UNIVERSIDAD CARLOS III DE MADRID

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

TESIS DE MÁSTER

MORPHOLOGY BASED LOCOMOTION CONTROLLER FOR MODULAR ROBOTS

Autor: Avinash Ranganath
Director: Luis Moreno Lorente

MÁSTER OFICIAL EN
ROBÓTICA Y AUTOMATIZACIÓN

LEGANÉS, MADRID
JUNIO 2013

UNIVERSIDAD CARLOS III DE MADRID
MASTER OFICIAL EN ROBÓTICA Y AUTOMATIZACIÓN

El tribunal aprueba la tesis de Master titulada "**Morphology Based Locomotion Controller for Modular Robots** " realizado por **Avinash Ranganath**.

Fecha: Junio 2013

Tribunal:

Maria Dolores Blanco Rojas

Alberto Brunete González

Cristina Castejón Sisamón

Contents

List of Tables

List of Figures

Acknowledgements

The author wants to thank all the staff of the Robotics Lab of the University Carlos III, for their warm welcome and their precious help.

He also wants to thank Dr. Juan González Gómez for his invaluable help and supervision provided during the project.

Abstract

Stigmergy is defined as a mechanism of coordination through indirect communication among agents, which can be commonly observed in social insects such as ants. In this work, emergence of coordination for locomotion in modular robots, through indirect communication among modules is being investigated. Due to the embodiment of the robot in the physical world, forces between physically connected modules in a modular robotic configuration exist as a result of interaction between modules, as well as between modules and their environment. These forces can be seen as analog communication between modules, and used as information for self-organization in a modular robotic organism. Also, experiments suggesting a strong interdependency between the morphology of the robotic organism and the emerged global behavior, have been conducted in this work.

Resumen

Estigmergia se define como un mecanismo de coordinación a través de la comunicación indirecta entre los agentes, que puede ser comúnmente observada en insectos sociales tales como las hormigas. En este trabajo se investiga la aparición de coordinación para la locomoción de robots modulares, a través de la comunicación indirecta entre los módulos. Debido a la forma de materialización del robot en el mundo físico, existen fuerzas, entre módulos conectados físicamente en una configuración robótica modular, como resultado de la interacción entre los módulos, así como entre los módulos y su entorno. Estas fuerzas pueden ser vistas como la comunicación analógica entre los módulos, y utilizadas como información para la auto-organización en un organismo robótico modular. Además, los experimentos que sugieren una fuerte interdependencia entre la morfología del organismo robótico y el comportamiento global surgido, se han llevado a cabo en este trabajo.

Introduction

Modular robots are systems composed of several unit modules which, provided self-reconfigurable capability, can autonomously change their morphology. Modular robots can be broadly classified into lattice-type and chain-type systems. Lattice-type systems achieve locomotion through continuous self-reconfiguration, where each module has the ability to move independently in the configuration. Locomotion in lattice-type systems gives the notion of modules flowing on the ground, which is visually similar to the locomotion of an amoeba, or to the flow of puddle of water on a flat surface. Locomotion in a chain-type system is achieved by controlling the actuator of individual modules in a fixed configuration.

Each module, in a modular robotic system, is a robot which has its own actuator, computation unit, sensors, power unit, and connectors to physically connect to two or more modules. A single module, due to its simplicity, is limited in its capability. But several modules connected together to form a bigger robotic organism, is capable of performing more complex tasks. An advantage of a modular robotic system is the ability to change the robot morphology to suit the task at hand. Also, if one or a few modules in a robotic configuration fail, they can be easily replaced with another module, and the cost of the entire system can be bought down by mass producing unit modules.

A modular robotic organism can either be composed of homogeneous modules, where all the unit modules are identical, or with heterogeneous modules. In an heterogeneous modular robotic system, a robotic organism could be composed of two or more kinds of modules, where each type of module is designed for a specific task.

In a modular robotic configuration, modules can either be controlled distributedly or in a centralized manner. Centralized control can be implemented by having one module in the configuration make the decisions and send the control signal to rest of the modules in the configuration, in a master-slave like method. But in this case, if the central controlling module fails, the entire configuration seize to operate. On the other hand, a robotic configuration can be controlled distributedly, where in each module would have its own controller, in which case, if one or some modules in a given configuration fail, the robotic organism could still function, although not at its full potential. Another advantage of distributed control is that it can scale well along with the size of the configuration, where as with a central controller, the load on the central controling module increases as the size of the robotic configuration increases, and so does not scale so well. Although, implementing a centralized controller is much simpler compared to a distributed controller.

Distributed controller in a homogeneous modular robotic system can either be homogeneous or heterogeneous. With an homogeneous controller, all the modules in a configuration would have clone controller with exactly the same parameters. Where as with a heterogeneous controller, modules could have specialized controllers for a specific task, based on their location in the configuration.

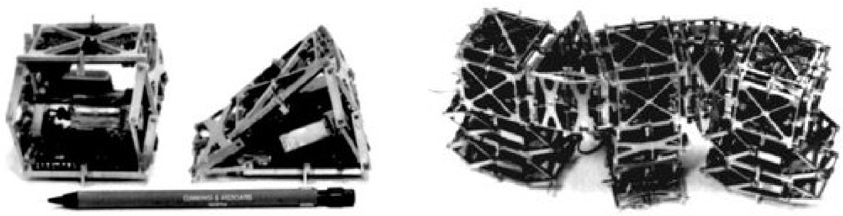
In this work, the motivation is to develop a locomotion controller for chain-type modular robots that is distributed, homogeneous and which does not rely on explicit communication between modules. Since modular robots do not have a fixed morphology, the focus is on developing a controller that is morphology independent, such that the controller can adaptive to change in morphology. Chapter 1 contains a review of the state of the art in modular robotic systems, distributed locomotion controllers for modular robots, and morphological computation, followed by an overview of the robotic platform and the modular robotic configurations that has been used in this work, in chapter 2. Chapter 3 explains the developed controller model, and chapter 4 contains the experimental results. Chapter 5 contains discussion, and the work is concluded in chapter 6.

Chapter 1

Related work

1.1 The origins

1.1.1 *Polypod-Bot*



(a) *Main modules* **(b)** *Configurations*

Figure 1.1: *PolyPod*

The earliest chain-type modular robot, in literature, appear in the year 1994 with *Polypod* Figure 1.1] [Yim, 1994] which is a bi-unit modular robot, developed by Mark Yim at PARC [Palo Alto Research Center]. *Polypod* was succeeded by *Polybot* Figure 1.2] [Yim et al., 2000] series of modular robots, with the first version created in 1998. Each module consists of a single unit with a single

DOF, which has its own computation, actuator and sensor, making each module a robot by itself. In some cases, power is supplied off board and passed from module to module. Each module can connect to other modules to form chain-type configurations, to form arms, legs, spine, etc. in a larger robotic configuration.

(a) *Configurations*

(b) *Module design evolution*

Figure 1.2: *PolyBot*

1.1.2 *CONRO*

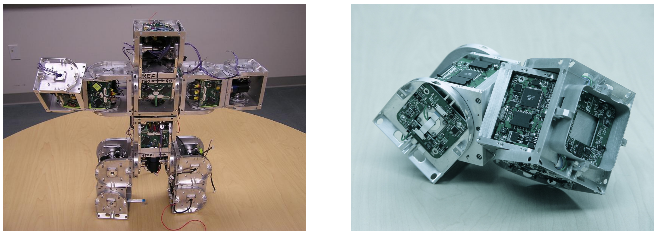
The *CONRO* self-reconfigurable robot[1] [Castano et al., 2002] was developed by P.Will at the Polymorphic Robotics Laboratory, University of Southern California, in 1998. Each module is an autonomous unit with one STAMP II micro-controller, two motors, four pairs of IR transmitters/receivers and four docking units for connecting with other modules. Each module has two DOF, one that rotates in pitch axis and the other in yaw axis. *CONRO* modules can be connected to each other to form into different types of configurations, some of which could be seen in Figure 1.3. Two connected *CONRO* modules can communicate with each

[1] http://www.isi.edu/robots/conro/

other using IR transmitter and received. To discover their position in a given configuration, modules communicate with each other, as well as for coordination during locomotion.

Figure 1.3: *Diffrent configurations in CONRO modular robot.*

1.1.3 *Superbot*



(a) *Humanoid configuration* **(b)** *A dual-unit Superbot module*

Figure 1.4: *Superbot modules*

The *SuperBot* [Salemi et al., 2006a] modular robots (Figure 1.4), developed at Polymorphic Robotics Laboratory, University of Southern California, is the descendant of the *CONRO* modular robot, and it also draws inspiration from *M-TRAN* modular robots. These robots were developed as part of a project funded by NASA, with space application as the final goal. One of the main design objectives of *SuperBot* was to make the modules sturdy by being able to deal with dust, moisture and strong light source, so that they can work in uncontrolled outdoor environments.

A *SuperBot* module is formed by linking two cube shaped bodies, which rotates in different axis, and a rotating central part connecting the two cubes. Each module has 3 DOF, which according to the authors, is the most for a single module ever developed. Each module has six docking elements, through which each module can be connected to at most six other modules at the same time.

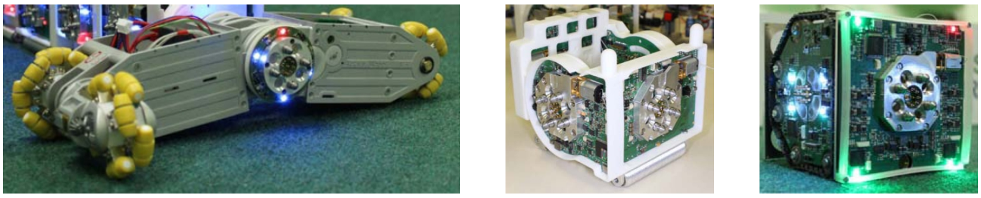
Unlike most other designs, a single *SuperBot* module can move on a 2D plain by itself, without being connected to any other modules, because of its 3 DOF. When connected to other modules, it can produce a range of locomotion gaits from simple crawling gait in a linear configuration, to more complex quadruped walking gait in a 3D configuration.

Along with the docking unit, each of the six sides on a single module has four infra-red receiver LEDs and a transmitter LED, used for communication between distant modules for docking to each other in 3D space. Modules unto 1 meter apart from each other can communicate and autonomously dock to each other.

Also, through a docking unit, connected modules can share power between each other. Current can always flow in from one connected module to another at any time, but the out flow of current is controlled by each module as an high-level behavior. In this way, when the power is critical in a modular robotic organism, on finding a power source, one of the modules in the configuration can connect to the power source, and eventually all the modules in the configuration can be charged by activating the current out flow from module to module as their batteries get charged.

For controlling high level behaviors of locomotion, reconfiguration and manipulation, a distributed control method inspired by biological hormones called Digital Hormone Method [Salemi et al., 2006b] has been utilized. Using Digital Hormones, which are messages generated by one module in a configuration, and diffused to other modules via message passing, a module can know its location in the topology, which the authors coin as topology mapping. Once all the modules know their topological position in a given configuration, they choose their optimal local action which would produce the desired global behavior of the organism. Through continuous topology mapping, if the morphology of the robot were to change, either by self-reconfiguration or due to failure of module(s), then modules will be able to adapt their behavior to the change in the configuration/morphology of the robotic organism.

1.1.4 ***Symbricator***



(a) *Active Wheel* **(b)** *Backbone* **(c)** *Scout*

Figure 1.5: *Symbricator modules*

Symbricator[2] [Ker, 2011] [Matthias et al., 2012] [Russo et al., 2012] is a heterogeneous, self-reconfigurable modular robotics platform developed mainly at Institute of Parallel and Distributed Systems, University of Stuttgart and Institute for Process Control and Robotics, Karlsruhe Institute of Technology, as part of the European funded projects *Symbrion* and *Replicator*. As part of this 5 year project, three distinct modular robot platforms, *Backbone*, *Active wheel* and *Scout*, were developed, with each platform designed for a special purpose in the end mission.

The research objectives of the *Symbricator* (*Symbrion* + *Replicator*) project is to investigate reconfigurability, adaptability and learn-ability in modular robotic

[2] `http://www.symbrion.eu/`

organisms on one hand, and to investigate evolve-ability of symbiotic [3] systems, and analogies to biological systems with respect to long-and short-term evolution, on the other hand.

The *Backbone* platform has an open ended cubical shape, and has 1 DOF for 3D locomotion. It has a very strong brush-less drive capable of lifting several connected modules, and is specialized for 3D locomotion. Each individual module can also perform locomotion in 2D, in all four directions, using a pair of specialized screw-driver wheels. Connected modules can also move in 2D using these wheels. This platform has four symmetric, genderless docking elements for connecting to other modules, and some basic sensors for perceiving its environment.

