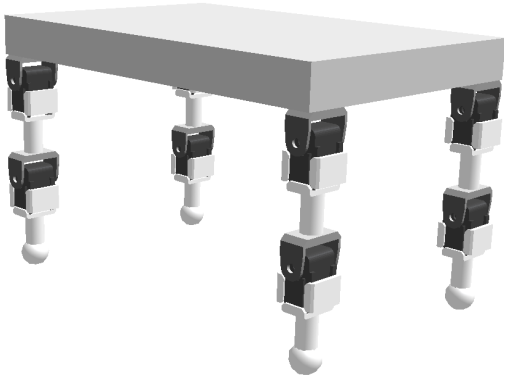
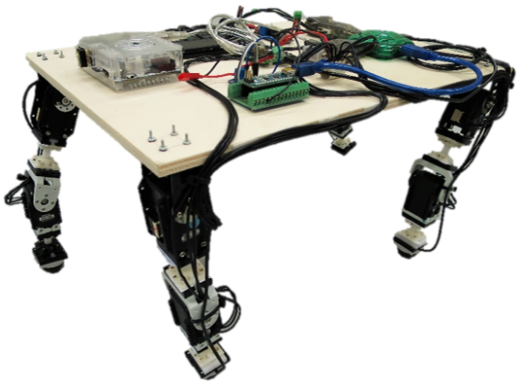
TÉCNICO LISBOA





CPG and Tegotae-based Locomotion Control of Quadrupedal Modular Robots

Rui Filipe Morais de Vasconcelos

Thesis to obtain the Master of Science Degree in

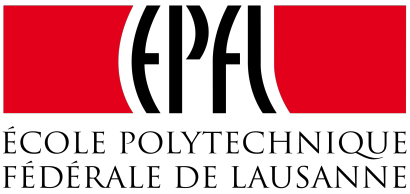
Mechanical Engineering

Supervisors: Prof. Paulo Jorge Coelho Ramalho Oliveira
Prof. Auke Jan Ijspeert

Examination Committee

Chairperson: Prof. João Rogério Caldas Pinto
Supervisor: Prof. Paulo Jorge Coelho Ramalho Oliveira
Member of the Committee: Prof. José Alberto Santos Victor

November 2016



This master project was developed at École Polytechnique Fédérale de Lausanne (EPFL) in the sphere of BioRobotics Laboratory (BIOROB) and founded by the Swiss National Science Foundation.

Acknowledgments

Initially, I would like to thank professors Paulo Oliveira, José Santos-Victor and Auke Ijspeert for the help provided to allow the realization of my thesis inserted in Biorobotics laboratory and EPFL.

To professor Auke and all the BIOROB team members for the exceptional reception and help throughout the whole project, without which I would not have been able to accomplish this task.

A special thanks to my supervisors at EPFL, Simon Hauser and Florin Dzeladini, for the great patience and availability during the project, they really made my learning curve steeper, and to Professor Paulo Oliveira for the incredible availability even at distance.

I am grateful for the amazing friends I have in Portugal and that I met in Switzerland, whose great support and motivation revealed to be crucial in my happiness and consequently working capacity.

Finally, in I would like to thank my family in Portuguese: Um enorme agradecimento à minha mãe, Maria Dulce, pelo apoio incondicional e exemplo de trabalho incansável, ao meu pai, José Fernando, por me ter transmitido a sua paixão por matemática, espirito crítico e foco nos objectivos e à minha irmã, Sara, pelos ensinamentos e amizade.

Resumo

Animais apresentam capacidades incríveis do ponto de vista de controlo em locomoção, que a Ciência procura compreender e replicar em robots. A habilidade de adaptar movimentos com base em mudanças morfológicas é especialmente interessante no ramo da robótica reconfigurável.

É comumente aceite que dois componentes representam um papel preponderante no controlo de movimentos: Geradores de Padrões Centrais (CPG) e mecanismos de reflexos. Respectivamente, duas estratégias de controlo inspiradas nestes conceitos foram estudadas no âmbito desta dissertação, uma rede CPG optimizada usada em anel aberto, e um método descentralizado baseado em regras de feedback "Tegotae".

Um robot quadrupede foi escolhido para o estudo, sendo modelado no simulador *Webots*, onde algoritmos de optimização e procura foram aplicados, e implementado em hardware, possibilitando experiências de validação.

Este projecto teve como foco analisar performance de modos de locomoção impostos usando a primeira estratégia, comparando resultados com locomoção emergente da abordagem descentralizada. Esta comparação provou que controlo baseado em Tegotae é capaz não só de conduzir o sistema para um comportamento de ciclo limite estável em estado estacionário, como o conduz para um extremamente eficiente. Diversos modos de locomoção encontrados na natureza foram reproduzidos com uma parametrização de trajectória simples, revelando custo energético e estabilidade mecânica destes.

Controlo baseado em Tegotae foi criteriosamente analisado, sendo propostas melhorias, nomeadamente escalamento com frequência e coeficiente de atracção variante no tempo. Feedback binário foi posto à prova pela primeira vez com sucesso, revelando contudo perdas na performance.

Diversas ferramentas de *software* e *hardware* foram desenvolvidas para uso futuro.

Palavras-chave: CPG, Tegotae, Controlo Descentralizado, Locomoção Emergente, Quadrupede.

Abstract

Animals exhibit astonishing capabilities in terms of locomotion control which Science aims at understanding and reproducing in robots. The ability to adapt movements based on morphological changes is specially interesting in the field of reconfigurable robotics.

It is commonly accepted that two components play an important role in the way animals perform control: Central Pattern Generators and Reflex Mechanisms. Respectively, two bio-inspired control strategies were studied under the scope of this thesis work, an optimized open loop CPG network, and a closed loop decentralized control of the limbs by Tegotae feedback rules.

A quadruped robotic platform was chosen for the study, being modelled in *Webots* simulator, where optimization and search algorithms were employed, and implemented on hardware, where validation experiments took place.

The focus of this project was on analyzing performance of imposed gaits, with the first strategy, comparing it with emergent gaits from the decentralized approach. This comparison proved that Tegotae-based control is not just able to drive the system towards a stable steady state limit cycle behaviour, as it does so for a very energy efficient one.

Several gaits found in nature were reproduced with a simple trajectory parametrization method, indicating energetic cost and mechanical stability of these.

Tegotae-based control was thoroughly analyzed, and some improvements are proposed, namely scaling with frequency and variable attraction coefficient. Binary feedback was also put at proof for the first time, showing feasibility but some performance loss.

Diverse software and hardware tools were developed for use in future work.

Keywords: Bio-inspired, Biorobotics, CPG, Tegotae, Decentralized Control, Emergent Gaits, Quadruped.

Contents

List of Figures

List of Tables

Nomenclature

A_i	$\rightarrow$	desired amplitude of oscillation for oscillator i
B_i	$\rightarrow$	binary force feedback for oscillator i
BH	$\rightarrow$	body height
BL	$\rightarrow$	body length
BWd	$\rightarrow$	body width
BWg	$\rightarrow$	body weight
c_l	$\rightarrow$	step cycle number of limb l
CoT	$\rightarrow$	cost of transport
D_t	$\rightarrow$	total distance traveled
df	$\rightarrow$	imposed duty factor
E_t	$\rightarrow$	total energy consumption
f	$\rightarrow$	frequency of oscillation
f_r	$\rightarrow$	reference frequency of oscillation
$g(\cdot)$	$\rightarrow$	generic coupling function
$G(\cdot)$	$\rightarrow$	generic function
h_{st}	$\rightarrow$	height relative to the workspace boundary performed during stance phase
h_{sw}	$\rightarrow$	height relative to the workspace boundary performed during swing phase
I_{in}	$\rightarrow$	input current
k	$\rightarrow$	time instant
k_e	$\rightarrow$	time instant of end of simulation
k_m	$\rightarrow$	time instant of measurement initialization
l_1	$\rightarrow$	length of section 1 of the planar limb
l_2	$\rightarrow$	length of section 2 of the planar limb
N_i	$\rightarrow$	force feedback for oscillator i
p_1	$\rightarrow$	transition point from swing to stance defined in cartesian space
p_2	$\rightarrow$	mid-point of stance phase defined in cartesian space
p_3	$\rightarrow$	transition point from stance to swing defined in cartesian space
p_4	$\rightarrow$	mid-point of swing phase defined in cartesian space

$\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}$	$\rightarrow$	position of limb end-effector in base reference frame
$\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}{}_x$	$\rightarrow$	x component of $\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}$
$\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}{}_y$	$\rightarrow$	y component of $\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}$
P_m	$\rightarrow$	mechanical power consumption
PI	$\rightarrow$	periodicity indicator of steady state gait
r_i	$\rightarrow$	current amplitude of oscillator i
std	$\rightarrow$	standard deviation
S	$\rightarrow$	instantaneous mechanical stability metric
S_t	$\rightarrow$	global mechanical stability metric
SN_l	$\rightarrow$	number of step cycles performed by limb l
SN_t	$\rightarrow$	average number of step cycles performed by all limbs
t	$\rightarrow$	continuous time
t_e	$\rightarrow$	time of end of simulation
t_m	$\rightarrow$	time of measurement initialization
t_{ss}	$\rightarrow$	time to achieve steady state
T	$\rightarrow$	discretization time step
v_d	$\rightarrow$	desired traveling speed
$\overline{v_{CM}}$	$\rightarrow$	average traveling speed of center of mass
$V(\cdot)$	$\rightarrow$	potential function
V_{in}	$\rightarrow$	input voltage
w_{ij}	$\rightarrow$	weight of coupling between oscillators i and j
x_i	$\rightarrow$	position output of oscillator i
$\overrightarrow{x_{CM}}$	$\rightarrow$	vectorial position of center of mass in global reference frame
X_i	$\rightarrow$	offset of oscillator i
Z	$\rightarrow$	fitness function

α_i $\rightarrow$ parameter for speed of convergence in amplitude of oscillator i

ϵ_t $\rightarrow$ energy efficiency metric

ϵ_t* $\rightarrow$ energy efficiency of best particle in results of PSO

θ_{max} $\rightarrow$ trajectory parameter describing hip magnitude during transition from swing to stance

Θ $\rightarrow$ pitch

λ $\rightarrow$ generic variable

μ $\rightarrow$ average value

ϑ_j $\rightarrow$ angular position of joint j

σ $\rightarrow$ Tegotae attraction coefficient

σ_b $\rightarrow$ Tegotae attraction coefficient for binary feedback

σ_c $\rightarrow$ Tegotae attraction coefficient specified as constant

σ_r $\rightarrow$ reference Tegotae attraction coefficient

τ_j $\rightarrow$ torque applied by of joint j

ϕ_i $\rightarrow$ current phase of oscillator i

ϕ_t $\rightarrow$ transition phase between stance and swing

Φ $\rightarrow$ roll

ψ_{ij} $\rightarrow$ desired phase shift between oscillators i and j

Ψ $\rightarrow$ yaw

ω $\rightarrow$ angular frequency of oscillation

Chapter 1

Introduction

1.1 General Introduction

During the past century, with the thriving development of Systems, Control and Estimation theories, increasingly complex systems have been effectively controlled, recurring to more and more elaborated strategies. With the evolution of these techniques, several attempts to mimic animal locomotion behaviours on robots, have increased in number and diversification, integrating fields of research such as Biomimetics and Biorobotics. The complexity and impressing capabilities of animals to perform predictions, control and decision making, represents therefore an amazing challenge for cooperation between Biology, Neuroscience and Systems Engineering, as well as an opportunity for development of new robots and strategies based on knowledge acquired from Biology, giving back to biologists the possibility to test hypothesis on robotic platforms.

From the control point of view, one of the most interesting features developed in nature through evolution is the competence for adaptability. Animals go through morphological changes during growth and aging, being able to readjust their actuation principles by sensing and learning, and in case of an injury, an immediate adaptation of gait is performed. In addition, the surrounding environment is taken into account allowing locomotion in rough terrain, constrained spaces or even in water, using gait transitions to match the desired speed, avoiding frequency saturation by the limbs.

Specifying on quadrupedal mammals, the anatomical properties responsible for control of body dynamics, such as brain, nervous, muscular and vision systems, are common despite the number and diversity of species. Additionally, a scaling of bone, body, and limb dimensions with body weight can be encountered [1] [2], showing the existence of general principals that define animal morphological evolution in favor of better locomotion. Also in the behaviour side, correlations between the speed, stride frequency and energetic cost of locomoting have been drawn with body weight and limb length in [3] and [4]. Relying on such body similarities and scaling properties, it is implicit that the control principles behind the incredible dynamical capabilities of animals should be universal.

Several experiments, including induced gait transition on a decerebrated cat (Brown, 1972), show that the underlying mechanisms responsible for locomotion are not fully handled by the brain. This is supported by the fact that having a big and highly developed brain does not seem to be mandatory to perform complicated movements. It is then currently accepted that four main ingredients take part in animal locomotor control:

1. A simple descending modulation from the brain, responsible for the locomotion trigger and having been reproduced by electrical stimulation of a section of the brain stem denominated Mesencephalic Locomotor Region (MLR).

2. **Central Pattern Generators (CPG)**, neural circuits found in both vertebrates and invertebrates that transform simple non-periodic descending stimulus from the brain into periodic locomotion movements, rhythmic patterns. These have been proven to play an important role in the control mechanisms used in animals from the lamprey [5], to cats [6], to humans [7]. See [8] for a biological overview on these neural circuits on mammals spinal chord.

3. Reflex mechanisms based on sensory feedback, responsible for the adaptation of movements.

4. Robust body mechanics necessary for stability, which include compliance and consequent passive dynamics, helpful for the execution of dynamical gaits (see [9]).

With an inspiration on the described components of animal locomotor control, several strategies were developed based on the concept of CPG and reflex mechanisms. Modelled by networks of phase-oscillators, CPG-based techniques have been employed to diverse control applications, from salamander [10], to quadruped [11] [12] [13], to humanoid robots [14] [15] [16] to exoskeletons [17]. In a similar fashion, a first strategy used in this thesis relies on such concept, with an open loop control, optimized recurring to Particle Swarm Optimization (see section 2.4). An overview on open loop CPG-based control can be seen in 2.2.

A second control approach employed in this master project is based on local reflex mechanisms, using external feedback to affect the limb phase through a Tegotae rule, in a decentralized fashion. Tegotae based control has been shown to result on the visualization of emergent gaits (diagonal sequence walk, lateral sequence walk, trot, pace, rotary gallop and bound) and gait transitions [24] by a decentralized control of each limb, with adaptation to the dynamical characteristics of the robot such as weight distribution [25] [26]. Detailed explanation on this approach can be found in 2.3.

As mentioned, besides the relevance of the neural system in the control of muscle activity, the passive dynamics present in the body through compliance plays a very important role in gait performance and transitions, allowing the necessary conservation of energy, as well as robustness. In [18], the dynamics of a compliant quadruped robot (*cheetah-cub*) is discussed, and this effect on snake and self-reconfigurable modular robots can be found in [19] and [20], respectively. In the scope of this project, a fully stiff quadruped robot was developed in simulation and hardware, with the integration of removable passive elements, creating a versatile test platform, but leaving the implications of such additional dynamics as future work.

1.2 Motivation

Self-reconfigurable modular robots are groups of equal systems, with locomotion and connection abilities which allow them to, starting from several decoupled modules, create highly complex structures capable of morphological changes. Examples of such robotic systems are M-TRAN [21] or Roombots [22]. This characteristic of reconfiguration creates an highly complex challenge where control and learning algorithms have an emphasized importance. Besides the high number of degrees of freedom and consequent redundancy that these systems acquire, upon each change of morphology, the differential equations that model the system dynamics have to be redefined. Therefore, a classical control approach becomes exponentially more complex with the number of modules involved, being unfeasible at a certain point. In addition, numerous mechanical problems limit the capabilities of this type of robots, such as backlash integration, and limited number of attached modules given a torque to weight motor relationship.

Classical control approaches rely on models derived mathematically or through identification methods and, given the necessary sensors and actuators, the dynamical response of such mechanisms can be driven towards a goal reference. However in the case of modular robotics, the complexity of the models for a given configuration may require the usage of computational methods instead of classic modelling. The usage of optimized CPG networks targets this problem, allowing the achievement of sub-optimal locomotion methodologies, for an n degrees of freedom structure, given a prior parametrization and reduction of the extensive search space. An example of this reduction and redundancy handling with CPG can be found in [23].

These computational strategies are therefore generic, working for any structure in the same manner. Although, they also require a re-optimization of the defined control parameters whenever a reconfiguration takes place. This represents therefore a high motivation for control strategies with emergent adaptability such as Tegotae based feedback. Tegotae-based feedback rules have been used on the control of quadruped, hexapod [27], and snake robots [28], nevertheless, always on robotic systems built for the effect, not being clear how specialized these need to be, and how suitable is this approach to more general structures as the ones present in modular robotics.

In order to allow locomotion in arbitrary robotic structures, two kinds of movements need to be handled: legged movements, where only the end effector has contact with the ground, and crawling movements, where the point of contact is not previously defined, creating additional problems in terms of feedback. Given the simplification on feedback gathering and extensive work developed in Tegotae-based legged locomotion, this was considered as the main focus, having included in the scope of this thesis an analysis on how to gather feedback from every module for arbitrary contact points with the ground – appendix E.

To sum, bio-inspired control techniques may have advantages in terms of control of complex robotic structures. With open loop CPG it is possible to control any morphology given a prior optimization process, however computationally expensive re-optimizations are needed after reconfiguration. With decentralized Tegotae feedback rules, emerging gaits are visualized, not being however clear what are the boundaries within which there is convergence for limit cycle behaviour, nor the quality of the solution, when compared with the optimized one. Hence, the focus of this project was to compare performance of these strategies in terms of speed, energy efficiency and mechanical stability, analyzing effect of step trajectories and showing feasibility of convergence to stable limit cycle behaviour.

For simplicity of the study, the capacity of self-reconfigurability was not a focus, having used modular robots reconfigurable by hand, but with the goal of deriving control principles applicable in both.

1.3 Contributions

This thesis aimed at studying two bio-inspired control strategies on a stiff quadruped model, an open loop CPG network and physical communication by decentralized Tegotae feedback rules. The capability of these techniques to drive a modular quadrupedal robotic system from any initial condition towards a steady state limit cycle behaviour was shown, followed by comparison of performance of both strategies in terms of locomotion speed, energy efficiency and mechanical stability.

A simple parametrization method for step trajectories is proposed, and a study on influence of such parameters was developed for the open loop approach. In the scope of this study, optimization procedures lead to the execution of several dynamical gaits in simulation, concluding that trot was the most indicated walking gait. Following such conclusion, a systematic search on the effect of step trajectory and duty factor in trot gaits resulted in the derivation of actuation principles.

Regarding Tegotae-based control, the emergent decentralized approach proved to work for a simple modular structure with steady state performance equal or higher than the optimized gait, due to the underlying effects in terms of duty factor and robustness to perturbations. The need for a scaling of the attraction coefficient with the intrinsic frequency is shown and a non-constant attraction is proposed. Finally, binary feedback was proven to be enough for convergence to trot gait, however, with negatively affected time response.

A versatile robotic platform was implemented on hardware in the scope of this thesis work, allowing validation of the above stated results, and is ought to be used in future studies on morphological changes and compliance, based on its reconfigurability properties and data analysis software developed in *MATLAB*.

Finally, a study on the possibility of usage of compliant elements for extraction of feedback from every module was performed, having resulted in a *MATLAB* tool for real-time analysis of torsion and bending, for application in future work.

The main contributions of this thesis work will be submitted to a highly rated international robotics conference.

1.4 Thesis Outline

This thesis report is composed by two main parts. A first one that aims at explaining all the decisions, methods and implementation procedures performed throughout the project, and a second one where the experiments and respective results are presented, discussed, and conclusions are drawn.

On chapter 2, necessary background concepts are introduced to allow a better understanding of the topic.

On chapter 3, implementation methods regarding simulation, hardware, parametrization, control and fitness computation are explained.

On chapter 4, optimization, search and validation experiments used to achieve the mentioned goals are introduced.

On chapter 5, the results of such experiments are exhibited and discussed.

On chapter 6, general conclusions and future work are presented.

Chapter 2

Base Concepts

The fields of Biorobotics and Biomimetics reveal an outstanding multidisciplinary character, having Mechanics, Electronics, Biology and Neuroscience merged in a symbiotic relation with the goal of understanding animal behaviour while reproducing it in robots. This chapter aims therefore at removing the need for the reader to be a specialist in any of these scientific fields, allowing him/her a better understanding of the topics ahead discussed. At first, quadrupedal methods of locomotion present in nature are exhibited, followed by introductions on the control, optimization and simulation techniques employed throughout the project.

2.1 Quadrupedal Locomotion Gaits

Ever since the first steps of Darwinist evolution, the animal ability of locomotion has shown to be a key aspect for survival. The process of moving in space went from stable swimming to very complex terrestrial and aerial locomotion methods, in a process of optimization of speed and energy efficiency that lasts for the past millions of years. Animals evolved in many directions, improving their ways of locomotion through both morphological and behavioral changes. However, besides this diversity, they possess universal capability of effectively controlling their body dynamics, even in scenarios that put their robustness at proof, by adapting their gait to the desired speed and existing surroundings.

Defining the locomotion gait by the order at which each of the four limbs are in stance phase, and by the time spent in it relative to the time in swing, characteristic denominated by **duty factor**, a group of gaits performed by animals is presented in figure 2.1 At low speeds, quadrupeds tend to use symmetrical walking gaits such as walking trot, lateral sequence (L-S) walk or diagonal sequence (D-S) walk, adjusting step size and frequency to different speeds. However, if the desired speed is high enough, the animal will go through a gait transition to running gaits, where the body dynamics has an increased importance and moments exist where none of the feet is touching the ground.

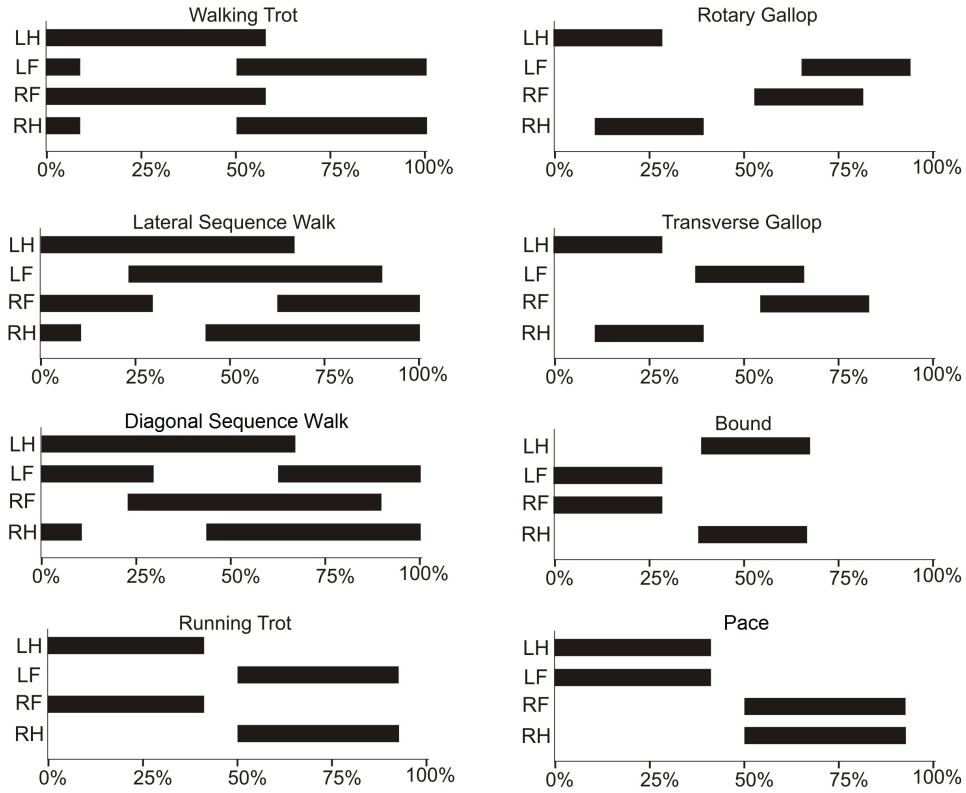


Figure 2.1: Walking and running gait types present in nature, defined by time of feet contact with the ground – left, right, fore and hindlimbs designated by L, R ,F and H respectively.

2.2 Open Loop CPG-based Control

With a base on the biological concept of Central Pattern Generators, and aiming at reproducing its capabilities of translating simple descending signals into complex rhythmic behaviours, a family of control approaches has been developed and used in bio-inspired robotics. Relying on the idea that, in nature, the patterns of movement are perfected through learning in the early stage of animals life-cycle, in robotics, this is done by parametrization and further usage of an optimization algorithm such as Particle Swarm Optimization (section 2.4). Hence, from the control point of view, the CPG is used as a reference generator block, having a simple input for multiple output references. Usually, these references serve then as an input for local controllers of each joint motors, leading to a technique non monitored through feedback, but whose actuation has been previously optimized. The block diagram of this approach is presented in figure 2.2.

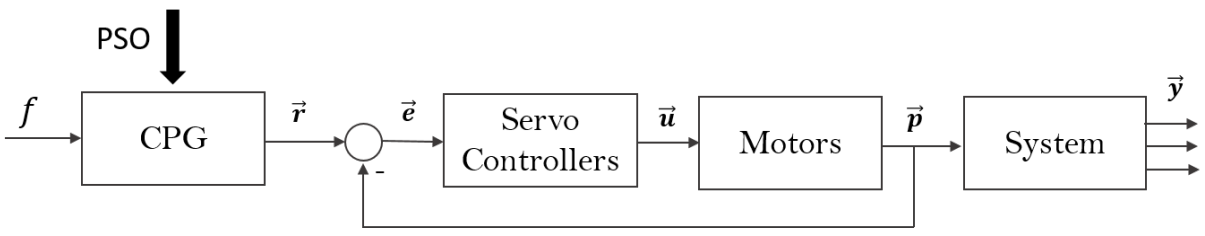


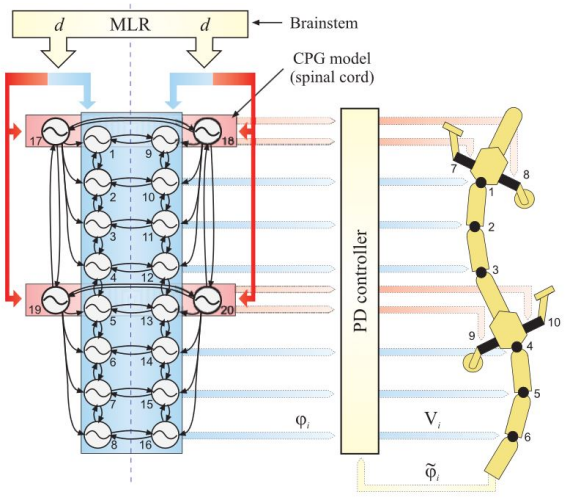
Figure 2.2: Block diagram of Open loop CPG-based control.

The neural system present in animals is modelled as a network of coupled phase-oscillators, as the one presented in figure 2.3(a), having the time evolution of each oscillator i defined by an intrinsic frequency $\omega_i = 2\pi f_i$ and by a coupling function g of the current phase of the oscillator itself, ϕ_i, and of its neighbours (equation 2.1). Having a good design of this function will result in phase-locking after a transition phase, described by a limit cycle behaviour, for regions of values of ω_i.

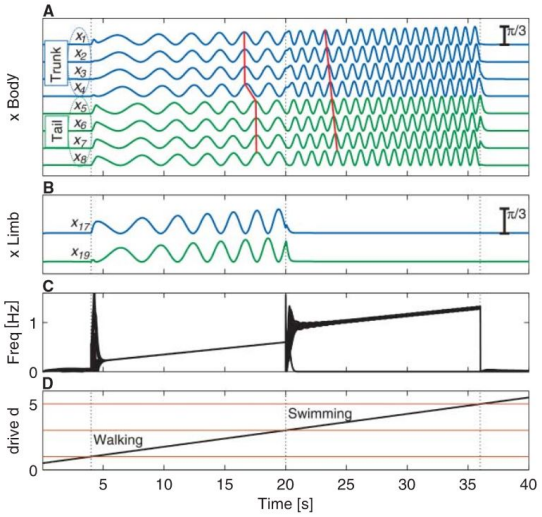
$$\dot{\phi}_i = \omega_i + g(\phi_i, \phi_j, \phi_k, \ \dots) \tag{2.1}$$

This type of control technique has the advantage of allowing distributed control with short feedback loops and ability to deal with redundancies, representing a good method to apply in robots with high number of degrees of freedom. Additionally it results, with the right combination of parameters, in limit cycle behaviour of the system, demonstrating robustness to perturbations. However, despite these advantages, a universal design methodology has not yet been set and a solid theoretical foundation for describing CPGs is still missing. For a review on CPG-based control see [29].

Taking into consideration the need for an optimization process of the actuation pattern, and the fact that the search space exponentially increases with the number of degrees of freedom of the robot to control, the usage of a good parametrization method is crucial. One of the ways to reduce the extensive search space, is to consider sinusoidal output of each phase-oscillator, having some or all these oscillators paired with the degrees of freedom (one phase-oscillator per DoF). This method was used in [10] to control a salamander robot, modelling the spinal chord through the CPG network presented in 2.3(a), and being able to reproduce gait transition from walking to swimming, simply by increasing the descending drive signal and consequently the oscillators frequency above the limbs bandwidth. The time evolution of this experiment is presented in 2.3(b).



(a) Salamander CPG model – drive signal *d* representing descending stimulation of the MLR in the brain stem.



(b) Experimental data of transition from walking to swimming given linearly increasing drive *d*.

Figure 2.3: CPG-based control of salamander robot – *"From swimming to walking with a salamander robot driven by a spinal chord model"* [10].

