BIOLOGICALLY INSPIRED
ROBOTICS GROUP (BIRG)

Participation to the Construction of a Salamander Robot: Exploration of the Morphological Configuration and the Locomotion Controller

Jérôme Braure

Logic Systems Laboratory (LSL)
Biologically Inspired Robotics Group (BIRG)
Swiss Federal Institute of Technology, Lausanne (EPFL)

Master Thesis

17th February 2004

Responsible
Prof. Auke Jan Ijspeert
EPFL / LSL / BIRG

Supervisor
Alessandro Crespi
EPFL / LSL / BIRG

Lausanne, 17th February 2004

Abstract

Simulators are very helpful tools in mobile robotics. They provide help and can accelerate the development process, they allow a broader variety of test-runs than the real robot does, without risking any damage. They can furthermore plot or record any type of data in a much easier and reproducible way than with a physical device. This document describes the development of a software model of an amphibious salamander/snake robot for simulation purposes with the mobile robot simulator Webots*. In a second phase, test runs are performed and the results analyzed. These experiments tend to characterize the influence of the different morphology and controller parameters on the locomotion, in particular on the velocity and its stability. This work's aim is also to provide a software model of a real robot currently under assembly in order to put the model and the simulator to the test for possible enhancement.*

Acknowledgments

I would like to thank Auke Ijspeert for allowing me to take part to the exciting and motivating research going on at the Biologically Inspired Robotics Group, Olivier Michel for his support and rapidity to implement the modifications we requested on Webots but also for the prompt and helpful answers given to all my questions, and Alessandro Crespi for his help, humour and great skills in problem solving, and all three of them for their friendliness which made my Diploma work very enjoyable!

Contents

1 Introduction

The tasks entrusted to robots are becoming increasingly varied and numerous. Programming an automaton to perform a given predefined movement can be fairly easy, whereas creating a robot that is autonomous and able to adapt itself to different kinds of environments is a completely different problem. One of the most concerned features is the locomotion. In this field, it is obvious that even extremely simple animals are nimbler than the most high-tech robot existing nowadays. Moreover, the past century has proved many times that taking inspiration from the nature for scientific and engineering purposes can yield great benefits. Therefore, roboticists consider more and more implementing in their projects the biological way used by the nature to solve problems.

The *Biologically Inspired Robotics Group* chose the salamander as research subject as it was the first amphibian that appeared during evolution some 370 million years ago (it was actually at the time the salamander's direct ancestor, named *Ichtyostega*). The salamander's locomotion (when it swims) is closely related to the lamprey, a primitive fish with anguilliform swimming, which is believed to be very similar to the ancestor of all vertebrates including the human being. Moreover, the lamprey's locomotion has been widely studied and decoded by neurobiologists. Thus, studying the salamander's locomotion will eventually improve the understanding of human control mechanisms and allow the development of biologically inspired systems with higher level of complexity.

There is at the current time (to my knowledge) no fully amphibian robot[1] taking inspiration from the nature. BIRG would thus become a precursor in that particular field, by creating the first amphibian biomorphic robot. The salamander robot will be able to perform exploration tasks and will serve as a test-bench for experimenting neural networks controlling the salamander's locomotion. It will also be a good basis for the experimentation of modular robotics, since the elements composing the body are all identical. The idea is that the body length (for example) could be easily modified depending on the tasks to be performed by just plugging some segments together.

2 Motivation

The goal of this work is to create a virtual model of the salamander robot in a commercial robot simulator and test morphological parameters as well as different settings of the locomotion controller. The results of these experiments will play a role in the design of specific parts of the robot, in particular the legs which will be shaped in a way that will optimize the velocity, stability, energy consumption and other features.

The usage of a simulator has many advantages: experiments can be performed without risking any material damage (real robots can be quite expensive, especially prototypes). Different scenarios can be tested using exactly identical initial conditions. Physical features can be dynamically modified without having to mechanically rebuild a piece. Various kind of data can be easily output or plotted. This work will also provide a good idea of how close to reality a simulator can be and thus allowing the possibility to bring enhancements to the simulator.

3 Literature review

This section will provide an overview of the current state of knowledge in the fields to which the present project is related.

[1] There is actually one, RHex, which is able to "swim" and trot but it is not considered to be a real amphibian. It moves forward with its six whegs (the contraction of *wheel* and *leg*, it denotes a wheel without rim, where the spokes are shaped like legs). It was not designed by copying an existing animal, in particular it uses almost the same gait for "swimming" and trotting.

3.1 Robot simulation

Robot simulation is a pretty recent activity as it is evolving along with the development of computer systems. Fast computers allow to perform more complex computations within a given time, thus providing a more accurate simulation, closer to reality.

Robot simulators can be classified in several categories: early simulators could only represent their world in two dimensions. Nowadays, with the fast progress of computer graphics, 2D simulators look very obsolete, and several 3D robot simulators are available, freely or commercially, through the Internet. A realistic simulator is essentially one that includes laws of physics (as many as possible), thus requiring a lot of computation time. Such a simulator is said to be *dynamic*. A simulator that does not handle physics, but emphasizes on trajectory or movement planning/handling is *kinematic*. Some simulators can even run several robots concurrently in the same world to experiment interactions between them.

2D simulators are becoming obsolete and fairly easy to create, so we will only focus on 3D simulators. Moreover, there is (to my knowledge) no 2D simulator that is dynamic. Let's have a look at some (3D) simulation softwares:

Webots is a commercial robot simulation software which was developed at EPFL [46]. It was initially designed for the KHEPERA robot, but it has since then been widely extended and is now able to simulate almost any kind of robot. Biped-, quadruped-, wheeled robots, small airplanes, blimps and of course an amphibian salamander robot (the present project) have already been implemented in Webots. It uses the Open Dynamics Engine (ODE) [40] libraries to simulate the rigid bodies' physical behavior. More details about this simulator will be given in the following sections, as it is the one that is used for the salamander project.
http://www.cyberbotics.com/

MATLAB has a free dynamic robot simulation toolbox which allows the user to easily create and manipulate data-types fundamental to robotics such as homogeneous transformations, quaternions and trajectories. It can compute inverse kinematics and includes a Simulink block library which brings Toolbox functionality into the Simulink environment.
http://www.cat.csiro.au/cmst/staff/pic/robot/

OpenSim is a free dynamic mobile robot simulator. It emphasizes on manipulator arms and implements some sensors. It also uses ODE for the dynamics. The input for the robot description and the initial values for the simulated run as well as outputs are XML files.
http://opensimulator.sourceforge.net/

EyeSim is a free kinematic multiple mobile robot simulator that includes simulation of the robot's driving actuators (differential steering), as well as all robot sensors, including: On-board vision (synthetically generated images), Infrared sensors, Bumpers, and Odometry.
http://robotics.ee.uwa.edu.au/eyebot/sim/sim.html

Yobotics' simulation software is a commercial dynamic robot simulator similar to Webots. Yobotics, a spin-off from the MIT, is a company specialized in building robotic legs for people suffering from weakness in their lower limbs. They are developing a wearable device called *RoboWalker* which augments or replaces muscular functions of the lower limbs.
http://yobotics.com

OpenHRP (Open Architecture Humanoid Robotics Platform) is a software platform for humanoid robotics, and consists of a dynamics simulator, view simulator, motion controllers and motion planners of humanoid robots. Robot models are defined using VRML as with Webots. Controllers are to be written in C++. This simulator emphasizes on humanoid robots, but custom robots can also be built.
http://www.is.aist.go.jp/humanoid/openhrp/English/indexE.html

MOBS is a kinematic simulator. It features modeled Sensors which are: odometry, bumpers, sonar sensors and cameras. MOBS has a complex sonar model, including multiple reflections, intensity calculations, and noise, which reflects real world behavior.
http://robotics.ee.uwa.edu.au/mobs/.

Juice is a free "skeletal animation workshop, with realistic physics". It is based upon ODE, emphasizes on walking robots (but wheeled robots can be implemented too) and provides a nice GUI through which all the development can be done. The concept is that the user builds up a robot model constituted of solid blocks connected together by different kinds of joints: hinges, sliders, etc. Each joint can then be actuated with a control function (for example a sine).
http://www.natew.com/juice/

ADAMS (Automatic Dynamic Analysis of Mechanical Systems) is the worlds most widely used software for mechanical system simulations. It is a high-level professional, commercial simulator that can be used to test any type of system, including mobile robots. Specific ADAMS products have been developed for aircrafts, cars, rail, engines, hydraulics and many more.
http://www.adams.com

3.2 Biologically inspired robotics

Many different biologically inspired robots have already been built. Some of them try to resemble as much as possible to a particular kind of animal while others only use certain features like the morphology of a leg or a type of gait.

There are two main kinds of legged locomotion: static or dynamic. Dynamic locomotion means that the walker passes from one unstable position to another. In such a situation, the center of gravity is not always right over the point (or area) where the foot has contact with the ground. In static locomotion, the walker remains constantly well balanced, at every instant of time.

The following (non-exhaustive) list describes some existing robots, classified according their type of locomotion and in particular the number of their legs. The aim of the following enumeration is to give the reader a fair overview of the steps which have been achieved in the field of biomorphic robotics. It also provides an idea about the trends by the little or high number of robots built for each category. The other types of bio-inspired robots which are less relevant for the salamander robot like flying robots, swinging robots (apes), will not be mentioned here. The focus is put on legged, serpentine and aquatic locomotion.

3.2.1 Monopods

Although monopod vertebrates are not the most common animals, their mechanical equivalents are quite interesting since they have to perfectly control their balance as it is the only way they can remain in the upright position and balance *is* a key feature in animal-like locomotion. Moreover monopods can only use dynamic locomotion. The study of these robots has paved the way for more complex machines by isolating the difficulties.

The Planar One-Leg Hopper was built in the years 1980-82 at the MIT Leg Laboratory [1] to explore active balance and dynamic stability in legged locomotion. It had one leg that changed length telescopically. As the robot was planar (it could only move along a circle) it was hold by a helper boom to provide stability on the other axis. The body carried sensors, interface electronics, and the hip actuators. The machine was powered pneumatically. Experiments with the Planar One-Leg Hopper showed that balance could be achieved with a simple control system. The control system had three separate parts: one controlling forward running speed, one controlling body attitude, and one controlling hopping height. These controllers worked independently, treating any coupling as disturbances. This robot, The MIT Leg Lab's first, was the ancestor of many successful bio-inspired robots.

The 3D One-Leg Hopper (1983-1989) had the same features as its predecessor, except a 2-axis actuated hip and no longer needed the helper boom; it could hop in any direction. The hip was powered hydraulically and

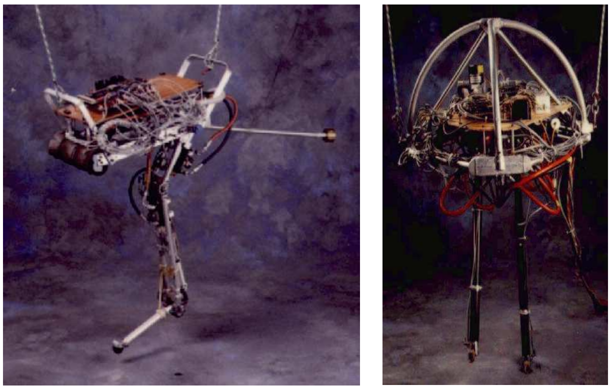

Figure 1: MIT's Uniroo monopod (left) and 3D biped (right).

actuated the leg by compressed air. This leg had two degrees of freedom (DOF). Its dimensions were 1.1 m for the height, a mass of 17.3 kg and it could achieve a running speed of 2.2 m/s.

Monopod (1988-1989) unlike its cousin the Planar One-Leg Hopper used an articulated leg instead of a telescoping one. It could run at a speed of 2.3 m/s. This type of leg, much closer to existing animal legs offered many advantages but the complexity was increased by the higher number of DOFs which require good coordination.

Uniroo (1991-1993) was their first robot patterned after an animal, the kangaroo. The Uniroo consisted of a body, a three-joint (hip, knee, ankle) articulated leg, and a 1 DOF tail. It was also planar and required a helper boom.

The Ambulatory Robotics Lab (ARL) at McGill University in Montreal [2] also worked on monopods between 1992 and 1997 by developing Monopod I, II and CARL. These robots served as platforms to study the control of passive dynamic running, using springs in the articulations to save energy. Monopod II became the most energy efficient running robot ever made. It ran up to 1.2 m/s using only 68 W of mechanical power.

3.2.2 Bipeds

Biped locomotion is probably the most trendy research area in biomorphic robotics as it is the type of locomotion that we, humans, use. Two-legged robots are becoming very numerous and popular, especially in Japan. The following enumeration will consequently only emphasize on innovative models as biped locomotion is not the main topic of this document. Many humanoid robot's real purposes are not to achieve a realistic biped locomotion but to perform other tasks like for instance playing an instrument, dancing or interacting verbally with people. These will not be mentioned in the following.

Ichiro Kato's robots WAP-1, 2 and 3 were created between 1969 and 1971 [3]. Kato, from the Waseda University, is considered to be the pioneer in that particular area, his robots were the first working models. WAP-1 had artificial muscles made of rubber as actuators. Planar biped locomotion was realized by teaching-playback control of its artificial muscles. WAP-2 had a pressure sensor under its feet and could hence control its posture. WAP-3 was able to walk up and down a staircase or a slope and to turn while walking. It was the first robot to achieve a 3D automatic walking. In 1973 WABOT 1 became the first full-scale anthropomorphic robot in the world. In 1979-80, Kato's lab developed WL-9DR, the first quasi-dynamic walker in the world, it had a total of 10 DOFs and was controlled by a 16-bit micro-computer. Its younger brother, WL-10R (1982-3) was able to turn, walk laterally and backward. Then, in 1984, WL-10RD had the ability of ascending and descending stairs and slight slopes. The Waseda University has since then created a very large number of biped robots.

The MIT Leg Lab's biped robots are quite remarkable although their initial leg morphology, telescoping legs like their monopod cousins, are rather far from biped vertebrates' legs. But as previously mentioned, this difference allowed the scientists to address issues separately. Their robots are most probably the fastest and nimblest walking machines. The MIT’s first biped named Planar Biped (1985-1990), a two-legged version of the Planar One-Leg Hopper, could change gaits while in motion, passing for example from the hopping gait to the running gait. It was a high-speed runner and a gymnast, demonstrating a flip. The next step was naturally the 3D Biped (1989-1995) which could hop around without boom. It was also able to perform a somersault and has been seen pulling a carriage with a man sitting on it. In 1994-95, a biped but non-walking robot was built, Geekbot, to study the transfer of support from one leg to another which occurs in dynamic walking.

Spring Turkey, a planar walking robot with bending knees using elastic actuation, was built in 1994-96. Right after that, they made the Spring Flamingo, a planar walker, their first robot to use feet and active ankles. Its name says it all, the robot’s walking look like the one of a flamingo, with knees bending backwards but a human-like version (forward bending knees) was also experimented. Spring Flamingo had force controlled actuators rather than position controlled, making it easier to coordinate the joints. One of the major discoveries in bipedal walking robotics is that the natural dynamics of the limbs can be used to simplify the control. This feature is exploited in that robot, in the case of the human-style knee bending. For instance, the leg can swing passively instead of requiring precise control to tell it how to move. This technique is called passive walking and has been used in unpowered walking machines back in the 80’s.

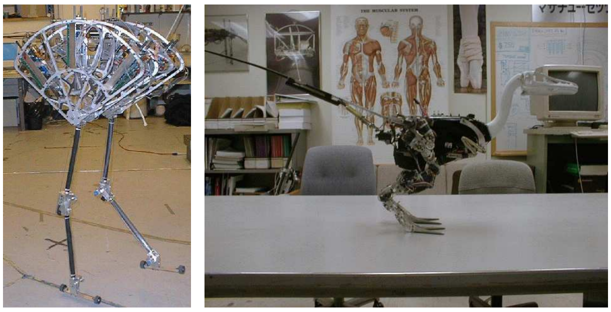

Figure 2: Spring Flamingo (left) and Troody, the MIT Leg Lab’s robotic troodon (right).

Troody, a robotic troodon (a dinosaur) was experimented at the Leg lab between 1996 and 2001. Unlike its predecessors, this 45 cm high biped with 16 DOFs was completely autonomous: it was untethered and could be tele-operated via an IR link. It could freely stand up, walk around and turn. The goal of this project was to bring back to “life” an extinct animal and to explore the application of walking robot technology to dinosaurs.

The MIT’s latest robot is named M2 and is a humanoid-legs 12 DOFs 3D robot. It will be used to investigate various walking algorithms, control techniques and automatic learning techniques.

Although Japan is the leader in humanoid robotics, some interesting models have been created in Europe. Here are the characteristics of some of them.

LUCY is a Belgian project [4] which has the goal of developing a robot walking in a dynamical stable way. Its muscles are pneumatic actuators and do not require complex gearing mechanisms for coupling. The dimensions are 150cm of height for 30kg. It cannot turn due to its unidimensional joints.

