Semester Project

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Locomotion in Modular Robots: YaMoR Host 3 and Roombots

Simon Lépine
simon.lepine@epfl.ch

Professor: Auke Ijspeert
Assistant: Alexander Spröwitz

30th January 2008

BIOLOGICALLY INSPIRED
ROBOTICS GROUP (BIRG)

Contents

Chapter 1

Introduction

Modular robotics is a relatively new but growing field constantly improving so as to give answers to some of the main issues of the future of robotics, like locomotion in extreme conditions (other planets, wars...), or more trivial but very challenging, like adaptable furniture. Roombots, a modular robot currently under development at the BIRG, a laboratory of the Swiss Federal Institute of Technology (EPFL), is a new type of modular robots with a very uncommon shape and 3 degrees of freedom per module.
The possibilities offered by this machine are difficult to measure yet, and there is a real need to understand and see what kind of locomotion these new modules can perform.
The first part of the project focused on YaMoR Host 3, a program that facilitates the implementation of a Central Pattern Generator on a YaMoR robot and provides an interface to run optimizations (for locomotion).
The second part was dedicated to Roombots, from the modelization to the characterization of the locomotion possibilities: degrees of freedom, motor forces, speed, collisions, etc.

Chapter 2

Guidelines

2.1 Task definition

- Reactivate the software framework around YaMoR Host 3.
- Model a Roombots (RB) module in Webots.
- Create several different robots exploring different locomotion principles, such as legs or wheels.
- Interface the robots with YaMoR Host 3, add CPGs and run optimization processes.

2.2 Schedule

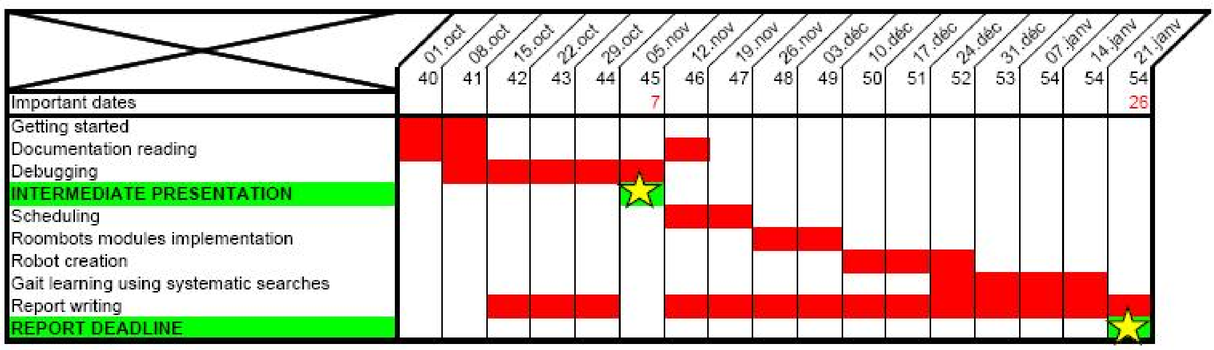

Figure 2.1: Schedule of the work.

Chapter 3

Background

This chapter summarizes the different elements that had first to be understood and handled in order to achieve this project.

3.1 Modular Robots

A robot is called 'modular' if it is constituted of several modules all interconnectable with each other, allowing a lot of different robot configurations. A modular robot is also self-sufficient, in terms of power, locomotion, actions, etc. Robustness, adaptability, multi-tasking are some of the main advantages of these systems. Self-reconfigurable modular robotics is a sub-domain of modular robotics, referring to robots which are able to change their structure on their own.
We generally distinguish two different classes in the architecture of modular robots: *lattice* architecture and *chain/tree* architecture.
Lattice architectured robots are usually assembled so as to create a more complex 3D structure (see 3.1).

Chain/tree architectured systems, on the other hand, are built in a 'linear' way, each of the modules being connected to less others in average.

3.1.1 History

In the late eighties, Toshio Fukuda created the first modular robot, called the CEBOT (for CEllular roBOT) [16]. It was composed of modules of different types, some designed to be only mobile parts, others being tools.
Greg Chirikjian, Mark Yim, Joseph Michael, and Satoshi Murata were the main figures in the further development at the beginning of the nineties. They had an important impact on the mechanical engineering aspect: Chirikjian (Metamorphic) [25], Murata (Fracta) [24] and Michael (Fractal Robots) [22] with their lattice reconfiguration systems and Yim with his chain-based systems, like the Polypod [28]. Daniela Rus (Molecule) [19] and Wei-min Shen (CONRO) [15] also had great influence on the programming part of the modular robots, working on distributed algorithms to control a large number of modules.
Recently, Satoshi Murata et al. developed a very interesting hardware platforms called MTRAN II [20] and III [23]. These systems blend the hybrid chain and lattice system paradigms, taking the best of both: like chain systems they can be controlled with Central Pattern Generators under different forms, and reconfigure like lattice systems would do.

Figure 3.1: Lattice architectured robot example: an assembling of Atron modules [1].

3.1.2 YaMoR Project

This acronym stands for 'Yet another Modular Robot' [2]. This project, developed at the BIRG, a laboratory of the Swiss Federal Institute of Technology (EPFL), aims to develop modules that can be quickly connected together to form various complex structures.

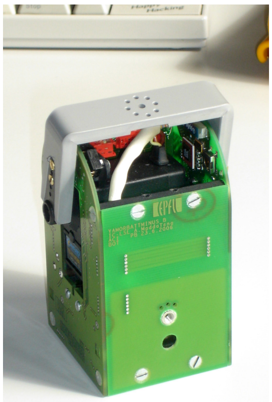

Figure 3.2: A YaMoR module [3].

The modules are autonomous and driven by strong motors (one module is able to lift 3 other ones). Each module can contain a FPGA and an ARM processor. The units don't have to be mechanically connected to communicate since they can use the Bluetooth protocol to do so.

The modules are assembled manually, and hung together via a screw and pin system.

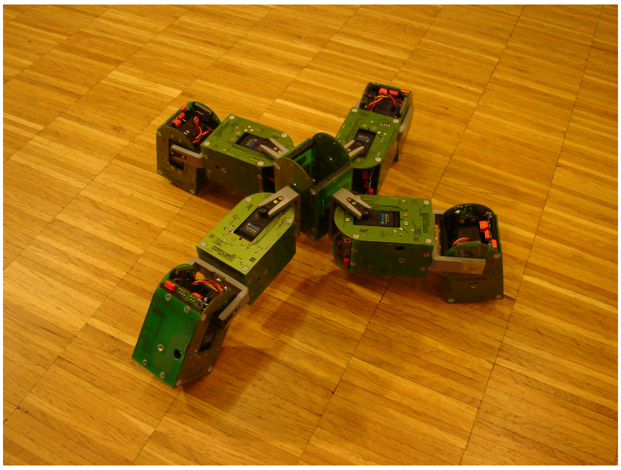

Figure 3.3: A YaMoR quadriped configuration. [4]

The locomotion of these robots is controlled by a *Central Pattern Generator* (CPG). The CPG model used for the YaMoR is inspired by nature. Indeed, we find CPGs in the locomotion system of many vertebrates [26]. The basic principle of the YaMoR CPG model is the following: each module contains a non-linear oscillator, coupled to some of the others. Each unit's motor is controlled by the output of its corresponding oscillator (see 3.2).
By applying some learning algorithms, like Powell's algorithm, the robot can learn gaits quite quickly for any usable configuration,as it has been shown by Jerome Maye [21]. The program YaMoR Host 3 developed by Michel Yerly [27] (see 3.2) allows us to build CPGs in a very easy way, thanks to its user interface, which did not exist before.