The described control method was implemented using the equations 2.2, where the desired amplitude A_i and offset X_i of each oscillator, and phase shift ψ_{ij} of each coupling, are optimized. The parameters α_i and w_{ij} define respectively speed of convergence in amplitude and weight of coupling from oscillator i to oscillator j.

$$\begin{cases} \dot{\phi_i} = 2\pi f + \sum_j w_{ij} \cdot r_j \cdot \sin(\phi_j - \phi_i - \psi_{ij}) & \text{Phase update} \\ \dot{r_i} = \alpha_i(A_i - r_i) & \text{Amplitude update} \\ x_i = r_i \cdot \sin\phi_i + X_i & \text{Oscillator output} \end{cases} \tag{2.2}$$

On this project, a network of one oscillator per degree of freedom was initially employed using the control method described by equation 2.2, however the usage of one oscillator per limb proved to be more adequate (see section 3.7.1). In the approach widely used in this thesis, step trajectories are parametrized and optimized along with desired phase shift ψ_{ij}, having a conversion between limb phase and end-effector position in cartesian space to be followed in joint space with aid of inverse kinematics (see 3.5, 3.6 and 3.7).

In this project, feedback was not introduced along with the neural couplings and only as a replacement of it by a Tegotae rule (2.3), however examples of this can be found in [30] and [31].

2.3 Tegotae-Based Control – Physical Interlimb Communication

"Tegotae" is the Japanese word that defines how well a perception matches an expectation. In terms of control, the concept means actuating on the unexpected feedback.

When an oscillatory actuation takes place, it is expected that parts of the oscillation will have more feedback than others. In crawling robots, the part of the oscillation that pushes the ground will have more force feedback, in swimming snake robots, the part of the oscillation that pushes the water towards the positioning of a sensor will have higher pressure measurements. On legged robots, force feedback from the feet will exist in stance phase, however it is not expected that it occurs during swing.

Based on the simple concept that, if there is contact force with the ground, then the phase of limb i, ϕ_i, should be the one correspondent to stance phase, an attraction component proportional to the ground reaction force N_i is added to the phase change rate $\dot{\phi_i}$ as described in equation 2.3, where ω represents an intrinsic oscillation frequency and σ the attraction constant, whose influence is thoroughly analyzed in this project. An explanatory diagram can be see in figure 2.4.

$$\dot{\phi_i} = \omega - \sigma\ N_i\ \cos(\phi_i) \tag{2.3}$$

By including this attraction to the phase expected given a measured feedback, independently in all the limbs, the body dynamics will induce physical communication between the limbs, dependent on the inertial characteristics of the robot, and on the passive dynamics in case of added compliance.

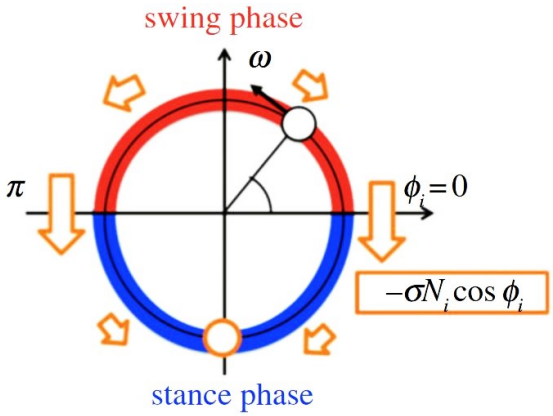


Figure 2.4: Dynamics of phase oscillator depending on local sensory feedback. Phase is pulled towards $3\pi/2$ – figure extracted from [25].

The feedback will hence play an active role in moving the phase difference between limbs to one that corresponds to a stable gait, as well as inducing a duty factor to each limb, generally slightly different, to adjust to asymmetries.

As an example of the resulting dynamics of one limb, on [25] the potential function 2.4 was used, showing two regimes, a fully oscillatory regime during swing phase (b), and an excitatory regime whenever feedback is present (c) – see figure 2.5. The resulting step phase evolved as described by the red line (a), having the slope of this potential function defined by the intrinsic frequency ω and the size of the protuberances in excitatory stage defined by the attraction coefficient σ.

$$\dot{\phi}_i = -\frac{\partial V(\phi_i)}{\partial \phi_i} \quad \rightarrow \quad V(\phi_i) = -\omega\phi_i + \sigma N_i \sin\phi_i \tag{2.4}$$

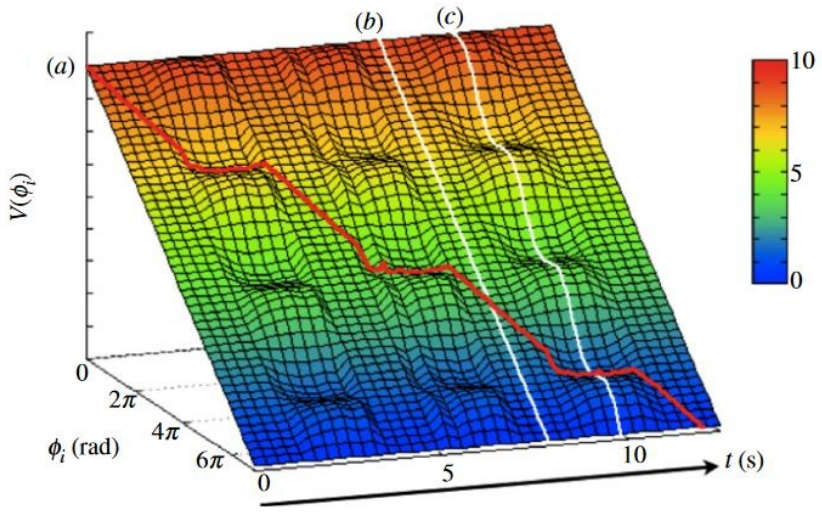


Figure 2.5: Potential function $V(\phi_i)$ obtained from the dynamics of the model defined by equation 2.3 – figure extracted from [25].

2.4 Particle Swarm Optimization (PSO)

Positioned in the category of the meta-heuristic algorithms, Particle Swarm Optimization is a method used to tackle problems of finding optimality in high dimensional and non-convex search spaces. First proposed in 1995 by James Kennedy and Russell Eberhart [32], this method is based on swarm behaviour observed in nature, where big groups of individuals (insects, birds, fish) move in a cooperative way, on a decentralized control fashion.

Following the analogy, a discrete number of particles is arbitrarily placed in the search space, having its initial position and velocity defined and the fitness corresponding to the current set of parameters computed. Based on the overall knowledge of the search space acquired after this initialization, a cyclic process begins, where the global best fitness position x^{gb} of all past iterations, as well as the personal best fitness position x_i^{pb} are saved, and the speeds and positions updated at every iteration, or time interval Δt. The position, x_i, and velocity, v_i, of particle i on iteration k are given by equations 2.5 and 2.6, respectively.

$$x_i(k) = x_i(k-1) + v_i(k-1) \cdot \Delta t \tag{2.5}$$

$$v_i(k) = wv_i(k-1) + c_1 q\left(\frac{x_i^{pb} - x_i(k)}{\Delta t}\right) + c_2 r\left(\frac{x^{gb} - x_i(k)}{\Delta t}\right) \tag{2.6}$$

The particles will then move in search space considering three phenomena:

1. Inertia, modelled by a weight w;
2. Cognitive learning of the individual, weighted by a factor c_1;
3. Social learning, weighted by a factor c_2.

The first component adds diversification to the search algorithm, allowing the finding of new regions and diversion from local minima, while the two remaining play an important role on intensification of solutions, intensifying convergence to the best parameters of one region of the high dimensional space. Random values of $r \in [0, 1]$ and $q \in [0, 1]$ are usually added at each iteration to increase the stochasticity of the algorithm, while a saturation value is added to the speeds to ensure stability of the search.

This method has successfully been used with several adaptations and applications, being a very efficient derivative free global search algorithm, with few parameters and simple implementation. However, a few drawbacks are known, such as premature convergence to sub-optimum points and difficulties in finding optima in the refined search stage.

One of the adaptations of this meta-heuristic is two-stage PSO, where the search is performed first to reach a certain boundary, and then to optimize a cost function within the predefined constraint. This method was used in 4.2.2, where a boundary of minimum speed was defined initially, following a minimization of energy consumption, within solutions presenting a valid speed.

The variables included in this section are purely explanatory of this background method, not being correlated with any other equations used in this dissertation, nor being included in the nomenclature.

2.5 *Webots* Robotics Simulator

Webots is a development environment which allows the modelling of complex robotic setups and the simulation of their dynamical interaction. The software interface, presented on figure 2.6, allows a graphical visualization of the time response of the modelled system, while updating the model itself or the control schemes.

This software allows the assembly and functioning of robots based on bodies, fixed connections, passive or active joints, sensors and communications. Bodies are defined by matrices of points extractable from CAD software and from which inertia is computed given a certain mass or density. Connections and passive joints result in physical constraints while active joints can be actuated as computed by the implemented controller. The resulting time response is then obtained the differential equations solver, *Open Dynamics Engine* (ODE), an open source library for simulating rigid body dynamics.

Having a model given by a tree of objects alike the ones above referred (figure 2.6 on the left) its interaction is ruled by cooperation between local controllers and a possible supervisor, which can be implemented using *C*, *C++*, *Java*, *python*, *Matlab* or *ROS* (figure 2.6 on the right). Having in mind the posterior hardware implementation, the control scheme implemented in this project was entirely developed in *C++*.

For a better understanding of the capabilities of this tool, as well as how to use it, read [33] and [34].

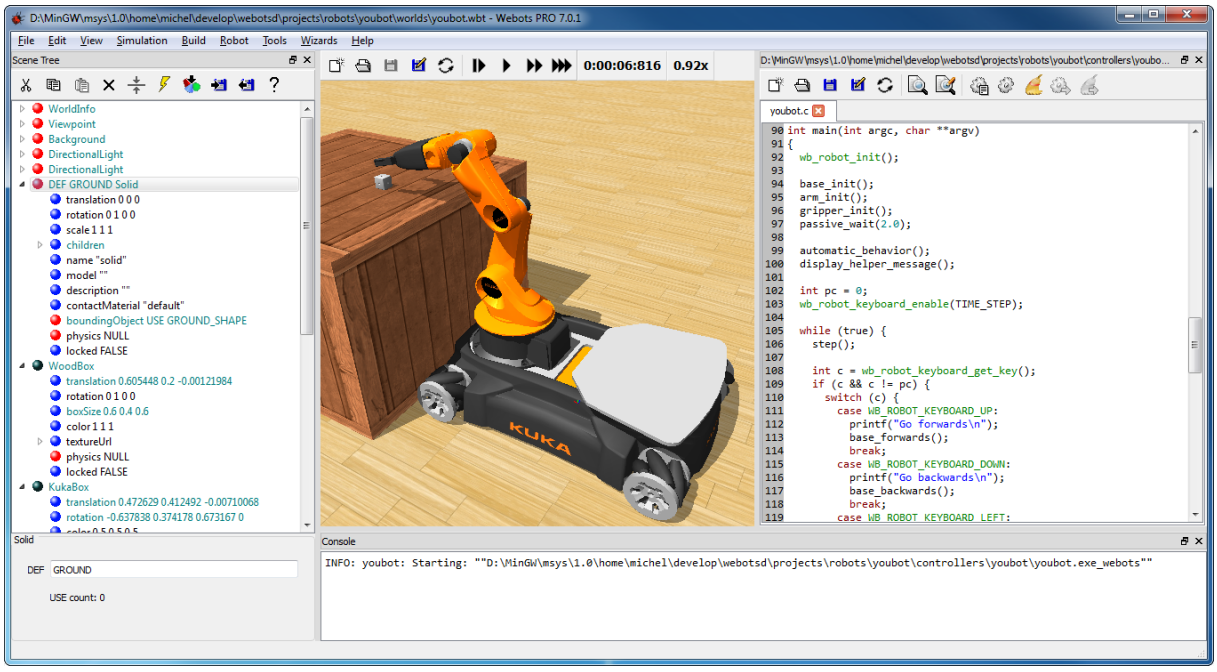


Figure 2.6: *Webots* simulator – tree of nodes on the left, physics visualization in the middle, and controllers coding on the right.

Chapter 3

Methods

This project aimed at increasing the understanding of the potential that the approaches introduced in 2.2 and 2.3 have to control modular robotic structures. With such a goal, models, procedures, control strategies and sensoring and measurement mechanisms were developed, being integrated and tested first on *Webots* simulation environment. Such iterative process represented the largest component of the project, leading to the creation of some hypothesis later validated on hardware.

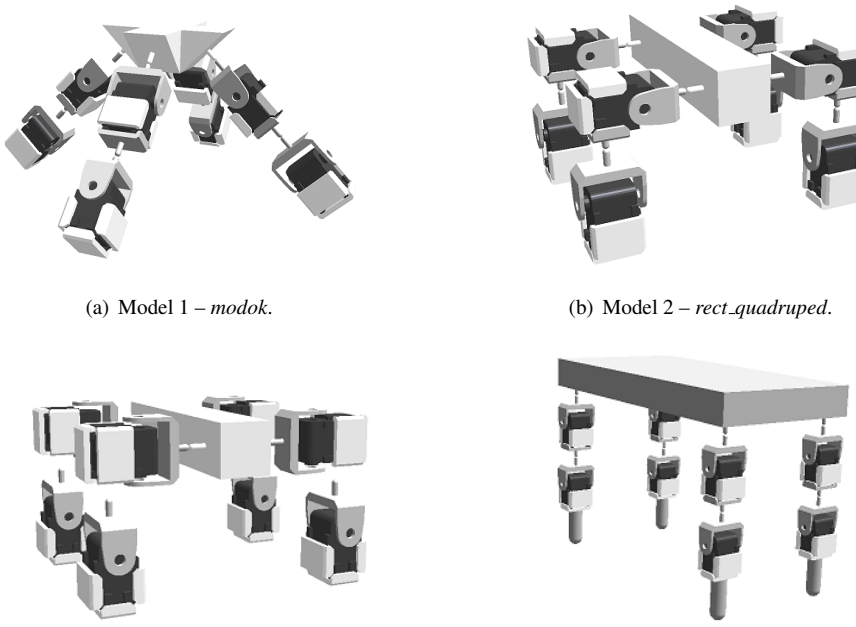
This chapter aims at showing an overview on the tools developed, referring some discarded attempts, decisions taken and methods used. It is here explained how the different pieces were put together to simulate, analyze and optimize real physics behaviour of the modular robots under study. Initially, an explanation on the models developed is given, followed by the methods used to perform simulations, hardware implementation and limitations, and simulation to reality gap. Then, kinematics and parametrization of the step movements of the base morphology presented in 3.1.2 are exhibited. In section 3.7, the implementation of the control strategies introduced in the previous chapter is explained. Finally, the fitness measurement in simulation and hardware is clarified, emphasizing their differences.

3.1 Robot Morphologies and Modelling

This thesis focused on quadrupedal modular robots, with the goal of accessing the capabilities of open loop CPG and Tegotae based approaches to control robotic structures in a universal fashion. While the first strategy allows the control of any structure recurring to optimization algorithms such as PSO, it requires extensive computation when any morphological change is considered, not representing a universal control method. On the other hand, the second strategy does not require any optimization but has only been proven to work in specific conditions, as the ones introduced in section 2.3.

In order to study the control capabilities of both methods, and given that several gaits were effectively reproduced on specialized quadruped robots with both approaches, the subset of morphologies under study was restricted to models of four 2-DoF limbs displaced in a symmetric shape. Some of these simple quadruped-like modular robots were modelled, making use of *Webots* software introduced in 2.5, and its behaviour with different types of controllers analyzed. The preliminary models are presented on figure 3.1.

3.1.1 Initial Morphologies – Iterative Process



(a) Model 1 – *modok*.

(b) Model 2 – *rect_quadruped*.

(c) Model 3 – *inverted rect_quadruped*.

(d) Model 4 – *lin_quadruped*.

Figure 3.1: Initial morphologies studied – process of achieving feasibility.

Given the goal of understanding to which extent does the morphology of the robot influence its locomotion capabilities, an iterative process of modelling different morphologies and optimizing the parameters of a neural network composed by one oscillator per degree of freedom and sinusoidal references (see figure 3.12) was performed. Such optimizations lead to the comprehension that morphologies have a great importance in determining the locomotion capabilities in terms of preferable direction of movement, as well as the influence of limb actuation in body dynamics.

These morphologies were discarded by the presence of two main problems:

1. Lack of limb workspace to perform step like trajectories in a plane perpendicular to the ground and parallel to the desired direction of locomotion;

2. Lack of mechanical robustness due to poor weight distribution, characteristic of having heavy modular limbs;

In this study, the presence of compliant elements in the modules was analyzed, as referred in 3.1.3, introducing passive dynamics present in animals. However, in order to have a thorough analysis on the performance of both control strategies, it became clear that a robust and stiff morphology should be chosen as a first science tool.

3.1.2 Result: Base Morphology

With the just stated purpose, the morphology in figure 3.2 was considered the ideal for having simple 2-DoF planar limbs, resembling hip and knee behaviour, and granting the possibility of slowly converging to more complex morphologies, analyzing robustness to mechanical changes.

Maintaining this morphology type, it is possible to study influence of body length (BL), body width (BWd), body weight(BWg), leg length ($l_1 + l_2$) and leg size ratio (l_1/l_2), in the dynamical behaviour of the robot. Additionally, this morphology clearly showed the desired feature of feasibility of reproducing stable steady state locomotion with Tegotae based control, being applied for the first time in a modular quadruped-like robot. This model was therefore the one selected to perform all the future analysis, both in simulation and hardware, having the possibility of changing either of the just stated morphology parameters.

Considering the reconfigurability of this morphology, a set of specific lengths was chosen heuristically to perform the desired experiments, being defined by the length of the passive stiff elements of each module – 25 mm. The dimensions are specified as follows:

- $l_1 = l_2 = 104.5$ mm
- BH $= l_1 + l_2 + \delta h = 235$ mm
- BWd $=$ BH $= 235$ mm
- BL $= 2l_1 \sin(\pi/3) + 2l_2 = 390$ mm
- BWg ≈ 1.4 Kg

The body height (BH) is determined by the length of the limb plus the height of the lever that attaches the limb to the body, while the body width (BWd) is set to the same value, forming a squared projection in the frontal plane. This choice allows a good level of mechanical robustness, and whose influence can be studied in the future. The body length, on the other hand, is the necessary not to have an overlap between the limb workspaces. Finally, the body weight (BWg) was derived by the weight of the modelled components and the estimated weight of the central body, including all the electronics and cabling.

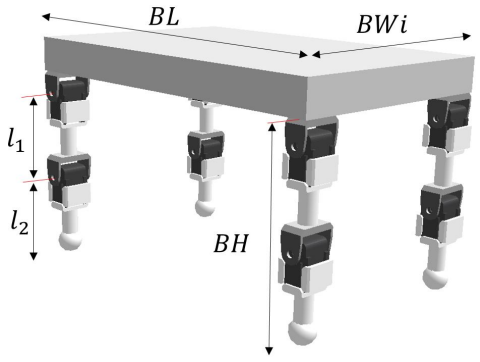


Figure 3.2: Base morphology under study – variable dimensions discriminated.

3.1.3 *Webots* Models Characteristics

Considering the positioning of the project in the modular robotics topic, the robots under analysis are mainly composed by equal modules, two per limb, as well as a rigid central body which allows an initial proximity with animal morphologies. Each of the mentioned modules is composed by the attachment of a passive component, with a defined level of compliance, to an active component, corresponding to a *Dynamixel AX-12A* motor. An example of one individual module and respective model developed on *Webots* is presented in figure 3.3.

(a) *Webots model.* (b) Hardware.

Figure 3.3: Module composed by *Dynamixel* motor, lever arm and passive element.

On *Webots*, each of the components, modules and central body, is modelled as a *ROBOT* node, including sensors, actuators, a local controller and radio communications to allow centralized control by a supervisor. The respective control scheme is further presented on section 3.2.

The central body, besides the diversity of shapes observable in figure 3.1, is always composed by a *BODY* node, GPS and IMU sensors and by a radio emitter to communicate with the supervisor. All the data collected by the central body is used solely in fitness computation, and never as feedback. The modules, on the other hand, apart from the several *BODY* nodes, include an actuated *HINGEJOINT* as motor with the moving bodies as children in the tree of nodes. Finally, several models included compliant elements, which were modelled as ball joints with predefined stiffness and damping coefficients, adding passive dynamics to the system behaviour, a phenomena briefly studied in simulation.

All the diverse bodies are merged together by connectors, which create no displacement boundary conditions to the differential equations of movement of each of the separate bodies.

3.2 Simulation Method

An essential component of this thesis work was developed using *Webots* simulation environment, where the further presented hypothesis were developed and tested, having been transported to hardware uniquely after extensive simulation tests. Together with the referred software, a few other tools were integrated in a *C++* framework, namely *Codyn* and *".xml job files"*, allowing the integration of coupled dynamics and optimizations, respectively.

In this section, the simulation framework is introduced as well as the tools named above, and the interactions between them is explained.

As presented in section 3.1, the quadruped robots under study are modelled by eight modules and one central body, corresponding to nine *ROBOT* nodes which include a set of children representing from solid bodies, to sensors, joints, connections, among others. Having this set of *ROBOT* nodes on *Webots* platform, each possesses a local controller responsible for managing sensor reading, actuation, and communication to a supervisor which coordinates all the events throughout the simulation time, according to the scheme in figure 3.4. In our case, only two types of different local controllers are needed – *middle_cube* for the central body and *robot_cluster* for the control of each of the modules.

In case of an optimization or systematic search process, a job file coordinates all the experiments, by requesting each simulation and inputing the previously specified parameters, to be optimized. For a systematic search the background program reading the job file simply requires simulations with all the combinations of parameters desired. However, for optimization procedures, an algorithm such as PSO is responsible for deciding the next set of parameters to be tested, based on the fitness collected after the simulations of the last iteration.

The *middle_cube* controller is only responsible for collecting and sending position and rotation information to the supervisor, collected by a Global Positioning System (GPS) and Inertial Measurement Unit (IMU) sensors, allowing it to compute total distance, D_t, and a stability metric, S_t, which are fed back to the job file during PSO.

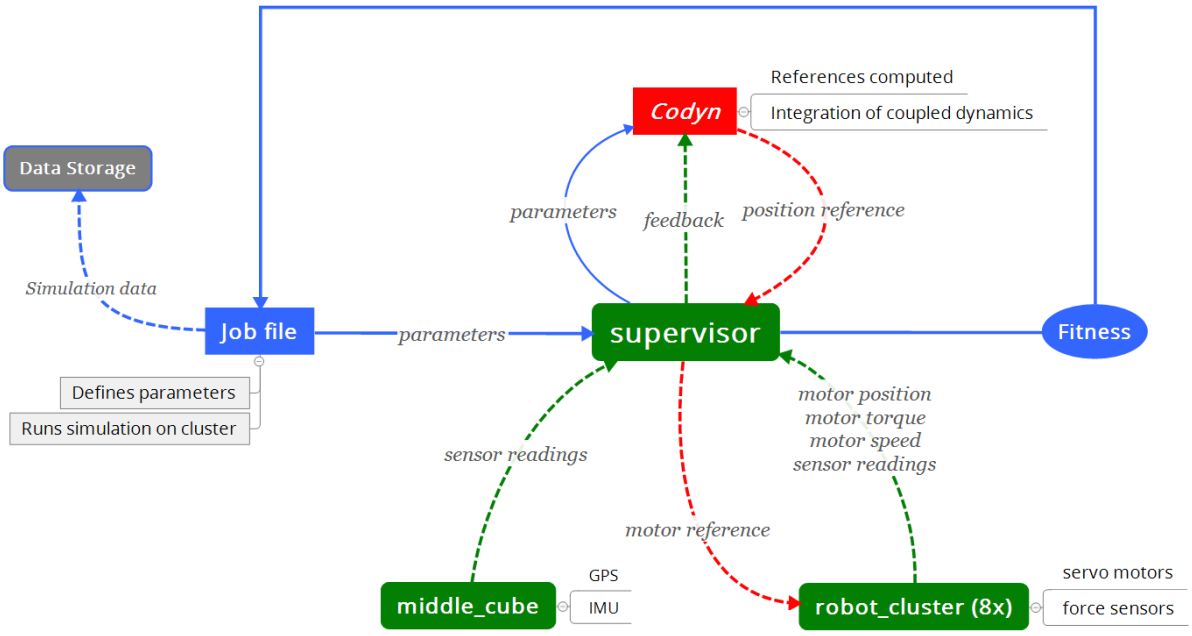


Figure 3.4: Optimization and simulation scheme.

On the other hand, the local controllers of each module, *robot_cluster*, receive from the supervisor a reference position, through a radio communication channel, and actuate the motors accordingly, modelling servo control trajectory following as perfect (no speed or torque saturation is considered in simulation – see section 3.9). Additionally, the lower modules include as children *TOUCHSENSOR* nodes gathering 3D-force information from contacts with the ground. In the end of every time step, both this information and the current state of the motors (position, speed, torque) is sent to the supervisor allowing the computation of the mechanical power consumption, P_m, and consequent energy consumption, E_t, by equation 3.17.

Besides the interaction with the local controllers, the supervisor is under control of all the simulation, determining what is done as initialization and in each time step. It sets the simulation parameters, or gets it from the job file, and computes both the actuation and the fitness variables relative to the sensor reading received from the local controllers.

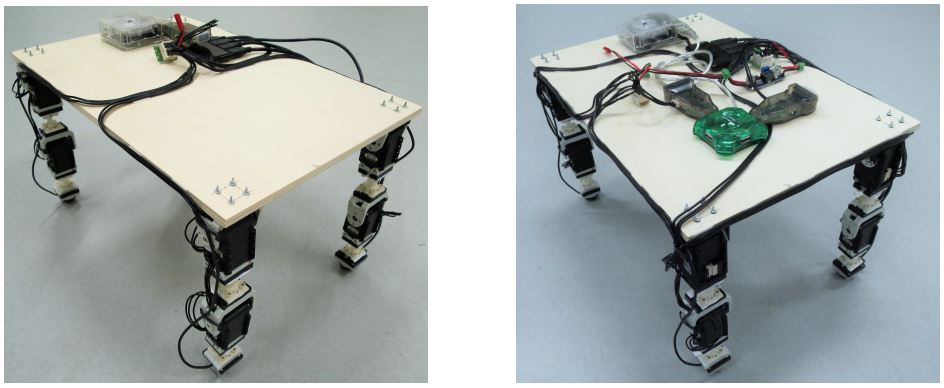
The *Codyn* library was used for integration of coupled dynamics, specially necessary in the case of open loop CPG, where the phase update rate $\dot{\phi}_i$ is dependent on the current phase, ϕ_i, but also on the state of the neighbour limbs, ϕ_j – equation 3.8. In the case of Tegotae based control, as there is no neural coupling between oscillators, this integration could be easily performed by the supervisor.

The network of phase-oscillators is defined in a *Codyn* file (*.cdn*), and its evolution is defined as a function of a set of constant and variable parameters. Later, the supervisor will define such parameters either upon initialization, or during simulation time in the second case. For the closed loop approach, the feedback is also sent to the *Codyn* network, for integration of $\dot{\phi}_i$ by equation 3.12. To sum, the *Codyn* network outputs the values of ϕ_i of the current instant, which are previously translated into reference positions in cartesian space by the supervisor, through equations 3.10 and 3.11. Having the reference position in operational space, the supervisor also handles the inverse kinematics and sends the references in joint space to the respective module. To end, the supervisor also calculates the fitness indicators based on the several data readings, and feedbacks them to the program running the job file, which stores all the data in a ".dat" file.

3.3 Hardware

Having developed a big set of simulation, optimization and search tools, an implementation in hardware was the missing component to validate hypothesis and better understand the dimension of the simulation to reality gap. This process represented therefore a continuous attempt to have a consistent reproduction of our base morphology on hardware, while maintaining the versatility to easily alter its dimensions and compliance.

Having fixed the base morphology as the one to target on this thesis, and after the extensive simulation tests, the implementation of our robot was object of the two iterations on figure 3.5. A first one, on the left, identical to the model used for simulations, using 8 *Dynamixel AX-12A* servo motors powered by 12V and controlled with TTL communication via one *USB2Dynamixel* driver. However, recalling that for study purposes the speed and torque saturation of the motors was removed from the model simulated (see section 3.9), a lack of torque necessary to perform well trajectories with $\theta_{max} > 0.2$ rad was observed. Therefore, a need for an improved solution resulted in the second version of the robot, with stronger hip motors, and whose schematic is presented in figure 3.7.



(a) Base morphology robot – version 1.

(b) Base morphology robot – version 2.

Figure 3.5: Intermediate versions of base morphology robot developed on hardware.

3.3.1 Final Robotic Framework

The final version of the robot is then composed by a set of 4 *Dynamixel RX-28* motors as hip and 4 *AX-12A* for the knee joints, running at different power levels, 18V and 12V, and with different communication protocols, *RS-485* and *TTL* respectively. This is allowed by having a second *USB2Dynamixel* driver and fully independent communications from the on-board computer, *Odroid XU4*. Although this solution adds the characteristic of having two types of modules, it was adopted by shortage of available *RX-28* motors for all the joints, and the results are not expected to differ significantly from fully upgraded limbs.

In addition, it is important to note that the passive components of each module are built in a way to allow an easy removal and replacement by elements with diverse dimensions and compliance levels, giving the robot an increased versatility to future studies, and that all the components were carefully placed to position the center of mass in the axis of symmetry.

Finally, with the goal of increasing the information collected in every experiment, and consequently the understanding of the body dynamics and energetic cost, a current sensor controlled by an *Arduino nano* board, and an Inertial Measurement Unit (IMU) sensor were added. The final robot is presented in figure 3.6.

In figure 3.7, a detailed view on the top of the robot, as well as on one of the limbs, is exhibited with all the including components numbered according to the table 3.1. An overview on the components with respective figures is presented in appendix B.

The on-board computer *Odroid XU4* receives all the force information from *Optoforce* sensors via serial communication, integrating the limb phases based on this feedback (example of closed loop) and translating the current limb phase into position references in cartesian space, converting posteriorly into joint references through inverse kinematics. These references are then sent to the two *USB2Dynamixel* drivers, which send the information to the respective motors via *TTL* or *RS-845* protocols. The actual position of each motor is measured by the integrated encoders, and sent back to the *Odroid*. All the data of force, phase and angles is stored in the local memory of the *Odroid* and accessible via Wi-Fi.