The University of Delft [5] has conducted experiments on walking robots with the aim of achieving a natural human-like walking pattern. The results of their research is intended to be applied to rehabilitation devices. One of these robots, Baps, is “the world’s first autonomous, biped robot driven by artificial muscles whose mechanical power consumption is less than a human’s” [35]. Baps stands for Biped with Adjustable Pneumatic Springs. Richard van der Linde, its creator, reused the principle of “ballistic walking” which was already known as early as 1836 and which has the important characteristic of minimizing power consumption. Ballistic walking, also known as “passive walking” takes advantage of the natural physical movements of the limbs by letting them swing like a pendulum. For comparison, Baps uses less than 1 W/kg, humans use about 2 W/kg and the Honda robot use 16 W/kg for walking.

Johnnie, the humanoid from Munich [6] is as tall as a man (180cm) and achieves a very smooth and realistic

3D walking. It remains balanced by using its arms and crosses obstacles (including stair climbing) thanks to its binocular vision consisting of two head-mounted cameras. The robot's gait is dynamically stable and its conceptors plan to implement a jogging motion. Johnnie can currently walk at a speed of 2.2 km/h.

Most biped robots are Japanese. Many of them focus on entertainment and are consequently less relevant to us. Among these are the Sony SDR-4X and Honda ASIMO.

3.2.3 Quadrupeds

The first computer controlled walking machine, the quadruped named Phoney Poney, was built in 1966 at the university of South Carolina.

Researchers from the MIT Leg Lab developed their Quadruped between 1984 and 1987 as an evolution of the previously built robots in order to demonstrate that monopedal running algorithms could be generalized to quadrupedal running. The robot uses the same kind of telescoping legs as its predecessors. The interesting part about four-legged locomotion is the transition between the different gaits: trotting, pacing and bounding. A decade later (1995-1996) they conducted experiments on a the Planar Quadruped, a robot featuring an articulated spine. Its movements could increase the general performances, among others the leg mobility, the power provided by the legs and the storage of energy using the elastic property of the spine.

GEO-I was another flexible-spine robot which was built by M.Anthony Lewis [7] in 1994. It also had reconfigurable legs which, along with the actuated spine, producing a salamander-like gait. This project lead to a second version, GEO-II, lighter and having force sensitive feet. This robot was used with an evolving neural controller taking inspiration from the lamprey with they idea of constructing a road map between this early vertebrate and more evolved animals.

Scout I, ARL's first four-legged robot [2] (1996-1998) was simplest quadruped ever designed by having only 4 DOFs (one per leg). It was nevertheless capable of many behaviors including walking, turning, side stepping, sitting and laying down, and climbing a 9cm step for a leg length of 20cm. Its younger brother Scout II received telescoping legs and became the world's first galloping robot. A version with passive knees was built to enable trotting.

The University of Zurich is working on the RunningDog project [8]. Their robot is designed to after the morphology of a dog, taking thus advantage of the naturally well-balanced canine proportions. The skeleton contains 28 passive joints, controlled by a system of springs imitating the animal's muscles and tendons. The concept of the project is to create a robot that is not specialized in performing only one task but rather able to cope with various assignments, just as real dogs do. It should help to understand general principles underlying intelligent behaviors of animals.

The Kimura laboratory [9] is very active in the field of quadruped robots using biologically inspired methods, especially for the locomotion controller. A first model named Collie-1 was built in 1984 to study dynamic walking. It was followed in 1986 by Collie-2 which provided useful information for instance concerning the gait to be chosen depending on the priorities like energy consumption, maximum speed or stability. 2D adaptive Walking on irregular terrain was achieved by the robot Patrush-I in 1994 and a little later by Patrush-II. A new robot called Tekken-I was then designed for 3D walking. It was able to pass obstacles, climb and descend slopes and could change gait while walking. Its successor, Tekken-II, was made self-contained (i.e. carrying itself its power supply and computer) and could consequently perform test runs outdoors. It also received a spring mechanism around the hip knee joint to reduce energy consumption. One of their latest robot takes inspiration from the salamander, not only from the point of view of the shape but also the controller which is directly inspired from the animal.

3.2.4 Hexapods

The six-legs morphology has the advantage of offering a stable, alternating tripod gait. The first hexapod robots appeared during the 70's, some of them were very large machines like for instance the Ohio State University Adaptive Suspension Vehicle weighing 3 tons. It was driven by a man but had automatic positioning of the legs. Another large hexapod was built at CMU in 1980-83, it could also carry a man.

The MIT Humanoid Robotics Group [10] constructed several hexapods in the early 90's. These robots, named Hannibal, Attila and Genghis were designed for autonomous planetary exploration. They were considered as being some of the most sophisticated autonomous robots, featuring over 19 DOFs, 60 sensors and 8 microprocessors.

In the same lab and in a collaboration with the Poly-PEDAL Lab Berkeley, the Boadicea project [11, 12] consists of building a small walking robot with better performances than the previous small robots. Its morphology directly inspired from the cockroach and is pneumatically actuated. Like the insect, its six legs are not all identical. furthermore, the robot has also the ability to continue walking when one or two legs are disabled without affecting the speed.

RHex [2, 13], a simple and small hexapod with rotary legs, is an extremely maneuverable robot with a very large variety of locomotion modes. It can run (3m/s), climb stairs, pass obstacles, bound quadrupedally, run bipedally, swim and even dive when the legs are replaced by fins. RHex's locomotion resembles that of the cockroach but was actually not directly copied from it. The robot has been developed in a collaboration between the universities of Michigan, Berkeley and McGill and may be sent to Mars on the 2007 mission.

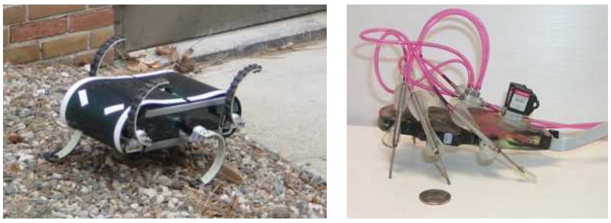

Figure 3: RHex (left) and Sprawlita (right).

The Sprawl family [14] of hand-sized robots has been designed by observing cockroaches with the goal of approaching its remarkable stability and speed, up to 50 body-lengths per second using the same morphology [31]. The fastest member of the family, Sprawley Davidson, achieves a velocity of 5 body-lengths per second. Each leg has two DOFs: a servo motor actuating the leg orientation and a pneumatic piston modifying the leg length.

The Case Western Reserve University (CWRU) [15], is working on impressive cockroach robots actuated by pneumatic artificial muscles. One of their models, Robot III, which is 17 times larger than its biological equivalent and totaling 24 DOFs, can lift as much as its own weight of additional payload. A micro-robot (smaller than 5cm in any dimension) based upon the cricket is under development. It should be able to locomote by both walking and jumping.

The University of Illinois is also developing a cockroach robot, the UIUC Hexapod [16], by taking as much inspiration from nature as possible. They are for instance implementing an insect-inspired binocular visual system to achieve agile locomotion over irregular terrain.

3.2.5 Octopods

The Fraunhofer Institute of Autonomous Intelligent Systems (AIS) together with the ARL and the MIT have developed the Scorpion octopod [17] at the beginning of the new century. It is using solar power to recharge its batteries and its conceptors plan to deploy it in the Mohave desert. The task of the robot will be to navigate to a target location 25 miles out in the desert without human interaction and to safely return to the deployment site demonstrating thus the ability for robots to survive several weeks without the help of humans.

The CWRU [15] is working with the company K^{2}T Inc. on a mine removal robot. It should eventually be able to clear mine fields and shallow underwater mines autonomously.

3.2.6 Snake robots

One could wonder why snake robots are interesting. They have no leg at all. The point is that sometimes legs can be penalizing, they add weight, require power, take more space, etc. Furthermore, it is well known that some snake are as well at ease in the water as on the ground and sometimes even on trees. They also have the ability to pass through tight holes which is an important advantage for exploration robots. Many studies of snake-like robots were conducted for the development of manipulators (robotic arms).

The Hirose + Yoneda Robotics Lab [18] has started creating snake-like robots as early as the 70's. The very first serpentine movement was achieved in December 1972 by the robot ACM III (Active Cord Mechanism). It measured 2 meters, had 20 joints with 1 DOF each and moved forward at a speed of approximately 0.4m/s. The differential friction required for the serpentine locomotion was recreated by simply fixing little wheels at the bottom of each segment of the body. The robot provided answers to their question: "How is it that a snake can go forward without legs?".

Among other robots by Shigeo Hirose and Kan Yoneda, the ACM-R3 (2001) had the capability of bending its body in tree dimension thus allowing it for instance to literally stand on its tail, like a cobra. It was wirelessly controlled consequently increasing even more its mobility.

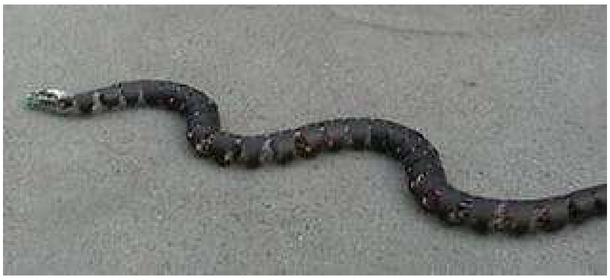

Figure 4: Gavin Miller's S5 snake.

Dr. Gavin Miller's Snake Robots [19], S1 to S5 are some of the most realistic in this category. The first model (S1, 1992-93) was a recreation of a robot made by Shigeo Hirose. Its speed and direction could be remotely controlled. A later model (S3, 1996-97) was featuring an additional degree of freedom on each segment, thus allowing movements in 3D. It demonstrated the conventional horizontal undulatory progression but also side-winding motion thanks to the new universal joints. The last one (S5, 1998-99) is a refinement of S3 with a larger number of segments and a smaller cross-section.

The GMD-snakes [20] rely on a different type of mechanical configuration: snake-1 (1996) is composed of a customizable number of bending segments which do not use wheels, unlike most of the previously mentioned robots. The robot's head carries sensors (e.g. optical sensors for orientation). The following model, Snake-2 (1998), is intended to be able to inspect tight tubes. This one presents 12 wheels around each of its segments and is equipped with many sensors: infrared and ultrasonic distance sensors, torque sensors, three sensors to measure the segment's position, and a head-mounted camera. Snake-2 is 10 times faster than its predecessor with a speed of 0.1 m/s.

NASA and JPL are working on snake robots [21] and manipulator arms respectively for planetary exploration and work on the International Space Station. The motivation for their interest in snake robots is their ability to creep into tight holes and pipes. They are also examining modular robotics e.g. a long snake that could brake apart thus becoming several exploration 'snakelets' and later regroup.

A worm-like type of locomotion has been developed [39] for a pipe inspection robot that can furthermore also demonstrate a serpentine locomotion thanks to three-DOF joints between the twenty segments. It was able to crawl through tubes of 25mm of diameter and had a micro grip plus a camera on its head.

Several other snake-like robots have been created but many of them are slightly off-topic since locomote in a way that differs more or less from the serpentine locomotion which uses the body movement to induce the speed. One of these is named Moira, is 2m long and is propelled by ridged belts like tank tracks. It should be used to locate survivors after earthquakes.

3.2.7 Eel-robots

Let us now move over to aquatic robots. Let's have a look at the few eel-like robots, whose characteristic is to use their entire body to provide thrust, like snakes. There seems to be very few of these robots.

The Northeastern University has created a lamprey-like robot intended for remote sensing operations in the water column with robust depth/altitude control and high maneuverability [22].

Franck Nickols has been working on a swimming eel-robot composed of four body sections but is providing very few information about it [23].

Ken McIsaac has also built an amphibious eel-robot named REEL for research in undulatory locomotion. This robot was able to swim and to crawl on the ground using four distinct gaits and their reverse motion [24, 38].

3.2.8 Fish-robots

In order to improve the capabilities of Autonomous Undersea Vehicles (AUV) the MIT developed a tuna-robot naturally named RoboTuna. The fish morphology is extremely well adapted to a liquid environment contrary to existing propeller-driven AUV's which operate at fairly low efficiencies e.g. short mission times, restricted payloads due to the large space required for the batteries. The robot's structure was made of a flexible hull and eight rigid vertebra ended by an oscillating foil. These vertebra were actuated by a system of tendons, passing through a mast to the support structure from which the body is hanging. One of the experimental result obtained was that the drag of the swimming fish RoboTuna appeared to be less than the drag on the straight RoboTuna [25].

PPF-09 is another fish robot but unlike RoboTuna, it is untethered and comprises only two joints. It can perform radio-controlled 3D swimming thanks to a vertical tail fin and two horizontal breast fins. This robot was designed for educational and ludic purposes although it is highly completed as a model fish robot [26].

Mitsubishi Heavy Industries has built a life like looking coelacanth robot which measures 120cm and weighs 40kg. The goal of that research is as for RoboTuna, to study the advantages of the fish morphology over usual underwater vehicles. Mitsubishi could take advantage of this characteristic for its submarines [27].

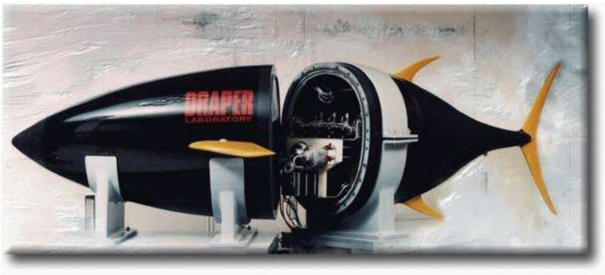

Figure 5: The Draper Lab Tuna Robot.

The Draper Lab has created a 2.4m long self-contained robotic tuna. It contains many sensors like internal thermometer, compass, accelerometers, leak detectors, etc and is able to swim in 3D at a speed of 2.4kt (4.3km/h). It shall be used in various missions like surveying, cable laying, mine/threat recognition [28].

3.2.9 Underwater walking robots

Some legged robots have been made waterproof to perform underwater tasks. The first example is an octopod based on the lobster and is intended for autonomous remote-sensing operations in rivers and/or the littoral zone ocean bottom with robust adaptations to irregular bottom contours, current and surge. It could for example search for dangerous underwater mines. Lobsters have the optimal body shape for walking underwater making them some of the most energy-efficient animals around. [22].

The Ariel robot is amphibious and can be compared to the crab. It is the first legged robot capable of walking on land or underwater. Furthermore, it is invertible i.e. if it flips over the top becomes the bottom and the robot can still go ahead by reorienting the legs. It shall also be used for the search of mines [12].

3.3 Animal locomotion

Locomotion is a key feature of an animal and can take many different forms: swimming, leaping, flying, crawling, walking, using two, four, or more legs, galloping, etc. As previously mentioned, modern robots tend to adopt the animals' locomotions, which are in many case, more appropriate and effective than simple wheels like mobile robots usually have. But the difference in agility is proportional to the difference of complexity of the underlying mechanisms. Consequently, when one tries to address this issue, some interesting questions arise: Do all the different types of locomotion have something in common? How does an animal control his gait[2]? Is it the brain that directly sends electric impulses to each muscle? How does an animal learn to perform his locomotion?

It has been shown that the control of muscles is only partially done by the brain; the spinal cord plays an important role in the stimuli generation for the muscles. Two different theories coexisted during the twentieth century. By electrically stimulating animals brains or spinal cords at precise locations, some scientists came to the conclusion that there was a *central control* while other believed in a *peripheral control* of the stimuli pattern generation. For example, as reported in [30], "Shik et al. (1966), for instance, showed that mesencephalic cats could exhibit a walking gait on a treadmill when an area of the mid-brain was electrically stimulated. Moreover, they found that such preparations could switch from the walk to the trot to the gallop if either the strength of the simulation or the speed of the treadmill was increased."

The theory that is commonly accepted nowadays is that the control (in the case of vertebrates) is central, i.e. there exists a circuitry in the spinal cord that is responsible for producing the oscillations which will generate the gait. These circuits, named *central pattern generators* (CPGs) receive in contrast rather simple inputs from higher-level control centers. We will come back to the CPGs in the following subsection.

3.4 Locomotion controllers

The design of locomotion controllers is a critical point and major difficulty in creating an robot using animal-like locomotion. It is one of the reasons why wheeled robots are much more numerous and usually used, as their conception is fairly easier. However, wheels have their disadvantages, they have for instance limitations on rough terrains.

There are currently three main approaches in locomotion control: trajectory based methods, heuristic control methods and CPG-and-reflex based methods. The task is principally to keep the robot in the upright position (especially for bipeds) and keep the balance or stability. This eventually requires to produce a controller that adapts the gait to the ground conditions when the robot has to walk on an irregular terrain.

3.4.1 Trajectory based control

This method consists of designing kinematic trajectories, by trial-and-error processes or by recording the movements of a real animal. Then the stability of the locomotion is to be tested using dynamic equations. One of the best methods for proving the stability of a trajectory is the ZMP (Zero Moment Point) approach. This method works like this: the ZMP is computed, it is defined as the point on the ground about which the sum of all the moments of the active forces equals zero. It is also known as the Center of Pressure. Then the stability is confirmed if the ZMP remains within the convex hull of all contact points between the feet and the ground.