The *Active Wheel* platform is designed with the main functionality of transportation task in mind. It is able to carry and transport a modular robotic organism made up of several *Backbone* and *Scout* modules, in an energy efficient way. This platform has two symmetric arms connected at the hinge, which can rotate between $0°$ and $180°$. Two omni-directional wheels are attached at the end of each arm, which provides the 2D locomotion capabilities for this platform. There are two docking elements on the hinge, which can be used to connect other modules. Each docking element can rotate independently, and has two separate motors for this.

The *Scout* platform is similar in shape to the *Backbone* platform, but it is designed with fast locomotion for exploration in mind. This platform has tracks for fast 2D locomotion on rough terrain. It has extra sensors compared to other two platforms, for sensing and exploration task. It has two laser-camera systems, on its front and rear, for far and short range obstacle detection. It also has a motor for 3D locomotion, but the motor is not as powerful as the one present in the *Backbone* platform, as the main purpose of this platform is not macro-locomotion, but exploration. Four docking units are available to connect to other

[3] Symbiosis is close and often long-term interaction between two or more different biological species

modules, and the docking unit on the left side has a separate motor for rotating between +/- $180°$. These modules are best suited to be placed on the outer edge of a modular robotic organism as it would then not have too many modules to lift for 3D locomotion, and can use the extra sensors for sensing the environment.

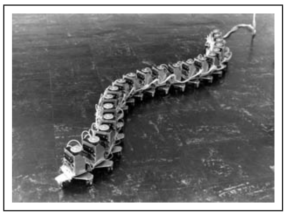
All three heterogeneous modular robotic platforms have some homogeneous elements, one of then being the common docking mechanism, required for connecting any combination of different types of modules to form a larger multi-robotic organism. A docking design called CoBoLD (Cone Bolt Locking Device) is employed for the docking units, which is genderless, $90°$ symmetric and can handle misalignment. The docking unit also provides wired communication and energy sharing between connected modules. As one of the main research goals of this project is energy sharing mechanism between connected modules, a unified energy sharing system is commonly implemented in all the three platforms.

1.1.5 Active Cord Mechanism

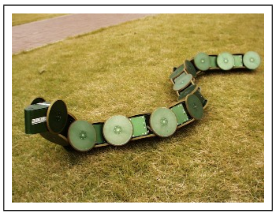
Although not completely classifiable as a modular robot, the original version of the AMC[4] (Active Cord Mechanism) was a milestone during its development, which was the first of its kind to implement a serpentine like locomotion gait, mimicking that of a snake, and an articulated mobile robotic design. It was created in 1972 by S.Hirose at the Tokyo Institute of Technology. The first version of the AMC had a fixed configuration, with each joint consisting of servo-mechanism that could bend left and right. Caster wheels were installed along the length of its body, which made contact with the ground. Locomotion in this robot was achieved by controlling individual joint with simple phase differed sinusoidal oscillators, which produced a propagating sine wave moving from the head to the tail of the linear robot, which propelled the robot forward. The emerged locomotion resembled that of an eel. In its latest version (ACM-R5), the design is modular, where each module has its own cpu, battery and motors, and can

[4] http://www-robot.mes.titech.ac.jp/robot/snake_e.html

operate independently. Although it can work only in a linear configuration, the length of the configuration can vary in the latest version, as individual modules can be added or removed from the configuration by an human operator. Connected modules can communicate with each other to determine the length of the configuration. AMC-R5 can operate both on land and under water. Evolution of the AMC design is as seen in Figure 1.6.

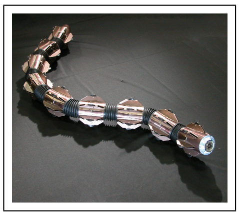


(a) *Active Code Mechanism No.3 ACM III.*



(b) *Self-contained Active Cord Mechanism-Revision3 ACM-R3.*

(c) *Practical 3-dimensional Active Cord Mechanism ACM-R4.*



(d) *Amphibious snake-like robot ACM-R5.*

Figure 1.6: *Different versions of the AMC robot.*

1.2 Locomotion controller classification

Locomotion controllers for modular robots can be broadly classified into two groups [Figure 1.7]; classical controllers and bio-inspired controllers. The former come from the industrial robotics domain and it is based on inverse kinematic and trajectory generation. These kinds of controllers are hard to scale with the increase in the degrees of freedom, and they require high computation

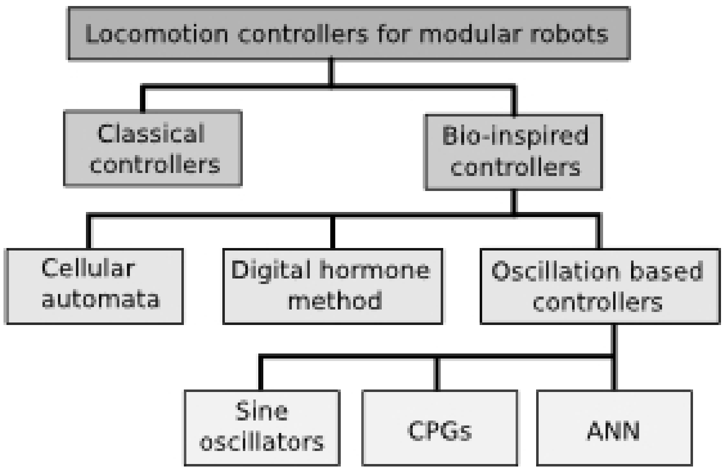


Figure 1.7: *A classification of locomotion controllers for modular robots.*

power. On the contrary the later class of controllers are inspired by biological processes. These controllers have been successfully implemented on different modular robotic platforms. Based on the method used, these controllers can be further sub-classified into Cellular Automata, Digital Hormone Method, and Oscillation based methods.

Lal et al. in [Lal et al., 2006] have implemented a Cellular Automata model for controlling locomotion of a five legged star shaped modular robot, where rules are evolved for controlling the actuator of each module, distributedly, based on the state of the module's actuator and that of its immediate neighbouring module's actuators, in the previous time step.

Shen et al. have used a biologically inspired method called Digital Hormone Method [Shen et al., 2000, Salemi et al., 2001, Hou and Shen, 2006] for adaptive communication of state information between modules, based on which a module can decide an action from a predefined gait table, which results in the emergence of locomotion. A particularly interesting aspect of this work is that if the configuration of the robotic organism changes, or if one or some modules fail, with adaptive communication, the locomotion gait is adapted to suit the change in configuration. Digital Hormones have been successfully implemented on two different modular robotic platforms called CONRO [Castano et al., 2002, Shen et al., 2002] and Superbot [Salemi et al., 2006a].

Gonzalez-Gomez et al. demonstrate in [Gonzalez-Gomez and Boemo, 2005] how simple sinusoidal oscillators can be used on minimal configuration modular robots with two and three modules for generating locomotion in once and two dimensions respectively, and in [Zhang et al., 2009] they study the locomotion of two different kinds of caterpillar gaits, from a kinematic perspective, and replicate the same on linear configuration modular robots, again using simple sinusoidal oscillators.

In [Spröwitz et al., 2008] Ijspreet et al. at the Biorobotics Laboratory, EPFL, have used Central Pattern Generators (CPG) [Ijspeert, 2008] for producing locomotory oscillations on their modular robotic platform called YaMoR [Moeckel et al., 2005], among other modular and non-modular robotic platforms. In [Pouya et al., 2010] they have tried similar CPGs for producing both oscillation and rotation in their second generation modular robotic platform called Roombots. CPGs are specialized neurons found in the spinal cord of vertebrate animals which have the capability of producing rhythmic output without rhythmic sensory or central input. The mathematical model of CPGs used for controlling locomotion in modular robots are usually one or two CPG neurons per module, which are coupled in different ways, based on the configuration, with similar neurons of other modules in the given configuration. CPGs were first successfully used on a modular robotic platform by Kamimura et al. in [Kamimura et al., 2003], where they use it for producing oscillations for adaptive locomotion on their M-TRAN modular robots.

Lal et al. in [Lal et al., 2008] have implemented an Artificial Neural Network [ANN] model as a locomotion controller for their brittle star modular robot. Here each module is modelled as a neuron in a fully connected neural network. Neurons sum their weighted input stimulus, which is the actuator phase angle that they share locally or globally based on their location in the configuration, and use a sinusoidal activation function to determine the next step. The authors have used Genetic Algorithm [GA] for evolving optimal synaptic weight vector of the ANN.

Though DHM, CPGs, and the Cellular Automata and ANN models used for locomotion controller by Lal et al., are all distributed control methodologies, they all rely on explicit inter-modular communication to adapt local behavior of individual modules to converge to optimal global behavior (Locomotion gait) of the robotic organism. The sinusoidal oscillators as locomotion controllers for modular robots, demonstrated by Gonzalez-Gomez et al. In [Gonzalez-Gomez and Boemo, 2005] and [Zhang et al., 2009], are distributed controllers as well, but the phase relation between modules are predetermined, making the controller heterogeneous. So in this work, the focus has been on developing a locomotion controller for modular robots that is distributed, homogeneous and uncoupled (which does not rely on explicit communication between modules).

1.3 Morphological computation

The term Morphological computation was coined by Chandana Paul [Paul, 2006], and it refers to outsourcing of computation to morphology and material property of an agent. Traditionally, intelligence and cognition have been associated solely with the brain of an agent, without taking into consideration its body and morphology. More recent works [McGeer, 1990], [Iida and Pfeifer, 2004] have shown that intelligence can arise as a result of interaction between the brain, body and the environment the agent is embedded in.

1.3.1 Passive dynamic walking

One on the earliest and a classical example of Morphological computation was demonstrated by Tad McGeer [McGeer, 1990] with the passive dynamic walker, which is a mechanical system that can walk down an inclined ramp with a slope of a certain degree, without any actuator, power supply, sensing or computation. In a sense, this agent is completely brain less, and the behavior of walking is produces solely based on its morphological properties, which is specifically tailored to produce the walking behavior. The energy requirement for walking in this design is minimal, as the walking is produced only by gravitation, and the walking gait produced seems very human like.

The drawback of the passive dynamic walker is that the conditions in which it operates, also called as its ecological niche, is very narrow. It means that the passive dynamic walker would cease to operate if anything in its environment or its morphology, like the slope on the ramp or the material property of its feet, is changed. *Denise*, an augmented passive dynamic walker with actuators, power supply and controller, was created by Martijn Wisse [Wisse, 2004] at Techinical University of Delft. The morphology of *Denise* was modified to walk on level surface, and the emerged walking gait was very human like and in-turn very energy efficient.

(a) *The classic passive dynamic walker* **(b)** *Denise*

Figure 1.8: *Passive dynamic walkers*

1.3.2 Puppy

Puppy [Iida and Pfeifer, 2004], a quadruped robot, built at the Artificial Intelligence lab, University of Zurich, mimics the morphology of a canine. Puppy has four limbs, and twelve joint (four each at the shoulder/hip, elbow/knees and ankle) in total. There are eight standard digital servomotors at the shoulder/hip and elbow/knee joints, and the ankle joints are connected via passive springs. A simple sinusoidal position controller was applied to each of the four shoulder/hip joints, where in the motor commands for the two shoulder joints were symmetrical, and the motor commands for the two hip joint were symmetrical as well. The elbow/knee joints were fixed to a constant position, and the ankle joints were passive. When evaluated by placing the robot on the ground, the robot displays a running gait, which is a result of the morphology of the robot (its shape and the passive spring joints), controller (parameters of the sinusoidal controller) and the environment (friction of the ground surface, gravity) the robot interacts with.

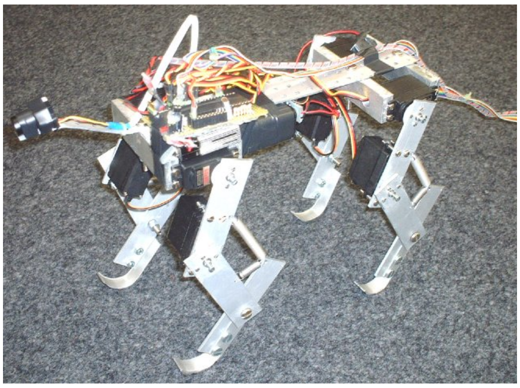


Figure 1.9: *The quadruped robot Puppy.*

1.3.3 WalkNet

Studies of insect locomotion [Cruse, 1990] [Cruse et al., 2002] has shown that the coordination between legs of an insect during walking, comes about as a result of coupled local neural circuits, and that there is no central controller that coordinates the legs during walking. When an insect, standing on the ground tries to move forward by pushing one of its leg backward, as a consequence of its embodiment, there is force applied on the rest of the stationary legs, and this information, in the form of joint angle measurement, can be used by the insect as a form of global communication between legs for producing locomotion, even without there existing a central controller that coordinates leg movements. Inspired by this, a distributed neural network architecture for controlling a six legged robot was developed [Dürr et al., 2003]. This is a very strong example of morphological computation, as the communication, and in turn the needed coordination between the legs for locomotion, comes about as a result the interaction between the insect and the real world.