The *Odroid* is connected to a computer via Wi-Fi, allowing a distant initialization of the robot, as well as real time change of the parameters that define a gait through a GUI (Graphical User Interface).

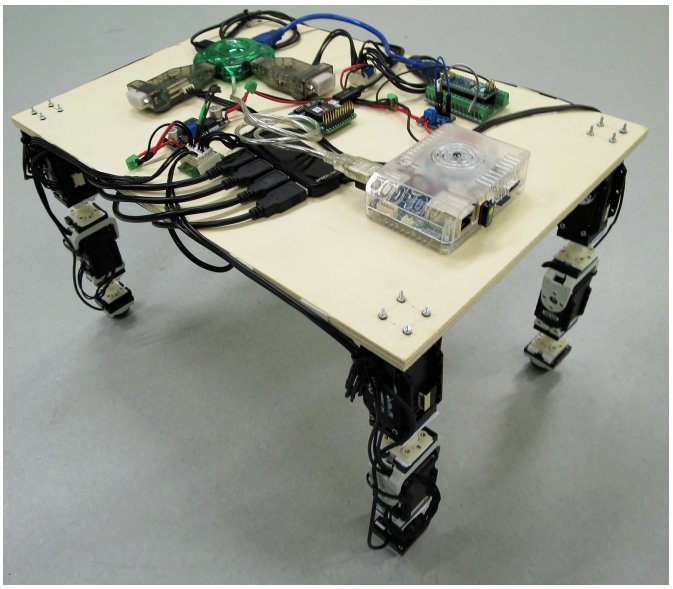
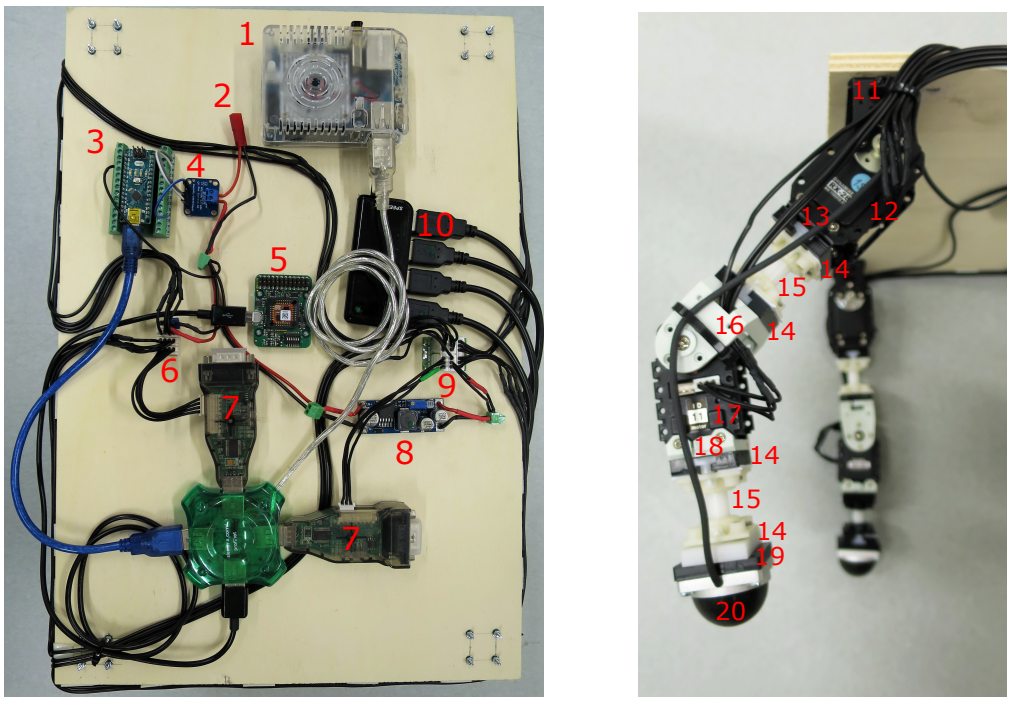


Figure 3.6: Base morphology robot – Final version.



(a) Robot top view.

(b) Limb.

Figure 3.7: Base morphology robot, final version – components identified.

Table 3.1: Robot components discrimination – detailed in appendix B

Nr.	Component	Figure
1	Odroid XU4 computer	B.7(a)
2	Power supply (18V)	–
3	Arduino nano board	B.7(c)
4	Current sensor	B.6(b)
5	Inertial Measurement Unit (IMU) sensor	–
6	4-pin cabling connection (RS-485)	–
7	USB2Dynamixel device (2x)	B.7(b)
8	Voltage regulator (18V-12V)	B.5
9	3-pin cabling connection (TTL)	–
10	Serial communication to Optoforce sensors	–
11	Bioloid FR07-H101K servo bracket (RX series)	B.2(a)
12	Dynamixel RX-28 servo motor	B.4(b)
13	Dynamixel FR07-S101K back frame (RX series)	B.2(b)
14	Universal connection – Bioloid to T-shape	–
15	Stiff element – $25mm$ height	B.3
16	Bioloid FP04-F2 servo bracket (AX series)	B.2(c)
17	Dynamixel AX-12A servo motor	B.4(a)
18	Bioloid FP04-F3 back frame (AX series)	B.2(d)
19	Optoforce to Bioloid connector	–
20	3D force *Optoforce* sensor	B.6(a)

3.3.2 Hardware Limitations

Once implemented the hardware representation of our base morphology, a process of characterization took place, being detailed in appendix **??**. In this procedure, the following hardware characteristics were identified:

- Resolution of ≈ 0.006 rad in the position measurements;
- Presence of mechanical backlash in the hip motors of $\approx \pm 0.015$ rad;
- Presence of reading noise in knee motor with amplitude of ≈ 0.02 rad;
- Settling time at step input of $\pi/3$:

  ≈ 0.25 seconds on *RX-28* hip motors, limited by speed saturation;

  ≈ 0.35 seconds on *AX-12* knee motors, limited by speed saturation;
- Percentage of overshoot:

  1.2 % on *RX-28* hip motors;

  2.2 % on *AX-12* knee motors;
- Motor non-linear compliant behaviour:

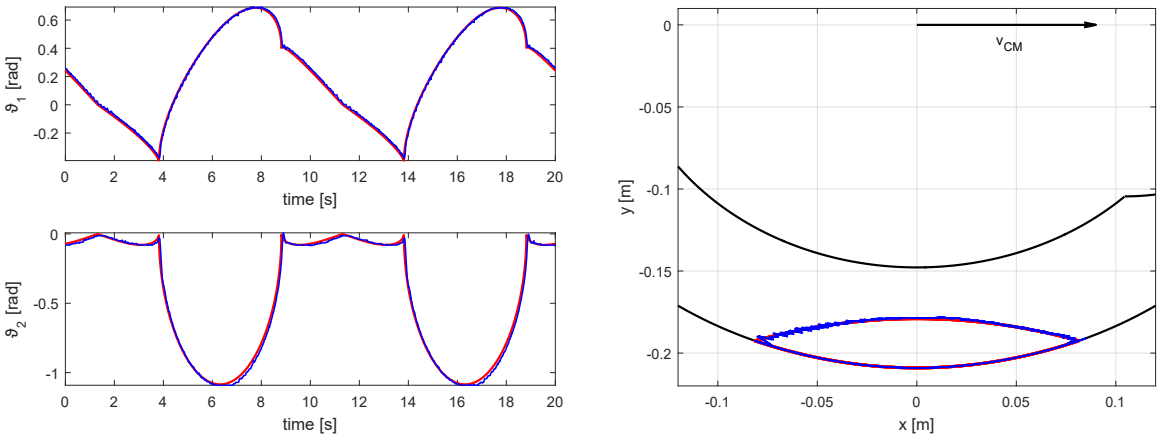
  $\approx 0.04\ [rad \cdot N^{-1} \cdot m^{-1}]$ on *RX-28* hip motors until $\approx 1N \cdot m$;

  $\approx 0.055\ [rad \cdot N^{-1} \cdot m^{-1}]$ on *AX-12* knee motors until $\approx 0.7N \cdot m$;
- Following error not highly influenced by trajectory shape;
- Following error increased by frequency due to speed saturation and delay on knee actuation.

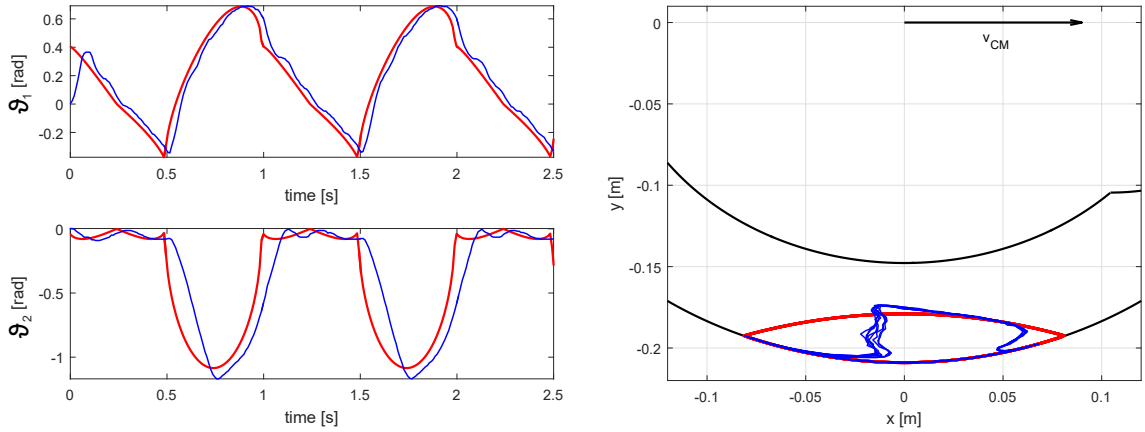
Finding the last point as the biggest limitation, an explanation for such problem follows. As visible in figure 3.8, at low frequencies, the local servo controllers manage to keep the tracking error almost null, presenting uniquely some vibration in the measurements due to the phenomena already mentioned of mechanical backlash and reading noise. Although, when the frequency is increased, speed saturation starts playing an important role, making the system non-linear, and leading to relevant trajectory following errors. Such effect appears first in the *Dynamixel AX-12A* motors, relative to the knee movements, creating a delay in the foot lifting. Hence, the trajectory following resulting from direct kinematics is greatly harmed.

As the motors are different and include diverse speed saturations, this effect does not affect both joints simultaneously, inducing a direct and more severe impact on the result of inverse kinematics. This effect should then be reduced in the presence of knee motors similar to the hip ones – *RX-28*.

Regarding the validation experiments, these were not fully compromised by this issue, being that the focus of this project was on walking behaviour and hence on not too high frequencies. The experiments performed, were developed to be inside the region where the trajectory followed reliably matches the ones simulated on *Webots* environment.



(a) 0.1 Hz.



(b) 1 Hz.

Figure 3.8: Negative effect of frequency on trajectory following due to speed saturation on the knee motors.

3.4 Simulation–Reality gap

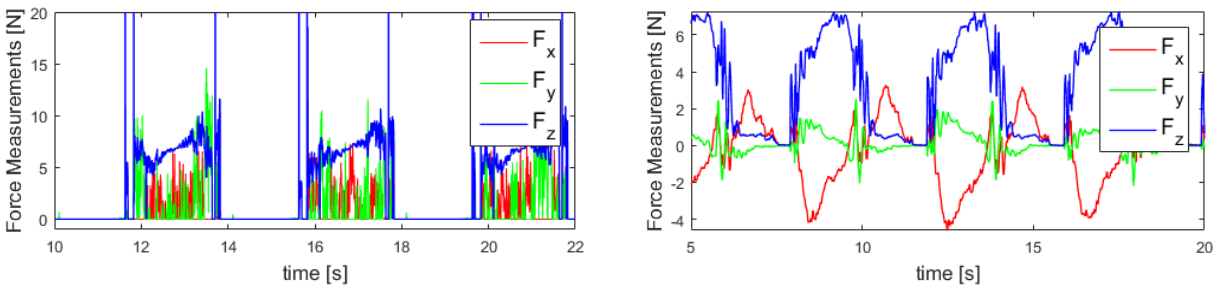
Despite of all the attempts to perfectly model and simulate the dynamical behaviour of robotic modular structures, a gap between what is described by the differential equations generated and solved by *Webots* and the actual response of the robot is always present. A collection of issues increases the dimension of such a gap, not allowing a fully accurate prediction of the time response of the considered group of systems.

At first, a set of simplifications are added to the models characteristics, such as considering homogeneous mass distribution in each motor case, neglecting cabling and electronic components positioning and mass distribution, and considering perfect actuation and feedback readings from the sensors. Then, all the mechanical issues such as backlash, motor compliance or speed saturation create non-linearities that are not accounted for.

In addition, problems regarding the contact friction between the robots feet and the ground were visualized. These are due to the restriction of not being able to model soft materials. The actual *Optoforce* sensors assembled in the edge of each foot, include a semi-spherical contact surface made of rubber, which slightly molds to the ground while in contact, creating an area of interception with the floor that highly increases static friction. However, these were modelled as stiff spherical surfaces with rubber-like friction specifications, leading to a unique contact point between each foot and the ground, and consequently slippery behaviour was observed, as well as noisy sensory readings. This problem leads to a big difference between the force feedback on simulation and hardware, presented in figure 3.9, where it is visible that the sensor readings on simulation present actually more peaks and noise that the real sensor.

Finally in terms of energy consumption measurements, introduced in section 3.8.2, on simulation this value is computed by integration of mechanical power (eq. 3.17), while for hardware it is computed by inserting the measured values of current in equation 3.19. The result of both approaches is quite different, given that in the second one, base current is accounted for and integrated even if the robot is stopped. Solutions for this problem are discussed in section 5.1.5.

Considering all the mentioned differences, the need for validating hypothesis on hardware is highly enforced.



(a) Simulation.

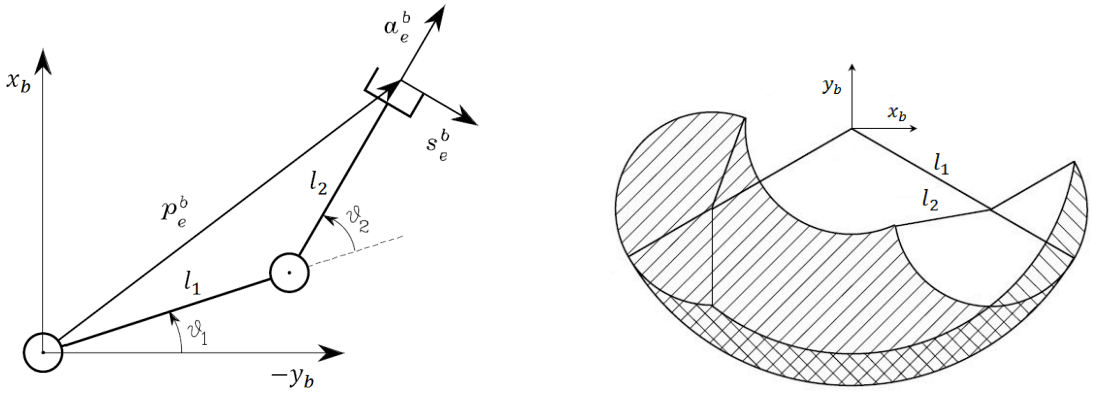
(b) Hardware.

Figure 3.9: Examples of 3-dimensional force feedback at steady state locomotion in simulation and hardware.

3.5 Limb Kinematics

Referring to the base morphology presented in 3.1.2, the direct and inverse kinematics of each limb are the ones of a 2-link planar arm, with a transformation of coordinates so that the x-y plane is the sagital plane and y is normal to the ground in initial position.

Considering the notation detailed in figure 3.10 (a), the obtained kinematic equations of each limb are the ones below presented, where both solutions for knee bending are referred in equation 3.3. In figure 3.10 (b), the workspace of the limb is presented, with regions relative to positive or negative knee bending differentiated. Both images in figure 3.10 can be found in [35], together with more detailed explanations.



(a) Limb configuration.

(b) Workspace.

Figure 3.10: 2-Link planar limb configuration and workspace.

With the mentioned transformation of coordinates, relative to the 2-link planar arm presented in [35], the kinematics of the limb can then be described in the following manner, where $s_i = \sin\vartheta_i$, $c_i = \cos\vartheta_i$, $s_{ij} = \sin(\vartheta_i + \vartheta_j)$, $c_{ij} = \cos(\vartheta_i + \vartheta_j)$.

- Forward Kinematics:

$$\mathbf{p_e^b} = \begin{bmatrix} l_1 \cdot s_1 + l_2 \cdot s_{12} \\ -l_1 \cdot c_1 - l_2 \cdot c_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} l_1 s_1 + l_2(s_1 c_2 + c_1 s_2) \\ -l_1 c_1 - l_2(c_1 c_2 - s_1 s_2) \\ 0 \end{bmatrix} \tag{3.1}$$

- Inverse Kinematics:

$$\mathbf{p_{e}^{b}}_x{}^2 + \mathbf{p_{e}^{b}}_y{}^2 = {l_1}^2 + {l_2}^2 + 2l_1 l_2(c_1 c_{12} + s_1 s_{12}) = {l_1}^2 + {l_2}^2 + 2l_1 l_2 c_2 \tag{3.2}$$

$$c_2 = \frac{\mathbf{p_{e}^{b}}_x{}^2 + \mathbf{p_{e}^{b}}_y{}^2 - {l_1}^2 - {l_2}^2}{2l_1 l_2}, \qquad s_2 = \sqrt{1 - {c_2}^2} \ \vee \ s_2 = -\sqrt{1 - {c_2}^2} \tag{3.3}$$

$$c_1 = \frac{(l_1 + l_2 c_2)\mathbf{p_{e}^{b}}_y - l_2 s_2 \mathbf{p_{e}^{b}}_x}{\mathbf{p_{e}^{b}}_x{}^2 + \mathbf{p_{e}^{b}}_y{}^2}, \qquad s_1 = -\frac{(l_1 + l_2 c_2)\mathbf{p_{e}^{b}}_x + l_2 s_2 \mathbf{p_{e}^{b}}_y}{\mathbf{p_{e}^{b}}_x{}^2 + \mathbf{p_{e}^{b}}_y{}^2} \tag{3.4}$$

$$\begin{bmatrix} \vartheta_1 \\ \vartheta_2 \end{bmatrix} = \begin{bmatrix} \arctan(s1/c1) \\ \arctan(s2/c2) \end{bmatrix} \tag{3.5}$$

3.6 Trajectory Parametrization

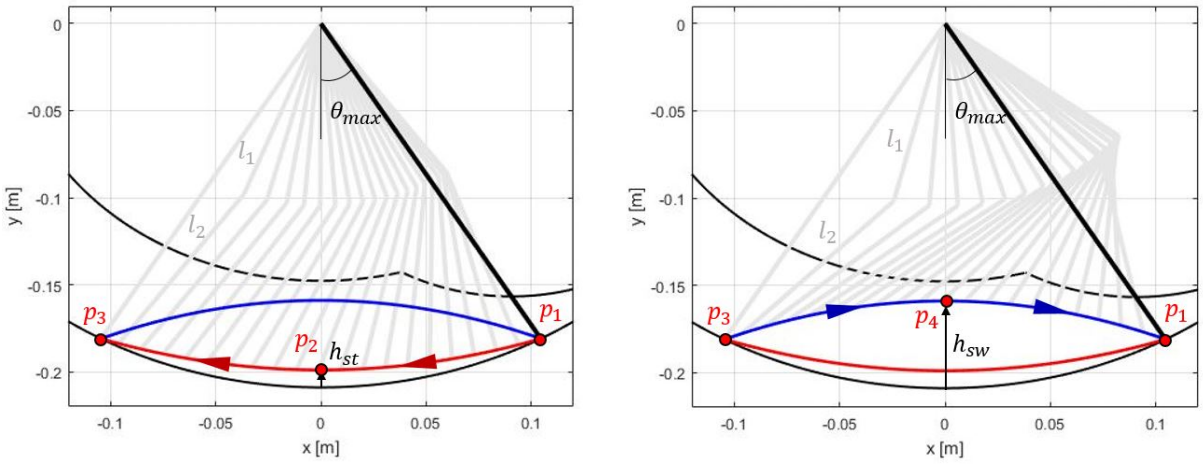
With a view to find optimal trajectories to be followed by the individual limbs, an iterative process of parametrization of trajectories and optimization using PSO (see section 2.4) was conducted. The referred iterations discarded the two tentative methods exhibited in appendix A, leading to the one used in the further work, described below. The tentative methods are included in this thesis as justification and comparison to the final methodology, which is presented as a suggestion for future work.

With the goal of overcoming the complications described in appendix A, the final parametrization method proposed (figure 3.11), and the one further used, includes a few simplifications. It considers that the touchdown and liftoff points, p_1 and p_3, are positioned in the lower boundary of the workspace, imposing a stretched knee in both cases. In addiction, a symmetry of trajectory is forced, having only one parameter regulating the size of the reference trajectory in the x axis – hip angle at p_1, θ_{max}.

Both stance and swing phases are then characterized by a height relative to the workspace boundary – h_{st} and h_{sw} – whose difference defines the ground clearance of the footstep (see figure 3.11). Having the points p_1 to p_4 defined by equation 3.6, stance and swing trajectories are finally designated by the quadratic function in 3.7, where h refers to h_{st} or h_{sw}, depending on the step phase.

$$\begin{cases} \mathbf{p_1} = (l_1 + l_2) \cdot [\sin\theta_{max},\ -\cos\theta_{max}] \\ \mathbf{p_2} = [0,\ -(l_1 + l_2) + h_{st}] \\ \mathbf{p_3} = (l_1 + l_2) \cdot [-\sin\theta_{max},\ -\cos\theta_{max}] \\ \mathbf{p_4} = [0,\ -(l_1 + l_2) + h_{sw}] \end{cases} \tag{3.6}$$

$$G(\lambda) = \frac{(l_1 + l_2)(1 - \cos\theta_{max}) - h}{(l_1 + l_2)^2 \sin^2\theta_{max}} \cdot \lambda^2 - (l_1 + l_2) + h \tag{3.7}$$



(a) Stance phase.

(b) Swing phase.

Figure 3.11: Parametrization of step trajectory in cartesian space.

3.7 Control Strategies Implementation

Having the trajectory shape defined as shown in figure 3.11, the speed at which the trajectory is followed at every time sample is given by the applied control rule. For both approaches under comparison in this thesis, Open Loop CPG and Tegotae-based Physical Interlimb Communication, the phase-oscillators network was composed of one oscillator per limb, as introduced in section 3.7.1.

Defining ϕ_i as the variable indicating the phase of limb i in its step trajectory, as represented in figure 3.14, this section is dedicated to explaining how both strategies influence the evolution in time of this variable, as well as its direct impact on the trajectory following.

3.7.1 From Eight to Four Phase-Oscillators Network

Robots with a big number of degrees of freedom and complex morphologies are very hard to effectively control, given the high dimensionality of the reference positions in joint space, and the coupled dynamics of their actuations. As introduced in section 2.2, one way to decrease the extensive search space is to impose sinusoidal outputs, defining the actuation by a set of two parameters per degree of freedom – amplitude A_i and offset X_i – and one parameter per neural coupling – phase shift ψ_{ij} – completing $3n - 1$ parameters. With such parametrization it is therefore possible to optimize the locomotion method of an arbitrary structure.

This was the first method used to control the morphologies presented in figure 3.1, with the phase-oscillator network of figure 3.12. By optimizing the mentioned $3n-1$ parameters, a visualization of preferable directions and methods of locomotion was possible, exposing some of the problems discussed in **??**, and leading to the choices made in terms of morphology.

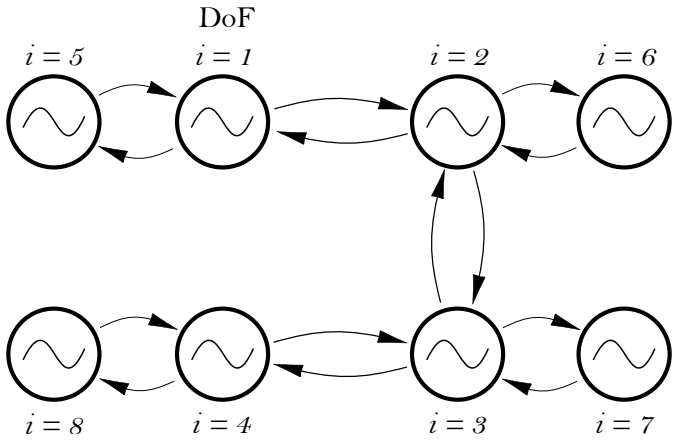
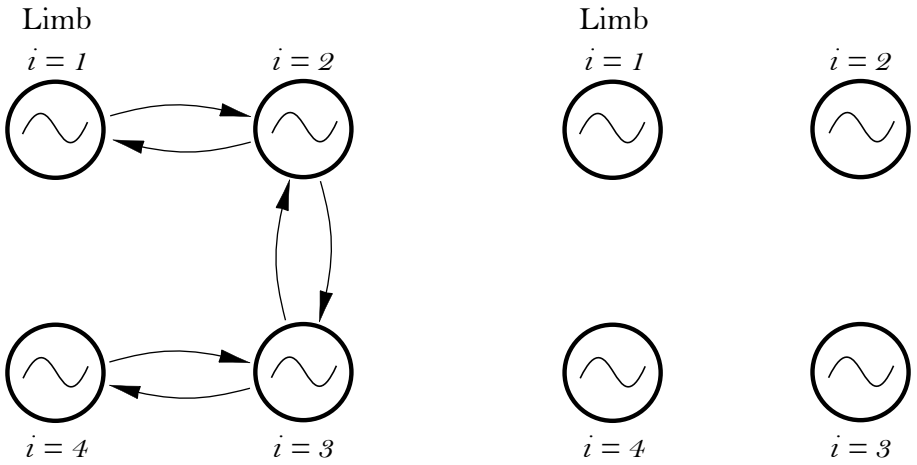


Figure 3.12: Initial neural network – one phase-oscillator per degree of freedom.

Regarding that the referred method includes for our network 23 parameters, requiring extensive optimization procedures for every small morphological change, it represents a highly non-general approach.

With the purpose of coming up with a more general approach, optimizations were performed during the parametrization process described in 3.6, discarding for example the attempt to parametrize in joint space presented in **??**, and leading to the final parametrization method. Finally, the usage of one phase-oscillator per limb, describing the step phase defined in operation space was used, leading to the networks presented in figure 3.13. On

the left, the network with neural communication used for optimized Open Loop CPG is presented, as well as the one without coupling, where the phase shifts are adjusted through a Tegotae feedback rule.



(a) Open Loop CPG – network of coupled phase-oscillators.

(b) Tegotae-based feedback – no neural coupling.

Figure 3.13: Final neural networks – one phase-oscillator per limb.

Having one oscillator per limb, the limb phase ϕ_i is directly translated into a position in cartesian space, as shown in figure 3.14, in a way that $\phi_i = [0, \pi/2, \pi, 2\pi/3]$ corresponds to p_1, p_2, p_3, and p_4 respectively, leaving the computation of joint references to the equations 3.10 and 3.11, and to the inverse kinematics.

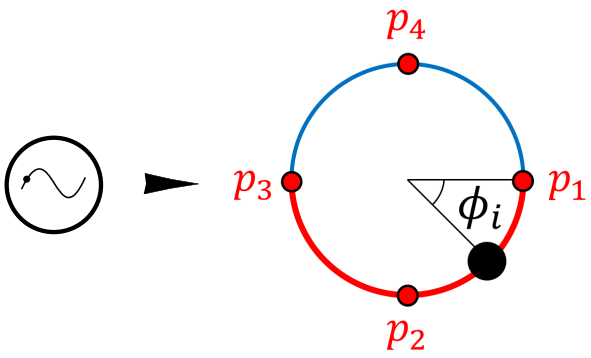


Figure 3.14: Limb phase-oscillation directly converted into points on cartesian space.

3.7.2 Strategy 1: Open Loop CPG – Network of Couple Phase-Oscillators

As introduced in chapter 2, the neural coupling term considered for our CPG network is given by the second term of equation 3.8. Given that we have a symmetrical and simple CPG network, the introduction of different weights w_{ij} for correlating body and limbs oscillations, as performed in [10], is not required. Hence, discriminating distinct values of w_{ij} would only result in an asymmetrical convergence of the limbs towards steady state. With such motivations, w_{ij} was fixed to a constant value of $w = 1$. The chosen value affects solely the speed of convergence to steady state, and unity allows to achieve fast, however stable convergence.

$$\dot{\phi}_i = 2\pi f + \sum_j w_{ij} \sin(\phi_j - \phi_i - \psi_{ij}) \tag{3.8}$$

The remaining parameter, $\psi_{ij} = -\psi ji$, refers to the desired steady state phase difference between limbs i and j, and is usually obtained through an optimization procedure. In an early stage of the project, optimizations were performed with these parameters included on the search (section 4.2), having been fixed afterwards as the ones of a trot gait, reducing the parameter space for the systematic search thoroughly performed (section 4.3).

Intrinsic duty factor – *df*

Assuming the control method just described, at steady state the rate of phase change $\dot{\phi}_i$ becomes constant, corresponding to a step trajectory followed always at the same pace. However, as described in section 2.1, animals present two stages in every step cycle, stance and swing, which take a different amount of time to be completed. In order to allow our robot to use this characteristic advantageously, a parameter of intrinsic duty factor, *df* is introduced.