[2] A gait is a specific type of coordination between limbs, e.g. walk, trot, gallop.

Another criterion that can be used for the stability of robot walking, instead of the ZMP, is the Center Of Gravity (COG) condition [37]. This method requires less (and simpler) computation thus quicker computation time but has some limitations. For instance, dynamic stable locomotion cannot be handled with the COG method but it is possible with ZMP. The ZMP method is well-suited when the stability is crucial. It requires however a perfect knowledge of the robot's dynamics and of the environment, additional on-line control to deal with perturbations, and enough time to define good trajectories [32, 34].

3.4.2 Heuristic control

This method's best example is the Virtual Model Control (VMC). The idea of VMC is to place virtual components which have physical counterparts (like springs, dampers or any other component that outputs a force based on its state) at intuitively chosen locations within the robot or between the robot and its environment. Then, a finite state machine (corresponding to the different phases of the locomotion) or other method of virtual component modulation is to be designed. Finally, the virtual elements are removed and the robot motor torques needed to replicate the effect (produced by known forces) of the virtual elements are computed using a simple transformation [42, 34, 43, 36]. The movements are produced according to a state machine representing the different phases of the locomotion.

Some benefits of that method are that it is compact, intuitive, requires low computation load, can be implemented in a distributed manner, does not need an accurate model of the environment and shows robustness against perturbations. However, as for ZMP, it requires a perfect knowledge of the robot's dynamics. Furthermore, one must make sure that the virtual forces to be replicated remain within the capabilities of the motors [34].

3.4.3 CPG-and-reflex control

This type of control is the one that is used by the nature in animal locomotion. Robots are currently being developed with such controllers after direct biological inspiration. Central Pattern Generators are, according to [33], neural circuits capable of producing the patterns of oscillations necessary for locomotion without oscillatory input from higher control centers or from sensory feedback. An example of input is the speed of motion, a simple signal which is transformed by the CPG into an oscillatory output yielding different gaits, depending on this signals amplitude, as previously mentioned in this document.

4 Software model implementation

The implementation of the salamander model is done in Webots [46]. Simulating a robot with Webots consists of (a) creating the robot's physical shape through the GUI, (which edits a VRML file with a `.wbt` extension) (b) writing the controller's C, C++ or Java source code and compiling it, and (c) launching the simulation.

4.1 Salamander body

The salamander was initially created using Webots' graphical user interface (GUI). The spinal cord is composed of several segments. One of these segments is depicted on figure 7.

In the Webots world description file, each body segment is a servo shaped by a group of two transforms, each containing a shape (respectively a box and a cylinder). Transforms allow to shift the shape's position, to move it away from the origin point. The legs are made up of two cylinders (femur and tibia) plus three spheres (hip, knee and foot) at the junctions, all contained in a servo. There are two other servos embedded in the leg (one for the hip and one for the knee), but they are not actuated. They are needed for setting the limb geometry, and can simulate a leg bending under compression. The legs rotate at a frequency of 1 Hz and must be coordinated with the body's oscillations to produce a coherent gait. We call them *legs* but they are actually whegs. Since salamanders have *legs*, we will keep that denomination even for the robot. There is a tricky part concerning the head segment. The root element of the salamander is a 'CustomRobot' node, to which all the

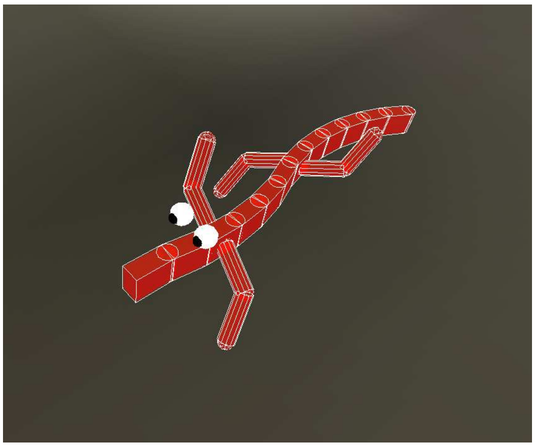

Figure 6: The salamander model in Webots.

other segments are attached like a chain. However, a CustomRobot node does not implement all the feedback functions that servos use to get their position, orientation, velocity, etc. which are required to determine each element's location relative to the water. Therefore, an additional servo is superimposed to the CustomRobot node, both being shaped as a regular body segment. This servo can then determine its position and velocity which will be exactly the same as the root element's.

4.2 Servo control

Servos contain PID controllers: PID stands for Proportional-Integrative-Derivative. It is a controller that adapts the input voltage of a small DC motor using a feedback loop, in such a way that the real position of that servo is as close as possible to the desired position. The input voltage u feeding the motor is computed using the following equation:

$$u(t) = K_p e(t) + K_i \int_{\tau=0}^{t} e(\tau)\partial\tau + K_d \frac{\partial e(t)}{\partial t}$$

where the error $e(t) = \widetilde{\theta}(t) - \theta(t)$ is the difference between the actual angle $\widetilde{\theta}$ and the desired angle θ. The scalars $K_{p,i,d}$ are called the *gains*. This equation is the sum of three terms: (a) the proportional term, which simply sets the voltage proportionally to the position error, (b) the integrative term which is proportional to the time integral of the error, and (c) the derivative term, which has a braking effect as it is proportional to the rate of change of the error. Graphical examples of differences between real and desired positions will be provided later in this document.

In Webots, the controller only has a proportional term, so it's input equation is: $u(t) = K_p e(t)$.

4.3 Desired position versus real position

An order given to a servo is executed within a certain delay due to the fact it can simply not switch instantaneously from one position to another that can be fairly distant. Moreover, if a second command is given before the first has been fully executed, the latter will not be completely achieved. These two effects result in a damping and a dephasing in the *desired position* wave, yielding the *real position*. The parameter that causes the most important variations between these two positions is the oscillation frequency of the desired position. The highest the frequency, the greatest the difference. Figure 9 shows the effect of the frequency on the real position. We can notice that the dephasing remains constant while the amplitude is seriously damped as the frequency increases. Figure 10 (left) shows that a variation in the amplitude maintains the similitudes, i.e. the

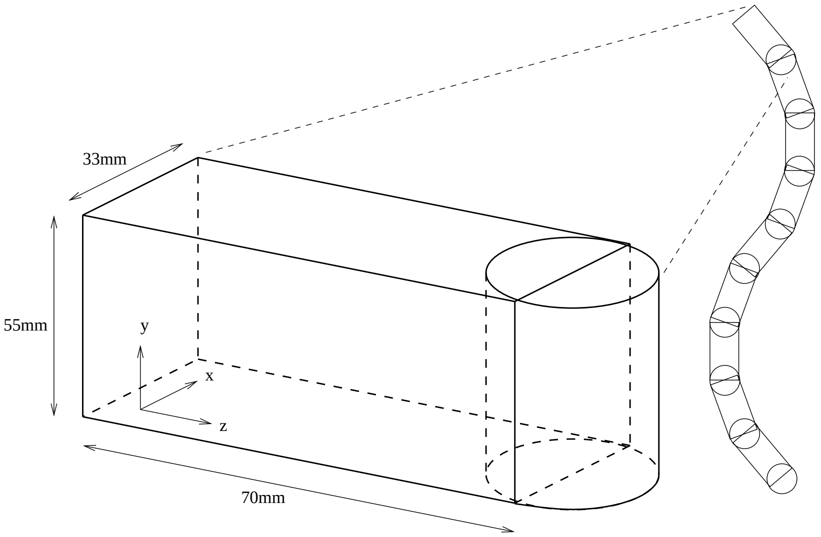

Figure 7: Salamander body segment with local coordinate system and top view of a salamander body.

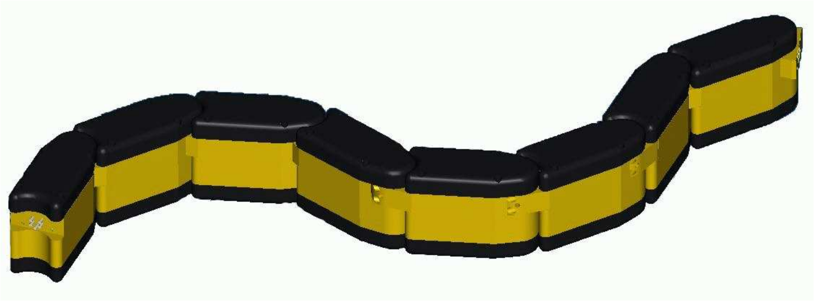

Figure 8: Computer graphics image of the real robot.

proportion of damping remains. Finally, on the right, figure 10 depicts the damping of the peak amplitudes as a function of the frequency. At ν = 3 Hz, the real position's amplitude is only half of the desired position.

4.4 Locomotion controller

The locomotion controller is a so-called 'open-loop' controller, it just actuates the servos, without sensory input. Its design is mainly based upon previous student work [29, 45] and can demonstrate legged or serpentine locomotion. In the latter case, which can be used for both swimming and locomoting on the ground, the body undulates with a traveling wave. The angle $\alpha_i(t)$ of servo i at time t is computed according to the following expression:

$$\alpha_i(t) = A \cdot \sin\left(2\pi\nu t - \frac{2\pi i}{\lambda}\right)$$

where λ is the wave length expressed in multiples of the segment length and ν the frequency.

In the case of legged locomotion, the body is separated into three parts (head, trunk and tail) which all oscillate as a standing wave with a phase shift of π between two contiguous parts. The parts' boundaries are defined as the locations where the legs are connected to the body. The equations controlling their servos are thus:

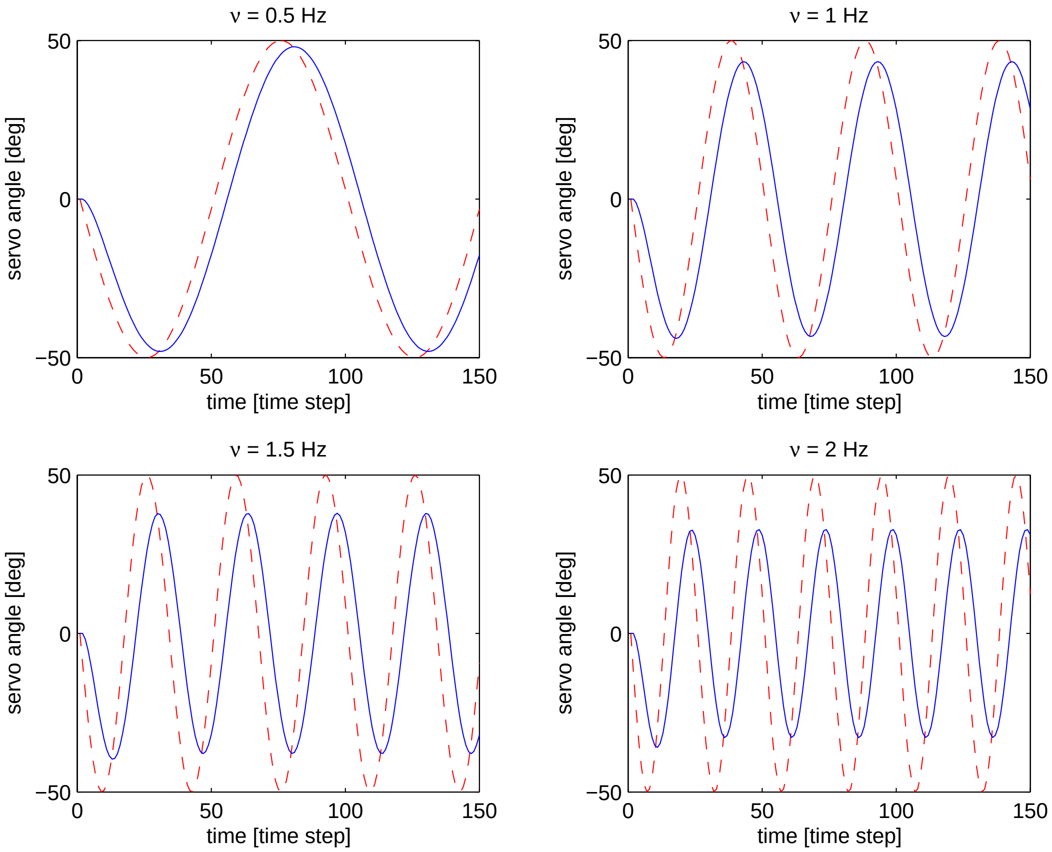

Figure 9: Real position (continuous line) versus desired position (dashed) of the servo located at the middle of the body during swimming. Frequencies are resp. 0.5, 1, 1.5 and 2 [Hz] but the desired amplitude ($50°$) remains the same.

$$\alpha_i(t) = \begin{cases} A \cdot \sin(2\pi\nu t) & \text{servo}_i \in \text{head}, \text{tail} \\ A \cdot \sin(2\pi\nu t + \pi) & \text{servo}_i \in \text{trunk} \end{cases}$$

Figure 11 depicts the locomotion controller's algorithm and figure 12 shows the movement of the salamander model trotting and performing serpentine locomotion. The salamander is said to be submerged when the head segment's altitude (Y coordinate) is below the water level. If it is above the surface, and if the robot is legged and the locomotion not fixed to serpentine, the legs are actuated and the body undulates as a standing wave. Otherwise, the legs (if any) are rotated and placed parallel to the body. Then, for all solids constituting the salamander, the altitude is queried and depending on it, Archimede and drag forces are computed and applied to them independently. When it is done, the loop starts over again.

4.5 Physics

ODE (Open Dynamics Engine) is taking care of the physical interactions between the bodies in our simulated world. It is used by Webots for simulating articulated rigid body dynamics. ODE's function library allows to create rigid bodies (solid objects), joints (between the objects), simulate friction, detect and handle collisions. However, neither ODE nor Webots have the notion of 'water'. Therefore, to be able to create a swimming

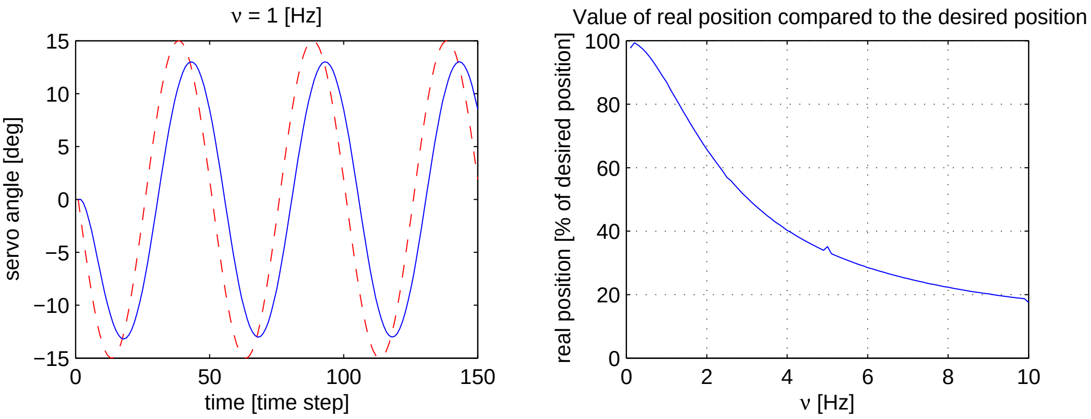

Figure 10: (left) same case as the top right of the previous figure but with an amplitude of $15°$. (Right) The damping of the real position (max amplitude) as a function of the frequency.

salamander, custom-made external forces have to be applied on the robot to simulate the effect of water. These are (a) the force due to Archimedes' principle and (b) the drag.

4.5.1 Archimedes' Principle

Definition: *When a body is wholly or partly immersed in a fluid, it experiences an upthrust or buoyant force equal to the weight of fluid it displaces.*

This force, usually denoted as Archimedes' force is computed using the following expression, which follows from the definition.

$$\vec{F_A} = -V_{body} \cdot \rho_{water} \cdot \vec{g}$$

Where V_{body} is the volume of the immersed body, ρ_{water} is the water density and $\vec{g}$, the gravitational force vector. One must notice that this force does not depend on the depth. Nevertheless, the case where the body is only partly immersed must be taken care of in order to avoid hectic oscillations. Indeed, without that case, the body would constantly oscillate between the *entirely immersed* and the *out of the water* positions, with a direct switch from one to the other. To prevent that behavior, a simple computation, can be made to approximate reality. It consists of replacing V_{body} by the volume of the immersed part of the body. Archimedes' force is computed independently for each segment of the body, including the legs.

The body's density is usually set to the same density as the water to assure neutral buoyancy during underwater test runs. A density lower than the water's will give the robot the characteristics of a floating body.