Chapter 2

Robot Configurations

Y1 modular robot is the creation of Juan Gonzalez-Gomez [Gonzalez-Gomez, 2008]. As could be seen in Figure 2.1, a *Y1* module has an open-ended cube shape, and it is made up of two 3D ’U’ shaped objects connected together to form an hinge. The module has a dimension of $52 \times 52 \times 52mm^3$, one Degree Of Freedom [DOF], and a rotational range of $180\,°$. A *Futaba 3003* servo motor is used as the module’s actuator. In this work, OpenRAVE [Diankov and Kuffner, 2008] an open source, Open Dynamics Engine based robotic simulator, is used for all the experiments.

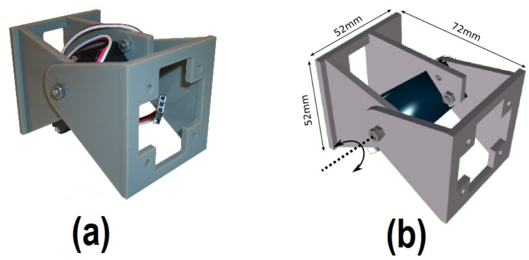


Figure 2.1: *Y1 module (a) Real and (b) Simulated versions.*

Two *Y1* modules can be connected to each other in several different ways, and each module can be connected with at most four other modules. Connection mechanism supported by *Y1* modules is very basic, and only static connections between modules are possible, making self-reconfiguration among modules not possible. Modules can be connected to each other either using nuts and bolts, or using zap-straps.

2.1 Modular robot configurations

In this work, five different modular robotic configurations have been experimented with. Dynamics of locomotion achievable by each robotic configuration, has been studied, by applying simple phase-differed sinusoidal oscillators to modules in a given configuration. In the following subsections, each configuration and their dynamics of locomotion has been described in detail.

2.1.1 *Minibot*

This is a two module one-dimension (1D) configuration [Figure 2.2], where in modules are connected to each other in series, and according to [Gonzalez-Gomez and Boemo, 2005] this is the smallest possible configuration that could produce a locomotion gait. Only locomotion in 1D can be achieved in this configuration, where the robot can either move forward or in reverse direction. Applying simple sinusoidal oscillators to modules, with predefined phase difference ($\triangle\Phi$), produces a caterpillar gait that resembles a traveling sine wave. The phase difference determines the direction of locomotion, with the robot moving in the direction of the module that has a negative phase difference with respect to the other module. Locomotion cannot be achieved in this configuration if the phase difference between modules is either around 0° or 180°.

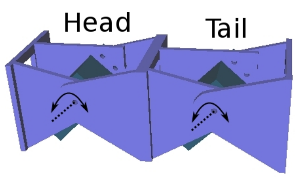


Figure 2.2: *Two module Minibot configuration.*

2.1.2 *Tripod*

This is a three module symmetric configuration, where in modules are connect to each other at an angle of $60\,°$, as shown in Figure 2.3. It is a 2D configuration and it can move on a 2D surface in three possible directions, as well as rotate on its own axis. When modules are applied with phase controlled sinusoidal oscillators, with two modules oscillating in phase and the third module oscillating with a $\triangle\Phi \in [100\,°, 150\,°]$, then the robot moves in the direction of the modules out of phase, and in the opposite direction if $\triangle\Phi \in [-100\,°, -150\,°]$. When no two modules oscillate in phase, and the phase difference between any two adjacent pair of modules is $120\,°$ [E.g. $\Phi_{M-1} = 0\,°$, $\Phi_{M-2} = 120\,°$, $\Phi_{M-3} = 240\,°$], then the robot rotates on its own axis in clockwise direction.

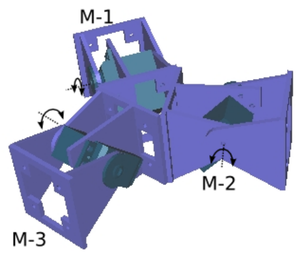


Figure 2.3: *Three module Tripod configuration.*

2.1.3 *Quadropod*

The *Quadropod* configuration is a extension of the *Tripod* configuration, which has an additional module, and the angle between modules is $90\,°$ [Figure 2.4]. It is a symmetric 2D configuration, which can move in eight possible directions on a 2D surface, depending on the phase difference between oscillating modules. If two opposite modules oscillate in phase, while the other two modules oscillate with a $\triangle\Phi \in [100\,°, 150\,°]$, then the robot moves in the direction perpendicular to the modules oscillating in phase. If two pairs of adjacent modules oscillate in phase, with a phase difference between these pairs [Eg: $\Phi_{M-1} = \Phi_{M-2} = 0\,°$

and $\Phi_{M-3} = \Phi_{M-4} = 120\,^{\circ}$], then the robot moves in the direction diagonal to itself. When no two modules are oscillating in phase, and the phase difference between any two pairs of adjacent modules is $90\,^{\circ}$ [Eg: $\Phi_{M-1} = 0\,^{\circ}$, $\Phi_{M-2} = 90\,^{\circ}$, $\Phi_{M-3} = 180\,^{\circ}$, $\Phi_{M-4} = 270\,^{\circ}$], then the robot rotates on its own axis in clockwise direction.

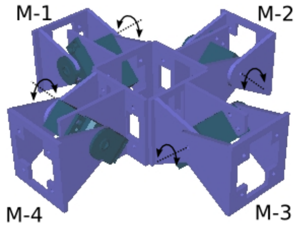


Figure 2.4: *Four module Quadropod configuration.*

2.1.4 *Y-bot*

The *Y-bot* configuration, as shown in Figure 2.5, is an extension of the *Tripod* configuration, which is conceived by adding an additional *Y1* module (*Tail*) to one of the three modules of the *Tripod* configuration, which becomes the *Spine* module. Locomotion in 2D is possible with this configuration, although only 1D locomotion gait is being focused upon in this work. When modules are applied with phase-differed sinusoidal oscillators such that there is an increasing phase difference between modules, starting from *Head* module to *Tail* module, while the two *Head* modules oscillate in phase [E.g. $\Phi_{Head-left} = \Phi_{Head-right} = 0\,^{\circ}$, $\Phi_{Spine} = 100\,^{\circ}$, $\Phi_{Tail} = 200\,^{\circ}$], the robot moves in the direction of the *Tail* module. The robot moves in the opposite direction, if this case is reversed [i.e. $\Phi_{Head-left} = \Phi_{Head-right} = 200\,^{\circ}$, $\Phi_{Spine} = 100\,^{\circ}$, $\Phi_{Tail} = 0\,^{\circ}$].

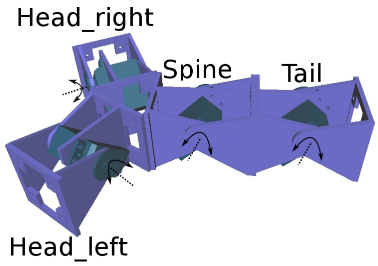


Figure 2.5: *Y-bot configuration.*

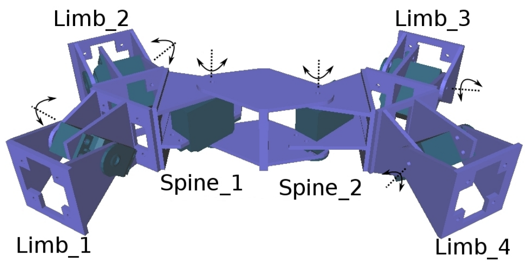


Figure 2.6: *Lizard configuration with four Limb modules and two Spine modules.*

2.1.5 *Lizard*

Lizard, as shown in Figure 2.6, is a six module configuration that has four *Limb* modules, and two *Spine* modules. This configuration is formed by connecting two *Tripod* configurations together, and then rotating the *Spine* modules by $+/-90^\circ$ respectively along the pitch axis, in relation with the rest of the configuration. This makes the two halves of the robot (Considering modules *Limb-1*, *Limb-2*, and *Spine-1*, as one half, and modules *Spine-2*, *Limb-3* and *Limb-4* as the other half) mirror images of each other. When modules in this configuration were actuated with phase-controlled sinusoidal oscillators, as shown in Table 2.1 (which is derived empirically), it resulted in a quadruped walking gait, resembling that of a reptile. The two spine modules wiggle side to side with a phase difference between them [1], while the limb modules move up and down, making the

[1]Although the control signal to the two spine modules are symmetric, they oscillate with a phase difference of 180° between them, since the two modules are connected to each other as a mirror image of each other

Module	Phase Angle
Limb-1	0°
Limb-2	160°
Spine-1	-80°
Spine-2	-80°
Limb-3	160°
Limb-4	0°

Table 2.1: *Phase relation between modules in a Lizard configuration with respect to the module Limb-1.*

robot produce a walking gait.

Chapter 3

Controller

Locomotion in general, whether a gallop of a horse, flapping of a bird, or bipedal walking of a human, can be seen as repetitive and coordinated movement of limbs, through which the locomotion gait emerges. Looking at locomotion as a collection of oscillators, the phase relation between these oscillators determines the generated gait. This phase relation can be brought about by sharing actuation information among modules through explicit inter-module communication in a modular robotic system. But since a modular robot is an embodied system comprising of physically connected robot modules, the proposed controller relies on the intra-configuration forces that exist among modules, for coordination.

3.1 Intra-configuration forces

Since modular robots are physically connected multi-robot systems, modules exert forces on one another when actuated. In a simulated *Minibot* configuration, when one module (*Head*) is actuated with a sinusoidal oscillator, with an amplitude of $60\,^{\circ}$, and the other module (*Tail*) is actuated as well, but made to remain at a constant reference position of $0\,^{\circ}$, the effects of the oscillating *Head* module is observable on the fixed position *Tail* module. As could be seen in Figure 3.1,

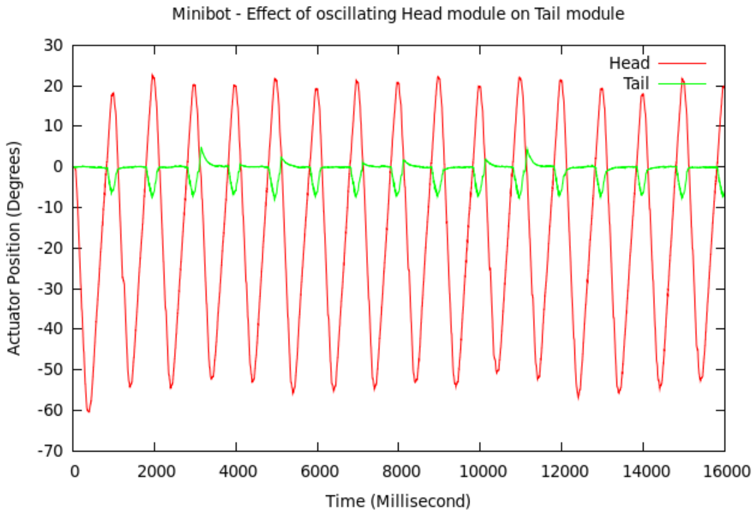


Figure 3.1: *Plot of actuator values in the Minibot configuration, demonstrating the effects of the oscillating Head module over the fixed position Tail module.*

the *Tail* module oscillates as well, with a low amplitude and an offset, due to the force exerted on it by the oscillating *Head* module. This phenomenon can be quantified by measuring the mean and standard deviation (SD) of the actuator value of the affected (*Tail*) module, which is as shown in Table 3.1. Similarly, when the roles of the *Head* and *Tail* modules are interchanged, the effects of the oscillating module is observed on the fixed position module, but with almost twice the mean and SD, which suggest that the effect of the oscillating *Tail* module on the *Head* module is twice compared to that of the oscillating *Head* module over the *Tail* module. This is because of the asymmetric mass distribution of the configuration, which is based on the way the two modules are connected to each other [1].