Having defined duty factor of a locomotion pattern as the percentage of stance phase in a step cycle, and describing now the transition phase from stance to swing as ϕ_t, the relation between trajectory following speed and the evolution of ϕ in time is corrected to intrinsically incorporate the effect of $df \neq 0.5$. This correction is visible in figure 3.15 and imposes that stance and swing phases correspond to $\phi \in [0, \phi_t]$ and $\phi \in [\phi_t, 2\pi]$, respectively. It is possible then to define duty factor as:

$$df = \frac{\phi_t}{2\pi} \tag{3.9}$$

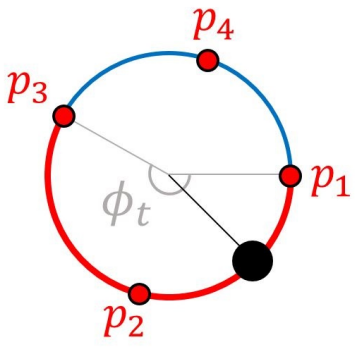


Figure 3.15: Reshape of phase-position transformation due to effect of intrinsic duty factor *df*.

The referred reshaping means that a constant phase derivative $\dot{\phi}_i$, does not imply constant rate of trajectory following if $df \neq 0.5$. In fact, the effect of duty factor changes in the joint space references for a constantly increasing ϕ are visible in figure 3.16, where the following of a trajectory with $h_{st} = 0$ and h_{sw} matching the height of the workspace in $x = 0$ is performed.

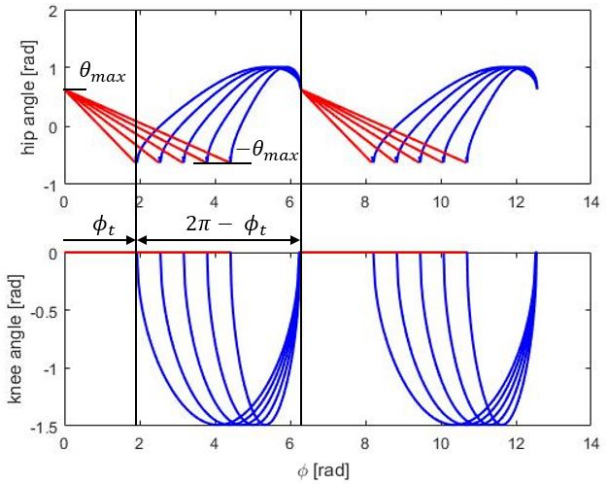


Figure 3.16: Trajectory following in joint space as a function of df.

The usage of this intrinsic reshaping of the relationship between ϕ and the actual trajectory following was used only in open-loop approach. In closed loop, $\dot{\phi}$ is not constant even in steady state, due to the direct consequence of feedback, having the duty factor emerge through such process.

Reference position as a function of ϕ and df

Once mentioned the outcome of introducing the duty factor intrinsically in the evolution of ϕ_i, the implementation of such result is ought to be presented. Considering the relation given by equation 3.9, it is possible to derive the equations for the evolution with ϕ of the reference position in operational space $\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}$, as a function of df. The result is presented in equation 3.10.

$$\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}{}_x(\phi) = \begin{cases} p_{1_x} - p_{1_x} \cdot \frac{\phi}{\pi \cdot df} & \text{if } \phi \in [0, \phi_t] \\ p_{3_x} + \frac{p_{1_x} \cdot (\phi - 2\pi df)}{\pi(1-df)} & \text{if } \phi \in [\phi_t, 2\pi] \end{cases} \tag{3.10}$$

Having performed the rescaling of figure 3.15 in $\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}{}_x$, the y component of the reference is simply given by reformulating equation 3.7:

$$\mathbf{p}_{\mathbf{e}}^{\mathbf{b}}{}_y(\phi) = \frac{(l_1 + l_2)(1 - \cos\theta_{max}) - h}{(l_1 + l_2)^2 \sin^2\theta_{max}} \cdot \mathbf{p}_{\mathbf{e}}^{\mathbf{b}}{}_x(\phi)^2 - (l_1 + l_2) + h\,, \tag{3.11}$$

$$h = \begin{cases} h_{st} & \text{if } \phi \in [0, \phi_t] \\ h_{sw} & \text{if } \phi \in [\phi_t, 2\pi] \end{cases}$$

3.7.3 Strategy 2: Tegotae-based Physical Interlimb Communication

Regarding mechanical coupling through a Tegotae rule, as indicated in section 2.3, the equation responsible for introducing force feedback in the rate of phase change is of the type of equation 2.3. However, since the used convention for ϕ in this thesis is the one in figure 3.14, the Tegotae rule was applied as in equation 3.12, where the attracted point is p_2, corresponding to a phase of $\pi/2$, and N_i corresponds to the component $\mathbf{a}_{\mathbf{e}}^{\mathbf{b}}$ (see figure 3.10 (a)) of the ground reaction force, measured in the limb's end-effector.

$$\dot{\phi}_i = 2\pi f + \sigma\ N_i\ \cos(\phi_i) \tag{3.12}$$

The referred attraction to $\phi_i = \pi/2$ is described in figure 3.17, where steady state is considered, existing no feedback forces during swing phase ($N_i = 0$). In this schematic it is easily visible that there will be an acceleration in $\phi_i \in [0, \pi/2]$ and a deceleration in $\phi_i \in [\pi/2, \pi]$ induced by the feedback during stance, $N_i \neq 0$.

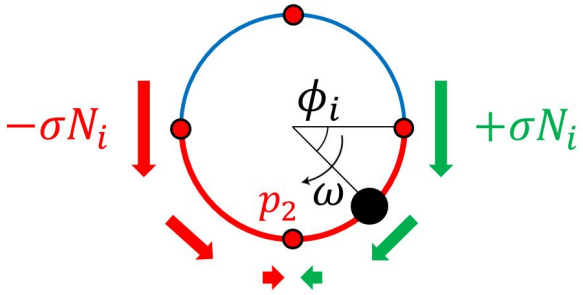


Figure 3.17: Effect of force feedback in dynamics of limb oscillation in steady state.

Therefore, $\dot{\phi}$ is not constant in steady state, having the duty factor emerging from the feedback itself and from the attraction coefficient σ, as just observed. Hence, no intrinsic duty factor is imposed and the relation between ϕ and the position in the step cycle is always the one in figure 3.14.

Changes in the attraction parameter σ, result in a reshaping of the steady state evolution of ϕ_i and consequently of the references in joint space. This time response can be observed in the feasibility analysis presented in 5.2.1.

Feedback of normal force – N_i

Regarding the possibility of using more force information, besides the normal one, tests were performed in simulation and hardware to analyze its potential, having the results presented in figures 3.18 and 3.19. On the left, 3-dimensional force information on the local reference frame of the lower limb is presented for both figures, while on the right, normal force data is compared with the average of the three components. In both cases, it is visible that considering the mean of components only corresponds to a scaling of the effect of feedback, not adding relevant information. Hence, only normal force is incorporated in the feedback rule.

It is also noticed that simulation data regarding ground collisions is much more noisy than the real sensor measurements, indicating problems of the collision simulation, discussed further in section 3.4.

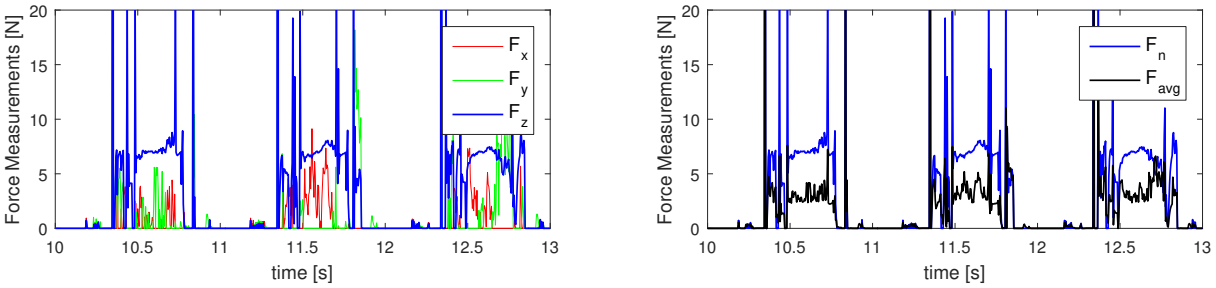


Figure 3.18: **Simulation:** Force measurements from simulation, of steady state open loop trot gait with $df = 0.5$, performed by base morphology at 1 Hz – On the left, the 3-dimensional components of force in local reference frame are presented, where F_z is the component in the axis of the lower limb. On the right, normal force $F_n = F_z$ is compared with average of force components, F_{avg}.

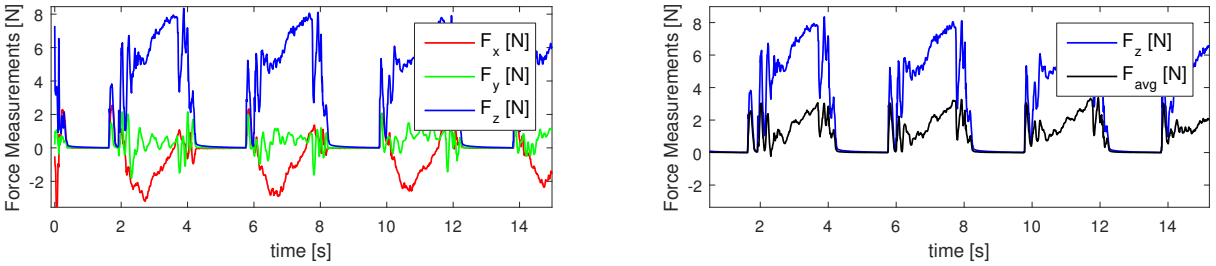


Figure 3.19: **Hardware:** Force measurements from *Optoforce* sensors of steady state trot gait performed at 0.25 Hz – On the left, the 3-dimensional components are presented. On the right, the normal component F_z is compared with the average of the components F_{avg}.

Binary Feedback

One of the goals of this thesis was to investigate on the possibility of using only binary feedback of contact force with the ground. This was implemented according to equation 3.13, where a threshold of 0.2N was considered in order to avoid incorrect classification caused by noise.

In order to compare results with the non-binary information, a scaling of σ has to be performed according to an average ground reaction force, denominating this scaled value by σ_b. Considering the reaction force profile, N_i, represented in figure 3.19, a relation of $\sigma_b = 6\,\sigma$ was considered.

$$\dot{\phi}_i = 2\pi f + \sigma_b\, B_i\, \cos(\phi_i)\,, \qquad B_i = \begin{cases} 1 & \text{if } N_i > 0.2\ \text{N} \\ 0 & \text{if } N_i < 0.2\ \text{N} \end{cases} \tag{3.13}$$

3.8 System Response Metrics

In this section the variables created throughout the project to be extracted from each simulation as indicators of gait performance are presented. These allowed the gathering of valuable information from a huge amount of executed simulations, without extensive visualization of their time response. A small comparison with hardware is present in 3.8.1 and 3.8.2.

For coherence in notation, the indicators are designed for a robot with n limbs of m degrees of freedom, denoting $(*_{j,i})$ as a characteristic of degree of freedom j, part of limb i. Additionally, discrete time is considered with time step T, in a way that simulation starts at $kT = 0$, measurements start at $t = t_m = k_m T$, and finally runtime finishes at $k = k_e$.

3.8.1 Distance – Average Speed

Giving use to a GPS sensor positioned in the center of the robot's central body, the distance traveled by the robot during the measurement time can be computed as follows, corresponding to an average traveling speed, as denoted in equation 3.15.

$$D_t = \|\overrightarrow{x_{CM}}(k_e T) - \overrightarrow{x_{CM}}(k_m T)\| \tag{3.14}$$

$$\overline{v_{CM}} = \frac{D_t}{k_e T - k_m T} \tag{3.15}$$

On hardware, this value was computed with use of time measurements, given a previously fixed traveled distance.

3.8.2 Gait Energy Efficiency

Having a measurement on how fast could a certain gait propel the robot in steady state, knowledge on the cost of the specified type of locomotion was necessary. With such purpose, an indicator of gait energy efficiency is introduced.

Considering mechanical power consumption, P_m, of a robot with n limbs and m degrees of freedom per limb as exhibited in equation 3.16, where $\dot{\vartheta}(kT)$ and $\tau(kT)$ refer to joint speed and torque in instant kT, it is possible to define all the energy consumed throughout the measured steady state response as E_t (eq. 3.17).

$$P_m(kT) = \sum_{i=1}^{n} \sum_{j=0}^{m} |\dot{\vartheta}_{j,i}(kT) \cdot \tau_{j,i}(kT)| \tag{3.16}$$

$$E_t = \sum_{k=k_m}^{k_e} P_m(kT) \tag{3.17}$$

Finally, the gait energy efficiency, ϵ_t, is defined as the inverse of the cost of transport (CoT), indicating how much distance can be traveled using the specified gait with one unity of energy.

$$\epsilon_t = CoT^{-1} = D_t / E_t \tag{3.18}$$

In the case of hardware, ϵ_t kept its definition, however the energy consumption was measured in a different way. As presented in section 3.3, a current sensor was attached to the power source and its readings saved. As the input voltage, V_{in}, is predefined in the power source, the total energy consumption only dependent of the input current, I_{in}, and can be computed as:

$$E_t = \sum_{k=k_m}^{k_e} V_{in} \cdot I_{in}(kT) \tag{3.19}$$

As referred in section 3.4, the different methods of computing energy consumption in simulation and hardware resulted in very different measures due to integration of base energy consumption on hardware, which is not considered in simulation. This issue becomes more relevant due to integration in time of the referred base cost.

3.8.3 Gait Stability

After an initial optimization process on distance and energy efficiency, it was clear that some solutions did not represent a normal way of locomotion, having jumps or flipping. To be able to filter such cases, measurements on the body dynamics were necessary and for that an Inertial Measurement Unit (IMU) positioned in the central body was of use. By collecting roll, pitch and yaw (Φ, Θ, Ψ) information throughout time, it was possible to analyze both the stability of the transient and steady state responses, as well as the periodicity of the second case.

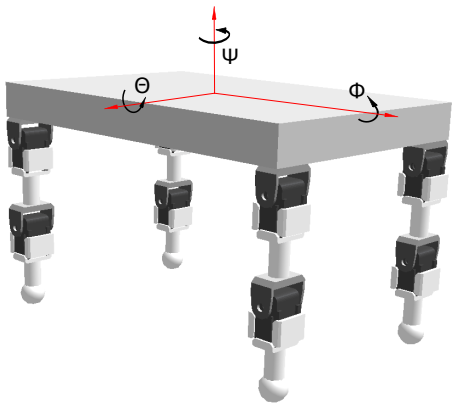


Figure 3.20: Inertial readings for IMU sensor on central body of base morphology.

Using this information, a stability metric in a time instant kT is defined as the inverse of the average of roll, pitch and yaw in that same instant (eq. 3.20). Furthermore, this value is averaged in time (eq. 3.21), introducing the idea of total average stability, S_t, which is extracted from every simulation as a fitness indicator.

$$S(kT) = \left(\frac{|\Phi(kT)| + |\Theta(kT)| + |\Psi(kT)|}{3} \right)^{-1} \tag{3.20}$$

$$S_t = \frac{\sum_{k=k_m}^{k_e} S(kT)}{k_e T - k_m T} \tag{3.21}$$

3.8.4 Average Step Number

Having a gait produced by an open loop CPG controller, the number of step cycles per time produced by the robot is constant in steady state. However, in closed loop, perturbations can have an influence in the speed of each step, making it important to know how many steps were performed during measurement time. Additionally, simulation problems like slipping can be detected by a difference between the number of steps performed and its corresponding traveled distance.

Defining the number of step cycles of limb i during the measurement time as SN_i, and a variable dictating the completion of a current step by limb i as st_i, the following is derived.

$$SN_i = \sum_{k=k_m}^{k_e} st_i \,, \qquad st_i = \begin{cases} 1 & \text{if } \phi_i = 2\pi \\ 0 & \text{if } \phi_i \neq 2\pi \end{cases} \tag{3.22}$$

Knowing how many step cycles each limb i was able to perform in the measurement time, the average step number, SN_t, is extracted as a scalar fitness indicator.

3.8.5 Periodicity of Converged Gait

With the purpose of understanding better the influence of the Tegotae-based feedback rule in the behaviour of our system, without looking at its time response, a measurement of periodicity of the steady state performance was required. To do so, the time taken to complete each cycle was used as baseline.

Considering $c_i \in [1, SN_i]$ as the variable for step cycle number, the time taken during a certain cycle c_i of limb i can be expressed as:

$$\Delta t_{c_i} = t_{(\phi_{c_i}=2\pi)} - t_{(\phi_{c_i}=0)} \tag{3.23}$$

Adopting the mean and standard deviation of this cycle time, we are able to compute the signal to noise ratio of each leg's periodic trajectory following, giving a measure of convergence to a limit cycle behaviour.

$$\mu_i = \frac{\sum_{c_i=1}^{SN_i} \Delta t_{c_i}}{SN_i} \,, \qquad \text{std}_i = \sqrt{\frac{\sum_{c_i=1}^{SN_i} (\Delta t_{c_i} - \mu_i)^2}{SN_i}} \,, \qquad SNR_i = \frac{\mu_i}{\text{std}_i} \tag{3.24}$$

Finally, a scalar fitness indicator, PI (for periodicity indicator), is extracted as a convergence measurement, given by the average of signal to noise ratios of the n limbs:

$$PI = \frac{\sum_{i=1}^{n} SNR_i}{n} \tag{3.25}$$

Regarding hardware implementation, only average speed and energy efficiency were computed for each experiment, however, values of reference and real position, force feedback and limb phase (ϕ_i) are extracted, exhibiting the information of step number and periodicity of steady state response.

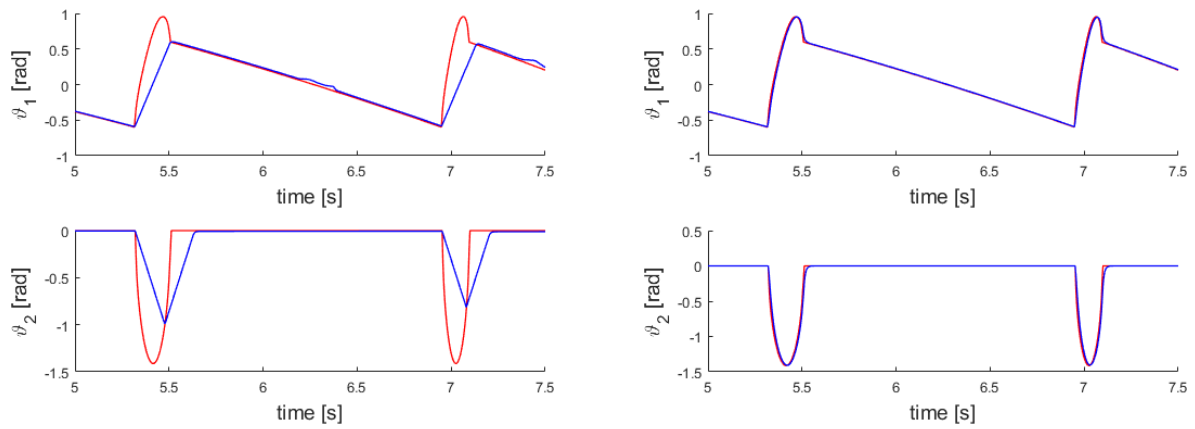
3.9 Experimental Procedures

3.9.1 Simulation

Once defined the control method among the ones in study, as well as the respective parameters described in section 3.6, a unique time response becomes fully characterized. However, finding the exact parameters that correspond first, to a stable locomotion method and second, to an optimal one in terms of energy consumption is not trivial.

With the goal of solving this type of problems, methods of optimization and data analysis that use thousands of simulations, alike Particle Swarm Optimization and Systematic Search are employed. However, the analysis of the system's response to each of the parameter combinations is impossible and the usage of fitness indicators as the ones described in 3.8 is therefore necessary. Nonetheless, for such indicators to promote a reliable understanding of the actual time response of the system, they must be extracted from a well designed experiment.

The resulting experimental procedure reflects the desire of observing the effect of trajectory and trajectory-following parameters in the steady state response of the system. With such intentions, speed and torque saturation of the motors were removed and the local controllers set to have very high gains. This way, no matter the trajectory that is set as reference, it is followed almost perfectly, having always the negative effect of excessive torque and speed revealed in the energy consumption and stability measurements. An example of the same reference trajectory followed with or without saturation is presented in figure 3.21.



(a) With speed and torque saturation.

(b) Speed and torque saturation not considered.

Figure 3.21: Trajectory following in simulation with and without speed and torque saturations – experiment with $f = 1.5$ Hz, $df = 0.9$

Moreover, in order to have reliable measurements of the steady state behaviour of the system, a gap of 30 seconds between the beginning of the simulation and the acquisition of the fitness indexes is given, and a total of 60 seconds of steady state response is measured:

$$t_m = 30\,\text{s}\,, \qquad t_e = 90\text{s} \tag{3.26}$$

In addition, with a view to benefit stable transients from standing to steady state gaits, imposed frequencies (f) are increased at a rate of 0.1 Hz per second until the reference frequency f_r is achieved. Finally, considering equation 3.12, in the case of mechanical coupling, also the σ variable is increased at the same rate, for stability purposes. This procedure is summarized in figure 3.22, having a steady state time given by equation 3.27. Note that, $t_{ss} < t_m$ for all the frequencies and values of σ under study.

$$t_{ss} = 10 \cdot \max(\sigma_r, f_r) \tag{3.27}$$

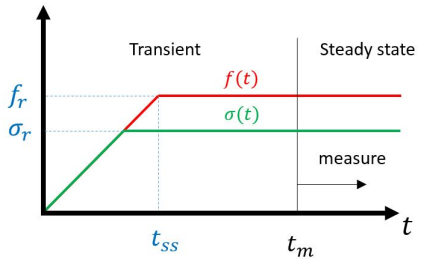


Figure 3.22: Test procedure to allow stable transient.

Upon performing an optimization procedure, it is also relevant to filter undesired solutions which may even present good fitness indicators. An example, is when the robot uses a lot of torque into a jump, traveling a good distance in an energy efficient way, but flipping over due to the highly unstable locomotion method. To prevent these cases to be considered as valid, if at any time instant, any of the rotations (Φ,Θ,Ψ) exceeds $\pi/2$, the specified gait will be considered as unstable and its traveled distance, D_t, will be given a null value.

3.9.2 Hardware

Following the development of the robotic framework introduced in 3.3, the design of experiments to validate conclusions was performed. Placing our robot in a previously marked treadmill, tests were performed with an heuristically predefined traveled distance, measuring time, reference and actual angles, 3D force in each foot and electric current drawn by all the motors. In the case of steady state analysis, initial distance was added to allow the non-consideration of transient phase. All the experiments were recorded, allowing posterior visualization of the gait properties.

The collected raw data was posteriorly processed in order to extract average speed $\overline{v_{CM}}$ and energy efficiency ϵ_t indicators comparable with the ones computed in simulation.

Chapter 4

Experiments

Having explained all the methods developed and implemented to allow the execution of reliable experiments, and the consequent draw of conclusions, the design of the test procedures is here exhibited. In a first stage of the project, extensive work was developed in simulation, being here introduced on sections 4.1 to 4.4. Posteriorly, after hardware implementation, a set of validation experiments was conducted, being resumed on section 4.5.

4.1 Search for Feasibility

Tegotae

One of the initial challenges of this thesis work was to achieve convergence to steady state walking emerging from physical communication through a Tegotae rule in a modular robot. This task was performed in combination with the choice of morphologies and lead to the one used in all the experiments.

Before any study on the influence of trajectories on the dynamics of the robotic system, a trajectory was heuristically defined to analyze feasibility of the convergence to steady state of a robot controlled by a Tegotae based feedback rule. This trajectory is the one of figure 4.1, where the limb workspace is shown in black.

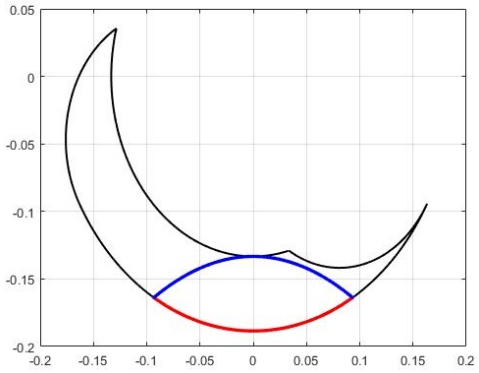


Figure 4.1: Hand-defined trajectory – $\theta_{max} = \pi/6$ rad, $h_{sw} = 60$ mm.

This experiment lead to the first limit cycle behaviour emerged from Tegotae based feedback, observed in simulation for base morphology. The results of this experiment are exhibited in section 5.2.1.

Open Loop CPG

Having performed trajectory parametrization, but preceding all the search and optimization methods used, the theoretical convergence of the system to limit cycle behaviour by usage of open loop CPG-based position control was tested. This experiment had as a goal, to show the resulting outputs of the strategy presented in 3.7.2, as well as its capacity to control our robotic system. With such a goal, df of 0.6 was imposed, as well as $\theta_{max} = 0.3$ rad, $h_{st} = 0$ and $h_{sw} = 0.02$ m.

The results of this experiment are presented in 5.1.1.

4.2 Particle Swarm Optimization of Open Loop CPG

Having as reference the experimental procedure described in section 3.9.1, as well as the simulation method described in 3.2, numerous optimizations were conducted in Biorobotics Laboratory cluster, implemented using job-files as described in section 3.2.

As mentioned in 3.7.1, in an initial phase of choosing a morphology to study and a corresponding parametrization method, a lot of optimizations were performed using PSO. Finally, after selecting a base morphology and the trajectory parametrization introduced in 3.6, two optimization procedures were chosen to be thoroughly repeated, having lead to important conclusions for this thesis work. These optimization problems are described in sections 4.2.1 and 4.2.2.

The referred Particle Swarm Optimizations were performed with a population size of 50 particles, and a number of iterations ranging from 200 to 700, having taken in consideration the computation time in such a choice. The possible lack of time for convergence of solutions may have an impact in the results obtained by these procedures, such as ability to evaluate enough search space and capability to get out of local optima. However, it is expected that, the successive repetition of optimizations combined with their intrinsic stochasticity shall lead to near-optimal solutions.

4.2.1 Energy Efficiency PSO For Constant Frequency

As a first optimization process, its goal was directed to accessing the best possible gait to be performed by the above defined base morphology at an imposed frequency. Regarding results of other tested cost functions, this optimization considers solely the maximization of energy efficiency as defined in equation 3.18, having been performed 6 times per frequency, for 3 distinct frequencies, $f_r \in [0.5, 1, 1.5]$ Hz. The optimization problem can therefore be expressed as below.

Maximize:

$$Z(\theta_{max}, h_{st}, h_{sw}, df, \psi_{12}, \psi_{23}, \psi_{34}) = \sum_{kT=30}^{90} \frac{v_{CM}(kT)}{P_m(kT)} = \epsilon_t \tag{4.1}$$

Subject to:

$$f = f_r$$

$$0 \leq \theta_{max} \leq \pi/3$$

$$0 \leq h_{st} \leq 0.06$$

$$0 \leq h_{sw} \leq 0.06$$

$$h_{sw} > h_{st}$$

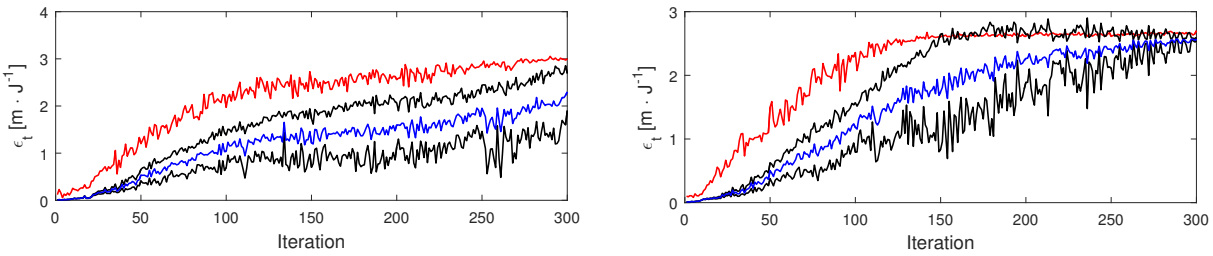
$$0 \leq df \leq 1$$

$$-\pi \leq \psi_{ij} \leq \pi$$

The number of iterations in each optimization process is defined prior to the optimization itself, in the respective job file. Consequently, some optimizations shown a stronger convergence to the presented solution than others. In figure 4.2, the evolution of the cost function throughout the iterations is exhibited for the second and third optimizations of table C.1, where it is visible that the first (figure (a)) is not fully converged, even though it represented the best solution of all the optimizations with $f = 0.5$ Hz. On the other hand the optimization of figure (b) shows a good convergence level to a local optima, given that the best particle does not significantly increase fitness in the last 150 iterations, having all the other particles approximate to this solution.

Besides the appearance of lack of convergence of the second optimization, described in figure 4.2, a global movement of the particles parameters to a region of the search space seems to occur in the first 150 iterations. This is visible on figure 4.3 where the evolution of the parameters of all particles can be seen, as well as the average value (in black). It is then clear the presence of an attraction point where almost all the particles are positioned in the end of the PSO.

The results of this optimization procedure are presented and discussed in section 5.1.2.



(a) Second optimization – $\epsilon_t* = 3.0232$.

(b) Third optimization – $\epsilon_t* = 2.7$.

Figure 4.2: Convergence of energy efficiency during PSO – fitnesses of top particle in red, average of all 50 particles in blue and in black average $\pm$ variance.