4.5.2 Drag

The drag force is more complicated to compute, since it depends on many more parameters. The model adopted in this project to compute it is the following:

$$F_{D_i} = \frac{1}{2} \cdot \rho_{water} \cdot S_i \cdot C_i \cdot v_i^2 \qquad i = x, y, z$$

F_{D_i} is the force to be applied along the body's i axis, where S_i is the body's section (surface) perpendicular to the axis, C_i is the drag coefficient relative to the i axis, and v_i the component of the body's velocity on its i axis. At each time-step and for each segment, the three components of the drag force are computed separately and applied to the body (see figure 13). The drag coefficients, usually denoted C_D in textbooks, are in our case only a rough approximation of the reality, the computation of their precise values would be

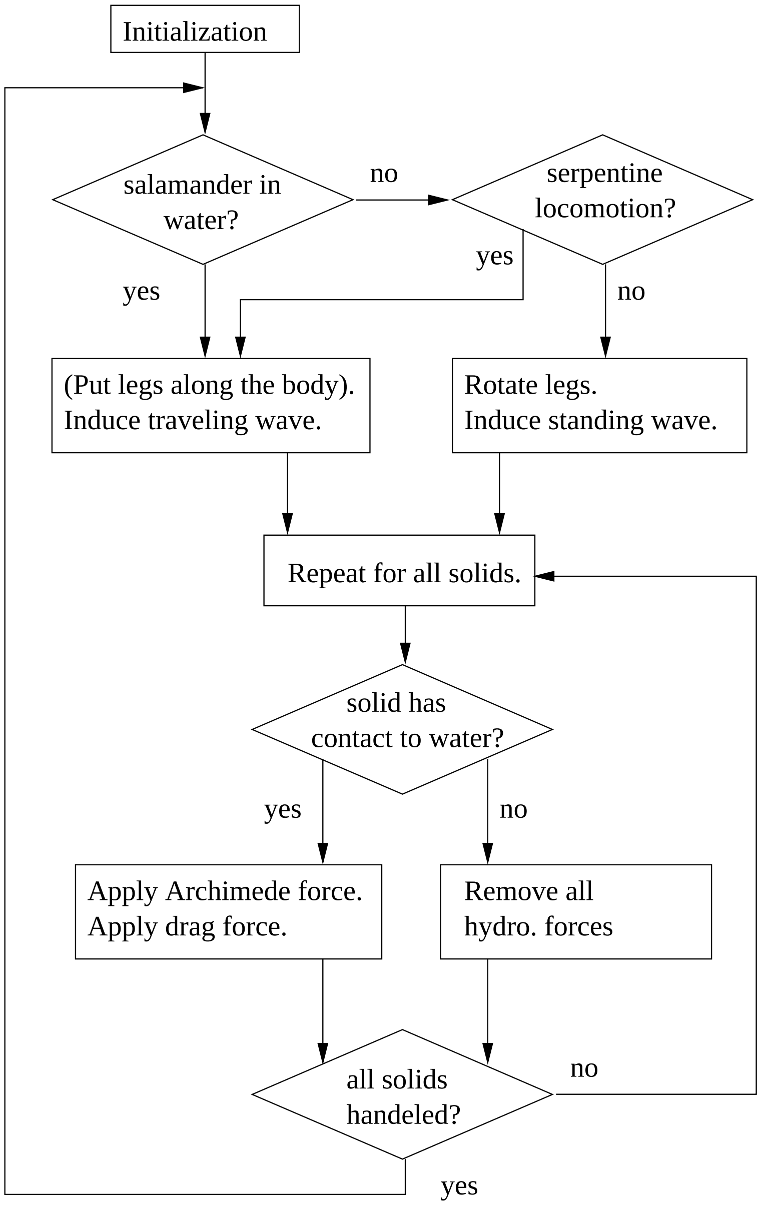
Initialization
salamander in water?
no
serpentine locomotion?
yes
yes
no
(Put legs along the body). Induce traveling wave.
Rotate legs. Induce standing wave.
Repeat for all solids.
solid has contact to water?
yes
no
Apply Archimede force. Apply drag force.
Remove all hydro. forces
all solids handeled?
no
yes

Figure 11: The locomotion controller's algorithm. After initialization, the infinite loop actuates the body and leg servos and applies hydrodynamical forces (if needed).

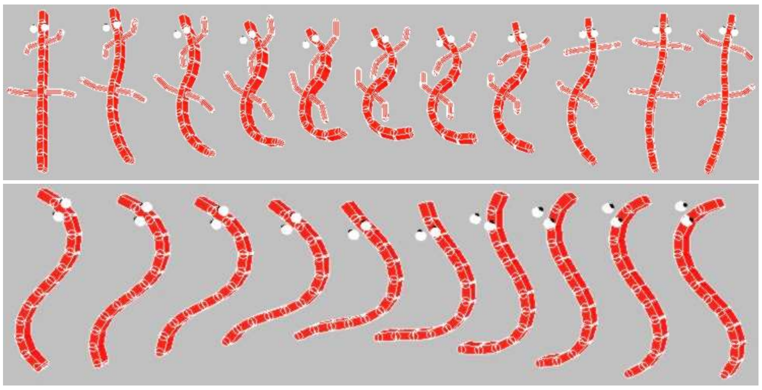

Figure 12: Sequence of salamander trotting and performing serpentine locomotion (half periods).

quite involved or should proceed by measuring the real elements. Fluids mechanics literature [47] provides the C_D's of various bodies, depending on their dimensions and shapes. The salamander body segments are approximated (for the x and y axes) as two-dimensional[3] bodies, since the flow is blocked at both extremities by the neighboring segment. On the remaining axis, z, The drag is null since each segment follows the preceding one and therefore has no direct contact with the water. Only the three leading segments have non-null C_Z coefficients, as described in [45].

4.6 Dynamic configuration generation

A program needed to generate custom Webots robot configuration files (`.wbt` files) was then developed. The features that can be set are: the body length, i.e. the number of elements constituting the body, the attach position of the legs, various leg characteristics as depicted on figure 14 as well as different angles as shown on figure 15. The software can easily be modified to allow the setting of other variables. The locomotion controller needs to be adapted to the configuration and to avoid a recompilation for each new individual, the parameters a written to a file which is read by the controller during run-time. This configuration file is usually edited by the same shell script that calls the 'robot generator' with the morphological parameters. The orientation of the leg-actuating servo can also be set, by modifying the angles mentioned on figure 16.

The legs are actually optional. A legless salamander (denoted as a *snake* throughout the rest of the document) can thus be generated. Some physical parameters can be customized, like for example the body density, the friction constant, and the bounce coefficient.

4.7 Batch mode

Webots can be launched in batch mode, in which the GUI is not created, thus saving computational time. This is useful when a large quantity of runs is to be performed. Furthermore, it allows Webots to be remotely launched.

4.8 Interface

In order to display data on a graph, *Gnuplot* can be launched from within the controller. It shows live streaming data and can also be useful for debugging purposes. Furthermore, the simulation usually launched via a shell script, outputs data to be analyzed in Matlab.

[3] In terms of fluids mechanics, two-dimensional means that the fbw (e.g. water) passes above and under the body, but not around the sides. For instance, a segment within a bar of infi nite length perpendicular to the fbw is considered as a two-dimensional body.

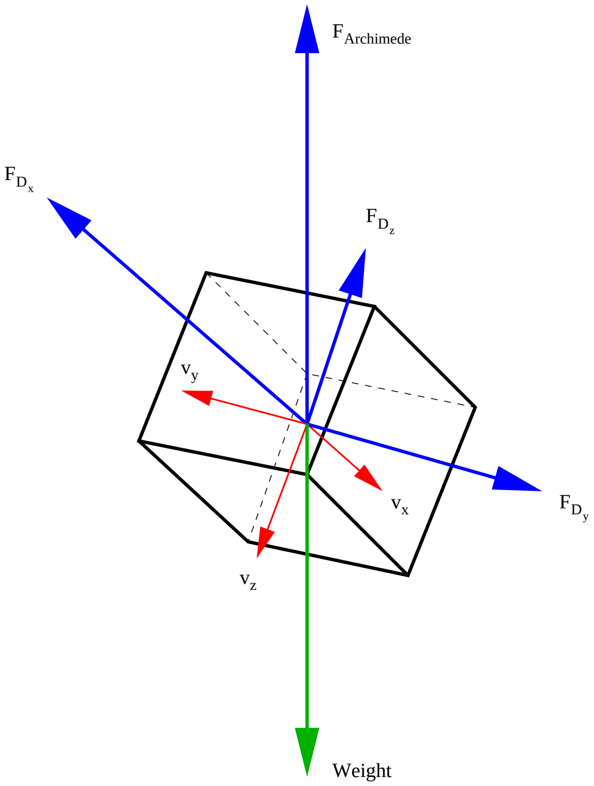

Figure 13: Model of the forces applied to a submerged body. Drag forces are handled independently along each axis of the body.

5 Research

We are now equipped with a fully customizable salamander model. But how do these parameters influence the locomotion? That is the main question we will try to answer in the present section, by characterizing the motion. The salamander's parameters can be divided in two categories:

1. The parameters concerning the physical shape, or morphology of our robot. In the Webots implementation, these are defined in the `.wbt` world description file.

2. The parameters influencing the movements. These parameters actually define the way the servos are actuated. In the Webots implementation, these are defined in the controller's C source file.

5.1 Robot morphology

In this part we study by which manner the morphology of our salamander affects its performances. The salamanders used for these batch test runs have a length of 12 segments, the front legs are attached to the third segment and the rear legs are connected to the seventh segment. For each run, which lasts seven seconds, the distance reached at the end of the run is logged as well as the number of time steps required for a stabilization of the windowed[4] average velocity[5]. A window of size 50 [time steps] gives a fair curve for this value; a too

[4] *Windowed* means that the average velocity is not computed over the whole set of samples, but only on the n last samples, in the case of a window of size n i.e. $\bar{v}(t) = \frac{1}{n}\sum_{i=t-n}^{t} v(i)$ with $v(i) = 0$ if $i < 0$.

[5] "Velocity" and "speed" are considered as synonyms in this report.

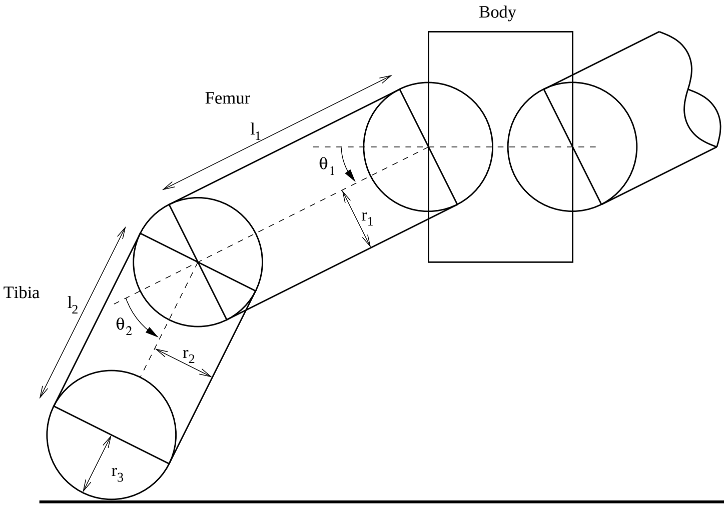

Figure 14: Leg characteristics. The rectangle represents the front of a body segment. The radii r_1, r_2 and r_3 are usually equal unless explicitly specified.

Experiment number	l_1 [mm]	l_2 [mm]	θ_1 [deg]	θ_2 [deg]	θ_3 [deg]	ϕ_1 [deg]	ϕ_2 [deg]
1	•	•	•	•	0	0	0
2	80	80	•	•	0	0	0
3	•	•	25	50	0	0	0
4	60	60	20	40	0	•	•
5	80	80	20	40	•	•	0

Table 1: Experiments on the legs morphology. The • indicates which parameters are varying during the experiment.

narrow window would prevent from ever stabilizing the average by keeping the oscillations and a too wide would lead to a strictly monotonous curve, without stable state. An example of windowed average velocity vs instant velocity is provided on figure 17. A run is said to be stable when it has acquired a value within 5% of the final average velocity and does not exceed this tolerance anymore. The reason why we study the covered distance instead of the top speed or the final speed is because the instant speed varies too much and too quickly to provide useful and comparable data. Moreover, privileging the instant speed could select an awkward configuration as being the best. For instance a salamander constantly walking in circles with a fast pace would be preferred to a sane individual trotting on a straight line with a lower velocity. Thus considering the distance reached after a given amount of time appears to be much more relevant in our case. The time to reach a stable velocity is also important, as it describes the way the robot accelerates and gives information about the stability of the gait. A stable motion will induce a constant (windowed) average velocity and thus a rather short time to stabilization. In the opposite case, an unstable gait will (in most of the cases) yield a non-constant average velocity and hence a long stabilization time.

Table 1 gives some details about the fixed and variable parameters in each experiment.

5.1.1 Experiment 1: Distribution of velocities for differently legged individuals

The software model of the salamander has so many degrees of freedom that it wouldn't be feasible to perform an exhaustive search in order to find the optimal configuration. The search space is not even discrete, it means

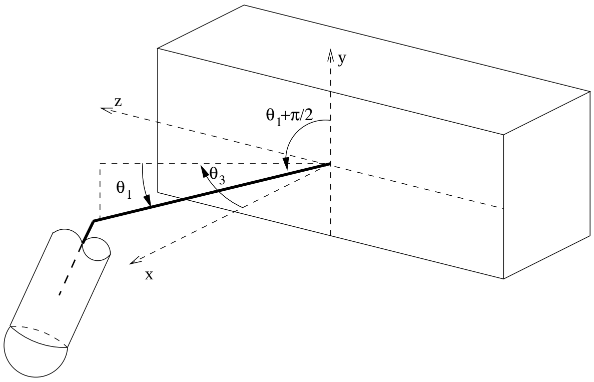

Figure 15: Leg angles. θ_1 is the angle between the xz plane and the leg, θ_3 is the angle between the x axis and the projection of leg on the xz plane. For computations, spherical coordinates (r, Θ, Φ) are used, where $\Theta = \theta_1 + \pi/2$ and $\Phi = \theta_3$.

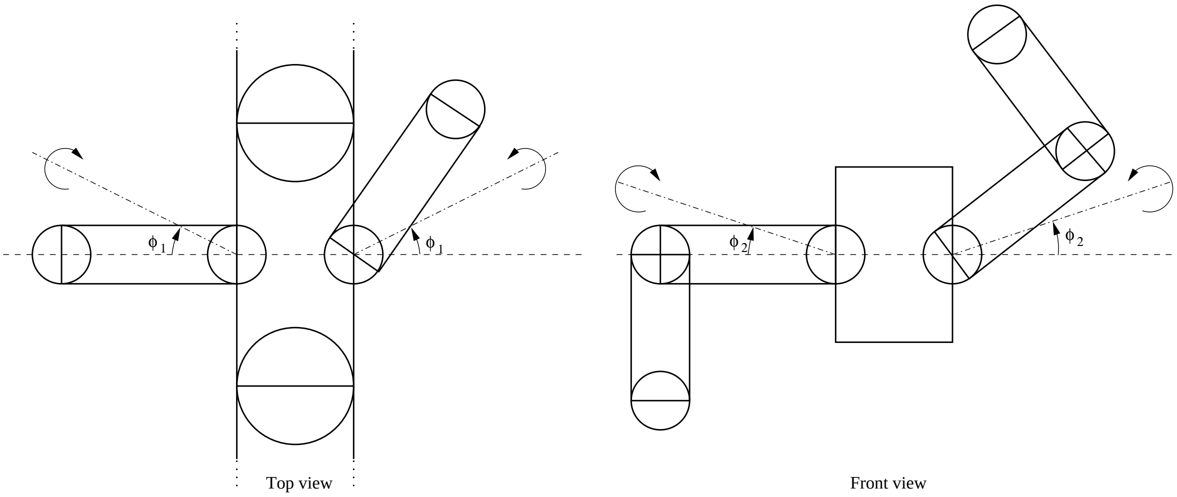

Figure 16: Leg rotation axes (dot-dashed) with their angles ϕ_1 in the horizontal plan and ϕ_2 in the vertical plan.

that the number of different possibilities is quasi infinite. We will begin by searching on a subset of the search space, by fixing some of the variable and examining sets of evenly distributed values. The variables tested in this preliminary study are: l_1, l_2, θ_1 and θ_2, respectively the femur and tibia lengths, the femur and tibia angles in the vertical plan. The results of this first experiment are displayed on figure 18. On the top left histogram, the peak around 0.1 is caused by all the individuals that cannot move themselves forward because they have either too short legs or angles that do not let the feet touch the ground.

On the top right histogram, the blank space between 0 and 50 is simply explained by the fact that no individual starts with a non-zero velocity; it is also due to the window size, which initially spans over only zero values. The first peak denotes that a fair number of salamanders stabilize their locomotion rather quickly. The peak at the end (at $t = 351$) consists of all individuals which never reach a steady locomotion. 351 is the total number of time steps for each run. One must however be careful when analyzing the time to velocity stabilization: an individual having a stabilization time of, for example, 300 time steps (according to the histogram) does not inevitably mean that after that time the velocity will remain stable; it only means that between the 300th time step and the end of the run, the windowed average velocity remained within 5% of the final value taken by this variable.

