrich, and Auke Ijspeert, *YaMoR and bluemove – an autonomous modular robot with bluetooth interface for exploring adaptive locomotion*, Proceedings CLAWAR 2005, 2005, pp. 685–692.

[Neu66] John Von Neumann, *Theory of self-reproducing automata*, University of Illinois Press, 1966.

[NM65] John A. Nelder and Roger Mead, *A simplex method for function minimization*, Computer Journal **7** (1965), 308–313.

[ODE] Open Dynamic Engine ODE, http://www.ode.org/ode.html, Open source library for simulating rigid body dynamics.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical recipes in C: The art of scientific computing*, Cambridge University Press, 1992.

[RV00] Daniela Rus and Marsette Vona, *A physical implementation of the self-reconfiguring Crystalline robot*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, IEEE, April 2000, pp. 1726–1733.

[She10] Charles Scott Sherrington, *Flexion-reflex of the limb, crossed extension reflex, and reflex stepping and standing*, Journal of Physiology **40** (1910), 28–121.

[Sim94] Karl Sims, *Evolving virtual creatures*, SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM Press, 1994, pp. 15–22.

[SWA+02] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura, *The intelligent ASIMO: system overview and integration*, Proceedings of the 2002 IEEE International Conference on Intelligent Robots and Systems, IEEE, October 2002, pp. 2478–2483.

[Web] Webots, http://www.cyberbotics.com, Commercial mobile robot simulation software.

[Wil99] Matthew M. Williamson, *Robot arm control exploiting natural dynamics*, Ph.D. thesis, Massachusetts Institute of Technology, June 1999.

[YDR00] Mark Yim, David Duff, and Kimon Roufas, *Polybot: A modular reconfigurable robot*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, IEEE, April 2000, pp. 514–520.

[YMK+02] Eiichi Yoshida, Satoshi Murata, Akiya Kamimura, Kohji Tomita, Haruhisa Kurokawa, and Shigeru Kokaji, *A self-reconfigurable modular robot*, The International Journal of Robotics Research **21** (2002), no. 10-11, 903–916.

Appendix A

Contents of the CD-ROM

- Electronic versions of the report (this document) and the presentation of the project: `report.pdf` and `presentation.pdf`.
- Implementation in C of the Matsuoka oscillator and a CPG made of coupled Matsuoka oscillators: directory `matsuoka-oscillator`.
- Webots worlds, controllers and customisation of the physics for the different experiments that were performed: directory `webots`. Most of the worlds do not work with versions of Webots equal to or above 5.1.6. Version 5.1.4 is recommended to properly simulate all the worlds.
- Libraries and additional software developed for this project, with documentation on how to use them in Doxygen format: directory `libraries`.
  - `robot-positioning`: library that is used to easily place `CustomRobot` nodes in a Webots world. The position of the robots can be indicated in global coordinates or in relative coordinates with respect to previously created robots, having the possibility of placing a robot that attachs another just with the information about the module the new robot will be attached to, and the faces and orientation of the attachment.
  - `create-world`: executable that uses the functions of `robot-positioning` to write a `.wbt` file with the specification of the Webots world and a C source file with the customisation of the physics.
  - `create-controller`: executable that creates a controller for a robot as the one explained for online optimisation, and the controller for a supervisor that implements the simplex method adapted for that robot. The structure of the robot can be either randomly generated by `create-controller` (and then the user only has to specify the number of modules the robot will have) or specified with a robot definition file (see examples in `libraries/create-world/def`).