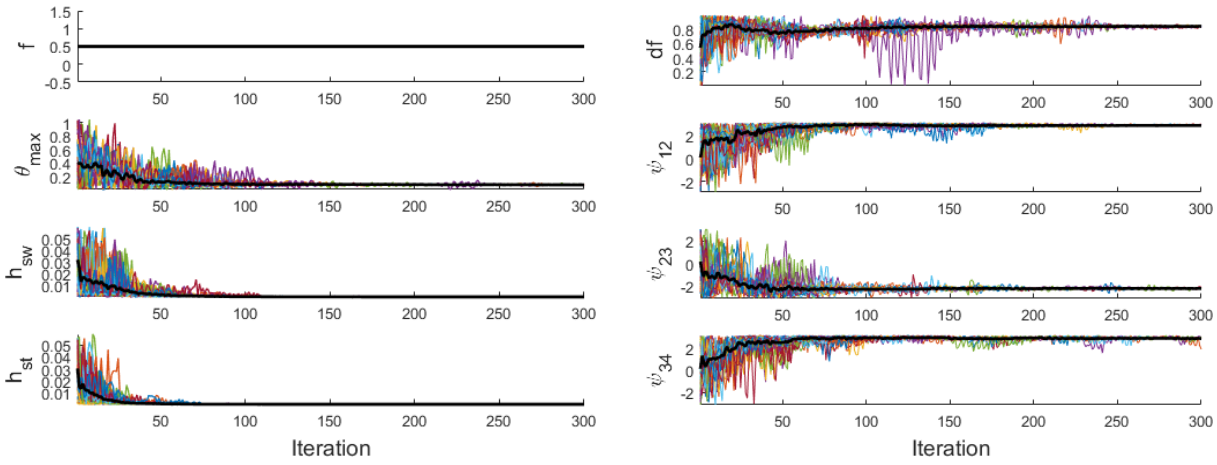


Figure 4.3: Convergence of parameters of all the particles during second optimization of table C.1 – average values in black.

4.2.2 Two-stage PSO – energy minimization for desired speed v_d

Taking into consideration that fast gaits are intuitively more energy consuming than slow ones, it is normal that the first optimization procedure does not result in fast locomotion methods. Furthermore, for control purposes both done by animals or in robots, it is not desired to travel at a certain frequency, but at a specific velocity, and for different velocities distinct gaits seem to be the best option. It is therefore interesting to investigate what are the optimal locomotion parameters for different velocities, both in terms of step trajectory shape and following speed.

With the specified target, stage PSO was implemented in a similar fashion to the previous optimizations, having now as first goal the achievement of desired steady state speed, and as second one the minimization of energy consumption. The first goal acts uniquely as lower boundary of speed, leaving the task of keeping the velocity near the desired value v_d to the intrinsically higher energy consumption of faster locomotion. Hence, the second problem can be defined as done below.

Maximize:

$$Z(f, \theta_{max}, h_{st}, h_{sw}, df, \psi_{12}, \psi_{23}, \psi_{34}) = \sum_{kT=30}^{90} \frac{1}{P_m(kT)} = \frac{1}{E_t} \tag{4.2}$$

Subject to:

$$\overline{v_{CM}} \geq v_d$$

$$0 \leq \theta_{max} \leq \pi/3$$

$$0 \leq h_{st} \leq 0.06$$

$$0 \leq h_{sw} \leq 0.06$$

$$h_{sw} > h_{st}$$

$$0 \leq df \leq 1$$

$$-\pi \leq \psi_{ij} \leq \pi$$

A total of 21 optimizations was saved, with desired speeds of $v_d \in [0.25, 0.5, 0.75, 1]\ m \cdot s^{-1}$, whose results are presented and discussed in section **??**.

4.3 Systematic Search of Trot Gaits in Open Loop CPG

The two optimization methods just presented lead to a group of conclusions exhibited in section 5.1.2. One of these, further validated in hardware, is that walking trot seems to be a very energy efficient gait for our stiff quadrupedal robot to walk at, given the above specified boundary conditions. In addition, the presence of knee bending during stance phase, given by $h_{st} > 0$, seems to only harm the energy efficiency of the gait in question.

Given these assumptions, later validated, the search for optimality can be reduced to 4 parameters – f, θ_{max}, h_{sw}, df – corresponding to a search space small enough to be systematically studied. Therefore, a grid of combinations of imposed trot gaits with stretched knee at stance was examined, collecting the respective fitness indicators.

As explained in section 5.2, with our stiff structure, the observed convergence to a steady state gait from Tegotae is always to trot. The goal of this search was then to clearly understand the influence of step trajectory and imposed duty factor on the speed and energy efficiency of the robot, regarding trot gaits. Values of $0.5 < df < 1$ were considered due to the results of optimization, and observation of animal walking gaits. Trot-like limb de-phasing was imposed ϕ_{ij}, not implying that this is exactly the gait in terms of stepping order.

The tested combinations represent a set of 32400 simulations, with the following range of parameters:

- 6 values of $f \in [0.25, 1.5]$ Hz;
- 30 values of $\theta_{max} \in [0, \pi/3]$ rad;
- 20 values of $h_{sw} \in [0, 0.06]$ m;
- 9 values of $df \in [0.5, 0.95]$.

The results of this search are exhibited in section 5.1.4.

4.4 Systematic Search on Tegotae

A Tegotae-based feedback rule allows the removal of neural coupling parameters ψ_{ij}, as well as intrinsic duty factor, df, having all their influence emerged from the sensory feedback itself. Therefore, and considering as well the effect of h_{st}, a set of only 4 parameters are involved in the process of reaching a near-optimal way of locomotion – f, σ, θ_{max}, h_{sw}.

Similarly to the systematic search performed for open loop, a set of three procedures of search were executed, including 72000 simulations each. These were:

1. Normal grid search;
2. Grid search with σ values scaled with frequency;
3. Grid search with σ values scaled with frequency and binary force information.

The range of parameters considered to perform the just mentioned search methods is listed below.

- 6 values of $f \in [0.25, 1.5]$ Hz;
- 20 values of $\sigma \in [0, 1] \cup [0, 1.2f]$;
- 30 values of $\theta_{max} \in [0, \pi/3]$ rad;
- 20 values of $h_{sw} \in [0, 0.06]$ m.

The results of the referred search procedures are presented in section 5.2.3.

4.5 Hardware Experiments

Having performed a big set of optimizations, systematic search, data and time response analysis prior to hardware implementation, the conclusions that had to be validated were defined, having shaped the experiments further executed.

The set of tests performed on hardware is below enumerated, being further inserted in the results, as validation procedures for simulation hypothesis, or as comparison.

1. Open loop CPG-based control:
   (a) Validation of locomotion with no knee movements – $h_{st} = h_{sw} = 0$;
   (b) Effect of bent knee at stance – $h_{st} \neq 0$;
   (c) Validation tests on trot being the most energy efficient walking gait;
   (d) Search on trajectories: effect of θ_{max} and h_{sw} on imposed trot gaits with $df = 0.5$.
2. Tegotae-based control:
   (a) Feasibility of convergence to limit cycle behaviour from several initial conditions;
   (b) Analysis of effect of σ in convergence time;
   (c) Search on σ *versus* frequency: effect on speed and energy efficiency;
   (d) Convergence and steady state analysis of binary feedback.

Chapter 5

Results and Discussion

Following the experiments introduced in 4, this chapter aims at exhibiting and discussing the results of all the optimization and search methods, as well as respective validation on hardware. The results relative to work developed with neural communication (Open Loop CPG) are presented in a first stage, in an order that benefits explanation and discussion. The results of physical communication with no neural coupling (Tegotae) are presented next in the same fashion.

The graphical representations of time response correspondent to each limb i, exposed in this chapter, consider the limb order of figure 5.1.

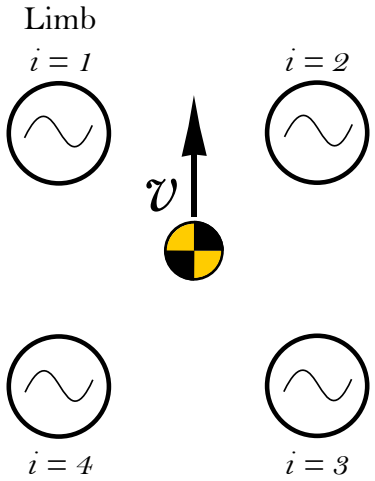


Figure 5.1: Description of limb numbering used for result exhibition.

5.1 Open Loop CPG-based Control

5.1.1 Feasibility

The actuation of the experiment described in 4.1 for open loop CPG is presented in figures 5.2 and 5.3. The convergence of limb phases and joint angles to steady state trot gait is observed, with a speed given by equal weights $w_{ij} = w = 1$ (equation 3.12). The initial increase of ϕ_i at a low rate is due to the experiment procedure described in 3.9, where the frequency is slowly increased, in this case up to 0.5 Hz. However the coupling influence

is very clear, specially in the limbs 1 and 3, where phase increase is delayed to allow convergence to the phase difference present in trot gait.

After transient, phase locking is exhibited, where the coupling term becomes null and all limbs oscillate at constant rate $\dot{\phi}_i = 2\pi f$, with a phase difference of $\psi_{ij} = -\psi_{ji}$. The trajectory following remains further constant, with references in joint space assuming the evolution in time given by the trajectory shape parameters θ_{max}, h_{st} and h_{sw}, as well as by an intrinsic duty factor parameter, df, introduced in section 3.7.2.

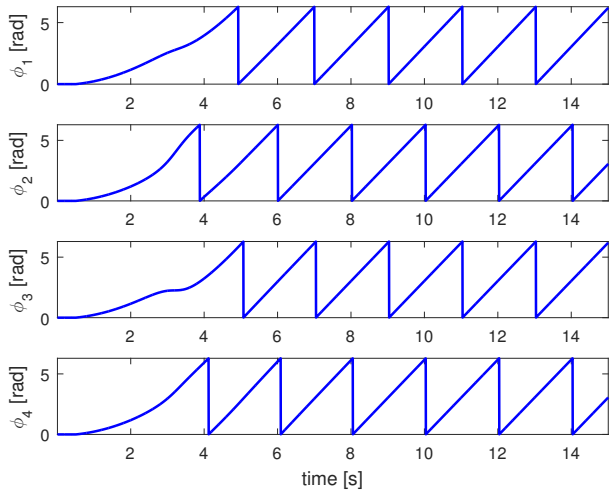


Figure 5.2: **Simulation:** Time evolution of ϕ_i showing influence neural couplings in convergence to trot gait, with posterior phase locking.

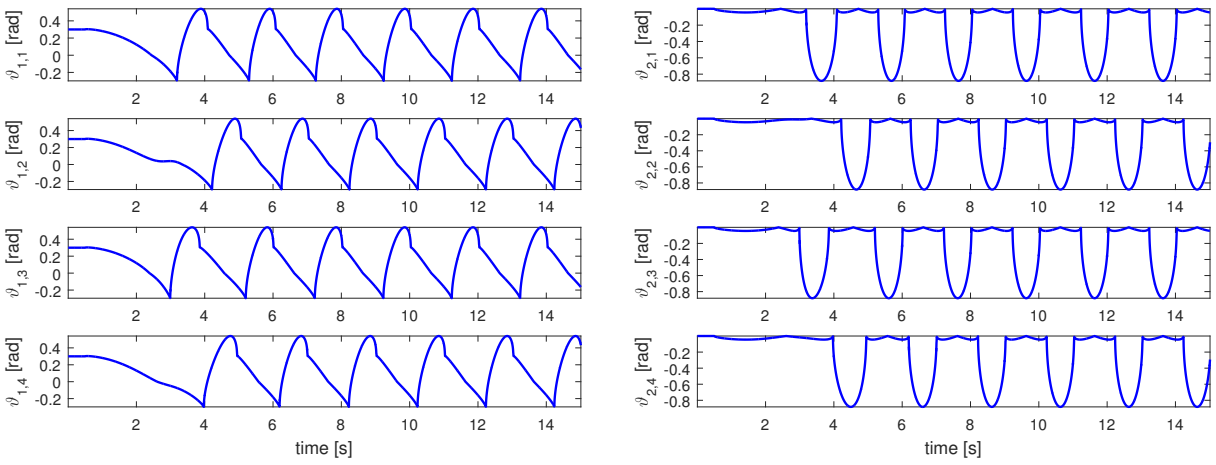


Figure 5.3: **Simulation:** Time evolution of joint references converging to trot gait.

This experiment is presented in order to allow a better understanding on the ability of the control strategy introduced in 3.7.2 to drive the limb phases towards a state where $\phi_j - \phi_i = \psi_{ij}$.

The presence of knee movement in stance phase visible in figure 5.3, even though $h_{st} = 0$, is due to a slight mismatch between the quadratic function defining stance and the workspace boundary expressed by a radial function. It can be proven that this problem never creates reference positions outside the workspace.

5.1.2 Particle Swarm Optimizations of Open Loop CPG

In this section, the results of the Particle Swarm Optimizations detailed in 4.2 are exhibited, being complemented by appendix C.

Energy Efficiency PSO For Constant Frequency

According to the optimization problem defined in 4.2.1, the maximization of energy efficiency, ϵ_t, was repetitively performed for 3 imposed frequencies [0.5, 1, 1.5] Hz, given the stochasticity of the optimization algorithm and the high dimensionality and non-convexity of the problem. The best particle of all the iterations of each of the optimization procedures is exhibited in table C.1.

During each of the thousands of simulations performed, distance traveled (D_t $[m]$), energy consumed (E_t $[J]$) and average stability index (S_t) were collected and used to compute the fitness function ϵ_t $[m \cdot J^{-1}]$, as well as the average travel speed, $\overline{v_{CM}}$ $[m \cdot s^{-1}]$. Hence, in table C.1 it is first indicated the characteristics of the performed optimization such as the fixed frequency f and the number of iterations executed (IT Nr.). Then, the iteration number where the best particle was found (Best IT) is indicated, followed by all the characteristics of the discovered solution. Among these, all the parameters that define the locomotion method and allow a repetition of the gaits are presented with angles in radians (θ_{max}, ψ_{ij}) and distance in millimeters (h_{st}, h_{sw}), as well as the corresponding gaits observed by analyzing the systems time response.

Taking into consideration the energy efficiency and speed of the gaits obtained in this optimization process, expressed in figure 5.4, it is visible that the most energy efficient ones correspond to trot gaits (in black), performed at very low speeds. There seems to be an exponential decay of ϵ_t with speed, and running gaits prove to be much more energetically expensive.

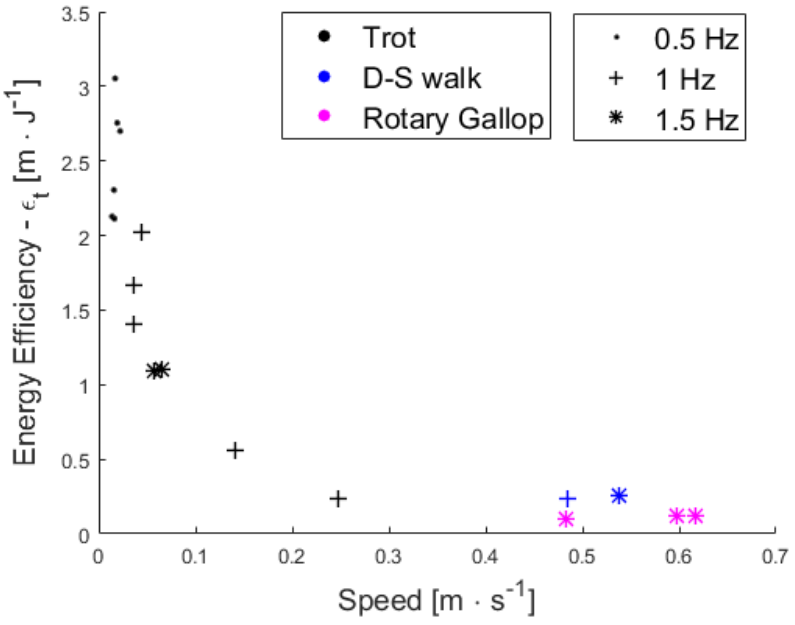


Figure 5.4: Energy efficiency of gaits derived from Particle Swarm Optimization versus locomotion speed – 3 imposed frequencies considered and 3 gaits encountered.

At low frequencies (0.5 Hz), all the solutions encountered correspond to trot gaits with very low speeds. Although, when the imposed frequency is increased, running gaits start to appear as local optima, showing some difficulty of keeping trot gaits at high frequencies, even without limb saturation.

As can be seen in table C.1, the best solutions acquire low speeds by having low values of θ_{max}, and have almost no knee movements ($h_{st} \to 0$, $h_{sw} \to 0$), creating ground clearance only through body dynamics. This is also aided by a value of intrinsic duty factor $df > 0.5$, which imposes a fast swing phase. Values of $df < 0.5$ are only encountered here in rotary gallop.

Two-stage PSO – energy minimization for desired speed v_d

Similarly to the first procedure, the results of the second optimization process, introduced in 4.2.2 and characterized by a two-stage PSO, are presented in table C.2 where now the desired speed v_d classifies the optimization and the gait frequency f is a search parameter. As previously described, the fitness to account for in this second method is energy consumption, $E_t\,[J]$, during one minute of measured simulation data.

As done before, in figure 5.5, the results of table C.2 are summarized, showing the different gaits achieved for each desired speed v_d, as well as their energy consumption. For each optimization group, having exactly the same cost function, initial and boundary conditions, a set of different gaits were achieved, showing the high non-convexity of the optimization problem.

In a first analysis, it is visible that the increase of minimum cost is proportional to the increase of desired speed. At lower speeds, up to 0.5 meters per second, trot gaits are present, being then replaced by diagonal sequence walk as the least expensive locomotion method. Even though there is no imposed limb saturation, the natural frequencies of the body dynamics do not allow trot gaits faster than a certain speed, and converging in this case to a less stable D-S walk. The pendular behaviours (roll and pitch) of these gaits are expressed in figure 5.6, where it is clear that D-S walking gaits with our stiff structure represent limit cycle behaviour with low stability or attraction, unlike trot gaits.

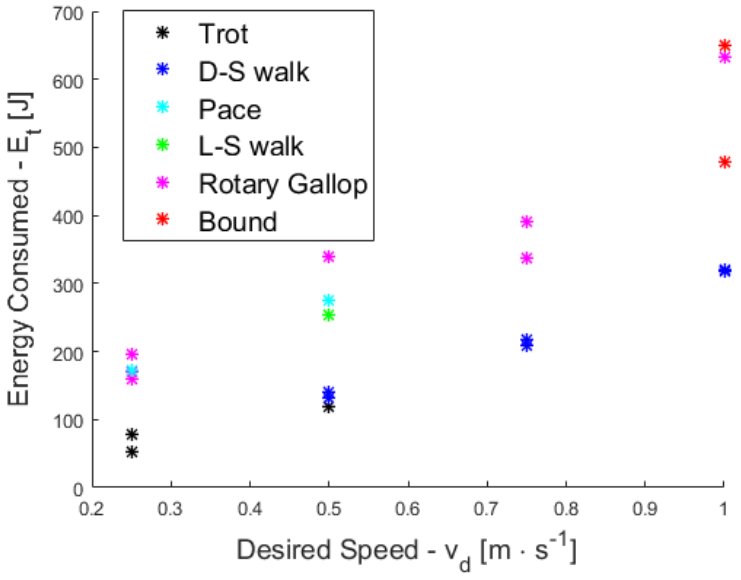
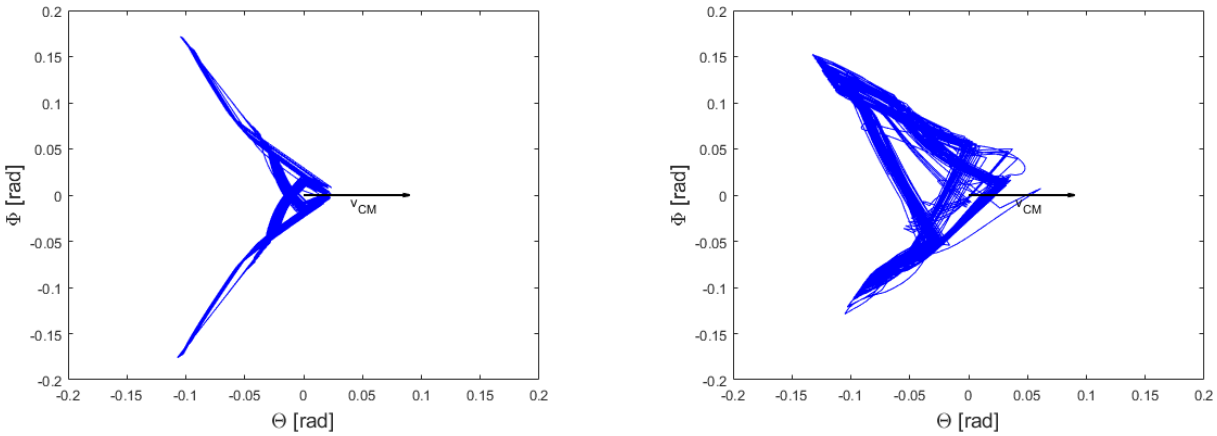


Figure 5.5: Energy consumption of gaits derived from two-stage Particle Swarm Optimization versus desired speed v_d – 4 desired speeds considered and 6 gaits encountered.



(a) Walking trot.

(b) D-S walk.

Figure 5.6: **Simulation:** Limit cycle behaviour observed on walking trot and D-S walk gaits, analyzed on roll (Φ) and pitch (Θ), representing the path followed by an inverted pendulum fixed to the upper body of the robot.

Finding the solutions of figure 5.6 as the most energy efficient with cost and stability decreasing with speed, it can also be noted that these solutions tend to have optimum frequency $f \approx 1.5$ Hz, increasing θ_{max} to match the desired speed. This occurs for $v_d \in [0.25, 0.5, 0.75]$. For $v_d = 1[m \cdot s^{-1}]$, a frequency increase is already necessary.

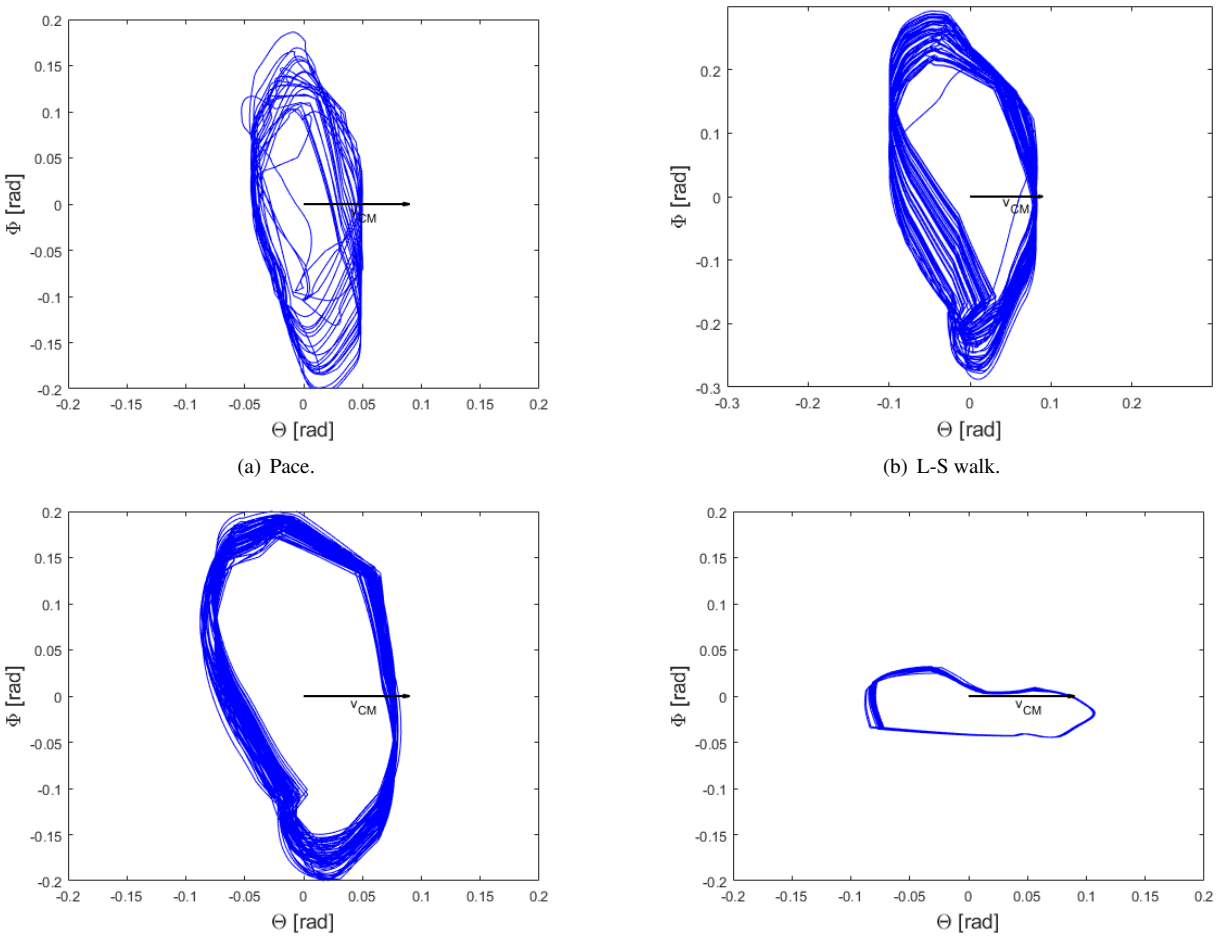
Unlike the results of pure optimization of energy efficiency, here the need for creating ground clearance through knee bending is clear on the optimized values of h_{sw}, having h_{st} assume values slightly higher than zero, increasing with speed to allow a better mechanical stability.

Even though diagonal sequence walk is present in nature, it does not show this kind of performance, since animals exhibit the possibility of creating torso movements necessary for a much more stable execution of this gait, as replicated by spinal compliance in [25].

In an upper level in terms of energetic cost, a second group of gaits appear, where the body dynamics has a much higher influence. This group of gaits is characterized by using the knee in an inverted way as before, having knee bending at stance, and stretching it for swing. A push-off is created in the hind limbs which inputs energy to the system dynamics. In an initial phase, where speeds are $v_d \in [0.25, 0.5]$, these gaits are less stable, showing a stepping order that corresponds to pace or L-S walk. However, as the desired speed increases, running gaits with a stable limit cycle behaviour tend to appear, such as rotary gallop or bound.

This type of solution is allowed in the optimization procedure by a switch between stance and swing phases, imposing a backwards locomotion, and having $df < 0.5$. This implies in reality that the new swing phase (prior stance) is now faster. Despite the non-intuitiveness of such solutions, the phenomena allowed here by a simulation artifact is present in nature, for example in cat galloping.

To sum, despite the diverse gaits obtained, trot is always the most energy efficient, being constrained to certain speed limits. At higher values of v_d, running gaits appear, with stability increasing with speed, but extensive energetic cost. Distinct solutions were encountered for different speeds, however the possibility for gait transitions from trot to running gaits was not part of the scope of this project.



(a) Pace.

(b) L-S walk.

(c) Rotary gallop.

(d) Bound.

Figure 5.7: **Simulation:** Limit cycle behaviour observed in dynamical gaits, analyzed on roll (Φ) and pitch (Θ).

Since trot was the most energy efficient gait, and feasible in a spectrum of speeds from 0 to 0.5 $m \cdot s^{-1}$ (1.2 body lengths per second) this was the gait further analyzed in the systematic search presented in 4.3, where the conclusions derived from the results in discussion lead to admit:

1. $h_{st} = 0$
2. $df \geq 0.5$

The first assumption was later validated on hardware, as well as the fact that trot was indeed the best walking gait for the considered speed range. Such validations are presented in the following section.

5.1.3 Validation Experiments on Hfardware

No knee movements ($h_{st} = h_{sw} = 0$)

According to the results presented in the previous section, the best fitness achieved by the Particle Swarm Optimization for energy efficiency corresponds to solutions of trot gaits with usage of only hip movements ($h_{st} = h_{sw} = 0$), being able to walk with very small ground clearance, achieving it only through the body dynamics. However, the possibility of locomoting in the described way seems to be caused by an artifact present in simulation. To confirm the result of such gaits on hardware, a set of experiments with null h_{st} and h_{sw} was conducted for different frequencies and values of θ_{max}. The results are presented in table 5.1, where it is confirmed that no walking gait is possible without imposed ground clearance through a big enough value of h_{sw}.

In these experiments two types of results were observed - either the absence of movement through the lack of capability to overcome the static friction imposed by the contact of the feet with the ground, or the presence of some movement, designated by "SM" in table 5.1, but without exhibiting a steady state like behaviour or converging to one. Some frames of the twelfth experiment are presented on figure 5.8, describing a movement classified as "SM".

Complementing the results of section 5.1.2, these experiments suggest that h_{sw} is mandatory for ground clearance and consequently for locomotion, leaving trot gaits with $h_{st} = 0$, low values of θ_{max} and high imposed duty factors (df) as characteristics that benefit energy efficiency of a gait.

Table 5.1: Experiments on hardware with $h_{st} = h_{sw} = 0$ – results.

Nr.	f [Hz]	θ_{max} [rad]	walks?
1	0.25	0.1	no
2	0.25	0.2	no
3	0.25	0.3	no
4	0.5	0.1	no
5	0.5	0.2	SM
6	0.5	0.3	SM
7	0.75	0.1	no
8	0.75	0.2	SM
9	0.75	0.3	SM
10	1	0.1	no
11	1	0.2	SM
12	1	0.3	SM

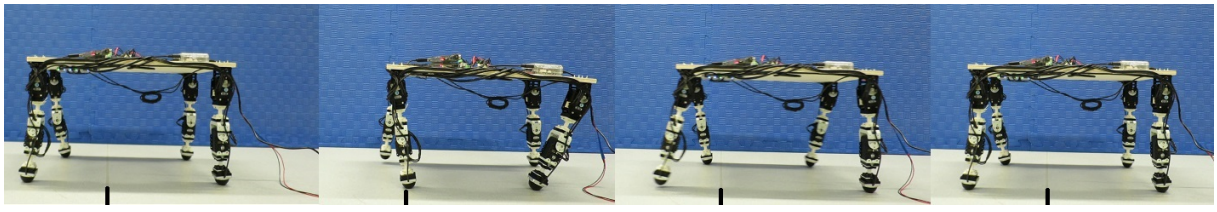


Figure 5.8: Frames from experiment 12 – Robot exhibits limb movements with phase differences of trot but no walking gait, moving back and forth.