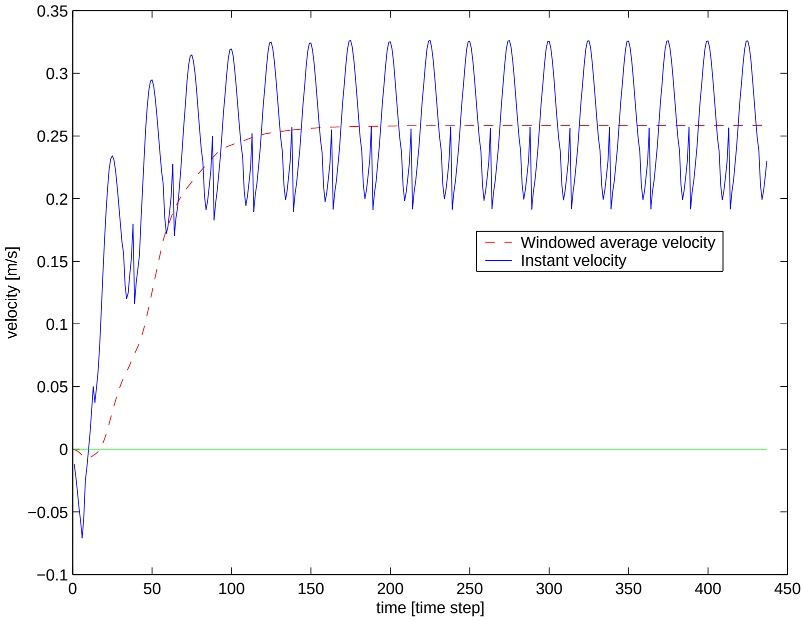

Figure 17: Example of instant velocity (blue) versus windowed average velocity (red dashed). The instant speed is here defined as the component of the velocity along the Z-axis of the segment on which the front legs are attached. The window size used here is 50 time steps.

Let us now consider only the good candidates and have a look at the stabilization time for the fast salamanders (the ones that walk a distance over 2m) which is shown on the bottom left histogram. As expected, these are not the quickest to reach the steady state since it naturally requires a longer time to approach the higher cruise speed. However, the largest subset has a stabilization time little over 100 time steps while relatively few individuals do not stabilize their velocity. This result denotes the healthy character of the fast candidates. The final configuration should be chosen among these fast-stabilizing elements.

Similarly, the distance covered by the individuals with stabilization time under 100 time steps is displayed on the bottom right histogram. It shows quite obviously that for this subset, the distance reached is inversely proportional to the quantity of candidates achieving it. The highest peak is caused by the "too short-legged" salamanders creeping on their bellies and barely moving forward. There is a large amount of them in the considered subset as they usually have short stabilization time since they quickly reach their very low "cruising speed".

5.1.2 Experiment 2: Influence of the leg angles

In order to study the influence of the angles θ_1 and θ_2 on the velocity and the time needed to reach a stable average speed, the two other parameters that were previously variable are now given a fixed value of $l_1 = l_2 = 0.08$m which is about 1.5 times the height of a body segment.

Figure 19 (left) shows the influence of the θ_1 and θ_2 leg angles on the distance reached by the end of the simulation run. One can clearly notice that as expected, $\theta_1 = \theta_2 = 0$ yields a very bad result, as the feet do not touch the ground, the salamander is incapable of moving forward. The best individual has $\theta_1 = 32$ and $\theta_2 = 34$, it reaches a distance of 2.63 m after 7 s. We can also see that the trend for the poor candidates is when $\theta_2 \approx -2\theta_1$. In that case, on one hand the foot is at the same height as the leg-body junction i.e. the foot has no contact on the ground and on the other hand, the "knee" does touch the ground, but with a dephasing of π with respect to the ideal position of the foot. The leg movement is hence uncoordinated with the body undulations thus preventing the salamander from walking away.

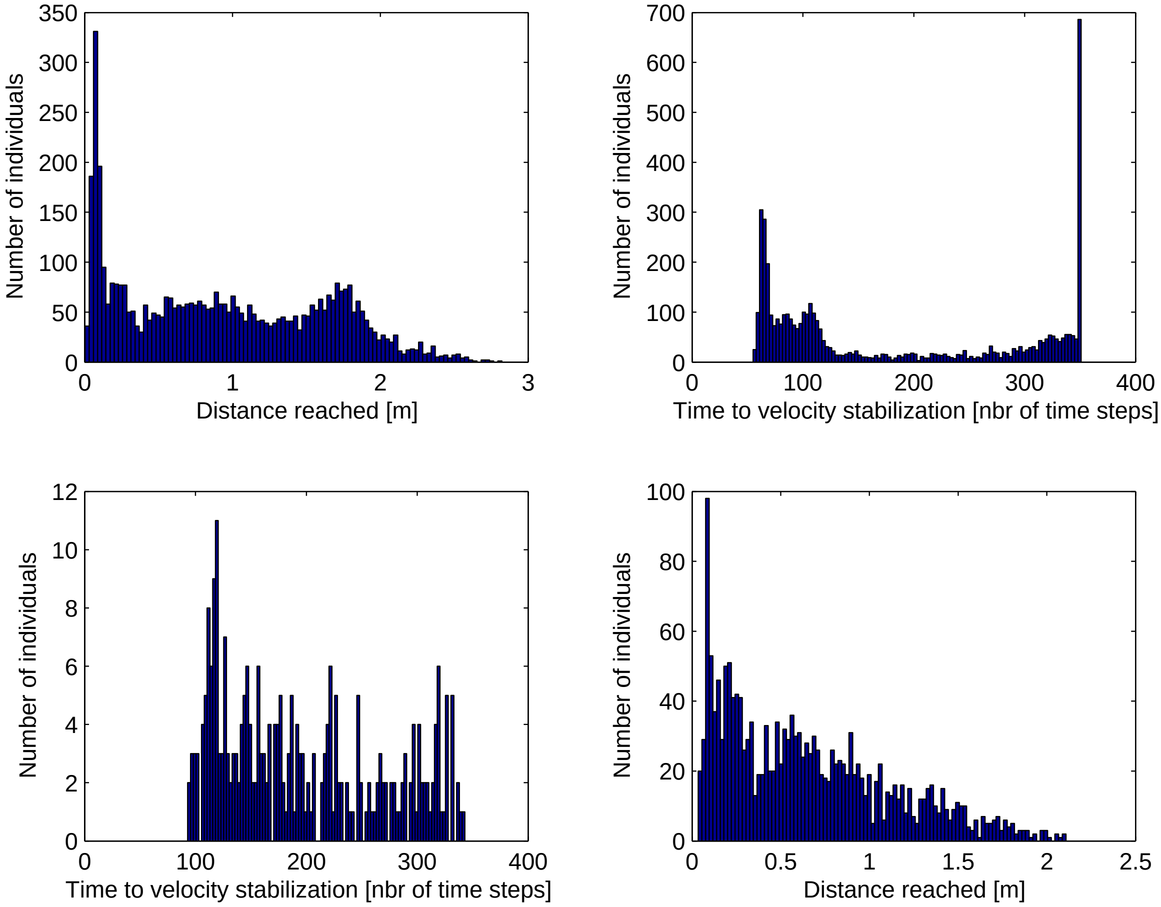

Figure 18: Experiment 1. Histograms showing the distribution of (top left) the distance reached by the individuals and (top right) the time needed for a stabilization of the windowed average. The two histograms on the bottom show (left) the time for stabilization for the individuals reaching 2m or more and (right) the distance reached by the individuals having a stabilization time less or equal to 100 time steps.

The good candidates stand along the line where $\theta_2 \approx -2\theta_1 + 100$ as this configuration tends to place the foot as far as possible from the leg rotation axis thus providing the foot a maximal velocity. We notice that the upper part of this line seems to yield a rather slow motion. This is simply induced by the lack of stability resulting from such a configuration, where the feet tend to take contact with the ground right under the body. The longest distance (2.63 m) is reached with the following values: $\theta_1 = 32^\circ$ and $\theta_2 = 34^\circ$. The stabilization time for this case is 179 time steps. The simulation for these individuals required very short time steps (1-5 ms) in order to avoid numerical explosions caused by the feet violently hitting the ground.

Figure 19 (right) depicts the time until stabilization of the windowed average speed as the leg angles θ_1 and θ_2 vary. The individuals having a short stabilization time are not inevitably the fastest ones in term of velocity. The uniform zone at the bottom left is a caused by all the candidates that never reach a steady locomotion, mostly because of the low values taken by their leg angles. They all get the same value (531) since it is the number of time steps elapsed by the end of the simulation. The individuals who quickly stabilize their motion have their angles standing along the line where $\theta_2 \approx -2\theta_1 + 20$. The sharp border (or steep slope) between the two regions mentioned above is explained by the fact that the robot achieves a steady locomotion as soon as its legs have a contact with the ground, which is only a matter of a fraction of a degree. The time required

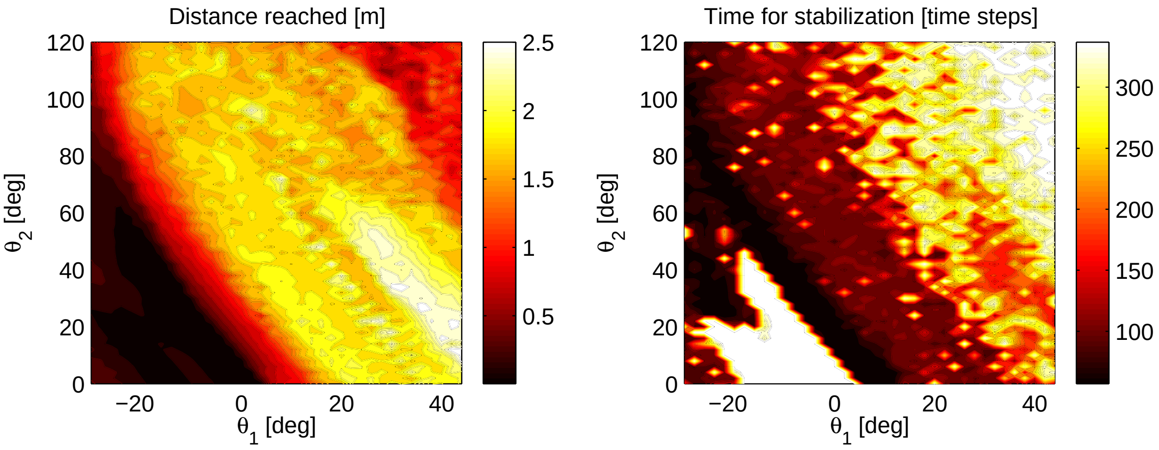

Figure 19: Experiment 2. Left: Distance reached as a function of the θ_1 and θ_2 leg angles. Right: Time needed to reach a stable velocity, as a function of the θ_1 and θ_2 leg angles.

for stabilization gradually increases as we reach the top right corner of the graph. Large angles cause the robot to loose stability. It then just crawls on the ground chaotically without ever stabilizing the speed.

5.1.3 Experiment 3: Influence of the leg lengths

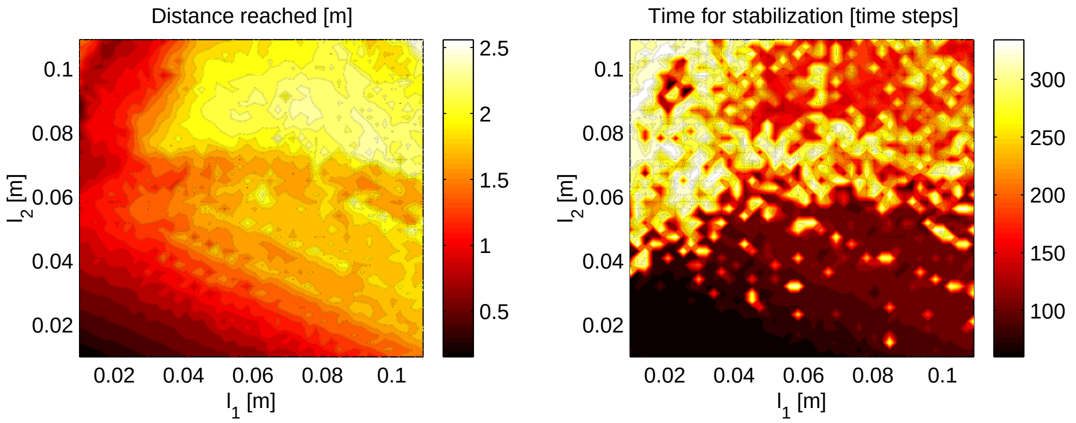

Figure 20: Experiment 3. Distance covered in 7s and time for stabilization as functions of the leg lengths l_1 and l_2. The leg angles are fixed to $\theta_1 = 25°$ and $\theta_2 = 50°$.

Figure 20 (left) shows how the covered distance depends on the leg lengths. As expected, the longest the legs, the farthest the final distance. There is however an exception to this rule: we can see that at the top left of the graph, configuration $l_1 \approx 0.02$ and $l_2 \approx 0.1$ yields poor results. This is caused by a loss of balance due to the relatively long size of the tibiae (l_2) which meet the ground almost right under the body. Consequently the salamander falls on its side and cannot move forward normally anymore. On the right part of the figure, we can notice that the time for stabilization depends more on l_2 than on l_1. Long femurs (l_1) seem to have a stabilizing effect on the average velocity as well as on the overall balance. As on the graph for the covered distance, the top left zone yields poor results for the same reason: a salamander crawling on its side can hardly achieve a steady locomotion. The settings for maximal value, in terms of covered distance are the following: l_1 = 0.107 m, l_2 = 0.109 m. This yields a distance of 2.68 m.

5.1.4 Experiment 4: Influence of the leg servo axis angles

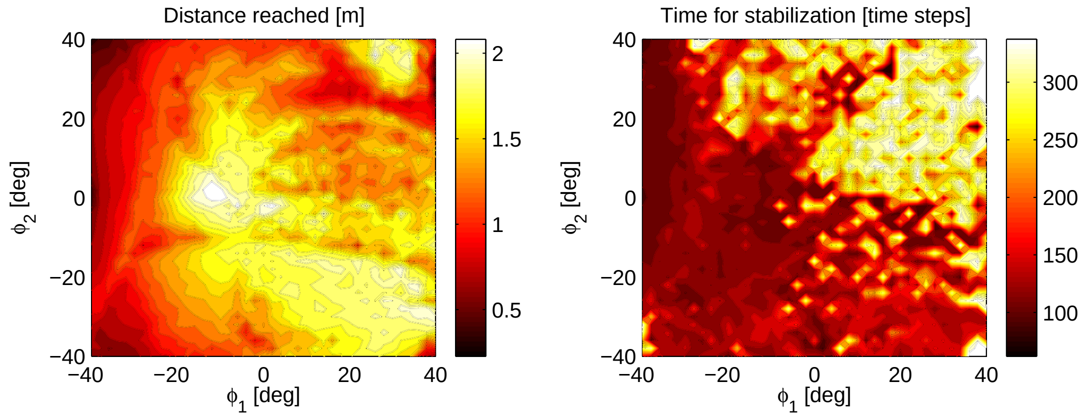

Figure 21: Experiment 4. Influence of the leg servo axis angles.

Figure 21 shows how the servo axis orientation affects the locomotion. Given this morphology, the optimum is reached when $\phi_1 = -10°$ and $\phi_2 = 0°$, the robot covers a distance of 2.17 m in 7 s with a stabilization time of 118 time steps. In that case, the leg rotation axis is slightly oriented to the back for the front legs (in the horizontal plan) and, symmetrically, to the front for the rear legs. There is a second zone of angles allowing the coverage of a long distance. It is located at the lower right angle of the graph. But in that case, the gait looks rather awkward. This configuration may even be impossible depending on the mechanical structure to be used on the real robot. Due to the high ϕ_1 angle, the feet are sometimes both on the same side of the body. For instance the left foot is on the ground while the right one is exactly above it, as a consequence of the high angle. A third zone provides good distances, but as previously mentioned, the high angle induces instabilities and this setting should be discarded. Moreover as can be deduced from the second graph, the time required for stabilization is visibly depending on the angles; high angles increase this value and should therefore be avoided.

5.1.5 Experiment 5: Influence of the leg servo axis angle and leg angle in the horizontal plan

Figure 22 depicts the way the modification of the leg and servo axis angles in the horizontal plan influence the covered distance and the stabilization time. There is a clear ridge running through the graph where $\phi_1 \approx \theta_3$ (femur axis and servo axis coinciding) and we see that the results tend to be better for positive values of θ_3 on both graphs. The optimal distance (2.32 m) is here achieved with $\theta_3 = 4°$ and $\phi_1 = 2°$. The time to stabilization is for that case 154 time steps.

5.2 Locomotion controller

In this part we examine the way the robot's performances are affected by modifying the controller's following parameters:

- A: Body Servo Amplitude. The amplitude is here defined as the angle between the two extreme positions of the body servos. An amplitude of $90°$ means that the servo rotates $90°$ in each direction.