3.1.3 Interesting Projects

Nowadays, about 30 modular robotic systems have been created in this field [5]. We will only focus on a few of them which are particularly interesting in our case.

Molecubes

The Molecubes are the main inspiration for Roombots. Each module looks like a sphere separated in 2 autonomous halves that can rotate around their axis (one degree of freedom). Each half has 3 connectors, as you can see on this picture (see 3.1.3). There are also videos available on this website [6]. This shape allows a broad variety of movements in a quite unusual way, due to the diagonal 'cut' of the modules. Indeed, the axis of rotation is aligned with the cube's longest diagonal [5].
The ancestor of the actual Molecubes showed very good self-replication properties, during several experiments with 3 and 4 modules [7].

Figure 3.4: Molecubes in action [8].

Figure 3.5: Former version of Molecubes [7]

Atron Modules

Roombots have some similarities with these modules (see 5) that have inspired at least one of the robot configurations. The Atron Modules have an ovoid form, and are made of 2 autonomous parts (there is one motor). One of the main features is the connecting system of these modules through a male and female system made of an active mobile hook (male) that grips to the other module connector (female). These modules proved very efficient in assembling altogether so as to create very complex structures. The Atron's team has for instance shown very interesting results with the simulation of a Boeing 747 entirely build up with virtual models of 'ideal' cubic modules, more powerful than any existing device [13] [9].

3.2 Central Pattern Generators

A Central Pattern Generator (CPG) is a neural network 'that can produce rhythmic patterned outputs without rythmic sensory or central input.' [18]
CPGs are for instance present in the vertebrates' spinal cord. Among other functions they control the locomotion mechanism, which treats the 'simple' information sent by the brain, like *walk*, *run* or *jump*. CPGs are made of interconnected neurons that produce these rythmic outputs used to control the muscles.
The main advantages of CPGs are their robustness and adaptability. Indeed, when the system is perturbated, it smoothly goes back to normality right after. The CPG model used will have an important influence on how the sytem adapts to changes, e.g. if the locomotion mode passes from *walk* to *run*. Depending on the model, the transition will be more or less smooth.

In our case, we use a model of CPG consisting in a network of coupled non-linear oscillators [14]. We use phase oscillators amplitude controlled, ruled by the following equations (for any oscillator i):
$\dot{\phi}_i = \omega_i + \sum_j w_{ij} r_j \sin(\phi_j - \phi_i - \varphi_{ij})$
$\ddot{r}_i = a_r(\frac{a_r}{4}(R_i - r_i) - \dot{r}_i)$

Figure 3.6: Snake configuration of the Atron Modules

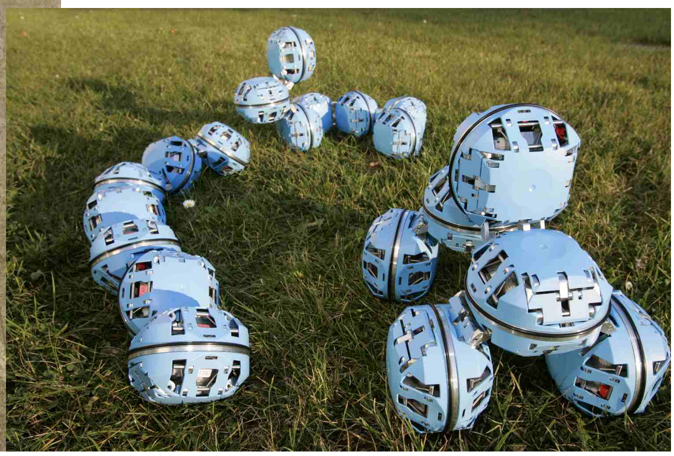

Figure 3.7: Several Atron configurations, including a car-like one. [10]

$\ddot{x}_i = a_x(\frac{a_x}{4}(X_i - x_i) - \dot{x}_i)$
$\theta_i = x_i + r_i \cos(\phi_i)$
Where ϕ_i [rad], r_i [rad] and x_i [rad] are the state variables of the oscillator, resp. its phase, amplitude and offset. ω_i [rad], R_i [rad] and X_i [rad] are resp. the intrinsic frequency, amplitude and offset of the oscillator. a_r and a_x are positive constants controlling the convergence speed.
The index j refer to variables of any other oscillator in the network. That brings us to w_{ij} and φ_{ij} [rad], resp. the coupling weight and the phase bias to the oscillator j. Finally, θ_i [rad] is the output of the oscillator.
A very complete description of CPGs can be found here in Jerome Maye's master thesis entitled 'Control of Locomotion in Modular Robotics' [21].

3.3 YaMoR Host 3

YaMoR Host 3 (see 3.3) is a software developed by Michel Yerly [27] at the BIRG. It is based on the previous work realized by Jerome Maye [21], called YaMoR Host 2.0.
Michel Yerly's program is an interface allowing the user to set up a CPG very easily and adjust all the connection parameters and the intrinsic oscillator properties. It has originally been designed to work with the YaMoR modules, and a part of the work would be to see what can be done with the Roombots.

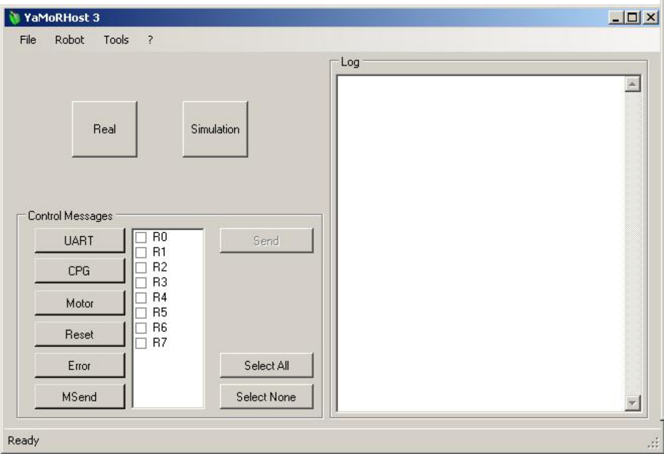

Figure 3.8: Main screen of YaMoR Host 3 [27].

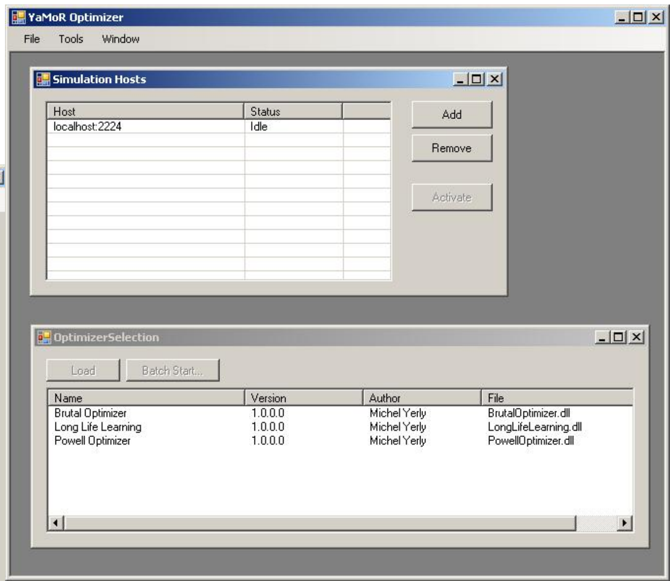

Figure 3.9: Main screen of YaMoR Optimizer [27].