Effect of h_{st}

Taking into consideration the results of the optimizations performed, presented in 5.1.2, h_{st} seems to play a relevant role in highly dynamical running gaits, however, for walking gaits this is not the case. Optimizations indicate that, in the case of walking trot, the best option for stance phase is to use a stretched knee ($\vartheta_2 = 0$). In order to confirm this hypothesis, a small set of experiments was performed at 0.25 Hz, comparing two trajectories with the same ground clearance (15 mm), but different values of $h_{st} \in (0, 5)$ and $h_{sw} \in (15, 20)$, followed in open loop. In figure 5.9, the results of computed energy efficiency, in meters per kJ, of gaits with $\theta_{max} \in [0.1, 0.5]$ are presented.

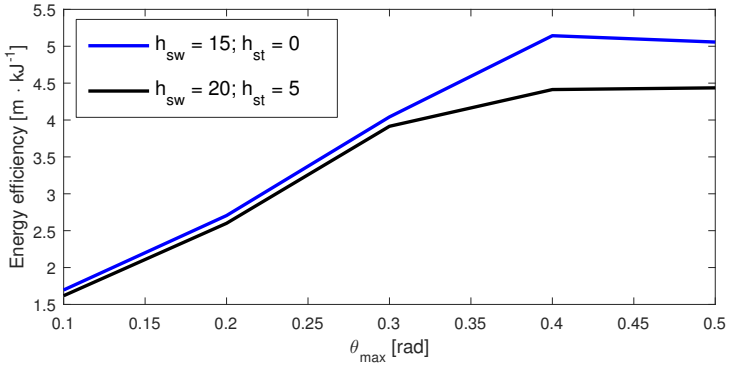


Figure 5.9: Negative effect of $\vartheta_2 \neq 0$ during stance phase of walking trot gaits.

This simple set of experiments allowed a validation on hardware that, indeed, a stretched knee is the most energy efficient in low speed walking gaits, specially at higher values of θ_{max}, indicating the cost of additional knee torque.

Trot as most efficient walking gait

To allow the further analysis of only trot gaits, a validation on hardware that trot is effectively the best walking method for our robot was necessary. Two sets of experiments were performed with this goal, one comparing trot with other symmetrical walking gaits present in nature, D-S and L-S walk, and a second one focused on understanding if trot was the best solution locally. These validations were performed with $f = 0.25$ Hz, $\theta_{max} = 0.3$ rad, $h_{sw} = 15$ mm and $df = 0.5$.

The results of the first set of experiments are exhibited in table 5.2, where it is clear the advantage of trot in terms of speed and energy efficiency.

Table 5.2: Comparative analysis between symmetrical walking gaits showing great advantage of trot.

Imposed Gait	$\overline{v_{CM}}\ [m \cdot s^{-1}]$	$\epsilon_t\ [m \cdot kJ^{-1}]$
Trot	0.059	4.042
D-S walk	0.035	1.583
L-S walk	0.032	1.87

Regarding now the possibility of trot being not exactly the optimum but near it, a set of experiments was conducted to understand if a small de-phasing could actually benefit the robots performance. The results are presented in table 5.3, where it is visible that trot is again the most energy efficient. In terms of speed it is shown that a small de-phasing does not affect much the performance, being even capable of improving it very slightly.

Table 5.3: Local analysis on performance by de-phasing of fore and hind limbs relative to perfect trot.

Imposed Gait	$\overline{v_{CM}}\ [m \cdot s^{-1}]$	$\epsilon_t\ [m \cdot kJ^{-1}]$
Trot	0.0555	2.977
+5% dephased	0.0555	2.953
-5% dephased	0.0557	2.874

The different results in terms of pure trot gait are due to different hardware versions. The first set was performed before the integration of the Inertial Measurement Unit (IMU), Arduino nano and current sensor in the robot, which increased the transporting weight.

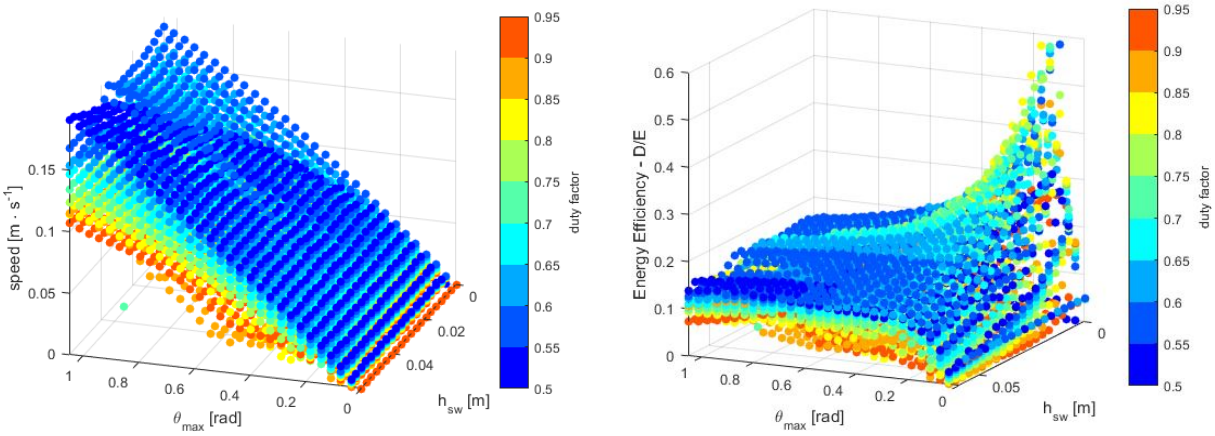
After the extensive optimizations and after these validation procedures, only four parameters were left under study by the systematic search that follows: two trajectory parameters (θ_{max}, h_{sw}), frequency f and imposed duty factor df.

5.1.4 Systematic Search of imposed Trot Gaits – θ_{max}, h_{sw}, f, df

Following the search method described in section 4.3, the fitness indicators corresponding to each of the combinations of parameters were extracted. Considering the 4 different parameters, as well as the fitness values of average speed, energy efficiency and stability, 7-dimensional data had to be analyzed to understand correlations and trends. Splitting the data of the gaits performed at different frequencies, the influence of the remaining parameters on the different fitnesses can be visualized in 3-Dimensional graphs with a fourth dimension given by color. An example of this can be found in figure 5.10, for speed and energy efficiency of gaits with frequency $f = 0.25$ Hz. The resulting graphs from this search are presented in appendix D.

In a manner coherent with the results of PSO, it is visible on trajectory that steps that represent almost no movement are the ones that show more energy efficiency, but no speed. Hip amplitude θ_{max} seems to correlate almost linearly with speed, having the necessity for h_{sw} to create ground clearance appearing only in values of $\theta_{max} > 0.4$ and for $df \approx 0.5$. In terms of speed h_{sw} does not play a role, affecting however the energy efficiency. A good value of h_{sw} seems to be the one minimum to create ground clearance for a fixed θ_{max}, but not bigger, as it implies added energy cost.

Regarding the effect of imposed duty factor df, a value around 0.5 seems to benefit speed, having a negative consequence proportional to the increase of this parameter. Such result can be explained by an increase of velocity during transition from swing to stance, creating forces opposite to the direction of movement. In terms of energy efficiency, values ranging from 0.55 to 0.8 are preferable, depending on the trajectory parameters.



(a) Average speed as a function of θ_{max}, h_{sw} and df.

(b) Energy efficiency as a function of θ_{max}, h_{sw} and df.

Figure 5.10: **Simulation:** Results of systematic search of open loop CPG-based control with imposed trot-like limb phases for $f = 0.25$ Hz.

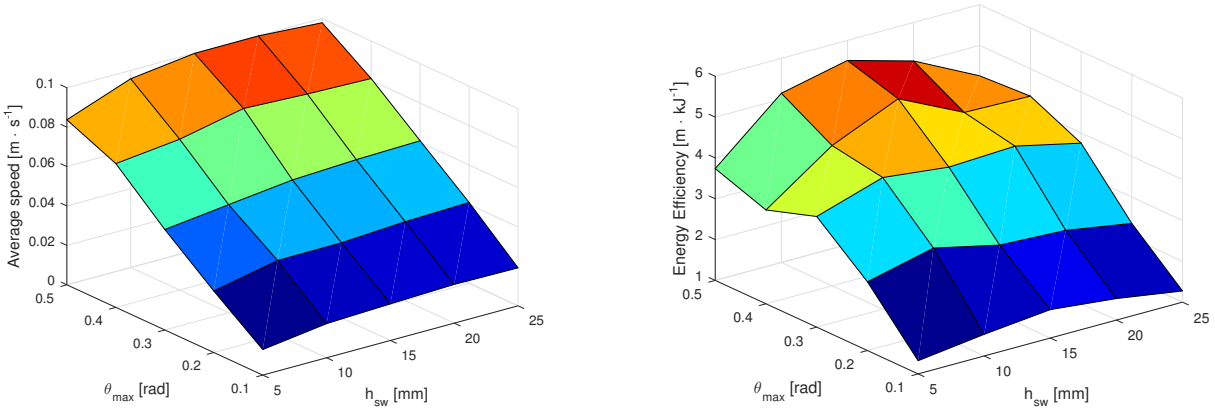
On appendix D, it is visible that the increase of frequencies has a hard implication on the clarity of the results, with the appearance of noisy regions.

5.1.5 Search on Hardware: Trajectory

In order to have a base of comparison between the results of simulation and the actual hardware, a small search process of 20 experiments was performed at a frequency of 0.25 Hz, for $df = 0.5$, $\theta_{max} \in [0.1, 0.5]$ rad and $h_{sw} \in [5, 25]$ mm. The results are exhibited in figure 5.11.

In terms of speed, the conclusions drawn in simulation are matched almost perfectly by hardware, with a linear increase of speed with θ_{max} and the need for a slightly higher h_{sw} at big hip amplitudes. This is observed in simulation for $df = 0.5$ with a decay of speed in the region characterized by $\theta_{max} > 0.3$ rad and $h_{sw} < 20$ mm.

Regarding energy efficiency, they are not comparable. In simulation the mechanical power is used to compute energetic cost by equation 3.17, however, on hardware, the current is measured at a time step of 1 millisecond and integrated in time, using equation 3.19 for total energy computation.



(a) Average speed as a function of θ_{max} and h_{sw}.

(b) Energy efficiency as a function of θ_{max} and h_{sw}.

Figure 5.11: **Hardware:** Search on trajectories performed for $f = 0.25$ Hz and $df = 0.5$.

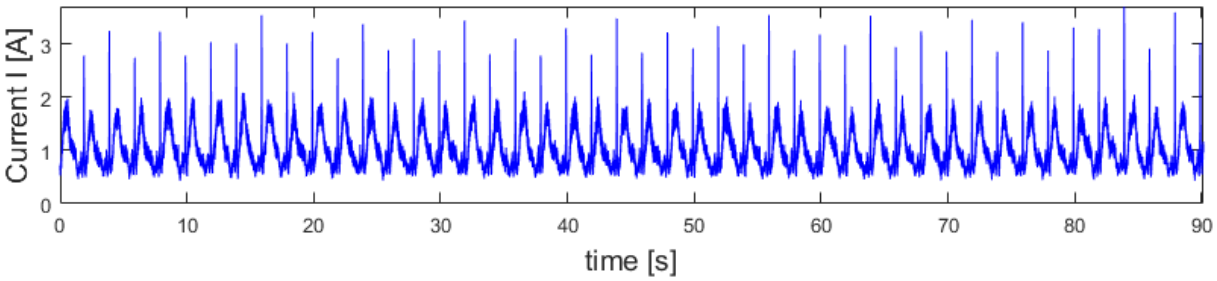


Figure 5.12: **Hardware:** Example of current measurement during walk – trajectory defined by $\theta_{max} = 0.3$ rad, $h_{sw} = 15$ mm, $f = 0.25$ Hz, $df = 0.5$.

In figure 5.12, the current drawn during a 90 second experiment of locomotion is shown at example. As it is visible, a base current is drawn from the motors, in a way that the lowest level of current during the experiment is ≈ 0.5 A. This base energetic cost, integrated in time creates the differences between the simulation and hardware.

How to proceed in such a case is fully dependent on the goal. If the goal is to understand animal locomotion, this base current should not be accounted for, or a value similar to the energetic cost of animal life should be kept, considering also the cost of being stopped, which depends on the current posture and corresponding motor torques. On the other hand, if the interest lies within the control of a specific robot, all the energetic consumption is ought to be accounted for, and in such a case, energetic cost in simulation should have an integrative base.

Given that energy efficiency is defined in equation 3.18 as total distance per energy unit, the integrative effect of base current will favor fast locomotion gaits, which in simulation, are much less energy efficient than slow ones.

Finally, the primitives that are possible to extract from both simulation and hardware, regarding trajectory, are that a value of h_{sw} should be big enough to create a ground clearance (≈ 15 mm on hardware), but not higher, since this imposes an energetic cost without influence in speed.

5.2 Tegotae-based Closed Loop Control

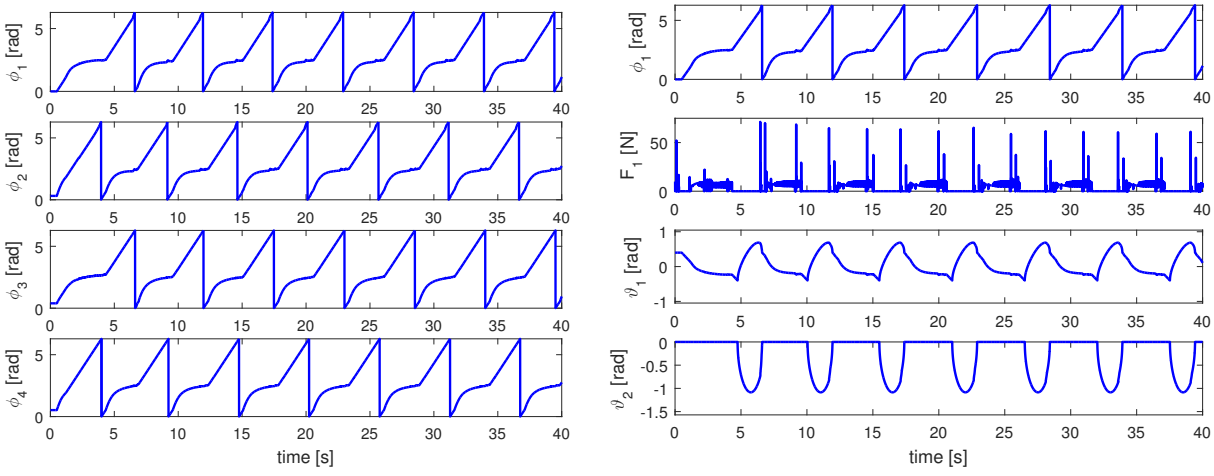
5.2.1 Feasibility

Taking into consideration the early experiment described in 4.1, the results of a first convergence to steady state limit cycle behaviour are presented in figures 5.13 and 5.14.

In figure 5.13 (a), the convergence of limb phases to trot gait emerging from the Tegotae rule is shown, having as initial conditions $\phi_i \approx 0$. In figure 5.13 (b), limb 1 is more thoroughly analyzed, allowing comparison between step phase, force feedback and joint references throughout the locomotion gait.

In both figures it is visible the acceleration followed by deceleration during stance phase, being visible in figure (b) force peaks at $\phi_1 \approx 0$, due to collision of limb 1 with the ground and at $\phi_1 \approx \pi$, due to collision of limbs 2 and 4 with the ground before limb 1 starts swing phase. These peaks have a direct influence in $\dot{\phi}_1$, amplified by high values of σ, mainly because of the absence of mechanical filtering, which is usually present through compliance in the works of gait replication [24] to [26]. In addition, in figure 5.13 (b) it is visible the emergent duty factor, explicitly clear in the force feedback measurements.

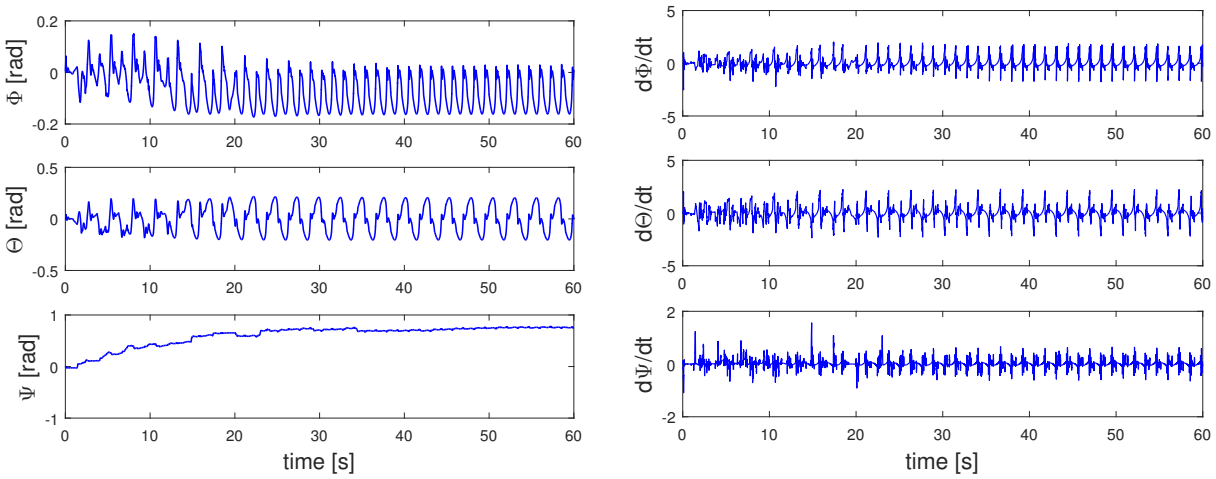
Regarding limit cycle behaviour, the periodicity of the gait can be observed in figure 5.14 where roll, pitch, yaw, and their derivatives were considered as state variables. In such figure it is possible to clearly distinguish between transition and steady state phases, having the second one showing cyclic evolution in time of all state variables (Φ, Θ, Ψ, $\dot{\Phi}$, $\dot{\Theta}$, $\dot{\Psi}$).



(a) Phase evolution of each limb.

(b) Phase, force feedback and joint references of limb 1.

Figure 5.13: **Simulation:** Time evolution of gait characteristics on first successful experiment with Tegotae-based control – convergence to trot gait observed.



(a) Orientation values.

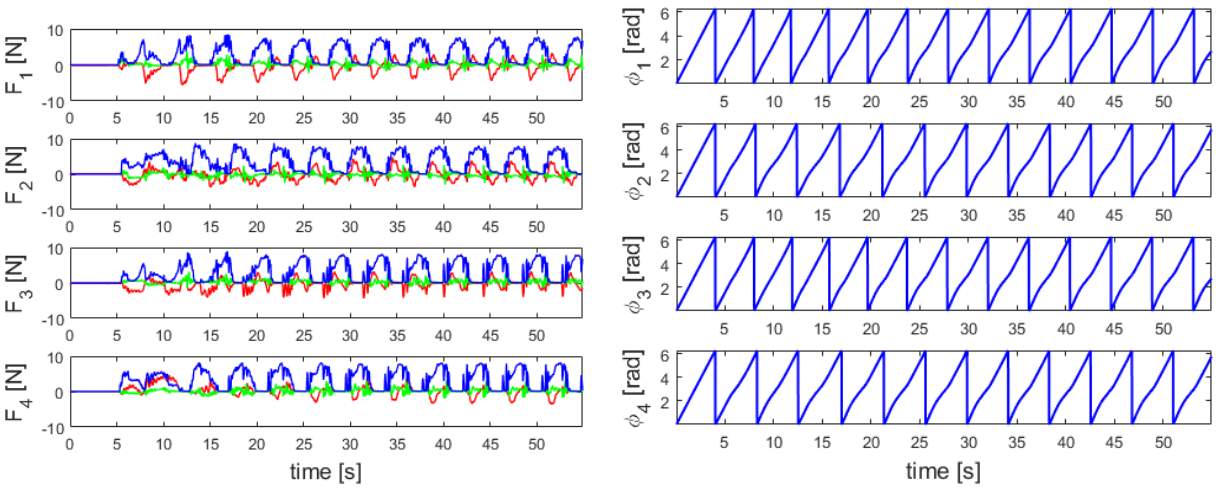
(b) Derivates of orientation values.

Figure 5.14: **Simulation:** Time evolution of inertial state variables – convergence to limit cycle behaviour.

5.2.2 Convergence to Trot gait

After hardware implementation, the feasibility of convergence to trot gait from an arbitrary initial condition, with decentralized control of each limb by a Tegotae rule was ought to be shown. In figure 5.15, the convergence from limbs in phase, to steady state trot is shown both in limb phases ϕ_i and in the 3-dimensional force measurements F_{i_x}, F_{i_y} and F_{i_z}.

Several initial conditions were tested, having always resulted in steady state trot gaits, which we have seen to be the most energy efficient. The speed of convergence from zero initial condition of all limbs was proven to decrease with σ, in the manner described in figure 5.16.



(a) Force measurements of each limb i – F_{i_x} in red, F_{i_y} in green and F_{i_z} in blue.

(b) Evolution of ϕ_i in time.

Figure 5.15: **Hardware:** Convergence to limit cycle behaviour correspondent to trot, shown in force measurements and limb phases – All limbs in phase initially, and $\sigma = 0.1$.

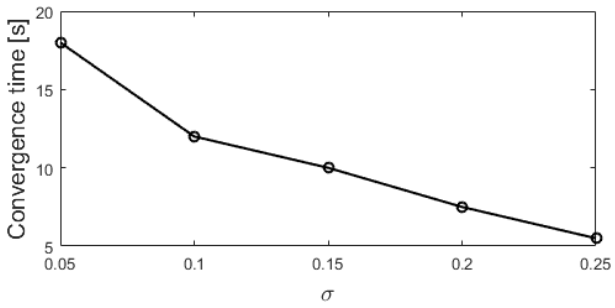
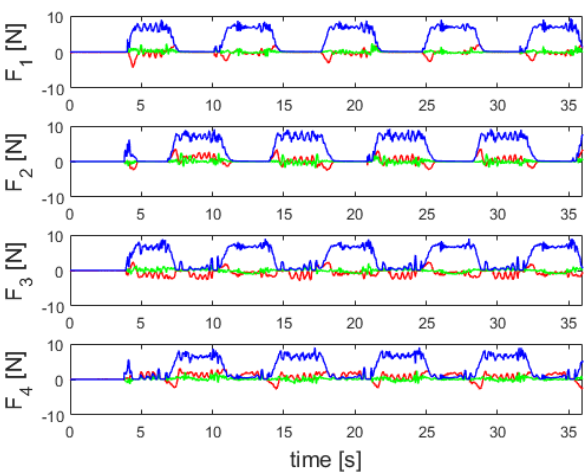


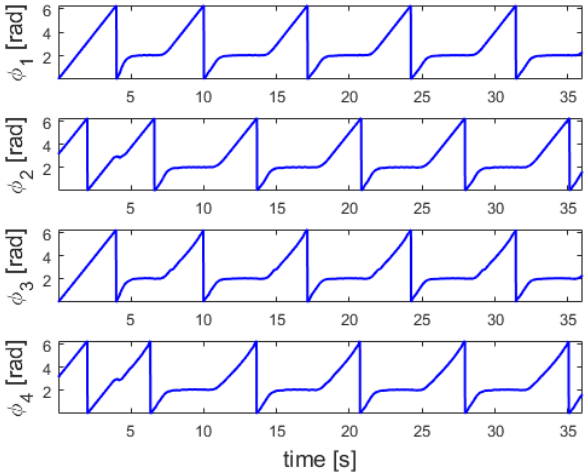
Figure 5.16: **Hardware:** Convergence time from in-phase limbs to steady state trot, dependent of attraction coefficient σ.

It is therefore clear that the biggest the attraction coefficient, the faster the convergence to trot gait. However, the steady state phase evolution, $\phi_i(t)$, is also reshaped due to this attraction coefficient. In figure 5.17, the phenomena of too much attraction in steady state is shown, where the speed of the gait is severely affected, reducing the effective stepping frequency to around half of the imposed one.

The aim of the search, whose results are presented in the next section, was therefore to analyze the effect of different σ values in the steady state response of the system, with the goal of identifying principles that benefit performance.



(a) Force measurements of each limb i – F_{i_x} in red, F_{i_y} in green and F_{i_z} in blue.



(b) Evolution of ϕ_i in time

Figure 5.17: **Hardware:** Time evolution of phase and ground reaction forces, when high attraction coefficient is used – $\sigma = 0.5$ for $f = 0.25$ Hz.

5.2.3 Systematic Search on Tegotae – θ_{max}, h_{sw}, f, σ

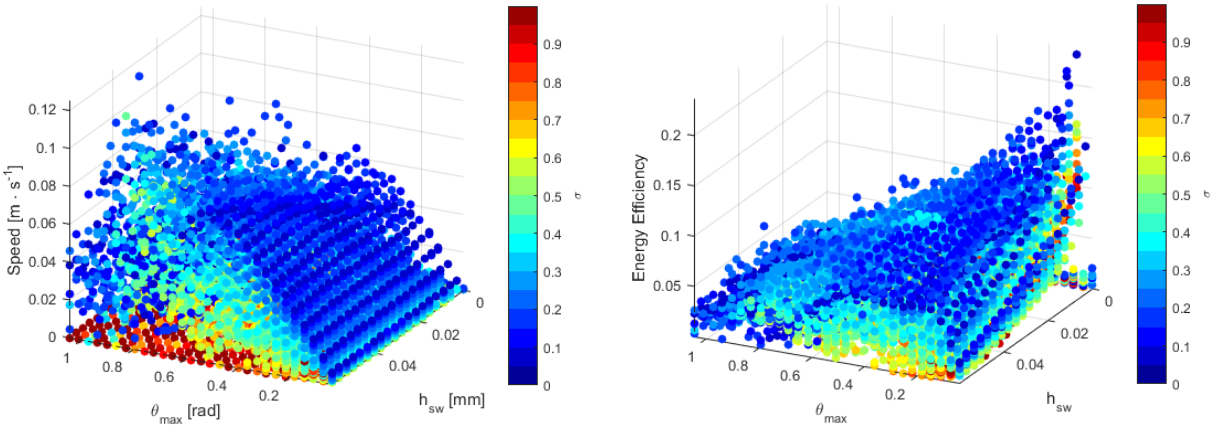
Similarly to the process performed for open loop, the results of the systematic search presented in 4.4 are here discussed, now emphasizing two topics: the effect of sigma on trajectory and the scaling of sigma with frequency.

Trajectory

Figure 5.18 describes the effect of trajectory in the case of closed loop on steady state performance of the robot for $f = 0.25$. The results for the remaining frequencies are exhibited in appendix D.2.

By comparison with the results on open loop, presented in figure 5.10, it is clear the reduction of limb trajectories that correspond to stable locomotion, emphasized for higher frequencies. However, the trends in the linear region, both in terms of speed and energy efficiency seem to be the same, validating the conclusions taken in terms of trajectory in sections 5.1.4 and 5.1.5. After this linear regime, for high values of θ_{max}, a lot of uncertainty appears, probably due to effect of noisy force feedback (visible in measurements presented in figure 5.19).

Very low values of σ seem to be the most energy efficient, as well as the fastest for a given trajectory. However, higher attraction allows a linear regime in an increased range of θ_{max}.



(a) Average speed as a function of θ_{max}, h_{sw} and σ.

(b) Energy efficiency as a function of θ_{max}, h_{sw} and σ.

Figure 5.18: **Simulation:** Results of systematic search on closed loop Tegotae-based control for $f = 0.25$ Hz.

Scaling with frequency

Considering the way equation 3.12 imposes attraction trough the coefficient σ, it is intuitive that at higher frequencies f, an equal value of σ will have a decreased effect. Such decrease is of course dependent on the fact that dynamically, the magnitude of force feedback N_i is not correlated to frequency. Figure 5.19 shows this independence in simulation.

Hence, scaling properties of σ with frequency should be observable on the performed systematic search. Taking into consideration the linear regimes of the diverse frequencies, the distributions shown in figure 5.20 can be extracted from the data set, where it is here clear the advantages in terms of both steady state speed and energy efficiency of having small values of σ, showing a negative effect of over-attraction to the stable point $\phi_i = \pi/2$.