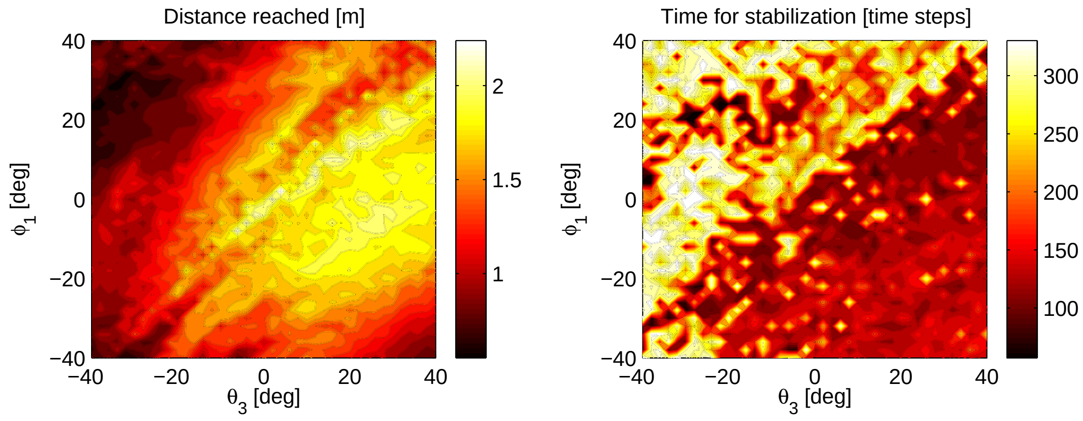

Figure 22: Experiment 5.Influence of the leg servo axis angle and leg angle in the horizontal plan.

Experiment number	A [deg]	$\Phi(t)$ [deg]	λ [body length]	ν [Hz]	environment	locomotion
6	•	0	def by leg pos	1	ground	trot
7	35	•	def by leg pos	1	ground	trot
8	30	no leg	•	•	water	swim
9	•	no leg	•	1	water	swim
10	•	no leg	•	•	water	swim
11	30	no leg	•	•	ground	serpentine
12	•	no leg	•	1	ground	serpentine
13	•	no leg	•	•	ground	serpentine

Table 2: Experiments on the locomotion controller. The • indicates which parameters are varying during the experiment. See subsection *Locomotion controller* for definitions of variables.

- $\Phi(t)$: Leg rotation function. This function defines the way the legs rotate around their axis. The leg rotation function is added to the monotonous leg rotation to yield the actual leg rotation angle in the following way: $\Theta(t) = 2\pi t + \Phi(t)$.
- λ: Body undulation wave length. For serpentine locomotion it can be set to any value but in case of legged locomotion it is defined by the leg attach position and cannot be arbitrarily chosen.
- ν: Body undulation frequency.
- environment, locomotion: the name says it all.

5.2.1 Experiment 6: Influence of the body undulation amplitude in trotting

This experiment consists of varying the amplitude of each spine-servo (thus the amplitude of the body undulations) during a legged run, causing the salamander to try all the possibilities, from a constantly straight body to a very pronounced S-shaped body swinging from side to side. That will allow us to examine the way the performances can be enhanced. Figure 23 (left) clearly shows that the body undulations help to augment the overall velocity. A large amplitude produces large steps, therefore the distance covered by each step is also larger. The distance covered with a straight (rigid) body is 1.16 m while the distance with a servo amplitude of

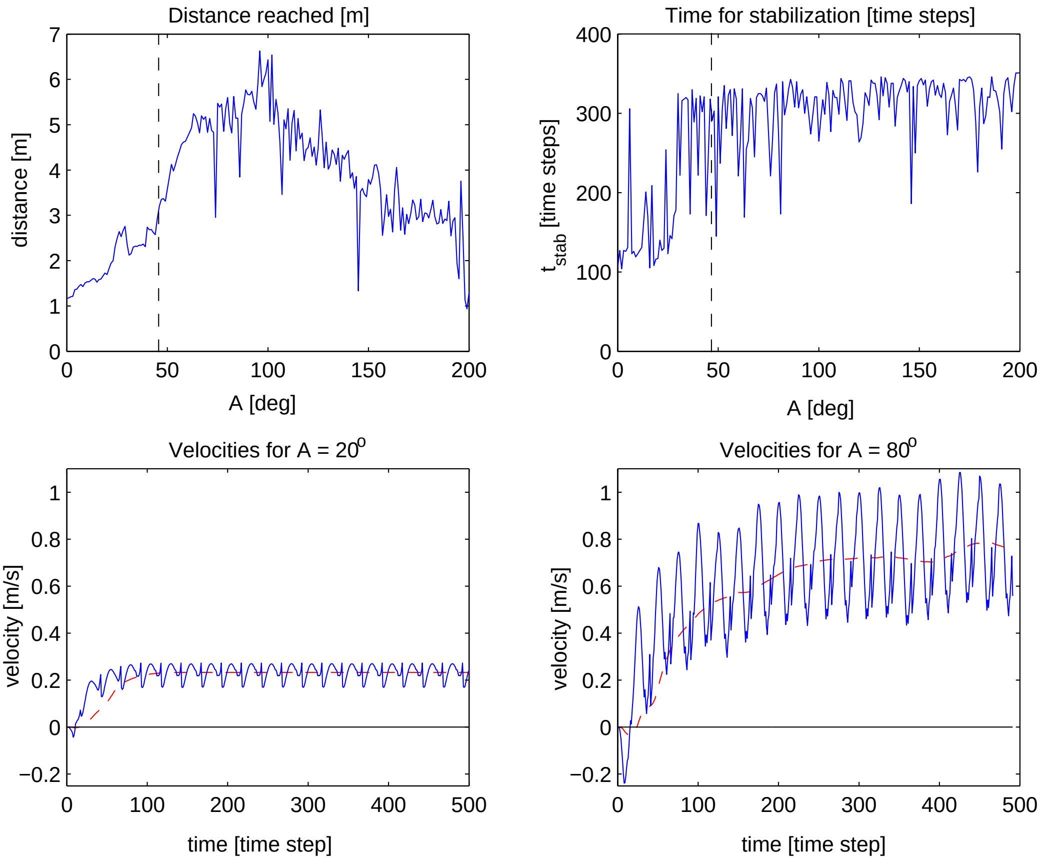

Figure 23: Experiment 6. Influence of the body undulation in legged locomotion. Distance reached (top left) and time for stabilization (top right) as functions of the servo amplitude. The dashed line marks the limit between the physically possible (left hand side) and impossible (right hand side) configurations. The two lower graphs show the instant and windowed average velocities for an amplitude of 20° (bottom left) and 80° (bottom right).

45° (about the maximal value to avoid inter-body collisions) is equal to 2.58 m thus the undulation multiplied the distance by a factor of 2.2.

On the right, we see that the stabilization requires roughly 120 times steps for amplitudes lower than 30 degrees but above that value, the graph becomes rather chaotic as the average velocity never completely stabilizes. There remains a low frequency oscillation preventing the stabilization in the way that was defined previously in the document. It *might* be a consequence of the tail swinging from one side to the other with more amplitude and consequently more force. But it is rather difficult to firmly establish the cause. According to the top right graph there seems to be a sharp change in the locomotion after around 30 degrees but actually, the gait does not *look* really different. The graph only reflects the fact that the low frequency oscillation's amplitude overlaps the ±5% strip used for this time computation. An example of that oscillation is provided on the bottom right graph, for which the servo amplitude is 80°. It clearly shows that the velocity never really stabilizes causing t_{stab} to be close to the maximal value. The graphs display data for amplitudes up to 200° which is actually a much too large value for a real robot, but it allows to have and idea about the location of the expected maximum, which is around 100°. This angle must in reality be carefully chosen since it can cause collisions between the legs and the body. The maximal servo amplitude, for this configuration, is approximately 45° (this value must be reduced if the robot should be able to turn) so the right parts of both graphs at the top can be discarded, since the simulation does not, in that case, reflect the reality: there is no collision detection between body elements in the current release of Webots. Nevertheless, it is interesting to conduct the experiment up to these high values in order to see the general shape of the curve which clearly increases, reaches a maximum and then decreases. This is a consequence of fact that there exists an optimal value for which the step length is maximal, thus letting the robot cover a greater distance by each step. The ideal candidate is to be chosen as the one with the largest servo amplitude (and low stabilization time) but without collisions between the limbs.

5.2.2 Experiment 7: Influence of the leg rotation function

Until now, the legs were rotating at a constant speed. Is this the optimal way to do? This experiment will try to provide an answer to that question.

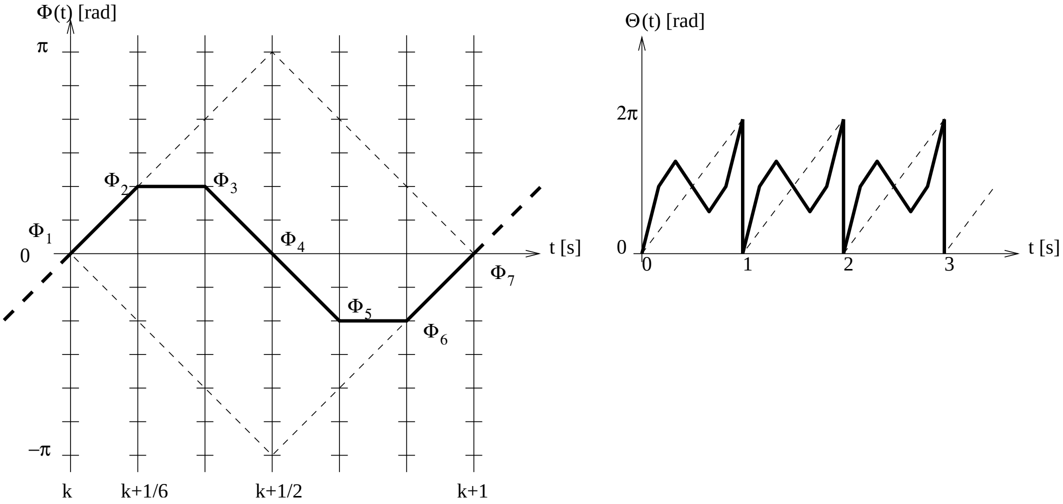

Figure 24: Example of leg rotation function (left). This function is defined by seven points, Φ_1 to Φ_7 which are respectively located (X-coordinate) at $t = k$ to $t = k + 1$, $k \in N$. Φ_1 can take any of the 13 evenly distributed discrete values between $-\pi$ and π as suggested by the ticks on the vertical axes. Φ_7 must be equal to Φ_1 since the function is periodic. Then, Φ_2 to Φ_6 can take any discrete value under the constraint that every variable can have a maximal difference of 2 'steps' with its direct neighbors. This defines the space of possible functions (dashed). The dashed area on the figure is the one corresponding to $\Phi_1 = 0$. A different value of Φ_1 causes the dashed area to shift up or down. On the left is shown the leg angle function for this particular example.

The leg rotation function $\Theta(t)$ will be modified by adding a new function $\Phi(t)$ to it. In order to keep a reasonable-sized search space, each single function $\Phi(t)$ will be defined by seven discrete points as explained on figure 24. This provides a total number of 22763 new leg rotation functions, including the usual case where $\Phi(t) = 0$. It does also handle simple dephasing by letting $\Phi(t)$ equal a constant.

The results of this experiment are displayed on figure 25, especially the four functions yielding the fastest individuals. The first thing that can be noticed is the fact that all of them are above the $\Phi(t) = 0$ case. This means that a slight positive dephasing (i.e. when legs rotate 'earlier' than in the usual case) may be beneficial. The fastest individual covers a distance of 2.52m. One important result is that the individual with $\Phi(t) = 0$ ends up at the 375th position (among the 22763 candidates, it is still in the top 1.65%) with a distance of 1.59 m as displayed on the histogram. The distance difference of almost 1 meter between this individual and the best one clearly shows that modifying the $\Phi(t)$ function can have a positive effect on the velocity, it this case, the covered distance has been multiplied by 1.58.

The time to perform the 22763 runs of this experiment was about 3 days on a Pentium 4 PC.

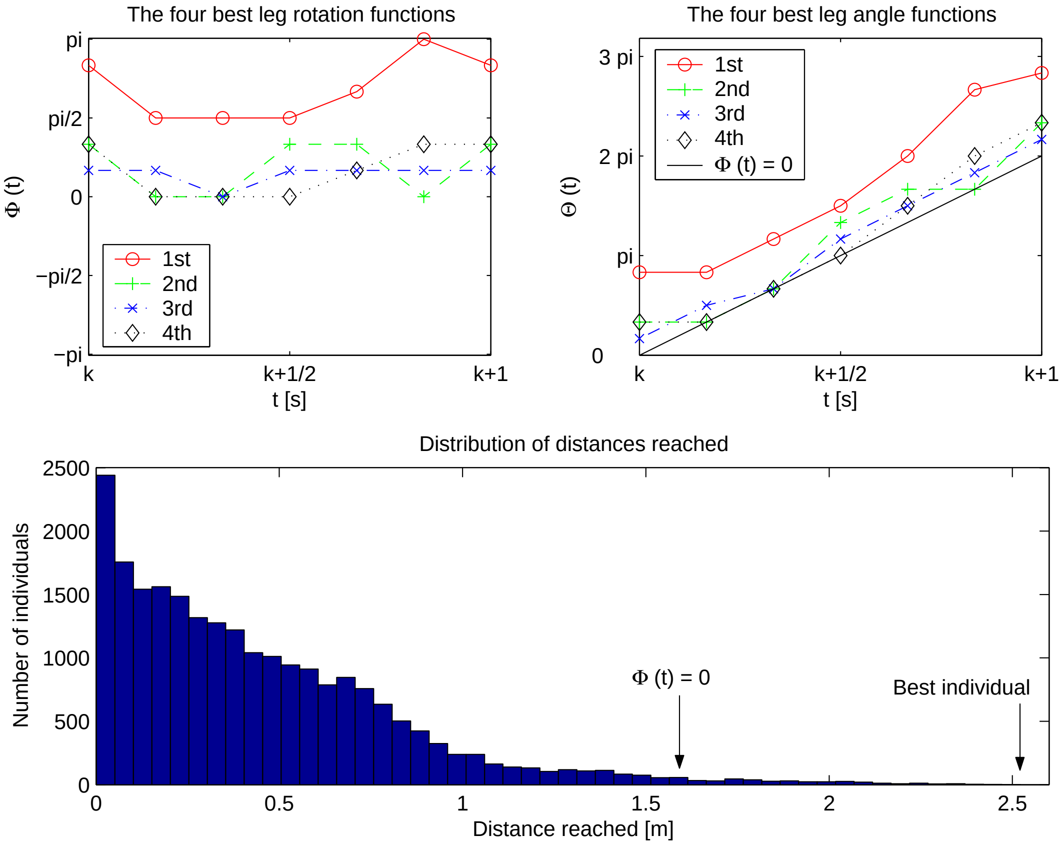

Figure 25: Experiment 7. Non-constant leg rotation. The leg rotation functions $\Phi(t)$ yielding the four fastest individuals (top left), the same functions added to the constant rotation (represented as well), thus giving $\Theta(t)$ (top right) and the distribution of the distances reached (bottom).

5.2.3 Experiment 8: Influence of the body undulation frequency and wave length in swimming

How do the frequency and wave length interact on a swimming 12-segment snake? In this experiment, the servo amplitude is fixed to $30°$. Figure 26 shows the results. The optimum for a $30°$ amplitude stands at $\nu = 5$ Hz and $\lambda = 2$ body lengths.

We can notice several interesting points: the expected null values along the lines of coordinate $\lambda \approx 0$ or $\nu = 0$ caused in the first case by the fact that a too short wavelength cannot induce a traveling wave, due to the “granularity” of the snake body (the segment length) and in the second case, a null frequency is a synonym of immobility. It is also interesting to see that for fixed frequencies under 2 Hz the body wave length has almost no effect on the velocity, and similarly, the frequency has no influence on speed for fixed wave lengths under 1 body length.

The performances decrease when the frequency increases mostly because each servo’s maximal amplitude (desired position) is only partially reached (real position), as it was shown on figure 10, and since a lower amplitude induces less power the covered distance gets shorter.