Michel Yerly has also developed an important functionality to YaMoR Host 3: the YaMoR Optimizer 3.3, which allows the elaboration and effective realization of an optimization procedure in an easier and more comfortable way.

Chapter 4

Debugging

Debugging was the first task. This part describes the different changes necessary to make YaMoR Host 3 run correctly.
It maybe doesn't look very long and difficult, but this part was really time-consuming and took several weeks to be finished.
The first problem encountered was how to make YaMoRHost3 working. Indeed, the CD version did not really work. Hopefully, Alexander Spröwitz had a back-up on his computer which appeared to work better.
The second problem arose when we tried to connect to the local host to launch a simulation in Webots. After having sent an e-mail to Michel Yerly (the creator of YaMoRHost3), we managed to make it work using his answer. The solution was to start Webots with one of the controllers he had developed: `test.wbt`
We also had to re-set up an older version of Webots (5.1.13), the one which was used for the development of YaMoRHost3.
The next difficulties came when we tried to use YaMoROptimizer. At some point we were not able to determine, the program was trying to copy some file into a folder which was not reachable, because it was Michel Yerly's personal one. This was defined in the Post-Build Steps of the project, adjustable in the project properties (here the project was 'MathEval'), in Visual Studio 2005.
After that, at the launch of the YaMoROptimizer, we were not able to start a lifelong learning scenario, the only ones available were a brute force optimization and a Powell optimization.
Michel Yerly then came down to Lausanne to help, invited by the BIRG.
He resolved all the problems encountered with the CD version, which were the following (*N.B.: This will work only if you create a folder* `C:\TEMP\CD version`):

- The file `default.xml` had to be moved from the folder `YaMoRHost3\YaMoRHost3\` to `YaMoRHost3\YaMoRHost3\bin\Debug\`
- Launch `YaMoROptimizer.sln`. Look into the properties of the `MathEval` project, more precisely in the Build Events. In the post build steps, replace `C:\temp\yerly\YaMoROptimizer\` with `$(SolutionDir)`.
- The first time YaMoROptimizer is launched, there is no optimizer available. The solution is to launch the desired optimizer directly. They are all in the folder `plugins`. Once the optimizer needed will have been launched, it will be automatically available in YaMoROptimizer the next time you use it.
- You also need to move the whole content of `plugins` one degree up, otherwise there are compiling errors with the optimizers, due to reference problems.

- In each of the optimizers you want to use, some changes need to be done: in the `Build Events`, replace in both commands `C:\temp\yerly\` with `$(SolutionDir)...`
  And in the Debug tab of the Project properties, change the 'start external program field' from `C:\temp\yerly\YaMorOptimizer\YaMoROptimizer\bin\Debug\YaMoROptimizer.exe` into `C:\TEMP\CD version\YaMorOptimizer\YaMoROptimizer\bin\Debug\YaMoROptimizer.exe`

Chapter 5

Roombots

The aim of the Roombots project [11] is to create modular robots that can move and assemble together so as to form furniture equipment, on the user's demand. It can be considered as a successor of the YaMoR project since it reuses some of its ideas. The main objectives are the design and construction of the Roombots prototypes, the control of locomotion of multi-modular Roombots, the control of self-reconfiguration and the design of a robot-user interface.

5.1 Modules

Each Roombots module could be roughly described as an assembling of 2 quasi-spheres, as you can see on the picture below (see 5.1). The inspiration comes mainly from Molecubes, but also from the Atron (gripping system) and M-TRAN. When looking closer, we can see that each of these quasi-spheres is 'cut' in 2 halves, in the diagonal. The diagonal chosen is the one with the longest distance between 2 edges. That makes a total of 4 autonomous parts able to rotate thanks to 3 DC-motors, hence we have 3 degrees of freedom (dof). There is no restriction in the rotation, each of the motors can rotate freely (360°) around its axis if needed.
Each quasi-sphere also has 5 connectors: 2 hermaphrodite ones and 3 passive ones, giving for each module a total of 10 connectors. The connection between 2 modules is not mobile, both modules are firmly hang together.
The specificity of this configuration makes possible, in live, the change relative position of the rotation axis B and C (see 5.1). They can be either parallel or orthogonal, depending on the position of the motor rotating around the A axis. This feature is an important advantage compared to the Molecubes (which have static connections between modules).

5.2 Robot configurations

It is not easy to imagine at first all the possibilities offered by the odd shape of the Roombots modules. Because this was the first attempt to do so, the configurations realized were all inspired by existing ones. Most of these configurations have first been created with Solidworks. The release of the new Webots version 5.7.0 made the creation process of these robot configurations much easier.

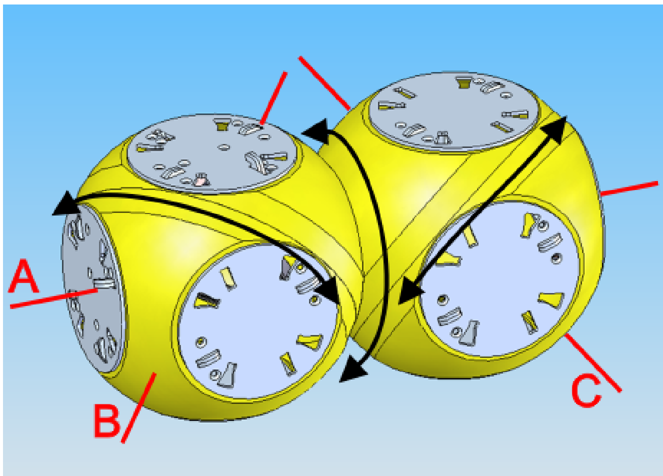

Figure 5.1: Degrees of Freedom of a Roombots module.

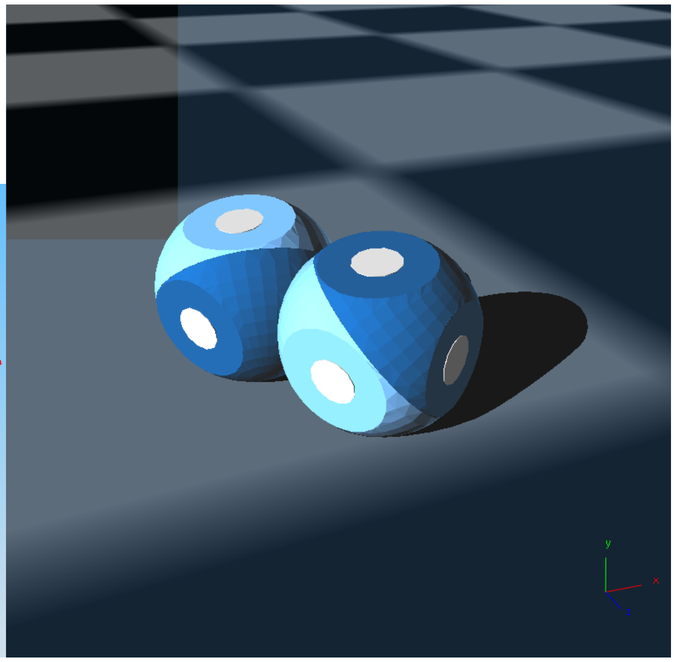

Figure 5.2: A Roombot module (Webots snapshot).