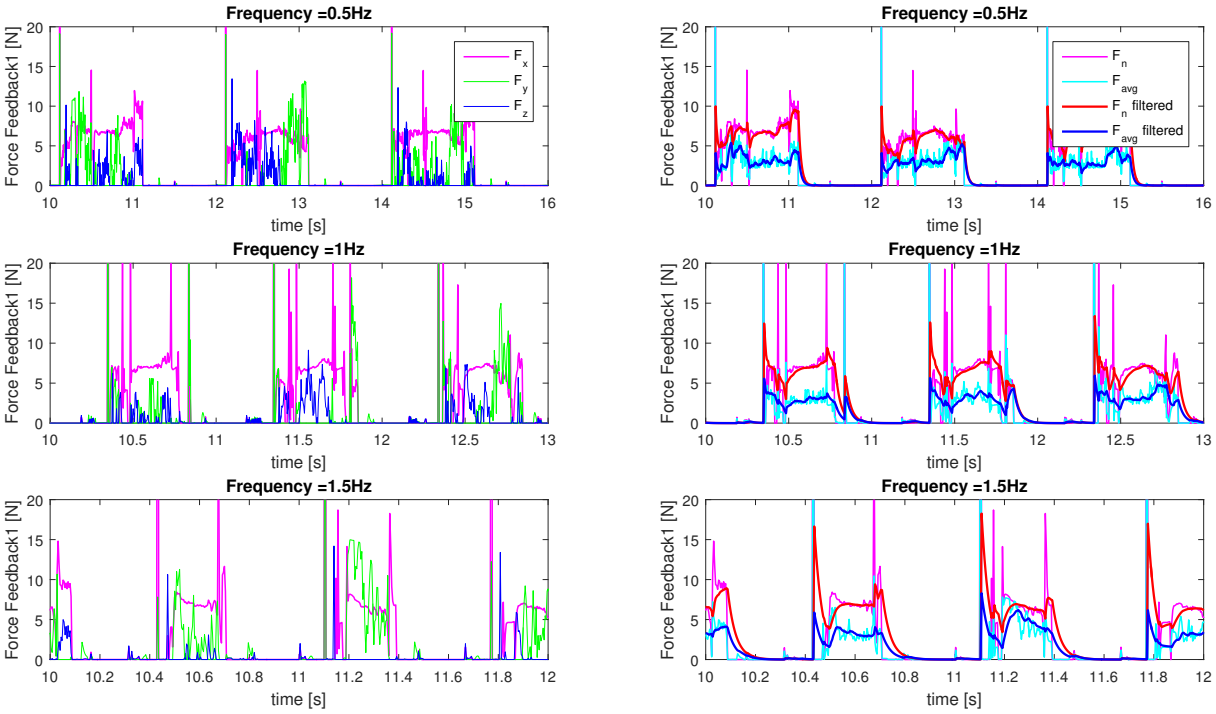
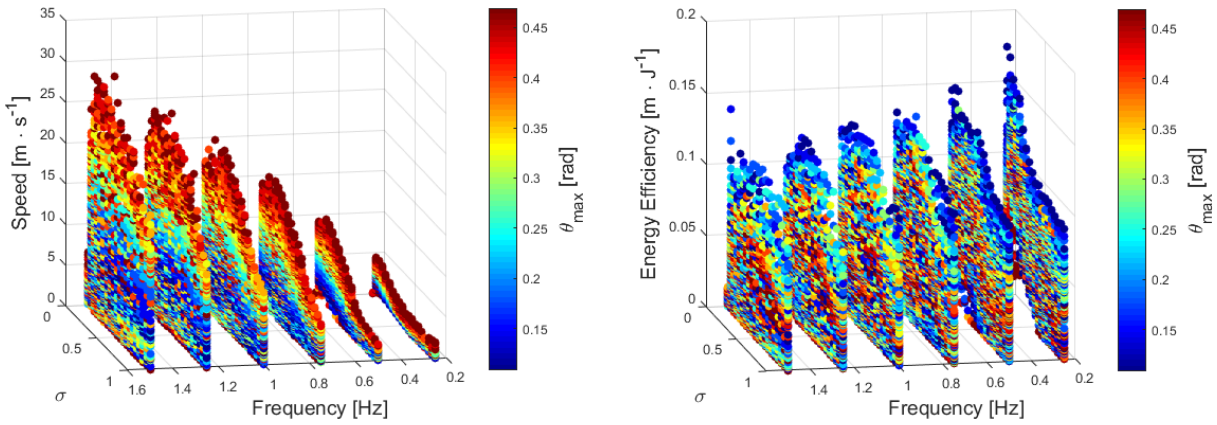


Figure 5.19: **Simulation:** Force measurements from Limb 1 of steady state trot gait performed by base morphology at different frequencies – On the left, the 3-dimensional components of force in local reference frame are presented, where F_x is the component in the axis of the lower limb. On the right, normal force $F_n = F_x$ is compared with average of force components, F_{avg}, before and after filtering by a first order low-pass filter.



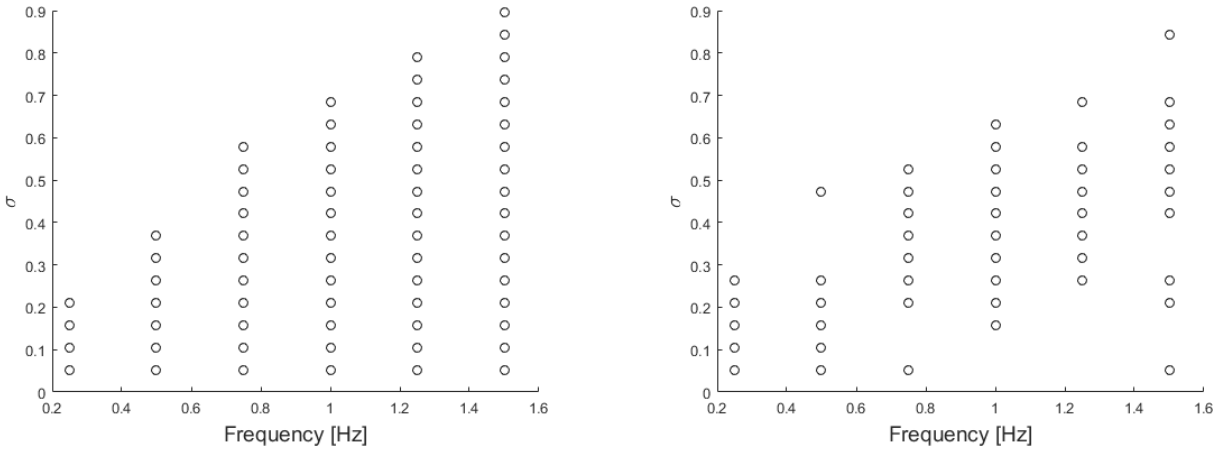
(a) Average speed as a function of σ, f and θ_{max}.

(b) Energy efficiency as a function of σ, f and θ_{max}.

Figure 5.20: **Simulation:** Scaling properties of σ with frequency observed in speed and energy efficiency of gaits from systematic search – data filtered by $\theta_{max} \in [0.1, 0.5]$ rad and $h_{sw} > 0.5$ mm to consider only linear regimes.

By collecting the best 15 values of each frequency set in terms of speed and energy, expressed in figure 5.21, the referred scaling properties are observable in speed. Here the best sigma is the one that allows a necessary convergence to trot gait, but that does not create an over-attraction, as the one described in figure 5.17.

In energy efficiency, a trend is visible, favoring higher values of σ for higher frequencies. This effect is explained by the a matching of σ to the one corresponding to an emergent duty factor near to the optimum, which we have seen to be around 0.6.



(a) Top 15 simulations in terms of average speed.

(b) Top 15 simulations in terms of energy efficiency.

Figure 5.21: **Simulation:** Top 15 simulations in terms of speed and energy efficiency of gaits from figure 5.20 – scaling properties of σ with frequency observed.

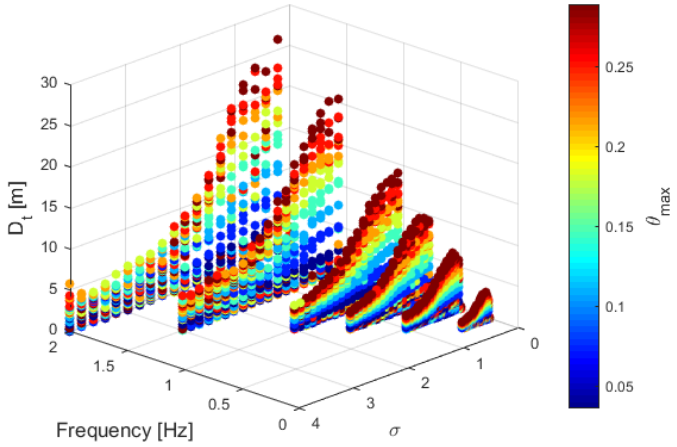


Figure 5.22: **Simulation:** Systematic search with scaled intervals of σ with frequency – scaling effect observed in distance traveled.

With a view on validating such scaling properties, a systematic search considering values of sigma scaled with frequency was performed, having the resulting data points exhibited in figure 5.22, where the conclusions taken for figure 5.20(a) are more clear.

5.2.4 Search on Hardware: Tegotae and Frequency

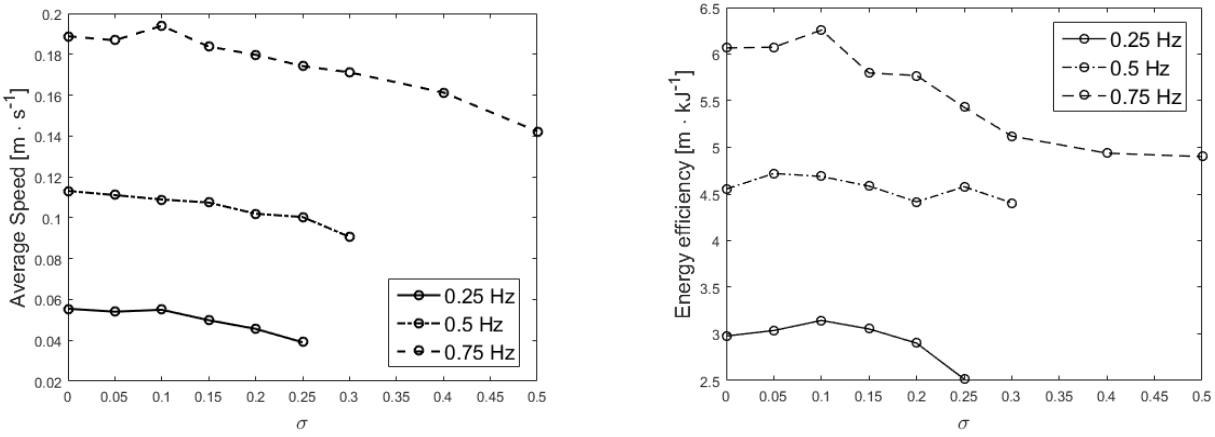
In order to validate the primitives encountered on simulation, a set of hardware experiments was performed at 3 different frequencies ($f \in (0.25, 0.5, 0.75)$ Hz), having the results expressed on figure 5.23.

Regarding speed, the negative effect of attraction is shown, inducing a slight quasi-linear decay, with some traces of scaling.

In terms of energy efficiency, it is important to note that decentralized control of each limb by a Tegotae rule, was able not just to drive the system towards a stable gait, as it forced convergence to gaits even more efficient than imposed trot. This was acquired by the emergence of a duty factor closer to the optimum, and by real time adjustments to perturbations or non-symmetries on the system dynamics.

As previously introduced, there is a big difference between energy computation in simulation and hardware, which here plays an important role as well. The integrative base current favors faster gaits, explaining the higher energy efficiency curves of the top frequencies. Additionally, because of such phenomena, high values of σ are penalized in energy efficiency. Having this in mind, and looking at figure 5.23(b), there seems to be an increase of the near-optimum region.

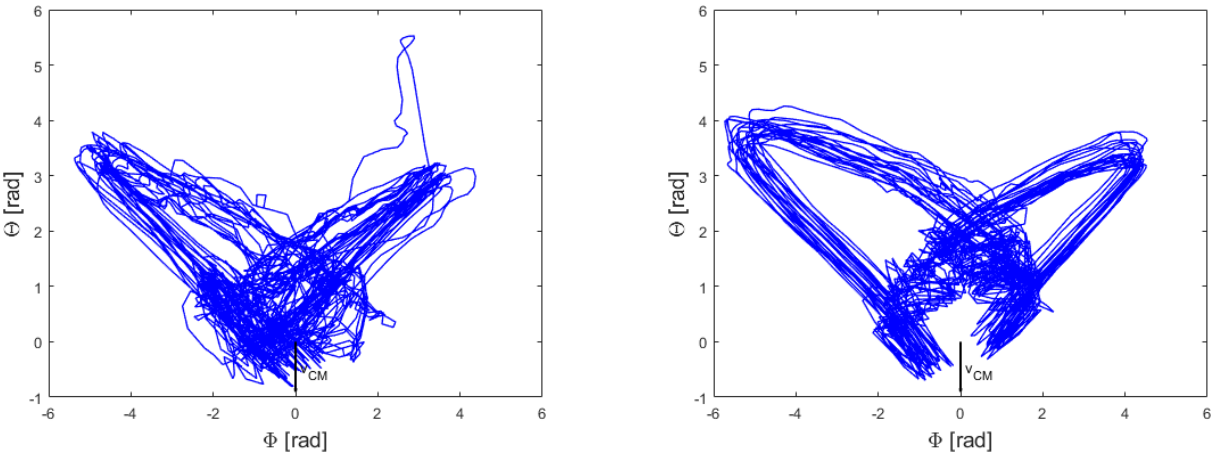
Besides allowing the emergent convergence from any initial condition to solutions which are very close to optimal in terms of efficiency, the simple Tegotae feedback rule also increases robustness of the system to perturbations. This effect can be observed in the limit cycle behaviour of the experiments correspondent to $\sigma = 0$ and $\sigma = 0.3$ – figure 5.24. In open loop, the transition from swing to stance phase is not smooth, with collisions against the ground that affect mechanical stability and direction of locomotion. However, with the introduction of σ, the gait becomes much more stable, with smoother stepping and body motion – figure 5.24(b).



(a) Average speed – $\overline{v_{CM}}$.

(b) Energy efficiency – ϵ_t.

Figure 5.23: **Hardware:** Analysis of effect of frequency and σ on average speed and energy efficiency – $\sigma = 0$ corresponds to imposed trot gait.



(a) Open loop trot gait: $\sigma = 0$.

(b) Closed loop gait with $\sigma = 0.3$.

Figure 5.24: **Hardware:** Comparison between open and closed loop pendular behaviours observed in roll (Φ) *versus* pitch (Θ) – frequency $f = 0.75$ Hz.

5.2.5 Binary Feedback – B_i

Finally, with a view on understanding if the force profile during stance phase has an influence in the quality of Tegotae-based control, or if binary information would be enough, a set of experiments considering the control method given by equation 3.13 was performed.

Simulation

In a similar fashion to the previous search procedures, the grid method was employed in binary tegotae, showing even more robustness than with non-binary information – figure 5.25. The size of the linear region is increased, due to filtering of the very noisy force feedback. This indicates the relevance of a better collision computation from the simulation algorithms (discussed in section 3.4).

Given the initial conditions far from trot, the similarities with the results presented in 5.2.3, show that binary information is in fact enough to drive the system towards a steady state limit cycle behaviour with trot gait and speed similar to the one observed before.

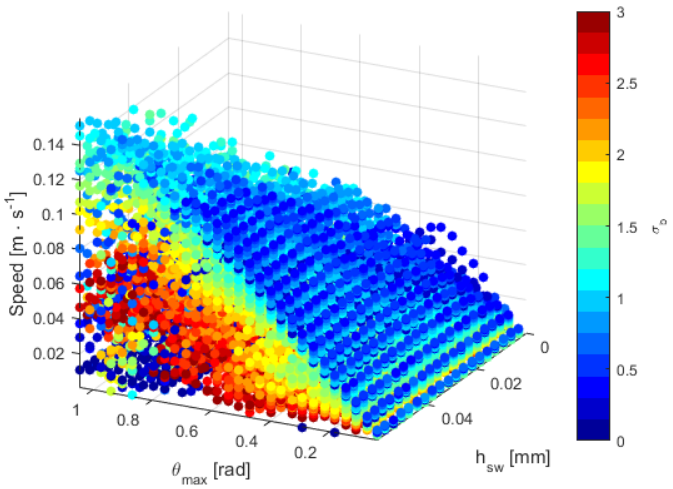


Figure 5.25: **Simulation:** Systematic search for binary feedback – Effect of trajectory and σ_b on steady state speed for $f = 0.25$ Hz.

Hardware: Convergence

With an aim at understanding the differences of this control technique in terms of the mentioned convergence, experiments in hardware were conducted for 3 different frequencies – $f \in (0.25, 0.5, 0.75)$ Hz.

Figure 5.27 shows the experiment performed at 0.25 Hz, with all limbs initialized with zero phase. Due to the threshold present in the control approach (see equation 3.13), a limit cycle occurs with all limbs in phase (figure 5.26), and the noise is not able to help the controller to drive the system away from such solution, as would occur in normal Tegotae.

However, when a perturbation was added (lifting of limbs 1 and 4 – second 17 of figure 5.27(a)), the system will get out of this locally stable limit cycle and converge to a steady state trot gait, proving the capacity for such convergence.

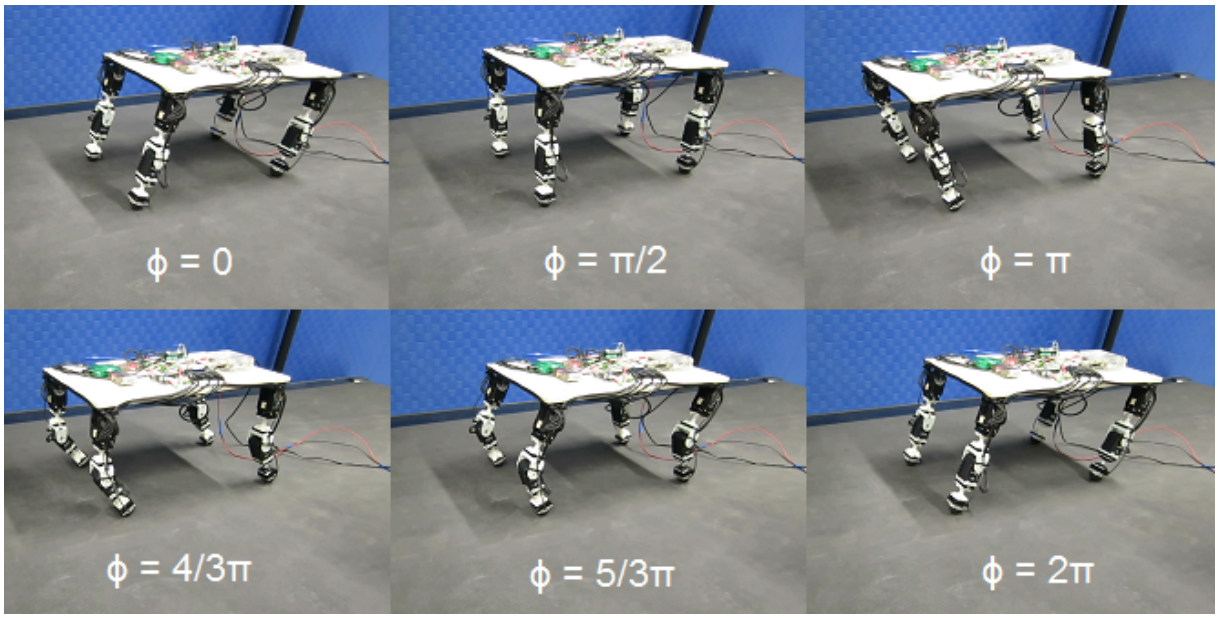
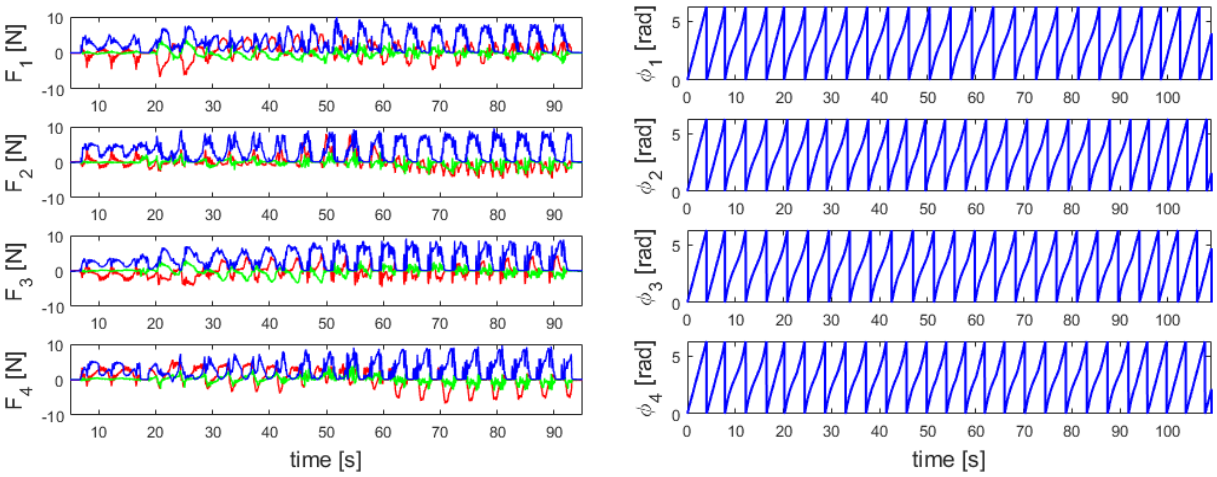


Figure 5.26: **Hardware:** Experiment frames of locally stable limit cycle behaviour – binary feedback not able to drive the system away from this solution.



(a) Force measurements of each limb i – F_{i_x} in red, F_{i_y} in green and F_{i_z} in blue.

(b) Evolution of ϕ_i in time.

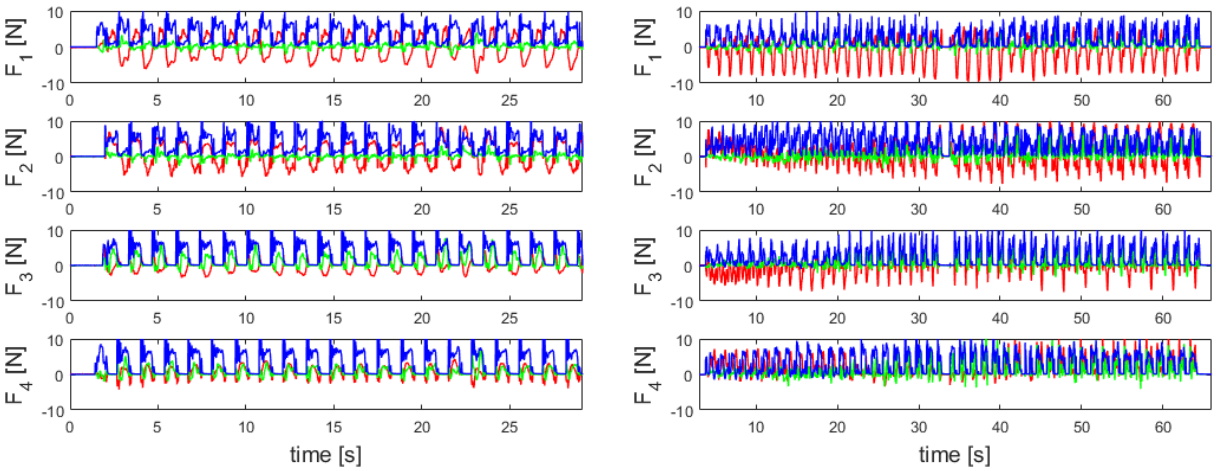
Figure 5.27: **Hardware:** Local limit cycle behaviour of limbs in phase ($t \in [0, 17]$s), followed by perturbation (lifting of limbs 1 and 4 at $t \approx 17$s) and convergence to limit cycle behaviour correspondent to trot (steady state reached at $t \approx 60$s) – $\sigma_b = 0.6$.

Having proven the feasibility of control with binary feedback, it should be however noted that the convergence time proves to be much higher than with normal feedback (compare figure 5.27(a) to 5.15(a). In fact the convergence times of the 3 experiments performed are presented in table 5.4, where for both 0.25 and 0.5 Hz, the time for convergence was around 40 seconds. Comparing with the times of normal feedback, given in figure 5.16, and specially with $\sigma = 0.1$ (correspondent to $\sigma_b = 0.6$), with only binary information, the convergence time is more than doubled.

In the case of 0.75 Hz, convergence to a nearly steady state gait is faster, but time response stays very noisy after transient. This can be seen in figure 5.28, where the comparison between force measurements on normal and binary Tegotae is performed, being visible the increased stochasticity of the second.

Table 5.4: Convergence times measured on hardware experiments with binary feedback at different frequencies.

Frequency [Hz]	Convergence time [s]
0.25	43
0.5	41
0.75	≈ 20



(a) Normal feedback – $\sigma = 0.1$

(b) Binary feedback – $\sigma_b = 0.6$

Figure 5.28: **Hardware:** Normal *versus* Binary feedback for $f = 0.75$ Hz observed in force measurements of convergence to trot gait (external perturbation added in experiment b for repositioning of the robot in treadmill at $t \approx 33$s).

Hardware: Steady State

Having analyzed convergence, the steady state speed and energy efficiency of binary Tegotae was compared with the one of normal Tegotae, showing lack of performance of the first. Table 5.5 shows the different results, implying that besides a slower convergence, binary feedback also loses relevant information in steady state, which does not allow it to achieve similar speed and efficiency.

Table 5.5: Binary *versus* normal feedback in terms of steady state speed and energy efficiency.

Feedback type	$\overline{v_{CM}}[m \cdot s^{-1}]$	$\epsilon_t[m \cdot kJ^{-1}]$
$\sigma = 0.1$	0.0559	3.146
$\sigma_b = 0.6$	0.0458	2.619

Chapter 6

Conclusions and Future Work

In a long term, this project aims at giving a contribution to the goal of controlling modular robotic structures with an arbitrary configuration. In an initial phase, throughout the numerous iterative processes and broad work developed to achieve convergence to steady state limit cycle behaviours, several problems that must be solved to reach this long-term goal were identified, being one of these, dynamical stability. Considering legged movements, the weight distribution of the robot and inertial characteristics play a very important role, and robust mechanics is essential to provide the necessary stability in transient phases. In addition, the quality of a morphology to effectively locomote by legged movements highly depends on the projection of the workspace of its limbs in a plane that includes the direction of transport. These represent characteristics that are not present in all configurations and therefore a morphological study on robustness is necessary, and can be started in the robotic platform developed in this thesis work.

Providing that the system includes the necessary mechanical properties, it is shown that a CPG-based control approach using a simple parametrization can result in many gaits found in nature and, as in animal kingdom, walking gaits are more stable at low speeds and running gaits at high velocities. Walking trot is proven to be the most energy efficient way of locomotion for our robotic platform, and in comparison with [25] this shows that spinal compliance plays an important role in walking gaits such as diagonal sequence and lateral sequence walk. The fact that no other gait besides trot was achieved by the decentralized approach, as done in [24], also reveals that compliance is crucial for dynamical running gaits.

Regarding step trajectory followed in open loop, it was possible to extract general principles for performance. Having a stretched knee during stance phase, and imposing a duty factor slightly above 0.5 increases energy efficiency. Having a swing phase characterized by the minimum ground clearance necessary to avoid stumbling is desired (around 15 mm for hip amplitudes up to 1 rad). The final traveling speed is linearly affected by hip amplitude and frequency.

During the project development, a lot of differences were encountered between simulation and hardware, discussed in 3.4, being the most relevant the different computation of energetic cost. A correct choice of a base cost for no movement is crucial and directly linked to the goal of the study. Knowing that stopped motors consume energy, and that standing in diverse postures has different associated consumption, a base cost should have a term purely dependent on the torque.

In the case of Tegotae, it was shown that a decentralized control of each limb, driven by sensory feedback, can result in convergence from any initial condition to a steady state limit cycle behaviour. Additionally, the resulting gait proves to be even more energy efficient than induced trot due to underlying effects that include emergent duty factor and increased robustness to perturbations.

High values of attraction coefficient σ are beneficial for fast convergence from an arbitrary initial condition to a dynamically stable limit cycle. In steady state, attraction favors mechanical stability and robustness to perturbations, however, low attraction proves to be more energy efficient. Hence, a non-constant value of σ should be used for control purposes. In the case of knowledge of the optimal gait, an attraction proportional to the difference between current and desired limb phases should be used, leading to a Tegotae rule alike the one of equation 6.1, where σ_c refers to constant sigma. In a more general case where the most efficient gait is not known, an exponential decay of σ with time between a value big enough to ensure convergence, and a value that represents a compromise between mechanical stability and energy efficiency should be used.

$$\dot{\phi_i} = \omega + \sigma_c \cdot \sum_j (\phi_j - \phi_i - \psi_{ij}) \cdot N_i \cos\phi_i \tag{6.1}$$

In addition, scaling properties of the attraction coefficient with frequency were observed, implying that this effect should also be accounted for in the design of a Tegotae rule that better fits the purpose of controlling a generic quadruped robot. The proposed alterations to the used feedback rule result therefore in equation 6.2.

$$\dot{\phi_i} = \omega \left(1 + \sigma_c \cdot \sum_j (\phi_j - \phi_i - \psi_{ij}) \cdot N_i \ \cos\phi_i\right) \tag{6.2}$$

As final conclusions of this project, binary feedback proved to be enough to drive the system towards steady state locomotion, having however loss of relevant information with consequences in convergence time, and steady state performance.

This thesis work represented an initial phase of a wider project, having included the implementation of software and hardware tools to be used in future studies and experiments. Among these, analysis on the mechanical properties required for feasibility shall be performed by changes in limb lengths and compliance levels.

Very important future work including Tegotae, is also to understand how morphological changes, such as non-symmetrical limbs, would affect the performance of the decentralized control approach, with the aim of confirming the adaptive capabilities of this approach. Analyzing adaptability to rough terrain would also be relevant, with attention on the fact that trajectory should be also adapted to increase ground clearance.

Finally, on longer term, strategies that include sensory feedback from each module are necessary, to allow combination of walking and crawling behaviours. Mechanisms for gathering feedback and more local Tegotae strategies have to be developed, and hopefully the additional simulation tools implemented on the scope of this master project (appendix E) will be of help.

Bibliography

[1] A. A. Biewener. Allometry of quadrupedal locomotion: the scaling of duty factor, bone curvature and limb orientation to body size. *Journal of Experimental Biology*, 105(1):147–171, July 1983.

[2] Nicolás E. Campione and David C. Evans. A universal scaling relationship between body mass and proximal limb bone dimensions in quadrupedal terrestrial tetrapods. *BMC Biology*, 10:60, 2012.

[3] N. C. Heglund and C. R. Taylor. Speed, stride frequency and energy cost per stride: how do they change with body size and gait? *Journal of Experimental Biology*, 138(1):301–318, September 1988.

[4] Herman Pontzer. Effective limb length and the scaling of locomotor cost in terrestrial animals. *Journal of Experimental Biology*, 210(10):1752–1761, May 2007.

[5] S. Grillner, T. Deliagina, A. El Manira, R. H. Hill, G. N. Orlovsky, P. Wallén, Ö. Ekeberg, and A. Lansner. Neural networks that co-ordinate locomotion and body orientation in lamprey. *Trends in Neurosciences*, 18(6):270–279, 1995.

[6] Jacques Duysens and Henry W. A. A Van de Crommert. Neural control of locomotion; Part 1: The central pattern generator from cats to humans. *Gait & Posture*, 7(2):131–141, March 1998.

[7] Milan R. Dimitrijevic, Yuri Gerasimenko, and Michaela M. Pinter. Evidence for a Spinal Central Pattern Generator in Humans. *Annals of the New York Academy of Sciences*, 860(1):360–376, November 1998.

[8] V. Dietz. Spinal cord pattern generators for locomotion. *Clinical Neurophysiology*, 114(8):1379–1389, August 2003.