Similar experiments were conducted on a simulated *Tripod* configuration, by actuating two modules with sinusoidal oscillators, which are in phase in the first experiment, and with a phase difference of 120° in the second experiment,

[1]The side of the *Tail* module holding the servo motor is connected to the side of the *Head* module that is free

		Fixed			
		Head		Tail	
		Mean	SD	Mean	SD
Oscillating	Head	-	-	-0.92°	2.16°
	Tail	-1.68°	4.39°	-	-

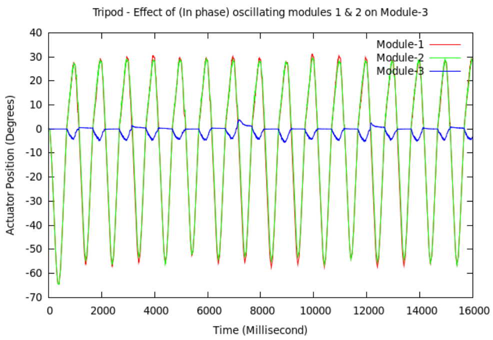
Table 3.1: *Quantifying intra-configuration force by calculating mean and SD of actuator position of the fixed position module, connected to an oscillating module in a Minibot configuration.*

while the third module remained at a fixed reference position of 0° in both the cases. In the third experiment a single modules was oscillated, while the other two modules remained at a fixed reference position of 0°. In all three cases, although the oscillating module(s) has an effect on the fixed position module(s), the effects were different on the fixed position modules(s) as could be seen in Figure 3.2. The mean and SD of actuator position of the respective fixed position modules(s) are as provided in Table 3.2.

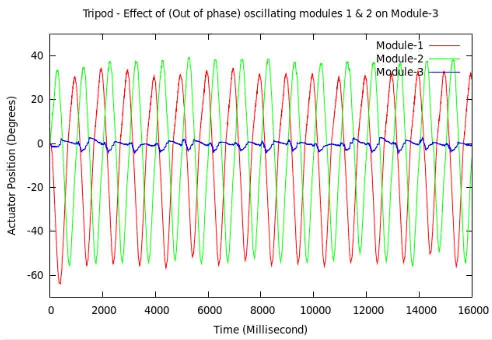
		Fixed			
		M-2		M-3	
		Mean	SD	Mean	SD
Oscillating	M-1 and M-2 $[\triangle\Phi = 0^\circ]$	-	-	-1.01°	1.74°
	M-1 and M-2 $[\triangle\Phi = 120^\circ]$	-	-	-0.53°	1.32°
	M-1	-0.47°	1.44°	-0.54°	1.71°

Table 3.2: *Mean and SD of actuator position of fixed position modules in a Tripod configuration.*

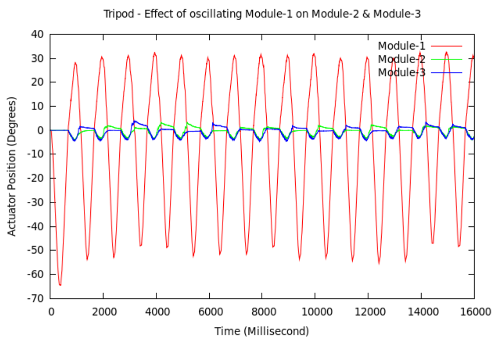
As could be observed in Table 3.2, there is a noticeable difference in the force exerted on the fixed position module, based on the phase difference that exist between the other two oscillating modules in a *Tripod* configuration. To further examine this relationship, in a *Tripod* configuration, between varying phase difference among oscillating modules, and the force exerted on the fixed position module, modules *M-1* and *M-2* were actuated with sinusoidal oscillators and phase value ranging between 0° and 330°, at an interval of 30°. The result of



(a) *Two modules oscillating in phase*



(b) *Two modules oscillating out of phase*



(c) *One module oscillating*

Figure 3.2: *Plot of actuator values in the Tripod configuration, demonstrating the effects of oscillating module(s) over fixed position module(s).*

this experiment is as shown in Table 3.3, which is plotted as a bar-graph in Figure 3.3. The force exerted on the fixed position module is at the highest when the two oscillating modules oscillate in phase, and at the lowest when the modules oscillate out of phase. The SD at a phase difference of $0\,^{\circ}$ is twice compared to SD at a phase difference of $180\,^{\circ}$, this is because both the modules oscillating in phase, exert force on the fixed position module at the same time, while when oscillating with a phase difference of $180\,^{\circ}$, each module exert force on the fixed position module at slightly different time.

The force on the fixed position module is exerted when an oscillating module pushes down on the ground surface. So, to study how ground surface friction

		Fixed position module M-3	
		Mean	SD
Oscillating modules M-1 and M-2	$\triangle\Phi = 0\,^\circ$	$-1.01\,^\circ$	$1.74\,^\circ$
	$\triangle\Phi = 30\,^\circ$	$-0.72\,^\circ$	$1.71\,^\circ$
	$\triangle\Phi = 60\,^\circ$	$-0.63\,^\circ$	$1.59\,^\circ$
	$\triangle\Phi = 90\,^\circ$	$-0.65\,^\circ$	$1.32\,^\circ$
	$\triangle\Phi = 120\,^\circ$	$-0.53\,^\circ$	$1.32\,^\circ$
	$\triangle\Phi = 150\,^\circ$	$-0.34\,^\circ$	$1.05\,^\circ$
	$\triangle\Phi = 180\,^\circ$	$0.31\,^\circ$	$0.84\,^\circ$
	$\triangle\Phi = 210\,^\circ$	$-0.50\,^\circ$	$1.01\,^\circ$
	$\triangle\Phi = 240\,^\circ$	$-0.84\,^\circ$	$1.27\,^\circ$
	$\triangle\Phi = 270\,^\circ$	$-0.71\,^\circ$	$1.32\,^\circ$
	$\triangle\Phi = 300\,^\circ$	$-0.72\,^\circ$	$1.39\,^\circ$
	$\triangle\Phi = 330\,^\circ$	$-0.64\,^\circ$	$1.69\,^\circ$

Table 3.3: *Mean and SD of actuator position of the fixed position module in a Tripod configuration, sampled over different phase value of oscillating modules.*

determines the force exerted, an experiment, similar to the one explained in the previous paragraph with the *Tripod* configuration, was conducted. In this experiment, module *M-1* and *M-2* oscillate in phase, and the mean and SD of fixed position module *M-3* was sampled over varying coefficient of friction of the ground surface. The results of this experiment is as shown in Table 3.4 and the same is plotted as a bar-graph in Figure 3.4. The results indicate that there exists a positive correlation between coefficient of friction of the ground surface and the force exerted on a module.

Based on several factors such as the morphology and mass distribution of a configuration, phase relation between oscillating modules and coefficient of friction of the ground surface, connected modules in a given configuration, exert force of each other. This could be seen as implicit, analog communication between modules, as a result of its embodiment, and could be used for controlling modules distributively such that difference in local behavior of individual modules would result in the emergence of global behavior of the robotic organism.

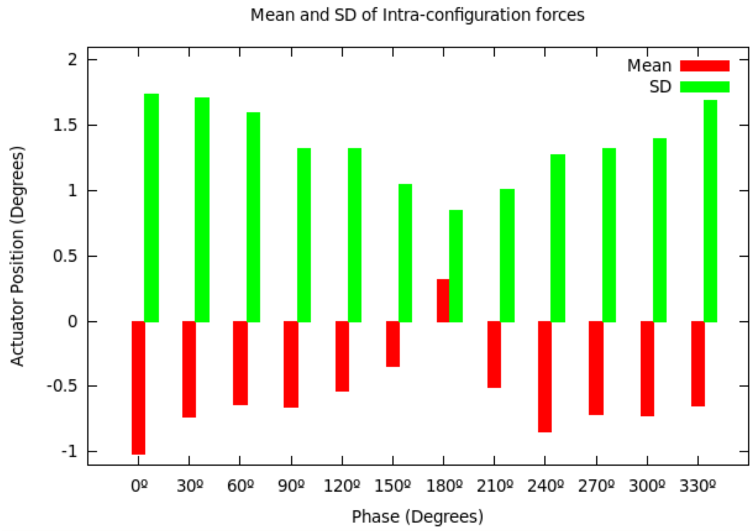


Figure 3.3: *Plot of mean and SD of actuator position of the fixed position module in a Tripod configuration, sampled over different phase value of oscillating modules.*

3.2 Simple controller

Based on the experimental results from section 3.1, a very simple distributed controller was developed that utilize forces existing among physically connected modules, as a way of implicit communication among modules in a configuration, for converging to and maintaining a steady phase relation among modules, which in turn result in stable locomotion gait. The controller employs a very simple oscillating mechanism, with fixed amplitude and offset as defined in equation (Equation 3.1). Each module is controlled independent by its own controller, although all the modules in a given configuration would have the exact same parameter values, and the controllers are uncoupled. Conditions (Equation 3.2) and (Equation 3.3) are used to determine if the module's actuator has reached the desired position, determined by equation (Equation 3.1), and if either of the two conditions satisfied, then the direction of rotational of the module's actuator is switched by obtaining the next oscillation angle from equation (Equation 3.1). Figure 3.5(a) depicts the control strategy. Condition (Equation 3.2) checks, at

		Fixed position module M-3	
		Mean	SD
Oscillating modules M-1 and M-2	$CoF = 0$	$-0.15\,^\circ$	$0.14\,^\circ$
	$CoF = 0.001$	$-0.15\,^\circ$	$0.15\,^\circ$
	$CoF = 0.003$	$-0.16\,^\circ$	$0.16\,^\circ$
	$CoF = 0.006$	$-0.18\,^\circ$	$0.21\,^\circ$
	$CoF = 0.009$	$-0.20\,^\circ$	$0.23\,^\circ$
	$CoF = 0.01$	$-0.20\,^\circ$	$0.25\,^\circ$
	$CoF = 0.03$	$-0.10\,^\circ$	$0.34\,^\circ$
	$CoF = 0.06$	$-0.18\,^\circ$	$0.63\,^\circ$
	$CoF = 0.09$	$-0.35\,^\circ$	$1.13\,^\circ$
	$CoF = 0.1$	$-0.44\,^\circ$	$1.33\,^\circ$
	$CoF = 0.5$	$-1.04\,^\circ$	$1.71\,^\circ$
	$CoF = 1.0$	$-1.01\,^\circ$	$1.74\,^\circ$

Table 3.4: *Mean and SD of actuator position of the fixed position module in a Tripod configuration, sampled over varying coefficient of friction of the ground surface.*

every time step, if the actuator is within a range of α and $-\alpha$ of the desired position determined by equation (Equation 3.1). Condition (Equation 3.3) checks if the speed of rotation of the module's actuator is above a certain threshold specified by β. The rate of rotation of an oscillating module's actuator is dynamically influenced as a result of intra-configuration forces that exist among oscillating modules, and this phenomena is captured by (Equation 3.3).

$$Y_i := (-1)^i A + o, \forall i \in \mathbf{N} \tag{3.1}$$

$$\mid Y_i - \theta_t \mid \leq \alpha \tag{3.2}$$

$$\mid \theta_{t-h} - \theta_t \mid \leq \beta \tag{3.3}$$

Where Y_i is the i^{th} input to the module's actuator, A is the amplitude, o is the offset, θ_t is the actual position of the module's actuator at time instance t. Parameters h, α and β are constants, and parameters α and β correspond to the absolute difference threshold between the actuator's actual and desired position, and actuator's rotation speed threshold respectively. The units of parameters h, α and β are milliseconds, degrees and degrees per second respectively.