5.2.1 Snake

This configuration is the easiest and most obvious we could imagine at first. But the possibilities offered by the Roombots has leaded to surprising locomotion possibilities.
The number of modules involved is easily modifiable, the principle being to connect modules in a row fashion. With only 2 modules, we can already have interesting results. There are 6 dof only, making the researches on locomotion not too expensive.
With 3 modules the possibilities are obviously augmented, along with the computational cost of research. Here the choice has been made to give an offset of $\frac{\pi}{2}$ to the central motor so as to have a symmetry between the 2 extremities, as you can see on the picture 5.3. The goal of this symmetry is to target gaits that don't go forward, like a snake, but on the 'side', perpendicularly to a snake gait.

5.2.2 Tripod

The locomotion of a tripod is naturally odd, since no living being has 3 legs. By itself this situation is full of uncertainty due to its shape. Here the emergence of some strange locomotion principle emerged with this one, totally different than the quadripod locomotion for instance, even though there is only one limb less.
The tripod is composed of 6 modules. It can be seen as a body made of 2 modules, to which 2 limbs are attached, each one composed of 2 modules (see 5.4).
The only problem will be the selection of parameters to explore, since it has 18 degrees of freedom (dof).

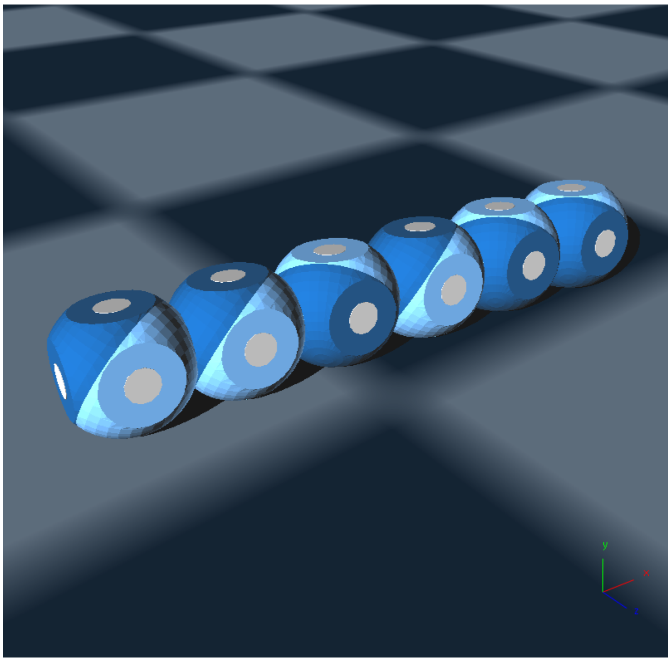

Figure 5.3: Snake configuration with 3 modules.

5.2.3 Quadripod

This quadripod is a 'classic' configuration but the gaits that can be realized with the Roombots modules are really interesting. This configuration is probably the one which could get the best results in terms of speed and efficiency in the locomotion.
Two versions of the quadripod have been created: a big and a small one. Their properties and goals are slightly different.

Big version

This version is made of 9 modules. 1 module is the 'body', around which are connected 4 limbs, a limb being constructed with 2 modules. As you can see on the picture 5.5, the 2 modules of each limb are placed perpendicularly to each other, to form an elbow. Its 27 dof maybe make the perspective of a perfect solution really difficult, but in fact, there are probably several very good gaits different from each other.

Small version

This configuration is very interesting, especially concerning the variety of locomotions possible. This version is made of 5 modules, the difference with the big quadripod being that a limb is made of one module only (see 5.6). We can see at least 2 radically different locomotion modes: a normal quadriped walking locomotion, or a car-like locomotion 6.1.1. The car is the perfect configuration to see what the wheel possibilities (through continuous rotation) of Roombots can afford.

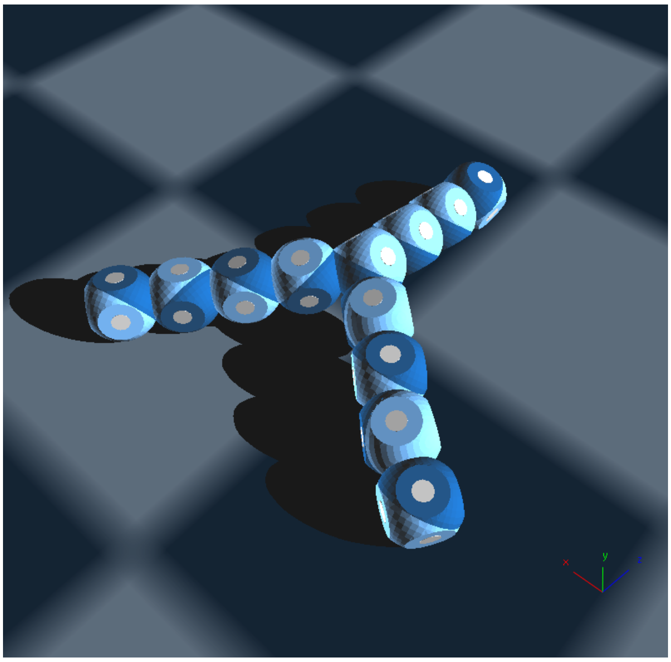

Figure 5.4: Tripod configuration with 7 modules.

And we can even imagine a gait combining both locomotion principles.

5.3 Controller

The controller is essential to make the robot move. All the instructions are given through it, and all the data is received through it. The design of the controller is up to the programmer, apart from the constraints of Webots.
In our case the choice has been made to transform one of the modules' type so as to simplify the code. Normally, a module is a *Custom Robot* (in Webots). The first module (*Module0*) has been changed into a supervisor, because we need one to get and set the position of the robot.
Indeed, this solution is much easier than a 'classical' solution with one controller for the modules, and one for the supervisor. The management of communication between all these entities is not straightforward.
So, for the sake of simplicity, the controller has been kept as simple as possible. The first version (used for the researches) is a sine-based controller (see 6), to become a CPG-based one after (see 3.2). Unfortunately, even if the code was ready, there was no time for the CPG implementation at the end of the project. It will still be provided on the CD enclosed with the report.

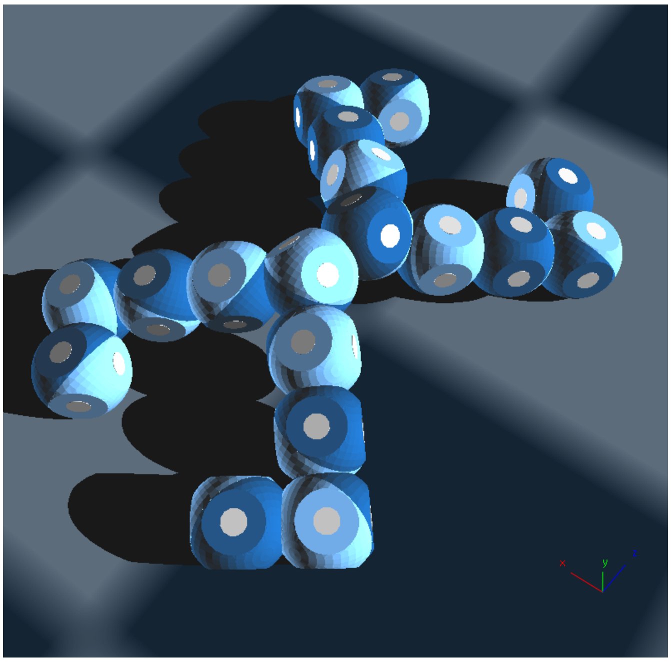

Figure 5.5: Big quadripod configuration with 9 modules.