5.2.4 Experiment 9: Influence of the body undulation amplitude and frequency in swimming

As displayed on figure 27, the augmentation of the oscillation amplitude first causes the velocity to increase up to a particular point, here around $20°$, where there is an optimal trade-off between large amplitudes which

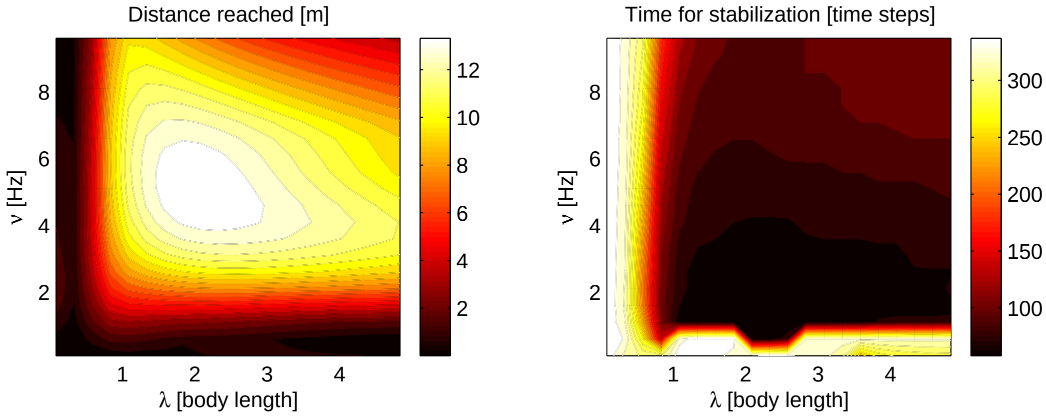

Figure 26: Experiment 8. Distance reached and time to stabilization as functions of the body oscillation wave length and frequency for a swimming snake.

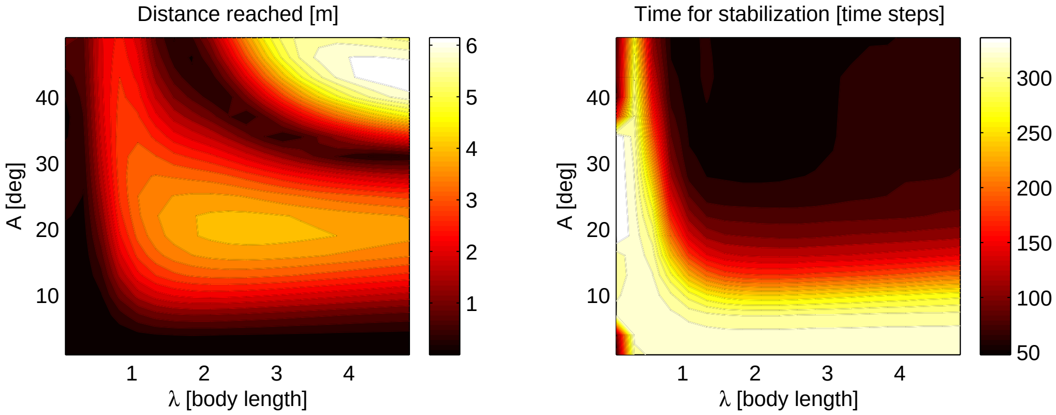

Figure 27: Experiment 9. Distance reached and time to stabilization as functions of the body undulation amplitude and frequency for a swimming snake.

induce more power in the movements but lengthen the way by creating S-shaped trajectories, and small amplitudes taking the shortest path but with less power. The 'valley' running from the top border of the graph to the right one is due to the fact that at a given amplitude, the snake's trajectory will resemble an '8' and thus make the robot remain at its original location. Past this point, the performances increase again as the snake starts to swim making loops which, when they 'unfold', provide a strong thrust to propel the robot. The top right corner of the 'distance reached' graph depicts this effect. The distance covered by the loop-making individuals is longer than the regular ones, however this performance would not be feasible in reality as it would induce inter-body collisions (currently not handled by the simulator).

The graph reaches its peak value at $\lambda = 2.6$ body lengths and $A = 5$ Hz. A body wave length of 2.6 means that the wave described by the body is shorter than an entire cycle, causing the tail to swing from side to side, the way fish do to propel themselves.

5.2.5 Experiment 10: The fastest swimmer with parameters bounded to real values

The low number of parameters (3) that enter into account in legless swimming allow an exhaustive search for the best swimmer, when the search space is bounded with the limitations of the real robot ($A_{max} = 30°, \nu_{max} = 0.5$ Hz). The best 12-segments individual, in this 'realistic' configuration has the optimal settings given in the following table.

$A^\star$ [deg]	$\lambda^\star$ [body length]	$\nu^\star$ [Hz]	$\bar{V}$ [m/s]
18	2.3	0.5	0.25

However, as seen in experience 8, for such low frequencies the wave length (above 2 body lengths) has almost no influence.

5.2.6 Experiment 11: Influence of the body undulation frequency in serpentine locomotion

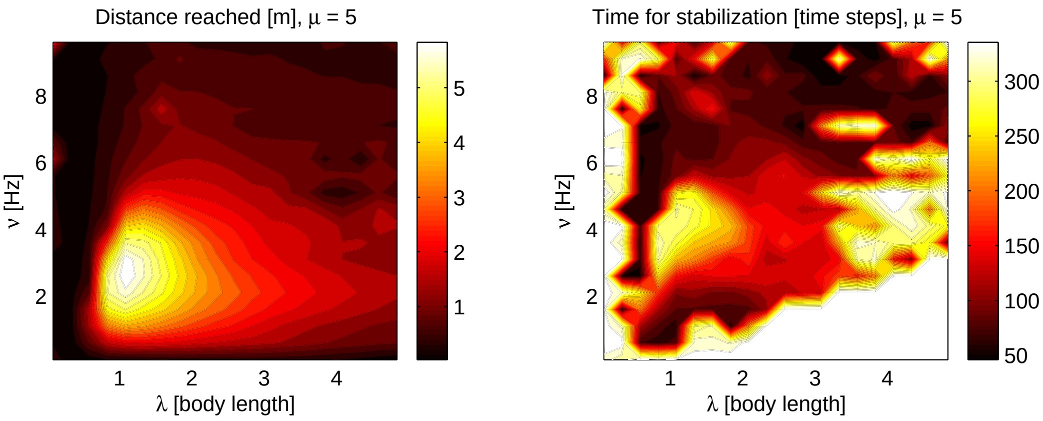

Figure 28: Experiment 11. Distance reached and time to stabilization as functions of the body servo amplitude for a snake locomoting on the ground.

As soon as Webots was made compliant to asymmetric friction, terrestrial serpentine locomotion could be implemented and tested. This experiment is the same as experiment 8, with the only difference that the snake locomotes on the ground, using differential friction coefficients. These are fixed to the following values: $\mu_x = 0.5$ (medium high friction) and $\mu_z = 0.1$ (low friction). Their ratio, the *friction ratio* is defined as $\mu = \mu_x/\mu_z$.

The highest velocity is reached by the individual with $\lambda = 1.08$ body lengths and $\nu = 2.6$ Hz, by covering 6.12 m, as displayed on figure 28. The interesting point is that we can now compare the aquatic and terrestrial serpentine locomotion. When compared to experiment 8, it appears quite obviously that for a same amplitude ($30°$) the optimal frequency as well as the optimal wave length are lower. The snake robot does not require a particularly high frequency to produce thrust, contrary to the swimming robot, whose drag forces are proportional to the square of the velocity, thus low for small speeds. We can also notice that only a short range of wave lengths provides results near the optimum. A body describing an almost entire wave length appears to be necessary for an efficient locomotion. Another noticeable feature, is the very poor results yielded by very short wave lengths: in those cases, and given the granularity of the body, the segments tend to rotate around themselves, letting the movement relative to the ground be rather transversal than longitudinal and therefore the high longitudinal friction coefficient μ_x will slow down the motion.

5.2.7 Experiment 12: Influence of the body undulation wavelength and amplitude in serpentine locomotion

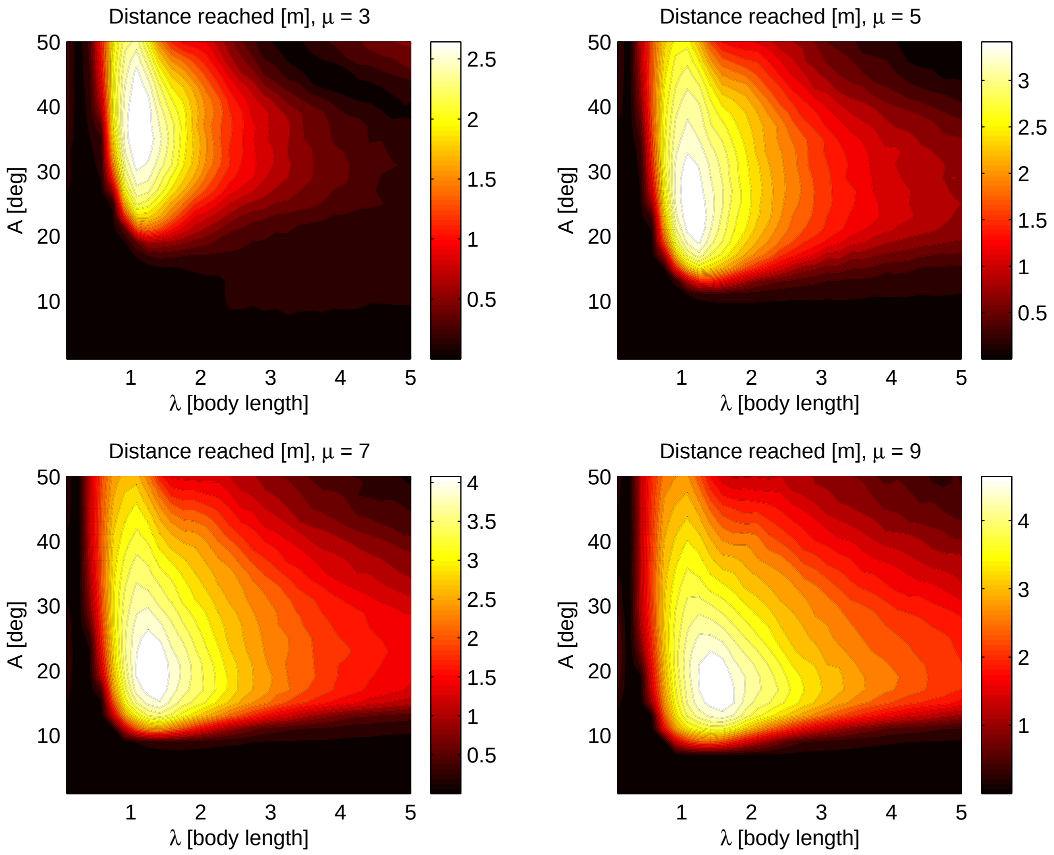

Figure 29: Experiment 12. Distance reached as function of the body undulation wavelength and amplitude for a snake on the ground. Each graph shows the distance covered with a different friction ratio μ = 3, 5, 7 and 9. The frequency is fixed to $\nu = 1$ Hz.

This experiment is basically the same as experiment 9, except the environment which is in this case the ground. The results are shown on figure 29, with different values of friction ratios. The performances seem to decrease much faster when the body wavelength increases as in the swimming motion. This means that the swinging tail motion does not work well on earth. The snake requires shorter wavelengths and higher frequencies (thus higher relative water velocity) in order to generate the drag forces which in turn will propel the body. This is a consequence of the drag forces being proportional to the squared velocity, hence low speeds will produce very low drag. In the contrary, the asymmetric friction of the snake's body on the ground can act even by very low speeds.

A low friction ratio apparently requires a higher amplitude to reach the optimum. On the top left graph, we see for instance that with $\mu = 3$ the peak exceeds the real robot's $30°$ amplitude boundary. However, in this case the frequency used is twice as high as the real servos highest frequency. The optimal configuration for the real robot will be determined in the next experiment.

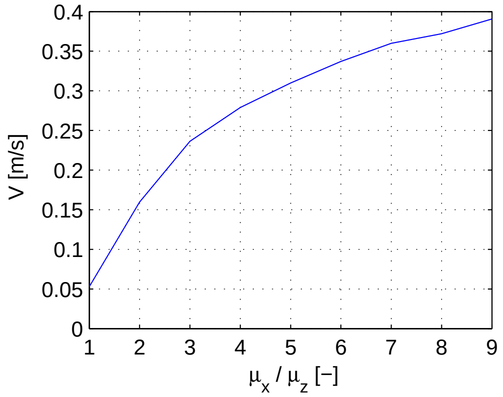

Figure 30: Experiment 13. Average velocity of snakes with servo amplitude and frequency bounded to the real values, as a function of the friction ratio.

5.2.8 Experiment 13: The fastest snake with parameters bounded to real values

This experiment is similar to experiment 10, except that it happens on the ground. There is however another difference: the friction coefficients μ_x and μ_z between the ground and the body, which allow serpentine locomotion, are still unknown, furthermore they can take different values depending on the materials chosen to produce them. The question is now: How important does the ratio $\mu = \mu_x/\mu_z$ need to be in order to induce a 'good' velocity? The answer should give an idea of the importance of the materials or the device (e.g. rubber wheels which can easily simulate a high μ) needed to produce the friction. Figure 30 Shows the average speeds (computed from 30 sec. runs) of the best individuals with $A \leq 30°$ and $\nu \leq 0.5$ Hz. The longitudinal friction coefficient μ_z has been set to 0.1 for all runs and the transversal one, μ_x, ranges from 0.1 to 0.9. The good surprise is that the curve is bent upward, it means that even low friction ratios can yield fair velocities. For instance, to achieve half of the average speed of the best individual with $\mu = 9$, you do not need $\mu/2$ but a lower value ($\approx \mu/3.6$) which is easier to obtain than $\mu/2$.

These 'best individuals' cited above all have different optimal amplitudes and wave lengths, provided in the following table:

μ [-]	$A^\star$ [deg]	$\lambda^\star$ [body length]	$\nu^\star$ [Hz]	$\bar{V}$ [m/s]
1	30	1.0	0.5	0.0530
2	30	1.2	0.5	0.1597
3	28	1.3	0.5	0.2363
4	24	1.5	0.5	0.2790
5	22	1.7	0.5	0.3100
6	20	1.7	0.5	0.3370
7	20	1.8	0.5	0.3600
8	20	2.2	0.5	0.3720
9	20	2.0	0.5	0.3907

According the experiment, the friction ratio μ should be as high as possible and the frequency as well. The amplitude and wave length should then be adapted depending on the value of μ to provide a maximal velocity.

From this table, we can deduce the limiting factors. The frequency is in every case equal to the upper limit, 0.5 Hz. Moreover, for friction ratios under 3, the amplitude also becomes a limiting factor.

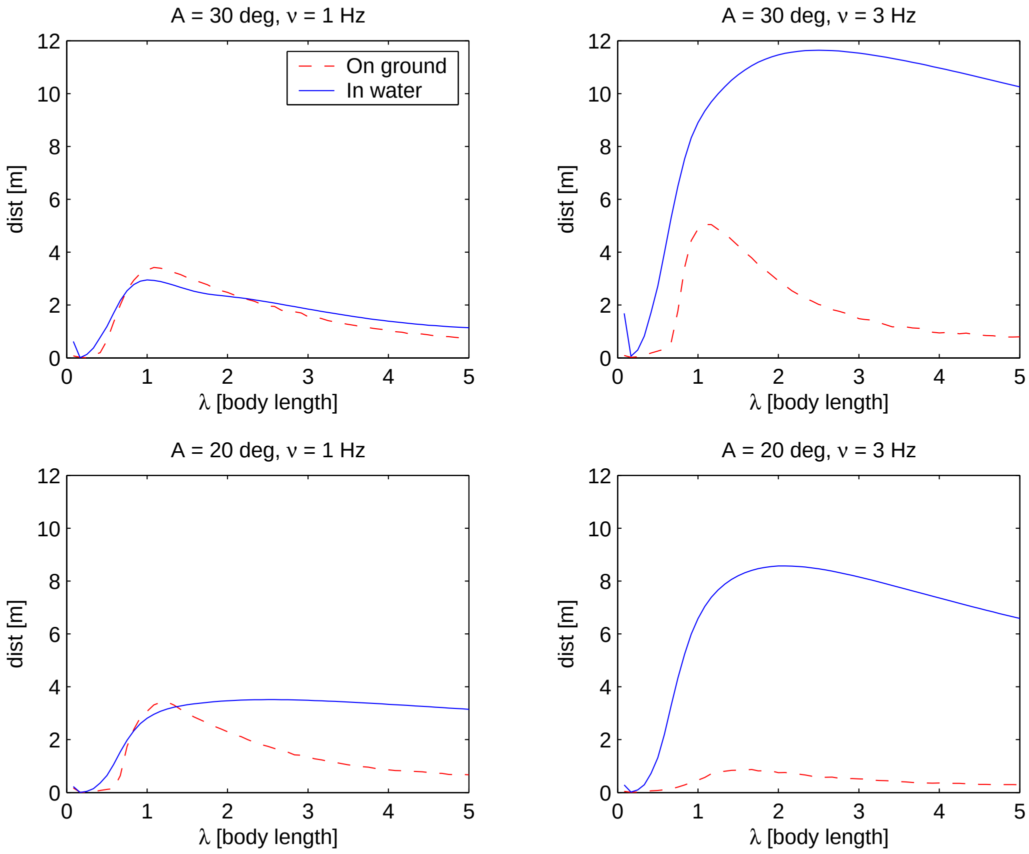

Figure 31: Experiment 14. Influence of the body wave length in terrestrial (dashed line) and aquatic (continuous line) locomotion.