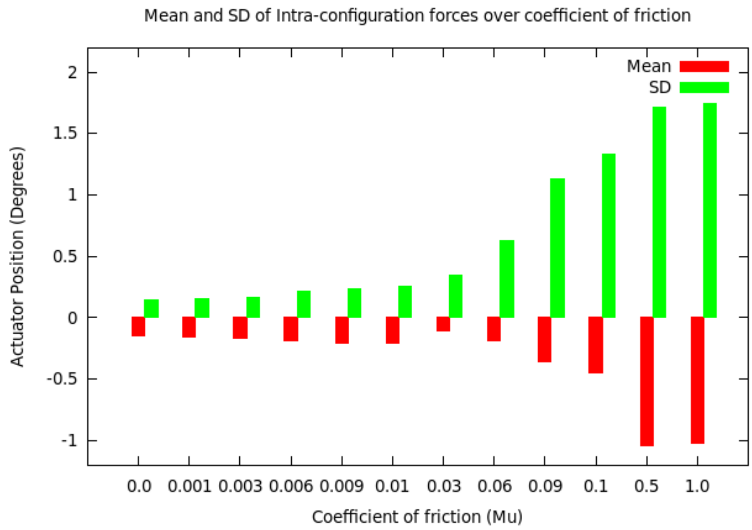
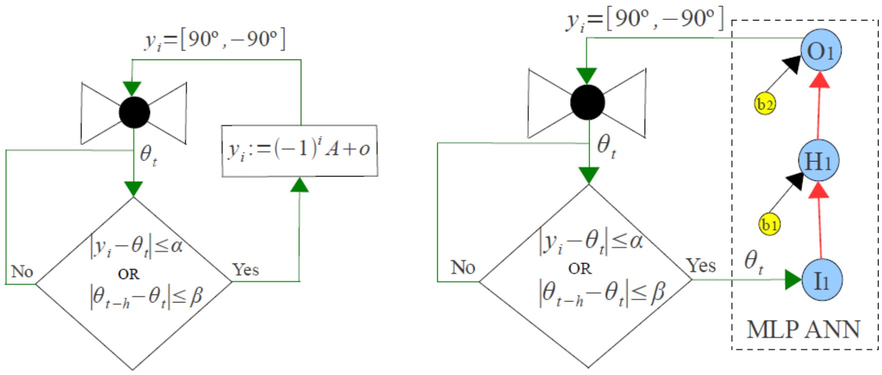


Figure 3.4: *Plot of mean and SD of actuator position of the fixed position module in a Tripod configuration, sampled over varying coefficient of friction of the ground surface.*

3.3 Neural controller

Extending the previous model to include adaptive oscillation rather than a fixed amplitude-offset oscillator, equation (Equation 3.1) is replaced with a fully connected feed-forward Multilayer Perceptron Artificial Neural Network [ANN], as shown in Figure 3.5(b). The ANN has one input neuron, one hidden layer with a single hidden neuron, and one output neuron. The input to the ANN is the actual position of the module's actuator, and the output of the ANN is the control signal to the module's actuator. The lone hidden neuron and the output neuron have one bias node each, and hyperbolic tangent activation function is used in all the layers. Flood [Flo], an open source ANN library, is used for implementing the ANN model. All the parameters of this controller are optimized using GA.



(a) *Simple controller* **(b)** *Neural controller*

Figure 3.5: *Control flow of the two controllers.*

Parameters	Value
Population Size	50
Evolution length	50 generations
Crossover percentage	50%
Elite population percentage	10%
Mutation rate	10%
Mutation rate for ES	25%

Table 3.5: *GA Parameter values used for evolution.*

3.4 Evolution

For a given configuration, parameters β and h, along with parameters A and o for the *Simple controller*, and synaptic weights of the ANN for the *Neural controller* are optimized using a method that combines GA and *Evolutionary strategy*. Parameter α was determined empirically, and fixed to a value of $3°$. Optimum parameters for the respective controllers were evolved offline for each robot configuration independently, by setting up the robotic configuration in the simulation environment, where in each module in a given configuration would have its own controller. Starting with random initial parameter values, each candidate solution was evaluated thrice, where in each evaluation lasted for a period of 50 seconds

[2]. The fitness function for selecting the best candidate solutions in each generation was the distance traveled by the robotic organism at the end of each evaluation cycle, measured as euclidean distance between the starting position and the final position, averaged over the number of evaluations [3]. A fairly standard GA/Evolutionary strategy approach was followed, with *Roulette Wheel* selection method and *Intermediate* recombination method, on real-valued genes, for reproducing new offspring. Table 3.5 contains the GA parameters employed, which were derived empirically.

Each new generation, after the first generation, was created by carrying forward the top 10% of the population from the previous generation. The next 10% of the new generation was populated by carrying forward mutated copies of the same top 10% of the population from the previous generation. The rate of mutation for this was set higher than regular, to 25%. The remaining 80% of the population was populated by producing new offspring through sexual reproduction between selected parents from the previous generation, and then mutating the offspring at a rate of 10%.

A random behavior is observed in the initial stage of the evolution process, where modules either oscillate erratically or in some case do not even oscillate. Figure 3.6 shows a graph plot of a *Minibot* configuration's actuator values, during the learning phase of it's *Neural controller*. Candidate solutions that result in some or all the modules oscillating, albeit suboptimally, which in turn result in displacement of the robotic organism, are carried forward, and over generations this behavior is fine tuned such that modules oscillate in a way that is optimal for the entire configuration, which in turn results in the emergence of a stable locomotion gait. Figure 3.7 shows the learning curve of *Neural controller* in a *Lizard* configuration.

[2]Each candidate solution was evaluated for a period of 50 seconds, but the evaluation was sped up in the simulation environment and took, on average, about 3 seconds in real time

[3]Each candidate solution was evaluated three times

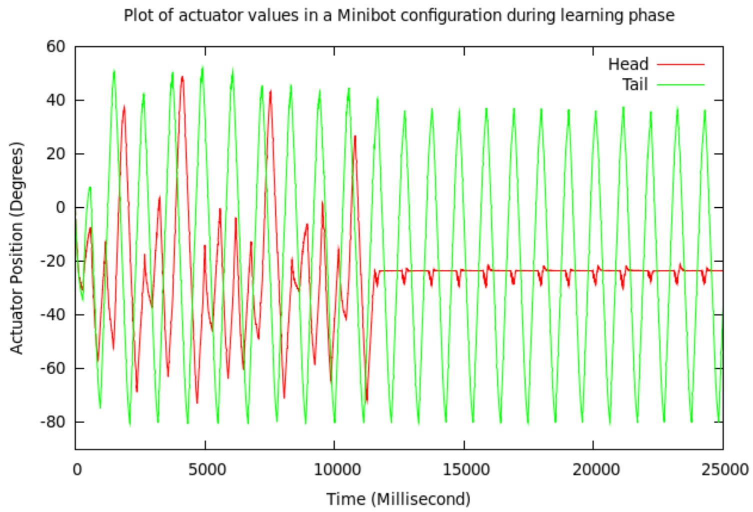


Figure 3.6: *Plot of actuator values of modules in a Minibot robot during the learning phase of it's Neural controller.*

Figure 3.7: *Plots of fitness value of best perfoming candidate solution and the mean fitness value of the population, over the course of evolution.*

Chapter 4

Experiments and Results

The best evolved *Simple* and *Neural controllers* were evaluated by applying the controllers independently on the respective configurations they were evolved for. The following sub-sections explain the observed behavior in each configuration. Also, results from cross-evaluating controllers, by applying best evolved controller for one configuration on a different configuration, is provided in the following.

4.1 Simple Controller

Applying the best evolved *Simple controller* on a *Minibot* configuration resulted in a caterpillar like locomotion gait. Since both the modules have exactly the same controller, and start at the same initial position, modules would start to oscillate in phase, but quickly develop and maintain a steady phase difference. The average phase difference between the two modules was $-139.68\,^{\circ}$, with a SD of $3.24\,^{\circ}$. The frequency of oscillation is not predefined in this control scheme, but is intrinsic to the system, and it is inversely-proportional to the amplitude. Similarly, the phase difference between modules is not predefined either, but is a result of the morphology of the robot, and the interaction of the robot with its environment.

A plot of the oscillation, frequency and phase values of the emerged locomotion gait in this configuration is as shown in Figure 4.1, Figure 4.2 and Figure 4.3 respectively. A sample of actuator plots in Figure 4.1 is magnified in Figure 4.4. It could be observed from the magnified image, that the control signal generated by the *Simple controller* is of the sinusoidal/triangle wave form.

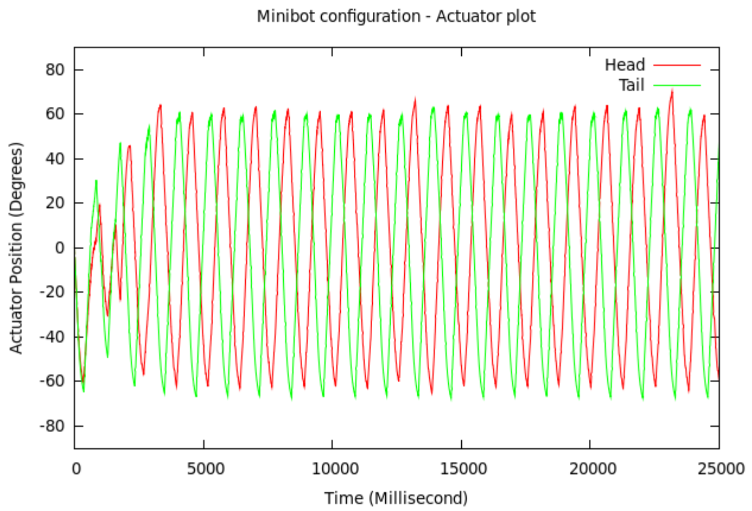


Figure 4.1: *Plot of actuator values of modules in a Minibot configuration when evaluated with the best evolved Simple controller.*

Similarly, evaluating the best evolved controller for *Tripod* and *Quadropod* configurations, respectively, resulted in a locomotion gait that made the robots move in circles, although the emerged gait was stable. Here, stability of an emerged gait is based on the consistency of the phase relation among modules. Looking back at the simple two module *Minibot* configuration, if the phase difference between the oscillations of the two modules varied largely over time, making the phase value either oscillate or erratic, then the emerged locomotion gait would be unstable making the 1D robot move back and forth on a straight line, rather than making the robot move consistently in one direction. But the SD of the phase difference between the *Head* and *Tail* modules was as low as $3.24\,^{\circ}$, resulting in a smooth locomotion gait.

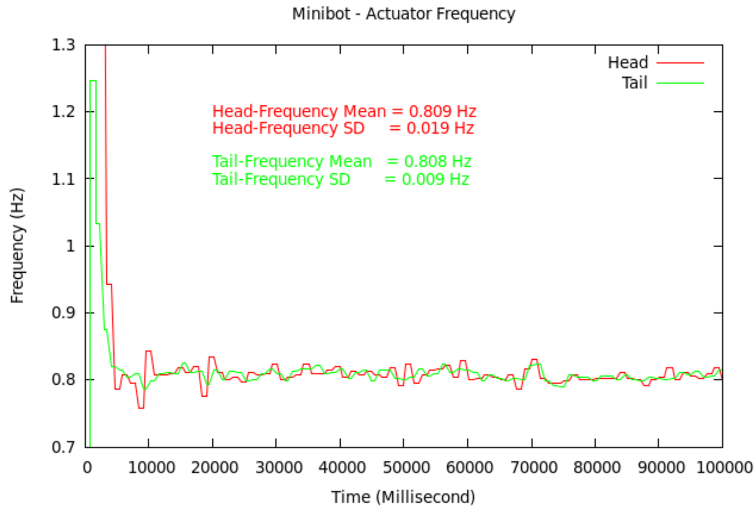


Figure 4.2: *Oscillation frequency graph of modules in a Minibot configuration when evaluated with the best evolved Simple controller.*

Looking at the phase graph of a *Quadropod* robot, as shown in Figure 4.5, it could be observed that the phase relation between modules, although vary slightly, are consistent enough to make the emerged locomotion gait stable. Table 4.1 [1] contains the mean and the SD of the phase difference between all the modules in this configuration, when evaluated with the best evolved *Simple controller* for a period of 200 seconds. The SD between any two modules in this configuration range between a maximum of $8.66\,°$ and a minimum of $4.89\,°$, resulting in a stable locomotion gait.As could be observed in Figure 4.5, modules *M-3* and *M-4* oscillate in phase, where as there is a phase difference between modules *M-1* and *M-2*, modules *M-2* and *M-3*, and modules *M-1* and *M-4*, resulting in the circular trajectory of the emerged locomotion gait. The trajectory of the robot recorded during this evaluation is plotted in Figure 4.6.

Evaluating the *Tripod* configuration with its best evolved *Simple controller*

[1] Phase values for the first 10 seconds of the evaluation were discarded while calculating the mean and SD, as the robot would take about 5 to 10 seconds to stabilize and settle into a smooth locomotion gait.

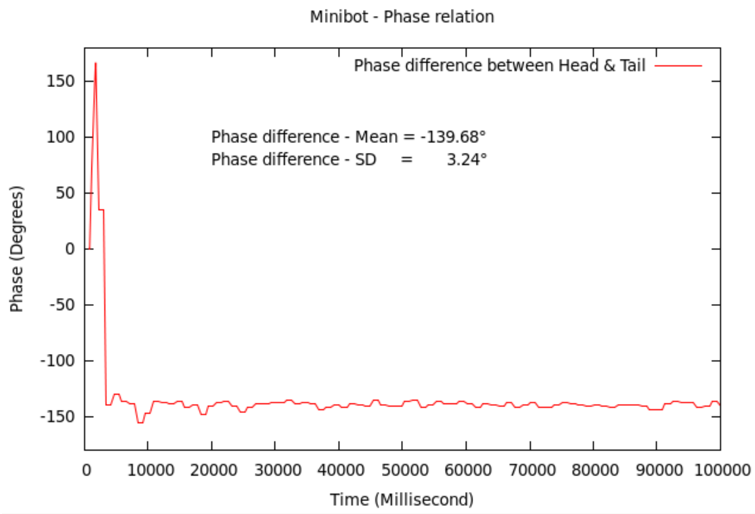


Figure 4.3: *Graph containing phase relation between modules in a Minbot configuration when evaluated with the best evolved Simple controller.*

produced a similar circular trajectory locomotion gait (Figure 4.8). The phase relation graph is shown in Figure 4.7 and the mean and SD of the phase difference between modules is as provided in Table 4.2, from which it could be observed that the phase relation between modules was consistency enough to produce a smooth locomotion gait.