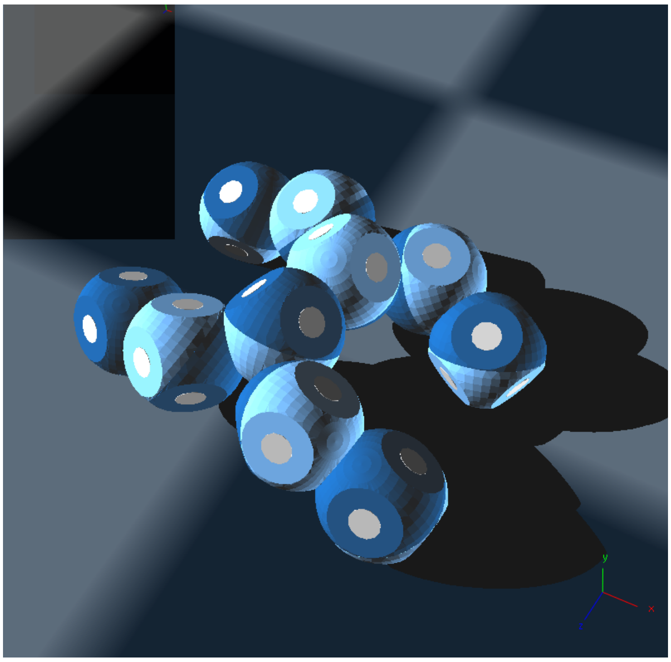

Figure 5.6: Small quadripod configuration with 5 modules.

Chapter 6

Gait Search

So as to find what kind of locomotion could be used with the Roombots modules, several systematic researches have been performed.
3 different types of configuration have been explored: snake, tripod and quadripod.
The videos of all the best gaits for each search are available on the disc enclosed.
The basic oscillator equation used for the researches (for each motor used) is this one (general case):
$Output = Amplitude * sin(\omega * t + phaseShift) + Offset$
Where ω is the frequency, almost alway equal to 2π, except for the tripod configuration. The other variables are the ones on which the searches are performed.
The process used for the search was this one:

- Select the relevant motors to use
- Define the different parameters for each of these motors
- Set the range of possible values for the different parameters
- Run the research
- Keep the best results

The selection of the motors and parameters to search on, and the choice of the range of values for each of them was the most tricky part. It would have been obviously computationally too expensive to run all the parameters, therefore choices had to be done. But since one of the principal goals of these researches was to find possible new modes of locomotion, a careful reflexion has been engaged to know what could be the best parameters to work on regarding the effectiveness of movements and the possibility for 'surprises' to emerge.
The measure of quality for a given set of parameters was the speed, i.e. the distance traveled over the time elapsed. The distance was not the 'final' distance ($finalPosition - initialPosition$), but was rather the sum of all the small distances traveled at each time step, i.e. the integration of the distance over a very small time step (16 ms). This approach doesn't discard the gaits that are less straight which are often very interesting as well. 16 ms has been chosen at the beginning, it seemed at that point a relevant value. This has maybe not be the case in reality, as we will see later.
All the results for every setting have been kept for the records. For each search, a file was created in which each line is a different set of values for the selected parameters, plus the speed measured.
'Keeping the best results' means that we look at the best solutions found, adn if possible, we try to detect some patterns, by looking at the frequence of appearance of the same value for one parameter (or a set of

parameters).
It is important to say that for these all these experiments, the maximum forces of the external motors have been set to 4 Nm, and to 2 Nm for the middle motor, for all the modules.

6.1 Quadripod

The quadripod robot can move in two radically different ways at least: like a car, or like a quadriped.
In the following, when we talk about the limbs 1, 2, 3, 4, we assume that 1 and 3 are diagonally opposed. And if we set that the limb number one is in the front of the robot, then it is also on the left of the latter.

6.1.1 Car

One of the first ideas about the locomotion of the Roombots was the possibility of a car-like robot. Indeed, the shape of the modules allows continuous rotation very easily, hence if an appropriate configuration is used, this is quite easy to do. This was initially inspired by the work done with the Atron modules [29]. The idea is simple: one half of a module rotates freely to act like a wheel. If we apply this to four modules attached to a static fifth one, the 'body', we obtain a rudimentary car 5.6. To do so, the small version of the quadripod has been used (see 5.2.3), because it seemed it was the most suitable for that. The result can be seen on a video (enclosed with the report). No exhaustive research has been done for this configuration, since the 4 motors that are used have a very basic linear behavior (The position of each wheel's motor is incremented every time step).
Most of the other motors set-points just have a constant value of 0, apart from the following ones:

- Body:
  - servo1: $\frac{-\pi}{3}$
  - servo2: $\frac{\pi}{3}$
  - m1: π
- Limb servo1:
  - Limbs 1 and 3: $\frac{\pi}{3}$
  - Limbs 2 and 4: $\frac{-\pi}{3}$

6.1.2 Walking Gait

The challenge in this case is to use efficiently the large amount of degrees of freedom (dof) to get a gait as smooth and 'animal-like' as possible.
The motors selected were the 3 body ones and the upper motors of the limbs, i.e. for each limb the servo-motor closest to the body. That makes 7 motors in total.
Then, on these motors, 13 parameters have been explored:

- Limb amplitude
- Limb phase shift (4 different parameters, different for each limb)
- Limb offset (4 times as well)

- Amplitude of the body's middle motor
- Amplitude of the body's 'outer' motors
- Phase shift of the body's outer motors (2 times, one for each motor)

First search

For the first research, the choice has been made to focus more on limb phase shifts, which seemed to be more important than amplitudes or offsets. We also wanted to see what could be the possibilities with the body's servomotors.
Apart from the 6 parameters studied during this first research, the other 7 had to bet set. The initial conditions then were:

- Limb amplitude: $\frac{\pi}{4}$;
- Limb offset 1 and 3: $\frac{\pi}{3}$;
- Limb offset 2 and 4: $\frac{-\pi}{3}$;
- Body's outer motors phase shift (2 times): 0.

The parameters explored are displayed here in detail:

parameter	range	step	Nbr. of steps
Limb phase shift (4 times)	$0 \rightarrow \pi$	$\frac{\pi}{4}$	5
Body's middle motor amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{4}$	3
Body's outer motor amplitude	$0 \rightarrow \frac{\pi}{4}$	$\frac{\pi}{8}$	3

The best result obtained after this first research was a robot going at a speed of 0.6504 m/s. More researches were needed, even though the robot was already walking in a promising way.

Second search

The 7 other parameters have then been processed, using the results of the previous research as initial conditions:

parameter	range	step	Nbr. of steps
Limb amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Limb offset 1 and 3	$\frac{\pi}{6} \rightarrow \frac{\pi}{2}$	$\frac{\pi}{6}$	3
Limb offset 2 and 4	$\frac{-\pi}{2} \rightarrow \frac{-\pi}{6}$	$\frac{\pi}{6}$	3
Body's outer motors phase shift (2 times)	$0 \rightarrow \pi$	$\frac{\pi}{4}$	5

At the end, a very smooth and efficient gait (see 6.1) arrived in top position: 0.8638 m/s.

Third search

The goal of this third research was to see if this gait could still be improved. We have used here the same parameters than for the first research, except that we took a smaller step for the body's middle motor amplitude.