5.2.9 Experiment 14: Comparison of optimal wave lengths in swimming and serpentine locomotion

We have roughly seen that the optimal frequencies, wave lengths and amplitudes differ depending on the environment. In order to provide a better view of the differences, comparative runs can be conducted. By fixing the amplitude and the frequency we can compare the effect of the body wave length on the velocity and by repeating the operation with different settings, the evolution can be guessed. The friction ratio μ is here fixed to 0.5. Figure 31 shows the performances for the same individual on the ground and in the water. As previously seen, the terrestrial locomotion has a narrow peak around the optimal wave length whereas the aquatic motion does not especially suffer from high λ's. This behavior can be seen in the nature, e.g. fish swinging their tail from side to side by bending their body in one way only, contrary to real snakes undulating by creating numerous traveling waves.

We can also notice that swimming can be much more effective, provided the frequency is over 1 Hz (which is unfortunately above the upper limit for the real robot's servos). This all shows that a modification of the undulation is recommended when passing from earth to water.

6 Discussion

The simulated salamander robot performs realistic motions, in the three different types of locomotion tested. However, a fair number of variables and coefficients have been rather arbitrarily set, since the exact value they shall take is still unknown. The determination of these values will precisely be one of the tasks to be done when the real robot's assembly will be achieved. Among these unknown parameters are: the friction coefficient between the robot and the ground, the mass and volume of the robot's segments (which will determine the density), the center of gravity's (CG) exact position.

Some other parameters have been approximated, like for instance all the drag coefficients, the external forces simulating the effects of water. Archimedes' force is an approximation when the salamander is only partially immersed, the drag force's computation is also an approximation since it is in our case a vectorial summation of forces separately computed along each axis. Moreover, As all body segments are considered to be two-dimensional (in terms of fluids mechanics), the head and the tail segments are slightly farther away from reality than the mid-body segments since the former have only one neighbor thus allow the water flow to pass around one of their sides. Similarly, when the body is only partially immersed, The drag forces are computed just as in the completely-immersed case.

The motion of the water itself is not modeled. There is no wave at the surface in the simulation, especially no "splash" effect with all the underlying physical characteristics (waves, water movement, turbulent flow, etc.) when the robot is dropped above the water from a given height. That approximation can lead to numerical explosions when the drop height exceeds a certain level. The drag force, which is proportional to the square of the velocity, suddenly passes from null to a very large value which can initially not be thwarted by the gravity force and hence catapults the salamander upward at high speed.

Some parameters have voluntarily been given a very high value, for instance the servo forces and their speed in order to perform experiments without being limited by this factor, which could interfere with the effects of the studied parameters.

Some inaccuracies are due to the simulator (Webots) itself, like for example the fact that the velocity of a solid object is defined as the velocity of the object's "origin" point [6]. External forces are also applied on this particular point. In our case, it is located at the middle of the axis around which the segment servo rotates. That detail can have catastrophic effects on the simulation (numerical explosions). For example, consider the case where one of the segments spins around its servo axis (which does not intersect the CG). The velocity returned when questioning the simulator is null and therefore the drag force applied on that element will also be null, although the body's CG velocity is everything but null. However, the importance of this effect on locomotion could not be clearly identified but seems to be rather low.

Webots does currently not handle inter-body collisions i.e. limbs belonging to the same body do not collide, they pass through each other just as nothing happened. This is completely unrealistic. However, if one takes care of not letting these situations happen (by fixing limits in the limbs' ranges) the simulation remains accurate.

Nevertheless, since every instance of the salamander model suffered from the same approximations, comparative tests (which were the core of the present work) are still valid and provide relevant information. There was however an exception, a case where some models had an advantage over others. At the beginning of every simulated run on the ground, the salamander is 'dropped' from a mid-air position, to avoid cases where the length of the legs cause the feet to be under the ground. That leads to unrealistic behaviors as the simulation starts. In the case of a fixed start-altitude, some long-legged specimen would touch the ground sooner than short legged individuals, and thus spend more time locomoting. To prevent that inequality, the salamanders were dropped from a height at which the distance between the feet and the ground was constant (with a special case for individuals having non ground-touching legs, i.e. crawling on their belly, which are dropped directly on the ground).

7 The future

This section proposes an overview of some of the experiments that should be conducted with the salamander robot, as a continuation of the present work.

[6] This origin point is different from the center of gravity when `transform`s are used, in order to shift the body's shape.

7.1 Power consumption

The power consumption is an important issue, which directly influences the robots mobility by reducing or extending its ’lifetime’. As mentioned at the beginning of this document, the limbs’ natural dynamics can spare energy if they are accordingly designed. the measurement of the power consumption is another feature that is currently unavailable in Webots.

7.2 Using the real parameters

The simulation shall in the future use more realistic parameters, bounded by some value. The body servos have a maximal oscillation frequency, angular velocity and the amplitude is limited by the segment’s shape. These parameters must be determined and integrated in the salamander model.

7.3 Friction

The friction coefficients between the ground and the robot will have to be determined on the real robot so that they can be accordingly adjusted within the simulation. They have up to now been set to an arbitrary value. For example, they could be experimentally adjusted by letting the robot slide on a slope and compare the average speeds in the real and simulated worlds.

7.4 3D locomotion

3D locomotion, i.e. having the ability to turn left and right, must definitely be studied, by taking into account the body’s dynamics. Performing a turn can be simply done by adding an offset value to each body servo’s α_i angle. A positive offset will make the robot turn right and a negative one will make it go left (it may be reversed, depending on the servo’s orientation). Early test runs performed during this work have demonstrated a very good behavior for turning in serpentine locomotion (both on the ground or in the water). Turning while trotting was however more difficult achieve and the observed effect was actually the opposite of the expected one. A “turn left” order sometimes resulted in a rather poor right turn. A Java user interface was created to allow a direction control of the robot.

7.5 Inter-body collision detection

As mentioned in the discussion, inter-body collision detection is currently unavailable in Webots. It should be implemented as this feature will play a more important role as the simulated runs shall come closer to reality. In particular, during turns, the probability of leg collisions increases and if such an event is not taken into account, the outcome of the simulation can be radically different.

7.6 Transitions between ground and water

Transiting from one element to the other, one of the key points in amphibious locomotion, must be carefully planned as it is a critical moment. Even the best swimmer *and* trotter would make a poor amphibious if it is unable to crawl out of the water. The parameters that must be tested are for example the strategy for detecting water, the timing of locomotion transition or the use of an ’intermediate’ gait (e.g. front legs rotating while rear legs and tail still in a swimming mode).

7.7 Max torque

The maximal torque needed to perform a given movement in the simulation may be unachievable by the real servos. This point has been discarded in the current work, by setting the maximal servo force to a very large value in order to explore the locomotion without being limited by this factor. But that limit shall be observed for a high-fidelity simulation.

7.8 Crossing

Crossing capabilities (passing over an obstacle or climbing a step) have not yet been addressed but shall be a one of the determining criteria in the search for the optimal leg and body configuration. Climbing is precisely one of the advantage of legged locomotion and motivates its choice.

7.9 Comparison with the real robot

Comparing the simulated model with the real robot will allow to tune the model's settings in order to approach reality. The model currently used to simulate water could also be enhanced (empirically) to approach a more realistic behavior, for example the effect of water motion, waves generated by the robot, etc. Comparative test runs should be performed in parallel, in the real world and in the simulator.

7.10 Genetic Algorithms

Because of the very large search space and the correlation between all the parameters, the most suitable tool to reach an optimum is certainly the genetic algorithm — optimization through evolution and selection, survival of the fittest. The parameters that should play a role in the fitness function are the velocity (or distance between the start and final positions, both on the ground or in the water), the time of velocity stabilization, the crossing capabilities, the power consumption, the stability, the ability to crawl out of the water, etc. The use of a genetic algorithm for optimization will become relevant only when all parameters have been correctly set in the simulation to match reality.

7.11 Concertina locomotion

Concertina locomotion can by achieved by undulating in the vertical plan, that is by initially laying the robot on its side. The disadvantage of this locomotion, is that with its 1 DOF joints, the robot would loose its ability to turn. It has already been tested in simulation.

8 Conclusion

A software model of the salamander has been created in the Webots robot simulator. The salamander can demonstrate trotting, swimming and serpentine locomotion. A program allowing the generation of custom-shaped salamanders (with or without legs, different dimensions and lengths, angles, controller settings) was developed as well. The robot software model and Webots both require some additional refinements to obtain a high-fidelity simulation and prediction tool, but the versions used in the present work were accurate enough to provide useful informative and comparative data. Test runs were performed in order to explore the characteristics of the salamander robot's locomotion using the three different motions. The influences of the different parameters on the velocity and its stability were analyzed.

Experiments showed that during legged locomotion, in order to maximize the velocity, the circle described by the feet spinning should be as far as possible from the body with a radius as large as possible. That means that long legs produce higher velocity, provided the stability is assured. Long femurs seem to have a stabilizing effect when the leg angles are set to a reasonable value. Slightly Changing the orientation of the leg rotation axes can be beneficial but it should remain in the horizontal plan to avoid instabilities. Body undulations can help to increase the velocity but they are limited by the risk of inter-body collisions, e.g. a leg hitting the trunk or the another leg. It has also been demonstrated that legs rotating at a constant angular velocity is not the optimal solution and the distance covered can be greatly improved by modifying the rotation function, causing for instance the legs to hit the ground with a higher velocity and hence produce an impulsion.

When swimming, the optimal body wave length appears to be over 2 body lengths in which case the tail swings from side to side the way fish do. The optimal frequency, body amplitude and wave length for swimming have been determined for a 12-segments swimming snake with the frequency and amplitude bounded to values corresponding to the real robot's limits.

In serpentine locomotion on the ground, the optimal wave lengths are generally shorter than in the water, to avoid, among other phenomena, a swinging tail which is not adapted to this terrestrial type of locomotion. It appears that in the water, contrary to the motion on the ground, a slow motion will hardly produce a forward movement as the 'propelling forces' are proportional to the squared relative velocity.

Simulations with different transversal-to-longitudinal friction ratios (which allow serpentine locomotion) have shown that the highest it is, the fastest the snake becomes. however, the augmentation of velocity does not grow linearly as the ratio increases; it is profitable for low ratios.

The real servo's top frequency of 0.5 Hz is in most of the cases a limiting factor. Faster servos would greatly improve the salamander's velocity. It would Naturally have an effect on the power consumption.

The present work has paved the way for later work aiming to define the salamander robot's definitive morphology and controller. It has provided a research tool which will allow optimization process to be conducted (e.g. genetic algorithms) in order to reach that goal.

References

[1] Leg Lab, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
http://www.ai.mit.edu/projects/leglab/robots/robots.html.

[2] Ambulatory Robotics Lab, Center for Intelligent Machines McGill University, Montreal.
http://www.cim.mcgill.ca/∼arlweb/.

[3] Humanoid Robotics Institute, Waseda University, Tokyo.
http://www.humanoid.rise.waseda.ac.jp/booklet/kato_4.html.

[4] LUCY, Multibody Mechanics Group, Department of Mechanical Engineering, Vrije Universiteit Brussel.
http://lucy.vub.ac.be/.

[5] Bio-robotics Laboratory, Technical University Delft.
http://mms.tudelft.nl/dbl/.

[6] Institute for Applied Mechanics, Technical University Munich.
http://www.amm.mw.tu-muenchen.de/index_e.html.

[7] Geo-I, II.
http://uirvli.ai.uiuc.edu/tlewis/pics/geoII.html.

[8] Running Dog Robot, Artificial Intelligence Lab, University of Zurich.
http://www.ifi.unizh.ch/ailab/people/iida/research/rdproj/.

[9] Kimura laboratory, University of Electro-Communications, Tokyo.
http://www.kimura.is.uec.ac.jp/.

[10] Humanoid Robotics Group, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
http://www.ai.mit.edu/projects/humanoid-robotics-group/index.html.

[11] Boadicea, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
http://www.ai.mit.edu/projects/boadicea/.

[12] Poly-PEDAL Laboratory, University of California, Berkeley.
http://polypedal.berkeley.edu/Bioinspire/Robotics.html
http://www.berkeley.edu/news/media/releases/2002/09/rfull/home.html.

[13] RHex. http://www.rhex.net/.

[14] Sprawl robots.
http://www-cdr.stanford.edu/biomimetics/documents/sprawl/.

[15] Biologically Inspired Robotics Lab, Case Western Reserve University, Cleveland Ohio.
http://biorobots.cwru.edu/.

[16] UIUC Hexapod, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign.
http://soma.npa.uiuc.edu/labs/nelson/hexapod.html.

[17] Scorpion, Group Biomimetic Autonomous Robots, Fraunhofer institute of Autonomous Intelligent Systems.
http://ais.gmd.de/BAR/research.htm.

[18] Hirose + Yoneda Robotics Lab, Tokyo Institute of Technology.
http://www-robot.mes.titech.ac.jp/home_e.html.

[19] Snake robots by Dr Gavin Miller.
http://www.snakerobots.com/.

[20] gmd-snakes, Group Biomimetic Autonomous Robots, Fraunhofer institute of Autonomous Intelligent Systems.
http://www.ais.fraunhofer.de/BAR/snake.html.

[21] *NASA's 'Snakebots' Slither to Life*. Space.com news, 2000
http://www.space.com/news/snakebots_000504.html.

[22] Biomimetic Underwater Robot Program, Marine Science Center, Northeastern University, Boston.
http://www.neurotechnology.neu.edu/.

[23] Frank Nickols Underwater eel robot, Nanyang Technological University, Singapore.
http://www.ntu.edu.sg/home/mfnickols/eel.

[24] Ken McIsaac, Department of Bioengineering, School of Engineering and Applied Sciences, University of Pennsylvania.
http://www.cis.upenn.edu/∼kamcisaa/.

[25] RoboTuna, Massachusetts Institute of Technology.
http://web.mit.edu/towtank/www/tuna/brad/tuna.html.

[26] PPF-09, National Maritime Research Institute, Tokyo.
http://www.nmri.go.jp/eng/khirata/fish/model/ppf09/index_e.html.

[27] *Robot fish to resurrect fossils*. BBC online news, 1999.
http://www.space.com/news/snakebots_000504.html.

[28] VCUUV: Vorticity Control Unmanned Undersea Vehicle, Draper Laboratory.
http://www.draper.com/tuna_web/vcuuv.htm.

[29] Simon Capern. *Modélisation d'un Robot Salamandre.* Semester project, EPFL, 2003.

[30] J.J. Collins and S.A. Richmond. Hard-wired central pattern generators for quadrupedal locomotion. *Biological Cybernetics*, 71(5):375–385, 1994.

[31] Jonathan E. Clark Jorge G.Cham Sean A. Bailey Edward M. Froehlich Pratik K. Nahata Robert J. Full Mark R. Cutkosky. *Biomimetic Design and Fabrication of a Hexapedal Running Robot*. IEEE International Conference on Robotics and Automation, 2001.

[32] Q. Huang, K. Yokoi, K. Shuuji, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation*, 17(3):280–289, 2001.

[33] Auke Jan Ijspeert. *Design of artificial neural oscillatory circuits for the control of lamprey- and salamander-like locomotion using evolutionary algorithms*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1998.

[34] Auke Jan Ijspeert. Biologically Inspired Artificial Intelligence Course (Uni Zurich, WS03 410), 2003.

[35] Harm Ikink. Baps lives in delft, and she's cool. *Delft Outlook*, 2003.

[36] C.M. Chew H. Herr J. Hu, J. Pratt and G. Pratt. Virtual model based adaptive dynamic control of a biped walking robot. *International Journal on Artificial Intelligence Tools*, 8(3):337–348, 1998.

[37] I.W. Park J.W. Lee, K.h Lee and J.H Oh. Walking and control of autonomous biped robot. *IFAC Workshop on Mobile Robot Technology*, pages 179–183, 2001.

[38] Tamara Knutsen. *designing an underwater eel-like robot and developing anguiliform locomotion control.* University of Pennsylvania.

[39] Doru Mihalachi. *Système de commande temps réel distribué pour un mini robot autonome de type serpent*. PhD thesis, Université de Metz, 2000.

[40] Open Dynamics Engine ODE. http://q12.org/ode/. Library for simulating articulated rigid body dynamics.

[41] Julien Perrot. *La Salamandre.* (158), 2003.

[42] J. Pratt, C.M. Chew, A. Torres, P. Dilworth, and G. Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001.

[43] Jerry E. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Massachusetts Institute of Technology, 2000.

[44] C.L. Shih, Y.Z. Li, S. Churng, T.T. Lee, and W.A. Gruver. Trajectory synthesis and physical admissibility for a biped robot during the single-support phase. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1646–1652. IEEE Publications, Inc., 1990.

[45] Anurag Tripathi. *Analysis of hydrodynamic forces on the model of a swimming salamander.* Internship, EPFL, 2003.

[46] Webots. http://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.

[47] Frank M. White. *Fluids Mechanics*. WCB/McGraw-Hill, fourth edition. Pages 458-460.