The *Y-bot* configuration, when applied with the best evolved *Simple controller*, produces a crawling gait very similar to the caterpillar gait produced by the *Minibot* robot. The emerged gait is again very stable, based on the consistency of the phase relation among modules, as could be seen in Figure 4.9. The two *Head* modules oscillate in phase, and there is a steady phase difference between the rest of the modules in the configuration, which produces a propagating sine-wave starting from the *Head* modules and moving in the direction of the *Tail* module, making the robot propel forward in the direction of the *Tail* module.

The *Lizard* configuration, on the best evolved *Simple controller*, produces a reptilian like quadruped walking gait. Modules start to oscillate in phase, but

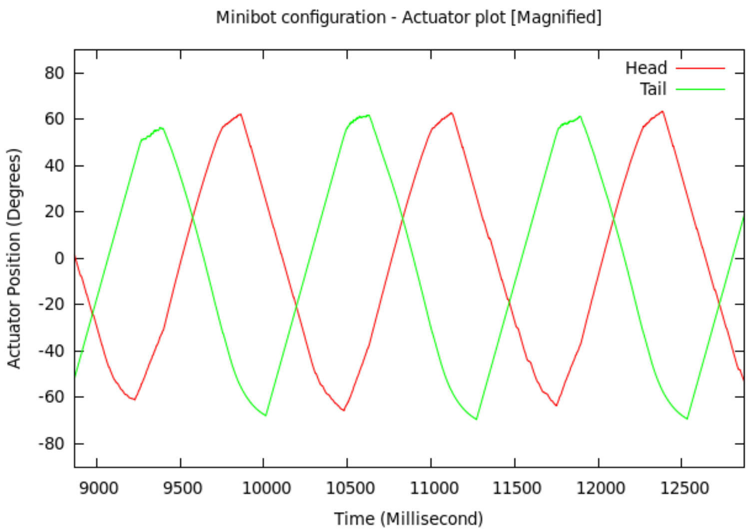


Figure 4.4: *Magnified sample of Figure 4.1.*

very quickly converge to and maintain a steady phase relation, as could be seen in Figure 4.10 and Figure 4.11. In the emerged gait, pairs of leg modules on the same side (*Limb-1* - *Limb-4* and *Limb-2* - *Limb-3*) oscillate in phase, while there exists a phase difference of about 180° between the two pairs. Although the two *Spine* modules seem to oscillate in phase, since they are connected to each other as a mirror image of each other, they oscillate side-to-side with a phase difference of around 180° between them. Considering module *Limb-1* as a reference module, a phase difference of about -100° exists between modules *Limb-1* and *Spine-1*, and about 100° between modules *Limb-1* and *Spine-2*. The robot would move in the opposite direction if these two phase values are interchanged.

4.2 Cross-evaluation

Given the self-reconfigurable capability to a modular robotic system, the controller must be able to adapt to changes in the configuration. To test how well

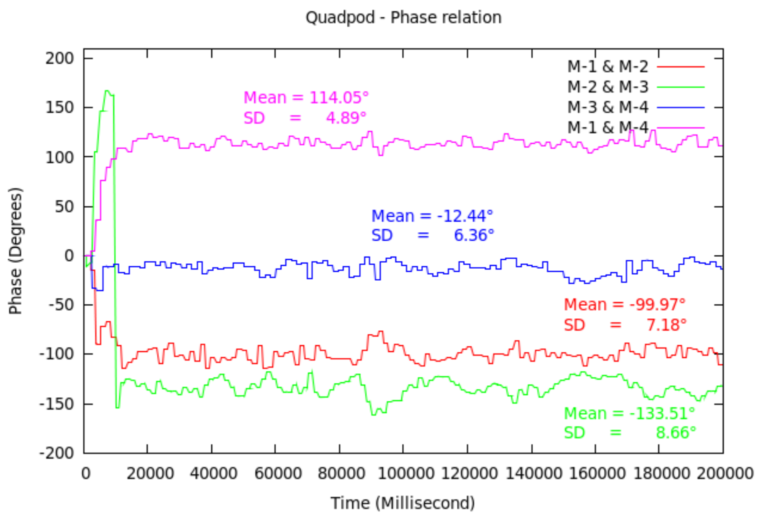


Figure 4.5: *Graph containing phase relation between modules in a Quadropod configuration when evaluated with the best evolved Simple controller.*

the proposed controller can adapt to a new configuration, the best evolved *Simple controller* for each of the five configurations was cross-evaluated on all five configurations. The result is as shown in Table 4.3.

Each controller was evaluated on each of the five configurations, and the values shown in Table 4.3 is the total distance travelled, in meters, after 100 seconds of evaluation. Each row consists results of one configuration evaluated with all five controllers. If the robot tipped over during an evaluation, then the result was not considered, and marked as N/A. Stability of a locomotion gait is based on the consistency of the phase relation among modules in a given configuration. So if the phase relation of an emerged gait during an evaluation was erratic, as shown in Figure 4.12, then that evaluation was not considered either, and marked as U/PR. Figure 4.13 is an example of stable phase relation among modules of a configuration when evaluated with a controller optimised for a different configuration.

Looking at Table 4.3, it could be determined that the most adaptive controller is the one evolved for the *Lizard* configuration, as the rest of the configurations

		Quadropod Modules			
		M-1	M-2	M-3	M-4
M-1	Mean	0°	-99.97°	126.49°	114.05°
	SD	0°	7.18°	5.31°	4.89°
M-2	Mean	-99.97°	0°	-133.51°	-145.98°
	SD	7.18°	0°	8.66°	6.23°
M-3	Mean	126.49°	-133.51°	0°	-12.44°
	SD	5.31°	8.66°	0°	6.36°
M-4	Mean	114.05°	-145.98°	-12.44°	0°
	SD	4.89°	6.23°	6.36°	0°

Table 4.1: *Mean and SD of phase difference between modules in a Quadropod configuration when evaluated with the best evolved Simple controller.*

		Tripod Modules		
		M-1	M-2	M-3
M-1	Mean	0°	47.86°	123.23°
	SD	0°	6.28°	8.43°
M-2	Mean	47.86°	0°	75.36°
	SD	6.28°	0°	9.0°
M-3	Mean	123.23°	75.36°	0°
	SD	8.43°	9.0°	0°

Table 4.2: *Mean and SD of phase difference between modules in a Tripod configuration when evaluated with the best evolved Simple controller.*

seem to produce some locomotion behaviour when evaluated with this controller, while the least adaptive controller is the one evolved for the *Minibot* configuration.

4.3 Neural Controller

The *Simple controller* model was extended by replacing the amplitude-offset controlled oscillator, (Equation 3.1) from section 3.2, with a ANN, which should in theory make the controller more adaptive, as in this model the output of the controller is not a fixed value, but can vary based on the dynamics of the system. Similar to the *Simple controller*, all the parameters, including the synaptic weighs

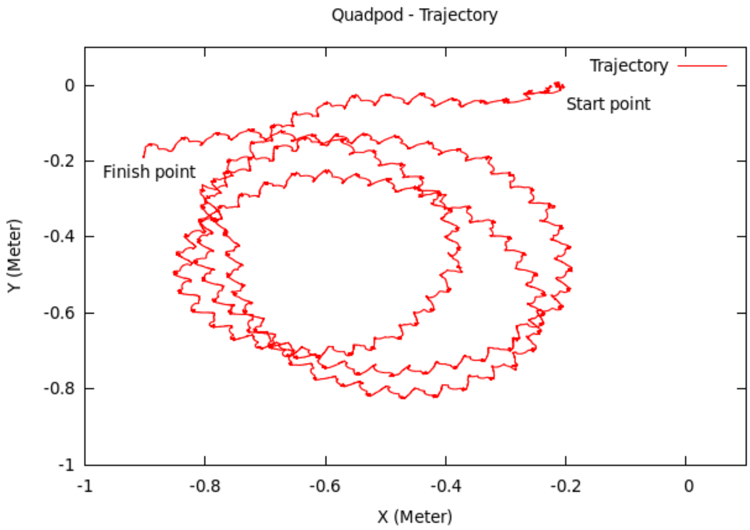


Figure 4.6: *Graph containing a plot of the trajectory of the Quadropod robot, when evaluated with the best evolved Simple controller.*

		Simple Controller				
		Minibot	Tripod	Quadropod	Y-bot	Lizard
Robot	Minibot	3.46605	N/A	1.36768	2.39138	2.88059
	Tripod	0.229033	2.12751	0.757838	1.24344	0.737008
	Quadropod	U/PR	3.09409	3.9091	U/PR	0.634184
	Y-bot	U/PR	U/PR	6.39763	6.9122	4.06972
	Lizard	U/PR	U/PR	U/PR	U/PR	5.41585

Table 4.3: *Cross-evaluation table of best evolved Simple controllers evaluated with every configuration.*

of the ANN, were optimised using Evolutionary algorithm. Cross-evaluating the best evolved *Neural controller* for every configuration on every configuration produced the results as shown in Table 4.4.

Comparing the diagonal components of Table 4.4 and Table 4.3, it could be observed that each configuration performs slightly better on its best evolved *Simple controller* compared to its best evolved *Neural controller*. But, adaptive oscillator of the *Neural controller* provides slightly better adaptability to change in configuration, as the total number of stable locomotion gaits obtained with this

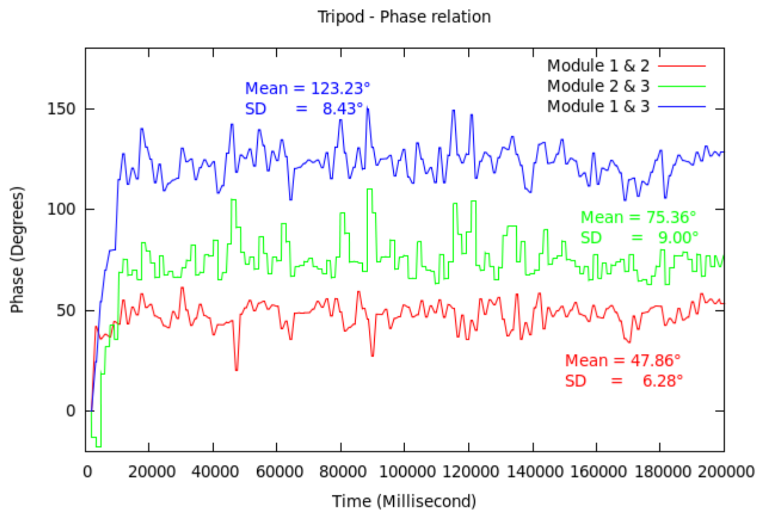


Figure 4.7: *Graph containing phase relation between modules in a Tripod configuration when evaluated with the best evolved Simple controller.*

		Neural Controller				
		Minibot	Tripod	Quadropod	Y-bot	Lizard
Robot	Minibot	3.08634	2.6017	N/A	N/A	2.59972
	Tripod	0.253527	1.93462	1.31414	1.15408	1.25913
	Quadropod	0.233509	0.748499	3.56558	U/PR	2.67733
	Y-bot	2.24279	1.75749	6.32351	6.54632	5.06163
	Lizard	U/PR	4.52042	U/PR	U/PR	5.71466

Table 4.4: *Cross-evaluation table of best evolved Neural controllers evaluated with every configuration.*

controller is 19 out of a total of 25 evaluations, in contrast to 16 stable locomotion gaits obtained when cross-evaluated with *Simple controller*. The graph in Figure 4.14 is indicative of the same conclusion as well, which shows that the average speed of locomotion is higher in three out of five best evolved *Neural controllers*, when compared with best evolved *Simple controllers*. Also, the over all average speed of locomotion, achieved with best evolved *Neural controllers*, when cross-evaluated, is higher in comparison to best evolved *Simple controllers*.