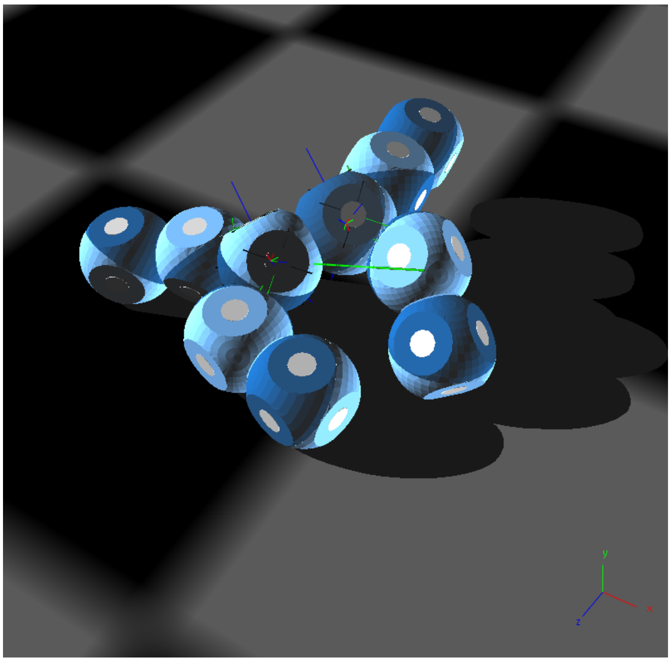

Figure 6.1: Second search result gait.

parameter	range	step	Nbr. of steps
Limb phase shift (4 times)	$0 \to \pi$	$\frac{\pi}{4}$	5
Body's middle motor amplitude	$0 \to \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Body's outer motor amplitude	$0 \to \frac{\pi}{4}$	$\frac{\pi}{8}$	3

The outcome was a faster gait (0.9870 m/s), unfortunately it seems there are some collisions between limbs which can be a serious problem with real robots.

Results

The second search gave the most interesting result, a smooth and efficient gait, that looks really 'animal-like'. It shows all the relevance of the Roombots architecture, that can simulate a fictive quadriped in a rather impressive way.

6.2 Tripod

The search for the tripod has to be very open. Indeed, its shape has more locomotion possibilities than the snake, but doesn't have as many collision constraints than the quadripod.
We use here 6 motors out of 18. It may seem low, but it's already an lot of parameters to handle. We obviously use the most central motor (see 5.4): the body's motor that is the closest to the 2 limbs. On the outer module of the body, we logically use the motor closer to the center of the tripod, so as to give greater amplitude to the moves. For each limb, and moreover for each module of each limb, we use again the motor that is closer to the center of the robot, following the same idea of maximizing the amplitude of the movements of a restricted set of motors.

The 12 parameters we will focus on during the searches are:

- Limb motors amplitude (same for all 4)
- Limb motors offset 1,2,3 and 4 (4 different variables)
- Body's outer motor amplitude
- Body's middle motor amplitude
- Body's outer motor phase shift
- Limb motors phase shift 1,2,3 and 4 (4 different variables)

An important point is the frequency used. For the other configurations, the frequency ω had the value 2π. We used this value here for the limbs, but for the body, we used a frequency 2 times higher of 4π. The initial intention was to see if some gaits where the body would work like a flagellum could emerge.

6.2.1 First search

As for the snake, we have preferred to run some searches on phase shifts first, since we believe it has a more determinant role in the gait process. Here again, the limb amplitude is the same for all the limb motors.
Therefore the initial parameters are the following:

- Limb motors amplitude (same for all 4): $3\frac{\pi}{8}$;
- Limb motors offset 1,2,3 and 4: 0;
- Body's outer motor amplitude: $\frac{\pi}{2}$.

And the parameters we want to know about:

parameter	**range**	**step**	**Nbr. of steps**
Limb phase shift (4 times)	$0 \rightarrow \pi$	$\frac{\pi}{4}$	5
Body's middle motor amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Body's outer motor phase shift	$0 \rightarrow \pi$	$\frac{\pi}{4}$	5

The best gait obtained with this research was really nice-looking, but not efficient at all: the robot was just oscillating without going forward (see 6.2). The parameters obtained were:

- Body's middle motor amplitude: $\frac{\pi}{4}$;
- Body's outer motor phase shift: $\frac{\pi}{4}$;
- Limb motor phase shift 1: π;
- Limb motor phase shift 2: π;
- Limb motor phase shift 3: π;
- Limb motor phase shift 4: $3\frac{\pi}{4}$;

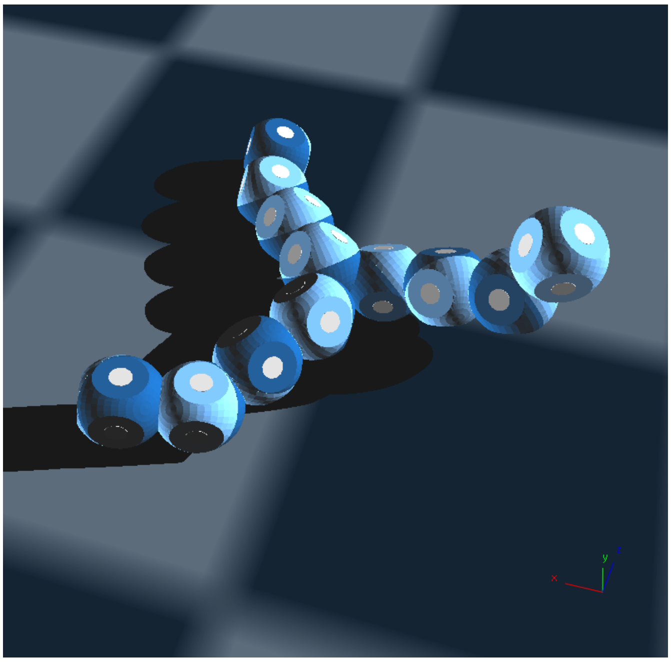

Figure 6.2: Tripod: first search result gait.

6.2.2 Second search

Because the gait was looking good, the results of this first search were kept, hoping that some adjustments of the offsets would do the rest. The other parameters were then explored as well:

parameter	range	step	Nbr. of steps
Limb motors amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Limb motors offset 1,2,3 and 4	$\frac{-\pi}{3} \rightarrow \frac{\pi}{3}$	$\frac{\pi}{6}$	5
Body's outer motors amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5

Again, the result was not very convincing, very similar to the previous one, with a little improvement only in going forward. The speed was 0.7921 m/s with this set of parameters found:

- Limb motors amplitude: $\frac{\pi}{2}$;
- Limb motors offset 1: $\frac{\pi}{6}$;
- Limb motors offset 1: 0;
- Limb motors offset 1: $\frac{-\pi}{6}$;
- Limb motors offset 1: 0;
- Body's outer motor amplitude: $\frac{\pi}{2}$.

6.2.3 Third search

The decision was taken to launch a last research on the same parameters than the first one, to see one last time if no gait could emerge, with the initial setting taken from the previous results, and the same

range of values to explore than in the first search (see 6.2.1).
The parameters obtained were the following:

- Body's middle motor amplitude: $\frac{\pi}{2}$;
- Body's outer motor phase shift: π;
- Limb motor phase shift 1: $\frac{\pi}{4}$;
- Limb motor phase shift 2: π;
- Limb motor phase shift 3: $\frac{\pi}{4}$;
- Limb motor phase shift 4: π;

The result was a total surprise: the robot was jumping, the 2 limbs working as legs, propelling the robot into the air, at a speed of 1.2066 m/s (see 6.3).