The *Neural controller* can be seen as a function with the current actuator

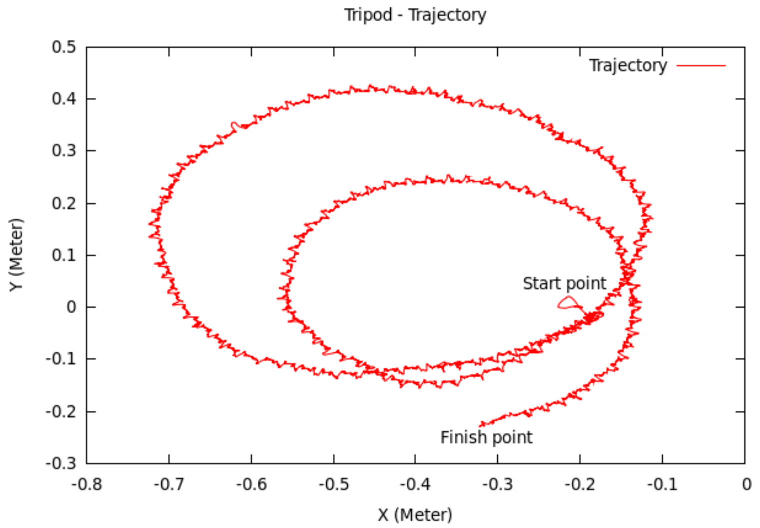


Figure 4.8: *Graph containing a plot of the trajectory of the Tripod robot, when evaluated with the best evolved Simple controller.*

position as its input, and its output used as the control signal of the module's actuator. To study the oscillatory characteristic and the adaptability of the *Neural controller*, an input-output map of the ANN optimized for each configuration was generated by discretizing the input, which range between $-90\,°$ and $90\,°$, at a resolution of $0.5\,°$. Figure 4.15 contains five plots, each of which is an input-output map of the ANN of the best evolved *Neural controller* of each of the five configurations.

Each input-output map of the ANN takes the form of (Equation 4.1), which explains the oscillatory characterstic of the ANN, as the sign of the output of (Equation 4.1) is opposite to that of its input. The input-output map of the *Lizard* ANN fits right in between the rest of the maps, which could be seen as the reason why *Neural controller* evolved for the *Lizard* configuration is the most adaptive, as the rest of the configurations produce a locomotion gait when evaluated with it.

$$f(x) = -1(tanh(x)) \tag{4.1}$$

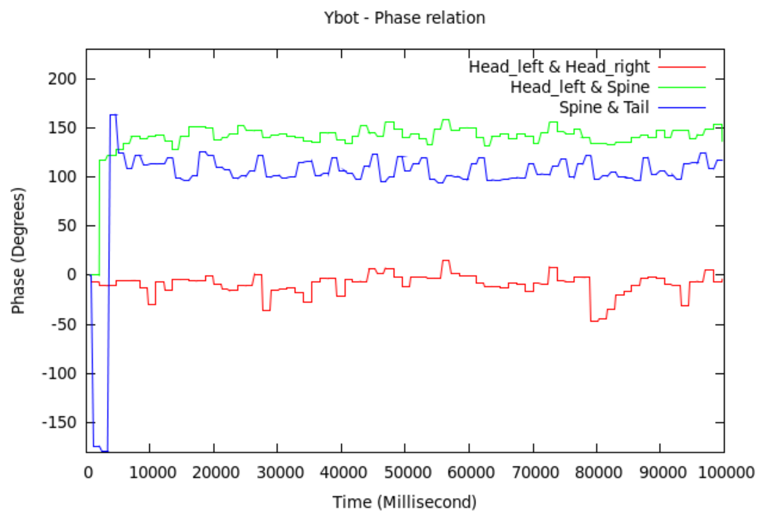


Figure 4.9: *Graph containing phase relation between modules in a Y-bot configuration when evaluated with the best evolved Simple controller.*

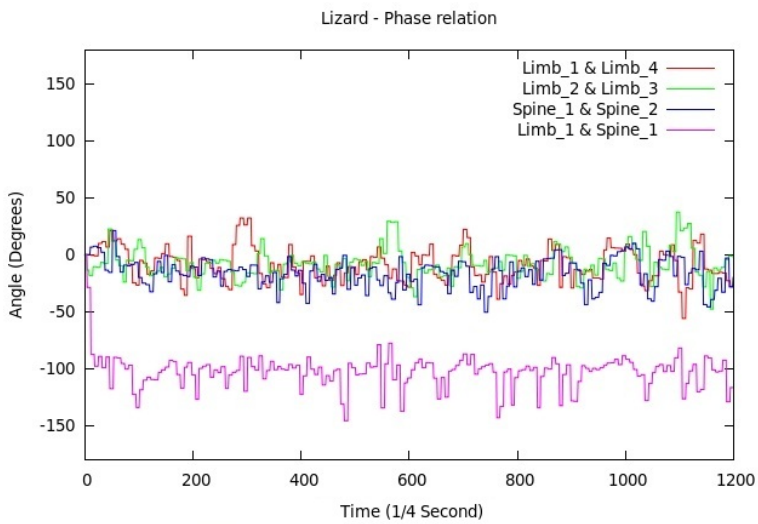


Figure 4.10: *Graph containing phase relation between some pairs of modules in the Lizard configuration when actuated with the Neural controller. The phase angle is represented as a value between -180 and +180 degrees for better visualization.*

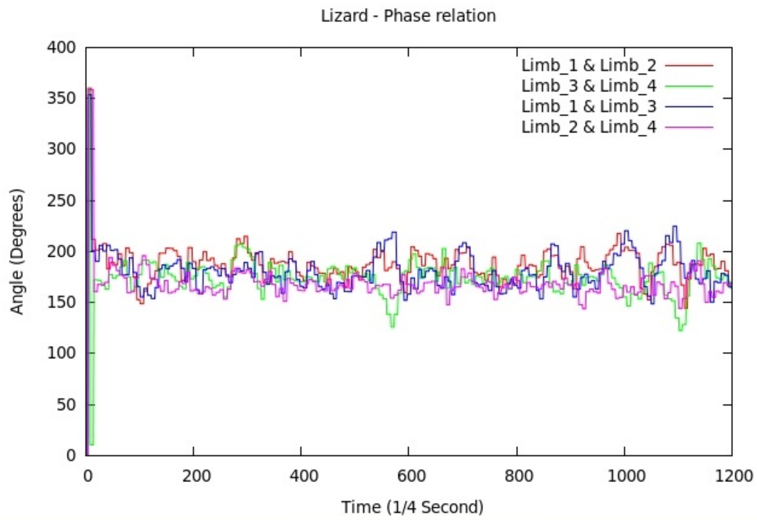


Figure 4.11: *Graph containing phase relation between a few other pairs of modules in the Lizard configuration when actuated with the Neural controller. The phase angle is represented as a value between 0 and +359 degrees for better visualization.*

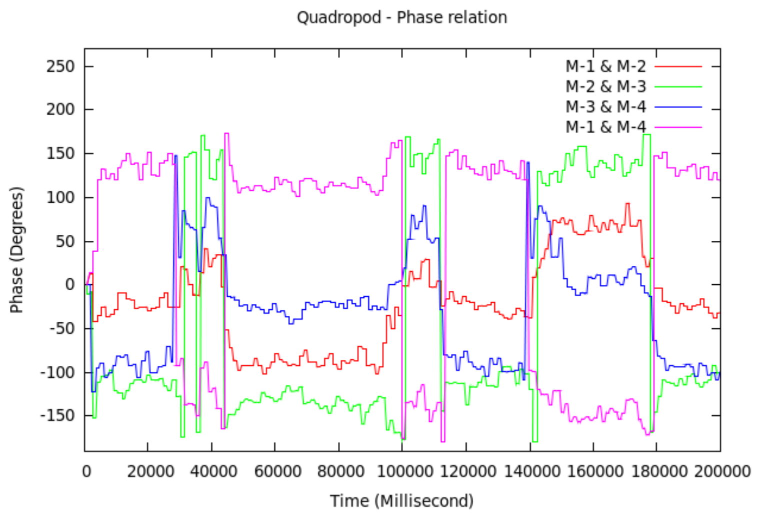


Figure 4.12: *Graph demonstrating unstable phase relation between modules in the Quadropod configuration when evaluated with Simple controller optimised for Y-bot configuration.*

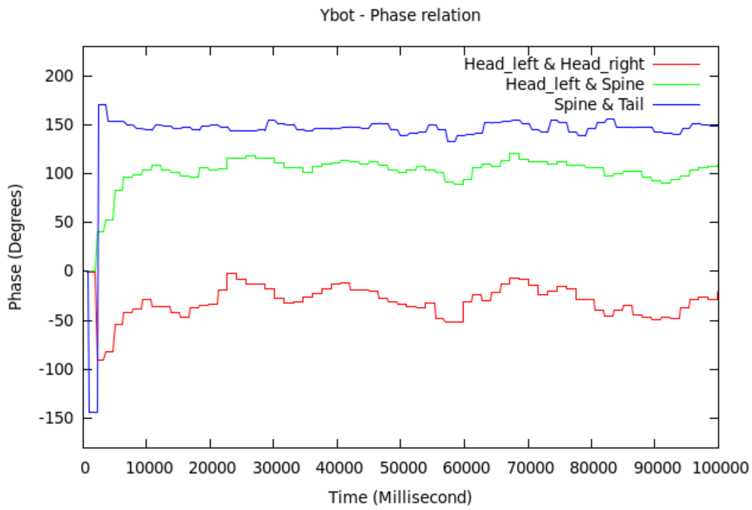


Figure 4.13: *Graph demonstrating stable phase relation between modules in the Y-bot configuration when evaluated with Simple controller optimised for Lizard configuration.*

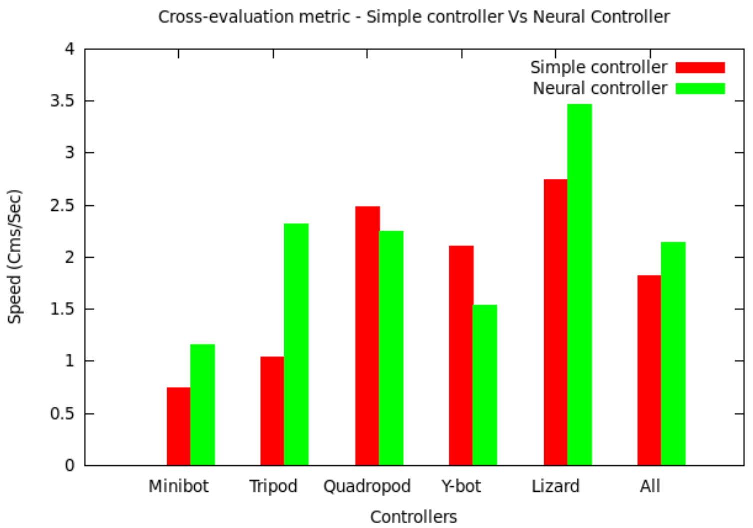


Figure 4.14: *Graph summarising cross-evaluation with average speed of locomotion between Simple and Neural controllers from Tables Table 4.3 and Table 4.4.*

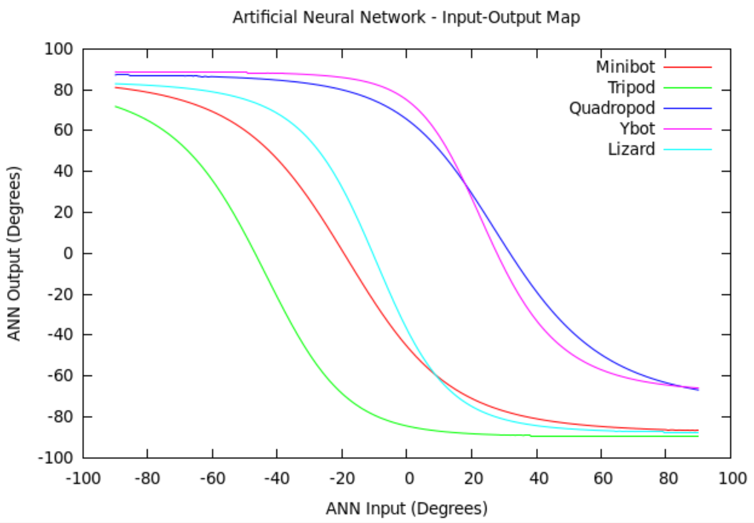


Figure 4.15: *Input-Output map of the ANN of the best Evolved Neural Controller of each of the five configurations.*

Chapter 5

Discussion

It has been possible to optimize, both the *Simple controller* and the *Neural controller*, for each of the five modular robotic configurations discussed in this work. In each evaluation, modules start to oscillate in phase, but then quickly develop and settle into a steady phase difference. The controller relies on inter-modular force that exist in a configuration, and these forces influences, as explained in section 3.2 (Equation 3.3), the module's decides on when the next control signal is sent to it's actuator. As explained in section 3.1, the intra-configuration forces on all the modules in a configuration, is not identical, due to which modules settle into oscillation with different phase, which can be seen as subtle difference in local behavior of individual robots in a multi-robotics system, that results in the emergence of a global behavior in the form of stable locomotion gait. The varying inter-modular force across a given configuration is a result of the morphology of the robotic organism, which also ensures a limit cycle behavior of the emerged locomotion gait.

Several different evolution of both *Simple* and *Neural controllers*, on *Minibot* and the *Y-bot* configurations consistently resulted in a caterpillar like locomotion gait. As a result of their morphological symmetry, the *Tripod* and the *Quadropod* configurations consistently produced a circular trajectory locomotion gait. The

Lizard configuration resulted in two different locomotion gaits. One, a reptilian like quadruped walking gait, which always emerged as a result of optimizing its *Simple controller*, and on the majority of occasions as a result of optimizing its *Neural controller*. Another kind of locomotion gait that emerged, on a few instances of optimizing its *Neural controller*, was a side walking gait, which made the organism move in the direction perpendicular to its body length.

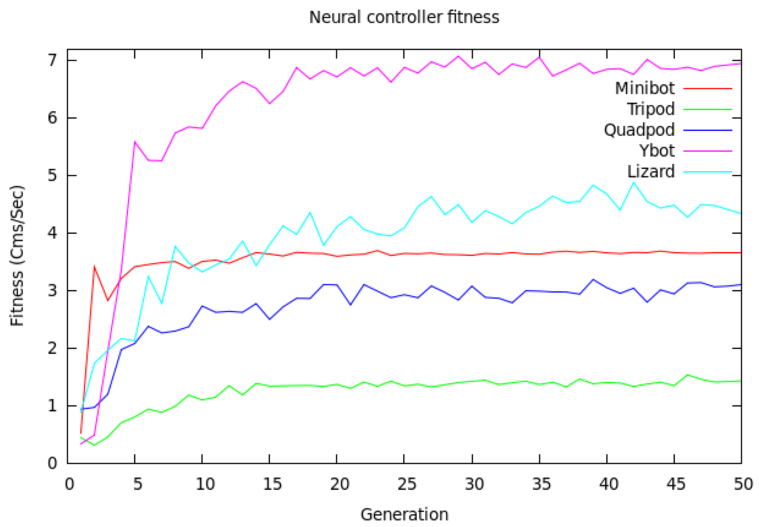


Figure 5.1: *Plots of fitness value of best performing candidate solution at each generation of the Neural controllers evolved for each of the five configurations.*

The learning curve of a controller differed for each configuration. The *Minibot* and the *Y-bot* configurations has the steepest learning curves, where as the *Lizard* configuration has the most gradual learning curve, as seen in Figure 5.1. Although the size of the search space for optimizing the respective *Neural controllers* is the same for any configuration, the *Lizard* configuration being a complex six module organism has a smaller optimum solution subspace, and so takes longer to converge to it's solution subspace.

Generating the input-output map of the ANN of the best evolved *Neural controllers* of each configuration shows that all the five controllers share a similarity between them, as they are all variants of the $-1(tanh())$ function. This similarity between the controllers in turn explains how a controller evolved for one configuration (e.g. *Lizard*) can produce a locomotion gait in another configuration (e.g. *Y-bot*), although the two configurations are very different from each other, as well as the two locomotion gaits that emerge in the respective configurations share no similarity between them as well.

Chapter 6

General conclusions and future work

In a multi-robot system like modular robots, coordination among modules is required to produce a stable locomotion gait, and with the proposed controllers, it has been possible to demonstrate how such coordination among modules can emerge based only on indirect local interaction between connected modules and between the modules and their environment, without the need for any direct communication between them. Furthermore, by cross-evaluating the controller, we have been able to demonstrate the dependency of the emerged locomotion gait on the morphology of the robot, supporting the notion of embodiment in a robot.

The proposed controllers have been successfully tested on simulated robot configurations, but it is very important to validate the controller on a real robotic configuration, built with real *Y1* modules, as the main concept of the proposed controller is based on physical interaction between connected modules, and between the modules and its environment. The best evolved *Y-bot Neural controller* was tested on a real *Y-bot* configuration. The evaluation resulted in a very unstable caterpillar locomotion gait, because a controller optimized for a virtual robot does not transfer well onto the real world robot, due to inconsistency in modeling friction, force, motor torque, etc. between the real world and the simulated world. Controller parameters, for a real modular robotic configuration, must be optimized in the real word, and this can be achieved through embodied evolution [Watson et al., 1999]. Currently, a setup for performing embodied evolution for optimizing a *Neural controller* for a real *Y-bot* configuration is underway.

In the current controller model, the actuator rotation speed threshold parameter β of the controller (as explained in (Equation 3.3) in section 3.2), although optimized for a particular configuration during evolution, is a constant during the control phase. If the value of the parameter β can adapt during the control phase, based on the robotic configuration, then it could increase the adaptability of the controller to changes in the configuration. This aspect will be studied and implemented in the next version of the controller.

Control signals generated by both *Simple* and *Neural controllers* take a sinusoidal/triangle wave from, which is simple and suffice crawling kind of locomotion gait in 2D robotic organisms, like the ones experimented with in this work. More complex oscillatory patterns would be needed for locomotion in 3D robotic organisms with multiple DOF limbs. For example, in a 3D quadruped robotic organism, which has 3DOF per limb (Each limb formed by 3 independent modules), would need both, coordination among limb modules (intra-limb coordination), as well as coordination among limbs themselves (inter-limb coordination), to produce a locomotion gait. It would not be possible to achieve locomotion by applying either sinusoidal or triangle oscillator, since locomotion in such a configuration would need some limb joints to oscillate with a pause, which cannot be achieved with a continues wave generator such as a sinusoidal or a triangle oscillators. A more complex controller, that can produce complex and adaptive oscillator patters needs to be developed in the future version of the proposed controller.

References

Heterogeneity of Autonomous Robotic Modules for the Reliability of a Self-Reconfigurable Multi-Robot Organisms, San Francisco, 2011.

Andres Castano, Alberto Behar, and Peter Will. The Conro modules for reconfigurable robots. *IEEE/ASME Trans. Mechatronics*, 7(4):403–409, December 2002.

Holk Cruse. What mechanisms coordinate leg movement in walking arthropods? *Trends in Neurosciences*, 13(1):15–21, 1990. URL `http://dx.doi.org/10.1016/0166-2236(90)90057-H`.

Holk Cruse, J. Dean, Volker Dürr, T. Kindermann, Josef Schmitz, and M. Schumm. *A decentralized, biologically based network for autonomous control of (hexapod) walking*, pages 383–400. Neurotechnology for Biomimetic Robots. MIT Press, 2002.

Rosen Diankov and James Kuffner. Openrave: A planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, Robotics Institute, Pittsburgh, PA, July 2008.

Volker Dürr, André Frank Krause, Josef Schmitz, and Holk Cruse. Neuroethological concepts and their transfer to walking machines. *International Journal of Robotics Research*, 22(3-4):151–167, 2003. ISSN 0278-3649.

J. Gonzalez-Gomez. *Modular Robotics and Locomotion: Application to Limbless robots*. PhD thesis, 2008.

J. Gonzalez-Gomez and Eduardo I. Boemo. Motion of minimal configurations of a modular robot: Sinusoidal, lateral rolling and lateral shift. In *CLAWAR*, pages 667–674, 2005.

Feili Hou and Wei-Min Shen. Hormone-inspired adaptive distributed synchronization of reconfigurable robots. In *The 9th Intl. Conf. Intelligent and Autonomous Systems (IAS-9)*, Tokyo, Japan, March 2006.

Fumiya Iida and Rolf Pfeifer. Self-stabilization and behavioral diversity of embodied adaptive locomotion. In Fumiya Iida, Rolf Pfeifer, Luc Steels, and Yasuo Kuniyoshi, editors, *Embodied Artificial Intelligence*, volume 3139 of *Lecture Notes in Computer Science*, pages 119–129. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22484-6. doi: 10.1007/978-3-540-27833-7_9. URL `http://dx.doi.org/10.1007/978-3-540-27833-7_9`.

Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, 2008.

Akiya Kamimura, Haruhisa Kurokawa, Eiichi Yoshida, Kohji Tomita, Satoshi Murata, and Shigeru Kokaji. Automatic locomotion pattern generation for modular robots. In *ICRA*, pages 714–720, 2003.

Sunil Pranit Lal, Koji Yamada, and Satoshi Endo. Studies on motion control of a modular robot using cellular automata. In *Proceedings of the 19th Australian joint conference on Artificial Intelligence: advances in Artificial Intelligence*, AI'06, pages 689–698, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-49787-0, 978-3-540-49787-5. doi: 10.1007/11941439_73. URL http://dx.doi.org/10.1007/11941439_73.

Sunil Pranit Lal, Koji Yamada, and Satoshi Endo. Evolving motion control for a modular robot. *Applications and Innovations in Intelligent Systems XV*, pages 245–258, 2008.

R. Matthias, A. Bihlmaier, and H. Worn. Robustness, scalability and flexibility: Key-features in modular self-reconfigurable mobile robotics. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 457–463, 2012. doi: 10.1109/MFI.2012.6342996.

Tad McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990. doi: 10.1177/027836499000900206. URL http://ijr.sagepub.com/content/9/2/62.abstract.

Rico Moeckel, Cyril Jaquier, Kevin Drapel, Elmar Dittrich, Andres Upegui, and Auke Jan Ijspeert. Yamor and bluemove - an autonomous modular robot with bluetooth interface for exploring adaptive locomotion. In *CLAWAR*, pages 685–692, 2005.

Chandana Paul. Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54(8):619 – 630, 2006. ISSN 0921-8890. doi: 10.1016/j.robot.2006.03.003. URL http://www.sciencedirect.com/science/article/pii/S0921889006000613. ¡ce:title¿Morphology, Control and Passive Dynamics¡/ce:title¿.

Soha Pouya, Jesse van den Kieboom, Alexander Spröwitz, and Auke Jan Ijspeert. Automatic gait generation in modular robots: "to oscillate or to rotate; that is the question". In *IROS*, pages 514–520, 2010.

S. Russo, K. Harada, T. Ranzani, L. Manfredi, C. Stefanini, A. Menciassi, and P. Dario. Design of a robotic module for autonomous exploration and multi-mode locomotion. *Mechatronics, IEEE/ASME Transactions on*, PP(99):1–10, 2012. ISSN 1083-4435. doi: 10.1109/TMECH.2012.2212449.

Behnam Salemi, Wei-Min Shen, and Peter Will. Hormone-controlled metamorphic robots. pages 4194–4199, 2001.

Behnam Salemi, Mark Moll, and Wei-Min Shen. SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system. Beijing, China, October 2006a.

Behnam Salemi, Peter Will, and Wei-Min Shen. Autonomous discovery and functional response to topology change in self-reconfigurable robots. In Dan Braha, Ali A. Minai, and Yaneer Bar-Yam, editors, *Complex Engineering Systems: Science Meets Technology*, pages 364–384. 2006b.

Wei-Min Shen, Behnam Salemi, and Peter Will. Hormones for self-reconfigurable robots. In *Intl. Conf. on Intelligent Autonomous Systems (IAS-6)*, pages 918–925, Venice, Italy, 2000. IOS Press.

Wei-Min Shen, Behnam Salemi, and Peter Will. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. 18(5):700–712, October 2002.

Alexander Spröwitz, Rico Moeckel, Jérôme Maye, and Auke Jan Ijspeert. Learning to move in modular robots using central pattern generators and online optimization. *I. J. Robotic Res.*, 27(3-4):423–443, 2008.

R.A. Watson, S.G. Ficiei, and J.B. Pollack. Embodied evolution: embodying an evolutionary algorithm in a population of robots. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages –342 Vol. 1, 1999. doi: 10.1109/CEC.1999.781944.

Martijn Wisse. Three additions to passive dynamic walking; actuation, an upper body, and 3d stability. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, volume 1, pages 113–132 Vol. 1, 2004. doi: 10.1109/ICHR.2004.1442118.

Mark Yim. *Locomotion With A Unit-Modular Reconfigurable Robot*. PhD thesis, 1994.

Mark Yim, David Duff, and Kimon Roufas. Polybot: A modular reconfigurable robot. In *ICRA*, pages 514–520, 2000.

Houxiang Zhang, J. Gonzalez-Gomez, and Jianwei Zhang. A new application of modular robots on analysis of caterpillar-like locomotion. In *Mechatronics, 2009. ICM 2009. IEEE International Conference on*, pages 1–6, 2009. doi: 10.1109/ICMECH.2009.4957149.