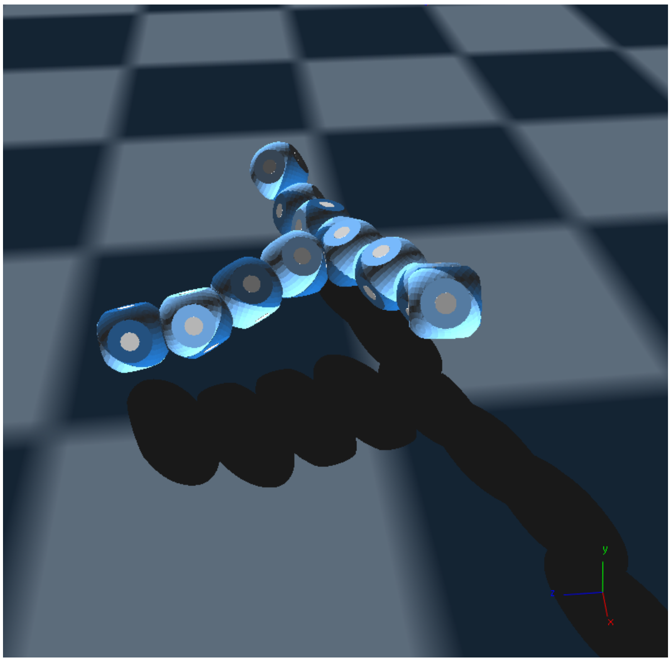

Figure 6.3: Tripod: third search result gait, the robot is actually jumping.

6.2.4 Results

We can see here that the Roombots allow some surprising gaits (the 5 first gaits were jumping ones, so it's not even an exception), which we hope will lead to even more odd locomotion modes. Nevertheless, we saw that the evaluation method is probably not the best, since, the best results of the 2 first searches were quite bad.

6.3 Snake

There are 2 ways of moving for this robot: it can move like a real snake or 'roll' on the side, in a way or another.
Here the *body* is the central module, and the *limbs* are the 2 outer modules, making a total of 3 modules. For this configuration, 5 motors over 9 are used. The ones used are in the 'middle' of the snake (the 3 motors of the central module and for each limb, the closest motor to the middle), the 2 motors at each extremity of the limbs modules have been discarded. The reason is the desire of focus on gaits that are not snake-like, the will to use the extremities as supports, and not too much as active actors of the locomotion.
The parameters chosen are:

- Amplitude of the body's middle motor
- Amplitude of the body's outer motors (same for both)
- Amplitude of the limb's inner motors (same for both)
- Offset of the body's outer motor 1 and 2 (2 different variables)
- Offset of the limb's inner motor 1 and 2 (2 different variables)
- Phase shift of the body's outer motor 1 and 2 (2 different variables)
- Phase shift of the limb's inner motor 1 and 2 (2 different variables)

The amplitudes have been assigned symmetrically to spare some computations and to aim specific gaits using this properties (like crawling). All the phase shifts are taken relatively to the central motor.

6.3.1 First search

The initial conditions have been assigned as following:

- Offset of the body's outer motor 1 and 2 (2 different variables): 0;
- Offset of the limb's inner motor 1 and 2 (2 different variables): 0;
- Amplitude of the body's middle motor: $\frac{\pi}{4}$;

The other parameters, chosen because considered as more important, are displayed here:

parameter	**range**	**step**	**Nbr. of steps**
Limb phase shift (2 times)	$0 \rightarrow \pi$	$\frac{\pi}{4}$	5
Body phase shift (2 times)	$0 \rightarrow \pi$	$\frac{\pi}{4}$	5
Limb's inner motor amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Body's outer motor amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5

The fastest gait was going at 0.7800 m/s but was not the most interesting in terms of locomotion (was just rolling on the side with a small movement amplitude). A 'less good' gait was going at 0.6566 m/s with the following parameters:

- Limb phase shift 1: $3\frac{\pi}{4}$;
- Limb phase shift 2: $\frac{\pi}{2}$;
- Body phase shift 1: 0;
- Body phase shift 1: $\frac{\pi}{2}$;
- Limb's inner motor amplitude: $\frac{\pi}{2}$;
- Body's outer motor amplitude: $\frac{\pi}{2}$;

In this gait, the robot is using much wider amplitudes, rolling on the side, using its extremities as supports (see 6.4).

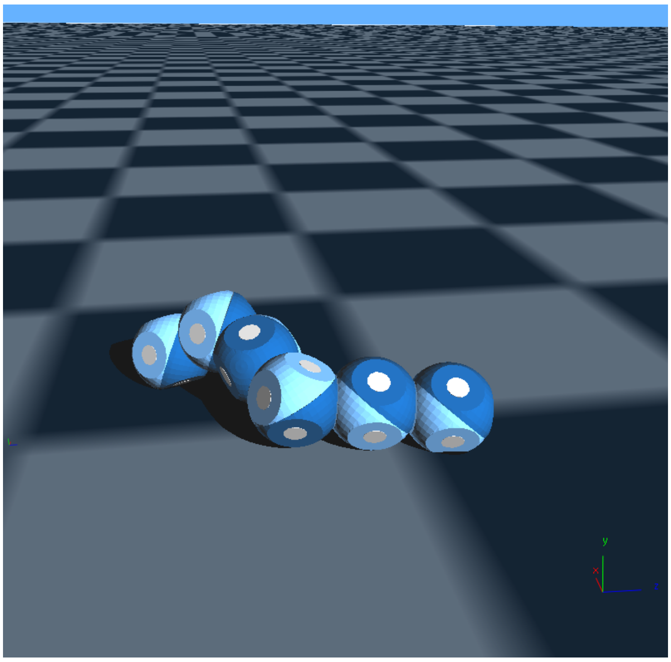

Figure 6.4: Snake: first search result gait.

6.3.2 Second search

The results shown in the previous parts have been reused as initial conditions, unless the body's outer motor amplitude, which has been ran again, because it didn't give satisfactory results during the first phase (most of the best results were using this parameter with a value 0). The details about the new search are:

parameter	range	step	Nbr. of steps
Body's middle motor amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Body's outer motor amplitude	$0 \rightarrow \frac{\pi}{2}$	$\frac{\pi}{8}$	5
Limb's inner motors offset (2 times)	$\frac{-\pi}{3} \rightarrow \frac{\pi}{3}$	$\frac{\pi}{6}$	5
Body's outer motors offset (2 times)	$\frac{-\pi}{3} \rightarrow \frac{\pi}{3}$	$\frac{\pi}{6}$	5

The result is a gait which looks a little bit like the previous one with even wider moves. It's a bit messy, but quite fast (0.7860 m/s). You can see it on the picture 6.5. The parameter results were the following:

- Body's outer motor offset 1: 0;
- Body's outer motor offset 2: $\frac{-\pi}{6}$;
- Limb's inner motor offset 1: $\frac{-\pi}{6}$;
- Limb's inner motor offset 1: $\frac{\pi}{6}$;
- Body's middle motor amplitude: $\frac{\pi}{2}$;
- Body's outer motor amplitude: $\frac{\pi}{2}$;

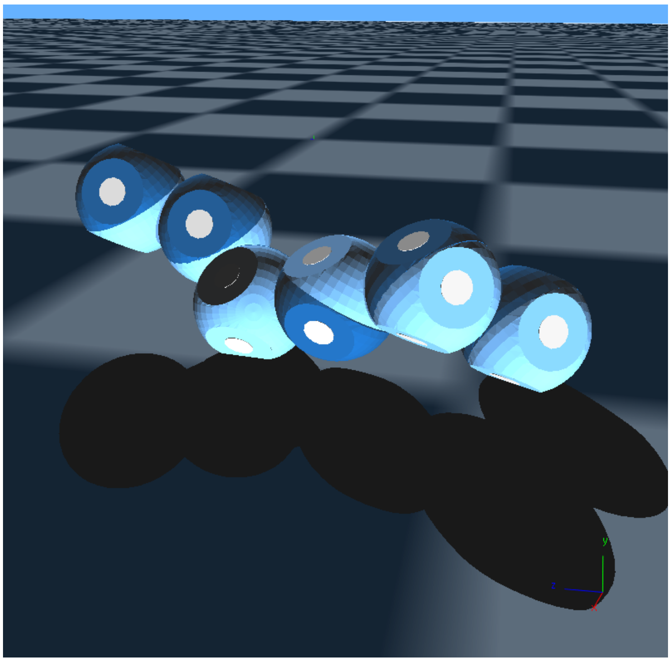

Figure 6.5: Snake: second search result gait.

6.3.3 Results

This snake configuration gives gaits that are not 'classical' snake gaits, using the properties of Roombots to perform rather strange moves to propel the robot.

Chapter 7

Future improvements

The quadriped configurations sound very promising, a combination of wheels and oscillatory behaviour could even be considered, as it has already been done with the Roller Walker robot of Pr. Hirose [17] [12]. It would also be really gripping to see what kind of locomotion modes the big quadriped configuration could use.
Concerning the tripod robot, many trails have not been explored yet, it could easily been done with more powerful optimization algorithms, like the Powell algorithm.
This could be done through the connection with YaMoR Host 3 and its YaMoR Optimizer.
In a very general way, the 360° rotation property of the Roombots motors should be carefully studied, it is definitely one of the most interesting features. The evaluation method for the gaits should also be modified. Instead of the method used in this report (speed calculated with the integration of the distance over a very small time step), a more 'weighted' measure could be used, a balance between this measure and a measure that favors more straighter gaits (like the strict distance traveled during the experiment).

Chapter 8

Conclusion

The results show that Roombots have a very large range of locomotion possibilities. Creating and using Webots models was a good way of experimenting different scenarios and configurations which will help in the conception of the real hardware. Especially using modules as wheels proved all the potential the Roombots have. They can roll, jump, walk, drive... a broad range of movements for a first search.
Several configurations have been created: a tripod, a snake, 2 quadripods. One of the quadripods, the smallest one, has even be used as a car, taking advantage of the 360° rotation properties of the Roombots.
Unfortunately, the evaluation method (speed computed with distance integrated over a very small time step) for the gaits was probably not the best (especially in the tripod case), even if it worked rather well in general.
The debugging of YaMoR Host 3 was a long operation, but it now be possible to focus on the interfacing with Roombots.
With some refining in the locomotion search, many other gaits can be found.
The author also deeply regrets not having had the time to implement the CPG model in the robot controllers, since they were almost finished.

Chapter 9

Acknowledgements

I would like to thank Joël, Aymone, Patrick, Clara, Caroline (x2) for the support they gave me in a way or another.

As always, my family was here too.

I can not forget my second family: the ACCE. And Mr. Heenok. And Takwala as well.

Yvan Bourquin has been an example of patience and help with me, I want to thank him for that.

Many thanks also to the merry crew of the lab: Aisha, Loic, Christophe, Julien, Michel and all the others.

And obviously, I thank Alexander.

Bibliography

[1] http://www.adaptronics.dk/Photos/Atron/jpg/Atron01.jpg.

[2] http://birg.epfl.ch/Jahia/site/birg/op/edit/pid/53469.

[3] http://birg2.epfl.ch/images/YaMoR/yamor$_{m}$*odule.jpg*.

[4] http://birg.epfl.ch/webdav/site/birg/users/147507/public/master/quad.jpg.

[5] http://en.wikipedia.org/wiki/Self-Reconfiguring$_{M}$*odular*$_{R}$*obotics*.

[6] www.molecubes.org.

[7] http://ccsl.mae.cornell.edu/research/selfrep/.

[8] http://www.molecubes.org/.

[9] http://www.adaptronics.dk/Videos/Atron/747.mpg.

[10] http://www.adaptronics.dk/Photos/index.html.

[11] http://birg.epfl.ch/page65721.html.

[12] http://www.youtube.com/watch?v=TYpGrsjeeLI.

[13] Self-reconfiguration using directed growth. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS 2004)*, pages 1–10, June 23-25 2004.

[14] J. Buchli and A.J. Ijspeert. Distributed central pattern generator model for robotics application based on phase sensitivity analysis. In A.J. Ijspeert, M. Murata, and N. Wakamiya, editors, *Biologically Inspired Approaches to Advanced Information Technology: First International Workshop, BioADIT 2004*, volume 3141 of *Lecture Notes in Computer Science*, pages 333–349. Springer Verlag Berlin Heidelberg, 2004.

[15] Andres Castano, Wei-Min Shen, and Peter Will. CONRO: Towards deployable robots with inter-robots metamorphic capabilities. *Autonomous Robots*, 8(3):309–324, June 2000.

[16] Toshio Fukuda, Martin Buss, Hidemi Hosokai, and Yoshio Kawauchi. Cell structured robotic system cebot - control, planning and communication methods. In *Intelligent Autonomous Systems 2, An International Conference*, pages 661–671, Amsterdam, The Netherlands, The Netherlands, 1989. IOS Press.

[17] Shigeo Hirose. Super mechano-system: New perspective for versatile robotic system. In *ISER '00: Experimental Robotics VII*, pages 249–258, London, UK, 2001. Springer-Verlag.

[18] Scott L. Hooper. Central pattern generators. *Embryonic ELS*, 1999.

[19] Keith Kotay, Daniela Rus, Marsette Vona, and Craig McGray. The self-reconfiguring robotic molecule: design and control algorithms. In *WAFR '98: Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective*, pages 375–386, Natick, MA, USA, 1998. A. K. Peters, Ltd.

[20] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji. Self-reconfigurable modular robot (m-tran) and its motion design. *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, 1:51–56 vol.1, 2-5 Dec. 2002.

[21] Jerome Maye. Control of locomotion in modular robotics. Master's thesis, Swiss Federal Institute of Technology (EPFL), 2007.

[22] Joseph Michael. Fractal robots. *Smart Materials System*, pages 56–65, February 1997.

[23] S. Murata and H. Kurokawa. Self-reconfigurable robots. *Robotics and Automation Magazine, IEEE*, 14(1):71–78, March 2007.

[24] S. Murata, H. Kurokawa, and S. Kokaji. Self-assembling machine. *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 441–448 vol.1, 8-13 May 1994.

[25] A. Pamecha, C. Chiang, D. Stein, and G. Chirikjian. Design and implementation of metamorphic robots, 1996.

[26] Grillner S and Wallen P. Central pattern generators for locomotion, with special reference to vertebrates. *Annu Rev Neurosci*, 8:233261, 1985.

[27] Michel Yerly. Yamor lifelong learning. Master's thesis, Swiss Federal Institute of Technology (EPFL), 2007.

[28] Mark Yim. Locomotion with A unit-modular reconfigurable robot. Technical Report CS-TR-95-1536, 1995.

[29] Esben Hallundbk stergaard, Kristian Kassow, Richard Beck, and Henrik Hautop Lund. Design of the atron lattice-based self-reconfigurable robot. *Auton. Robots*, 21(2):165–183, 2006.