[9] Thomas A. McMahon and George C. Cheng. The mechanics of running: How does stiffness couple with speed? *Journal of Biomechanics*, 23:65–78, January 1990.

[10] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model. *Science*, 315(5817):1416–1420, March 2007.

[11] Yasuhiro Fukuoka, Hiroshi Kimura, and Avis H. Cohen. Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts. *The International Journal of Robotics Research*, 22(3-4):187–202, March 2003.

[12] C. Liu, Q. Chen, and D. Wang. CPG-Inspired Workspace Trajectory Generation and Adaptive Locomotion Control for Quadruped Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(3):867–880, June 2011.

[13] Jiaqi Zhang, Feng Gao, Xiaolei Han, Xianbao Chen, and Xueying Han. Trot Gait Design and CPG Method for a Quadruped Robot. *Journal of Bionic Engineering*, 11(1):18–25, January 2014.

[14] Jiang Shan, Cheng Junshi, and Chen Jiapin. Design of central pattern generator for humanoid robot walking based on multi-objective GA. In *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings*, volume 3, pages 1930–1935 vol.3, 2000.

[15] Carla M. A. Pinto and Martin Golubitsky. Central pattern generators for bipedal locomotion. *Journal of Mathematical Biology*, 53(3):474–489, July 2006.

[16] Gen Endo, Jun Morimoto, Takamitsu Matsubara, Jun Nakanishi, and Gordon Cheng. Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. *The International Journal of Robotics Research*, 27(2):213–228, February 2008.

[17] J. Fang, Y. Ren, and D. Zhang. A robotic exoskeleton for lower limb rehabilitation controlled by central pattern generator. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 814–818, December 2014.

[18] Alexander Spröwitz, Alexandre Tuleu, Massimo Vespignani, Mostafa Ajallooeian, Emilie Badri, and Auke Jan Ijspeert. Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot. *The International Journal of Robotics Research*, 32(8):932–950, July 2013.

[19] M. Vespignani, K. Melo, S. Bonardi, and A.J. Ijspeert. Role of compliance on the locomotion of a reconfigurable modular snake robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2238–2245, September 2015.

[20] M. Vespignani, E. Senft, S. Bonardi, R. Moeckel, and A.J. Ijspeert. An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4308–4313, November 2013.

[21] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, 7(4):431–441, December 2002.

[22] A. Sprowitz, S. Pouya, S. Bonardi, J. V. Den Kieboom, R. Mockel, A. Billard, P. Dillenbourg, and A. J. Ijspeert. Roombots: Reconfigurable Robots for Adaptive Furniture. *IEEE Computational Intelligence Magazine*, 5(3):20–32, August 2010.

[23] Stéphane Bonardi, Massimo Vespignani, Rico Möckel, Jesse Van den Kieboom, Soha Pouya, Alexander Spröwitz, and Auke Ijspeert. Automatic Generation of Reduced CPG Control Networks for Locomotion of Arbitrary Modular Robot Structures. *Proceedings of Robotics: Science and Systems*, 2014.

[24] Atsushi Tero, Masakazu Akiyama, Dai Owaki, Takeshi Kano, Akio Ishiguro, and Ryo Kobayashi. Interlimb neural connection is not required for gait transition in quadruped locomotion. *ResearchGate*, October 2013.

[25] Dai Owaki, Takeshi Kano, Ko Nagasawa, Atsushi Tero, and Akio Ishiguro. Simple robot suggests physical interlimb communication is essential for quadruped walking. *Journal of The Royal Society Interface*, 10(78):20120669, January 2013.

[26] Dai Owaki, Leona Morikawa, and Akio Ishiguro. Why do quadrupeds exhibit exclusively either trot or pace gaits? *Dynamic Walking Annual Meeting*, 2013.

[27] Masashi Goda, Sakiko Miyazawa, Susumu Itayama, Dai Owaki, Takeshi Kano, and Akio Ishiguro. Understanding Interlimb Coordination Mechanism of Hexapod Locomotion via "TEGOTAE"-Based Control. In Nathan F. Lepora, Anna Mura, Michael Mangan, Paul F. M. J. Verschure, Marc Desmulliez, and Tony J. Prescott, editors, *Biomimetic and Biohybrid Systems*, number 9793 in Lecture Notes in Computer Science, pages 441–448. Springer International Publishing, July 2016. DOI: 10.1007/978-3-319-42417-0_44.

[28] Ryo Yoshizawa, Kano Takeshi, and Akio Ishiguro. Realization of Snakes' Concertina Locomotion by Using "TEGOTAE-Based Control". *Biomimetic and Biohybrid Systems*, 2016.

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

[30] Örjan Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69(5-6):363–374.

[31] L. Righetti and A. J. Ijspeert. Pattern generators with sensory feedback for the control of quadruped locomotion. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, pages 819–824, May 2008.

[32] James Kennedy and Russell Eberhart. A New Optimizer Using Particle Swarm Theory. *Micro Machine and Human Science, 1995. Proceedings of the Sixth International Symposium.*

[33] Webots Reference Manual - release 7.4.3, May 2014.

[34] Webots User Guide - release 7.4.3, May 2014.

[35] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics - Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London, London, 2009.

Appendix A

Discarded Parametrization Methods

A.1 Parametrization in Joint Space

Animal foot trajectories during walking include stance phase, where stretched knee is many times observed, and swing phase, where knee bending is crucial to allow ground clearance. In an attempt to mimic such behaviours, hip movement was modelled as a sinusoidal wave, while knee bending was parametrized using a sigmoid-type periodic function of $\phi \in [0, 2\pi]$ (figure A.1). This parametrization method includes only 1 parameter for the hip, amplitude A, and five parameters for the knee, phase displacement d, width w, height h, and two shape parameters s_1 and s_2. This method is described by the following equations:

$$\vartheta_1(t) = A \cdot \sin(\omega t) \tag{A.1}$$

$$\phi(t) = \omega t \in [0, 2\pi] \tag{A.2}$$

$$\vartheta_2(\phi) = \frac{h/2 \cdot s_1 \cdot (\phi + w - d)}{\sqrt{1 + (s_1 \cdot (\phi + w - d))^2}} - \frac{h/2 \cdot s_2 \cdot (\phi - w - d)}{\sqrt{1 + (s_2 \cdot (\phi - w - d))^2}} \tag{A.3}$$

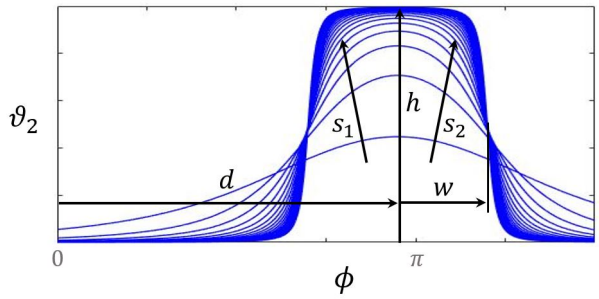


Figure A.1: Sigmoid type function for parametrization of knee bending.

This method led however to poor resulting gaits, indicating that considering the hip actuation as defined in A.1 is too restrictive. Having in attention that the knee bending parametrization was already very high dimensional, and that a simple representation was desires, further attempts on allowing more complex hip movements were discarded in favor of a parametrization in operational space.

A.2 Parametrization of Ellipse in Operational Space

One first approach to parametrize the reference trajectory in operational space was the use of an ellipse defined by four parameters – o_x, o_y, r_x and r_y – as in figure A.2. However this solution is not ideal, as only specific combinations of the parameters involved result in valid trajectories, in a way that the remaining are constrained by the available workspace.

Additionally, the cases of non feasibility would create impossible solutions of the inverse kinematics, sending non valid joint references to the motors, and such cases would have to be identified by the optimization method itself and highly penalized in the fitness. To avoid this need, a parametrization method that takes into account the workspace itself was seen as the most convenient procedure.

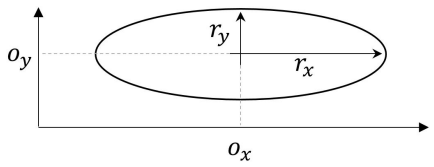


Figure A.2: Parametrized ellipse.

Appendix B

Hardware Details

B.1 Iterative Process

Regarding the mechanical part, the process of hardware implementation matched the iterative procedure of initial morphology choosing. An example of this is the mechanical assembly on figure B.1 of the *rect_quadruped* robot, whose model is presented in figure 3.1 (b). Building such morphology, even without the matching electronics to actuate it, was enough to comprehend that some problems can arise, in the more simple morphologies, as the one presented. These problems include bodyweight distribution and lack of limb workspace in the direction of desired movement.

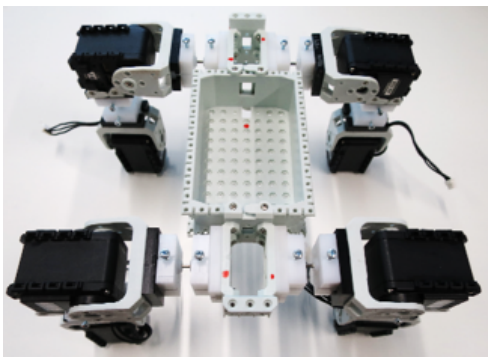


Figure B.1: Initial hardware tests: *rect_quadruped* morphology assembled.

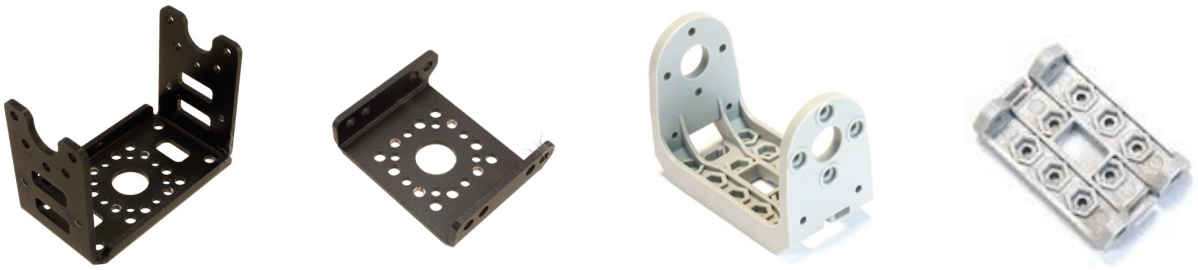
For implementation of the preliminary and final robots, *Bioloid* components allowing the construction of modular robotic structures with *Dynamixel* servo motors were employed.

B.2 Components

In figure 3.7(b) it is visible that each limb is composed by a big set of components. Attached to each of the two motors, the corresponding *Bioloid* levers and fixations of figure B.2 are screwed. Then, the passive sector of each module is composed by a replaceable element (B.3), which can easily be slid on and off. To allow this, two universal connectors which convert *Bioloid* fixations into T-shaped connections are placed between the passive

element and the back frames attached to the motors.

Figures B.2 and B.3 show respectively *Bioloid* and manufactured passive components present in the robotic framework.



(a) FR07-H101K (b) FR07-S101K (c) FP04-F2 (d) FP04-F3

Figure B.2: *Bioloid* mechanical components.

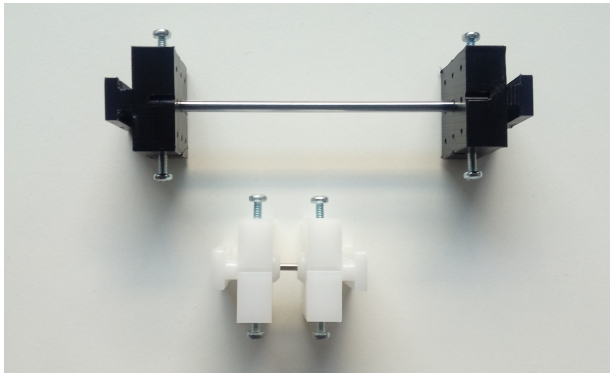
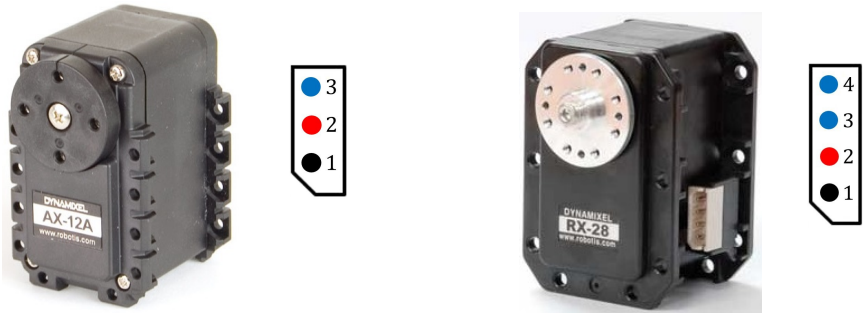


Figure B.3: Passive replaceable components – to be used in future work for morphological and compliance studies.

Actuators

According to the previously exposed, a first version of the robot included 8 *Dynamixel AX-12A* motors (figure B.4(a)) not presenting enough hip torque. To overcome such issue, the hip motors were replaced by *Dynamixel RX-28* ones (figure B.4(b)), which present a stall torque of 3.7 $N \cdot m$ opposing the 1.5 $N \cdot m$ of the first, while having similar dimensions.



(a) AX-12A – 3-pin *TTL* connection (b) RX-28 – 4-pin *RS-485* connection

Figure B.4: Dynamixel motors used as joint actuators.

Table B.1: Motor communication types – 3-pin (TTL) and 4-pin (RS-485)

Pin	**AX-12A (TTL)**	**RX-28 (RS-485)**
1	GND	GND
2	VDD	VDD
3	DATA	DATA+
4	–	DATA−

Such upgrade lead to two main sets of changes. First, as shown in figure B.4(b), the communication protocol of motor *RX-28* differs from the one used for motors *AX-12A* (see table B.1). In addition, the motors run at different voltages, creating the need for a voltage regulator (figure B.5). As visible in the schema 3.7(a), the input voltage ($18V$) directly powers the *RX-28* motors through the *RS-485* cabling, powering the voltage regulator in parallel, which drops to $12V$ the power supply for *AX-12A* motors.

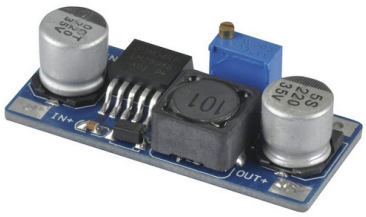
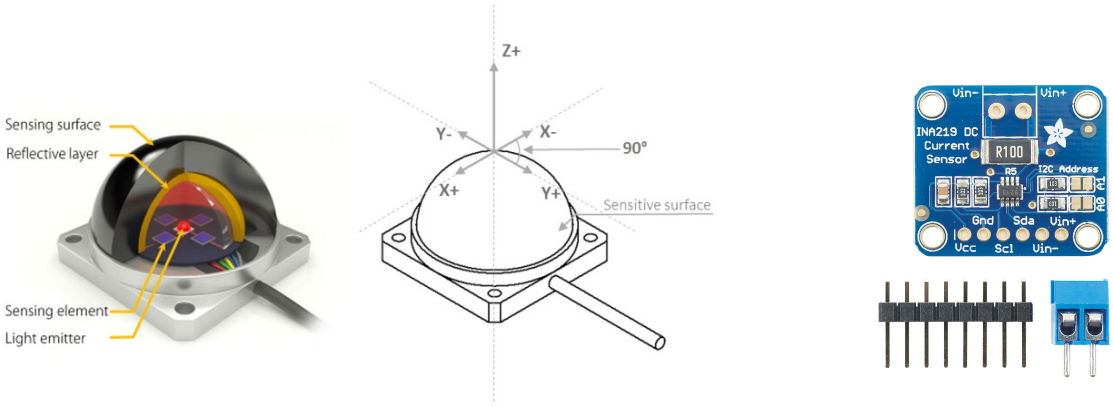


Figure B.5: Voltage Regulator.

Sensors

Regarding sensors, in a first stage only 3D force and current sensors were implemented (figure B.6), having been included an Inertial Measurement Unit (IMU) in the last stage of the project.



(a) Optoforce 3D-force sensor.

(b) Current sensor.

Figure B.6: Sensors used in robotic platform.

Controllers

(a) Odroid XU4. (b) USB2Dynamixel. (c) Arduino nano.

Figure B.7: Electronic components used for control and communication.

B.3 Hardware Characterization

With a view to identify the robots characteristics, posteriorly increasing the understanding of its dynamical behaviour, a few experiments were performed and analyzed. A collection of these tests and discussed results is further presented.

B.3.1 Backlash and Resolution

One of the issues identified after the full assembly of our quadruped robot was the presence of mechanical backlash in the *Dynamixel RX-28* motors, responsible for the hip movements. To identify the magnitude of such backlash the experiment whose results are presented in figure B.8 was performed. Both the *AX-12A* and the *RX-28 Dynamixel* motors present a position encoder resolution of around 0.007 rad however, given the exposed backlash, the position readings from hip motors are expected to present an accuracy within 0.03 rad.

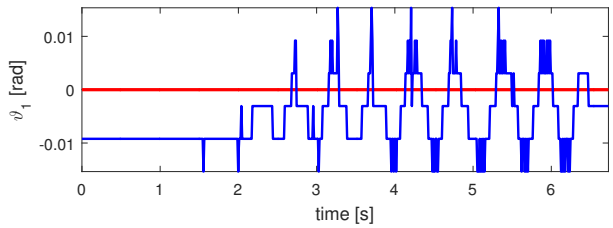
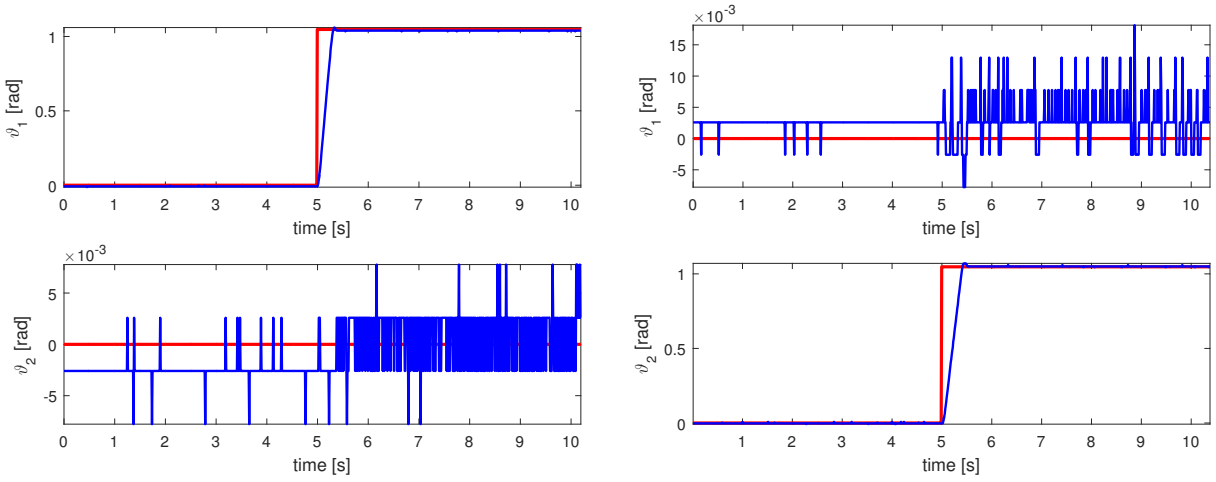


Figure B.8: Mechanical backlash identification – position reference in red and readings in blue.

B.3.2 Step Response

In order to analyze the performance of the servo controllers intrinsically implemented in the *Dynamixel* motors, the response of both hip and knee actuators to a step input of $\pi/3$ radians was collected.



(a) Hip step response.

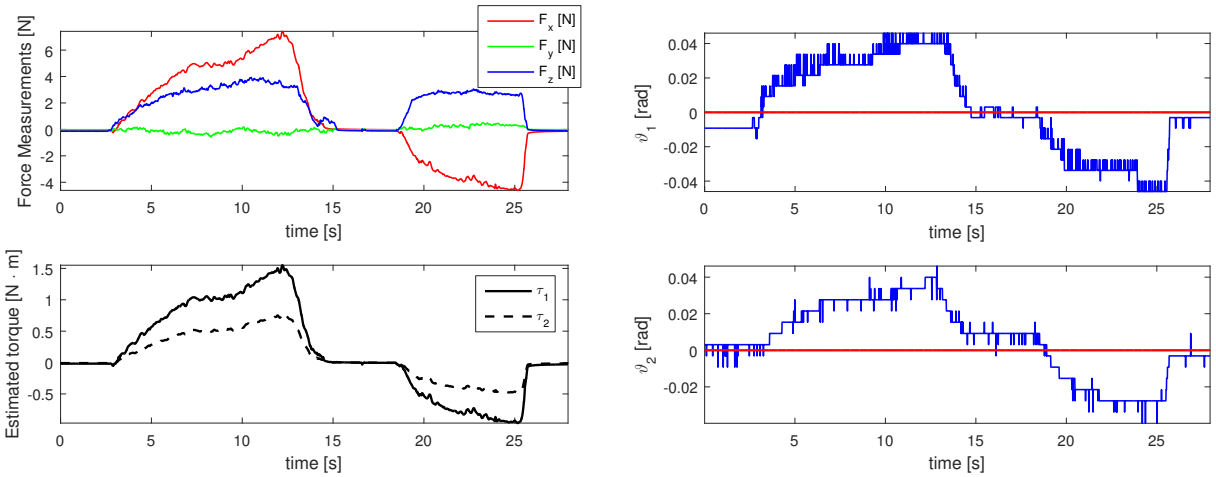
(b) Knee step response.

Figure B.9: Hip and knee motors step response – position reference in red and readings in blue.

In the actuated joints, the absence of steady state error is observed, having a very slight overshoot, and a settling time limited by speed saturation. As introduced before, the presence of backlash and measuring noise is now observed in both hip and knee motors with zero reference. The expected accuracy of the *Dynamixel AX-12A* motors is hence within 0.015 rad.

B.3.3 Motors Stiffness Identification

Adding to the mentioned experiments, a characterization of the impact of torque in the motors position, under a zero reference, is tested. With such purpose, positive and negative torque was induced to the motors through force applied directly in the *Optoforce* sensor present in the end-effector of limb 1. The motors presented a nonlinear level of compliance, showing stiff behaviour for torque levels above the ones applied. The experiment is described in figure B.10.



(a) Forces applied and respective torques.

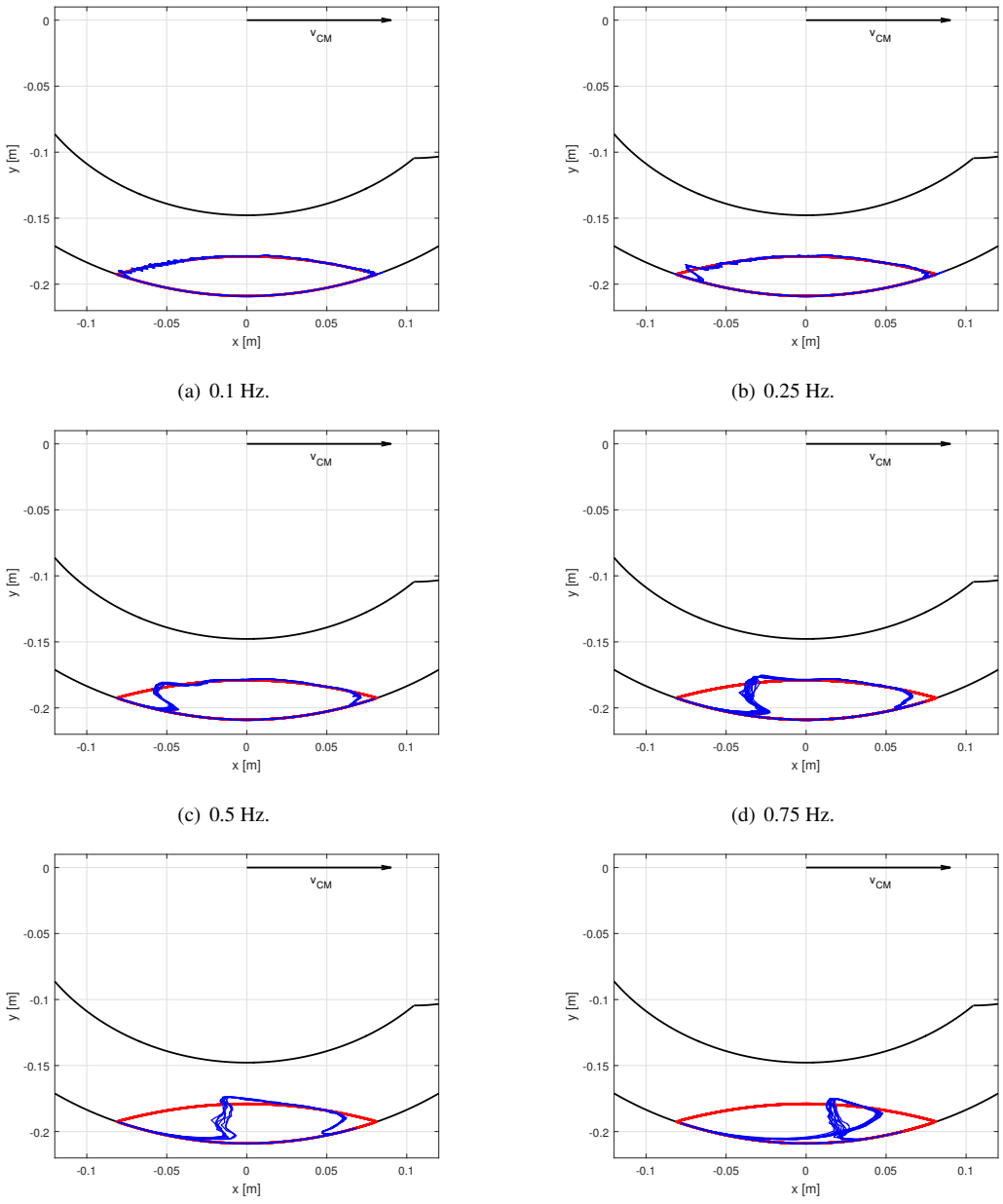
(b) Joint time response.

Figure B.10: Impact of torque application – identification of motor stiffness levels.

This experiment reveals a certain level of compliance present in the motors, which may affect the trajectory following by position errors in joint space of around ± 0.045 radians.

B.3.4 Bandwidth Characterization – No load

Having performed the mentioned static tests, the analysis of the dynamical behaviour of our system was required, namely the bandwidth of our actuators with no load. With such a purpose, a considerably large trajectory was used, being characterized by $h_{st} = 0$ mm, $h_{sw} = 30$ mm and $\theta_{max} = 0.4$ rad. In figure 3.8, the reference following in joint space and the respective trajectory following in operational space are presented for both 0.1 and 1 Hz. Here, in figure B.11, the effect of frequency in cartesian space is more thoroughly analyzed.



(a) 0.1 Hz.

(b) 0.25 Hz.

(c) 0.5 Hz.

(d) 0.75 Hz.

(e) 1 Hz.

(f) 1.5 Hz.

Figure B.11: Negative effect of frequency on trajectory following in cartesian space.

Appendix C

Particle Swarm Optimization Results

Table C.1: Results of energy efficiency PSO for constant frequency.

PSO Characteristics			Top Fitness				Best Particle Parameters						
f_r	IT Nr.	Best IT	ϵ_t	$\overline{v_{CM}}$	S_t	Gait	θ_{max}	h_{st}	h_{sw}	df	ψ_{12}	ψ_{23}	ψ_{34}
0.5	200	200	2.1121	0.0163	6853.4	trot	0.06	≈ 0	0.12	0.8395	-2.2621	2.8152	2.9894
0.5	300	290	3.0532	0.017	5771.1	trot	0.069	≈ 0	≈ 0	0.8455	2.9031	-2.2482	2.8776
0.5	300	300	2.7	0.0221	2859.2	trot	0.084	≈ 0	≈ 0	0.7914	-2.99	-2.2866	2.9785
0.5	400	341	2.1277	0.0138	7342.3	trot	0.05	≈ 0	0.184	0.7535	-3.1042	3.0667	3.1283
0.5	400	360	2.7518	0.0192	507.73	trot	0.073	≈ 0	≈ 0	0.7646	-2.2236	2.978	2.488
0.5	680	660	2.3044	0.0157	2700.5	trot	0.058	≈ 0	0.185	0.7655	2.8902	-2.7892	-2.9408
1	200	192	0.2353	0.2474	4.25	trot	0.418	1.7	12.9	0.731	-2.9936	-2.1185	1.8671
1	200	199	1.4028	0.0369	1104.5	trot	0.067	0.012	0.234	0.7626	-2.8172	2.3386	-2.4382
1	400	366	0.555	0.1411	342.6	trot	0.211	1.1	5.5	0.6219	-2.9386	3.056	3.1191
1	400	391	0.2389	0.4846	3.9652	D-S walk	0.914	6	47.4	0.6465	3.1398	-3.1012	3.1229
1	400	335	1.6683	0.0364	6556.3	trot	0.064	0.01	0.197	0.7387	-2.8655	3.9662	3.0158
1	680	617	2.0222	0.045	2520.4	trot	0.086	≈ 0	≈ 0	0.802	-2.1618	2.9419	2.5112
1.5	300	296	0.2524	0.5382	4.072	D-S walk	0.664	3.1	23.2	0.6406	3.1121	3.1311	-3.1076
1.5	300	266	1.094	0.0572	554.64	trot	0.066	≈ 0	0.198	0.724	3.062	-2.8667	2.8335
1.5	400	359	0.0951	0.4827	8.976	rotary gallop	0.467	0.456	59	0.4232	0.9652	1.3454	1.4183
1.5	400	365	0.1193	0.597	14.9134	rotary gallop	0.544	1.4	57.3	0.4593	1.5107	1.2875	2.5777
1.5	400	366	1.1057	0.0658	4254.2	trot	0.0728	0.013	0.22	0.6938	3.0971	3.081	3.055
1.5	680	623	0.117	0.6176	7.328	rotary gallop	0.541	2.32	57	0.4958	-1.6486	-0.9766	-2.4938

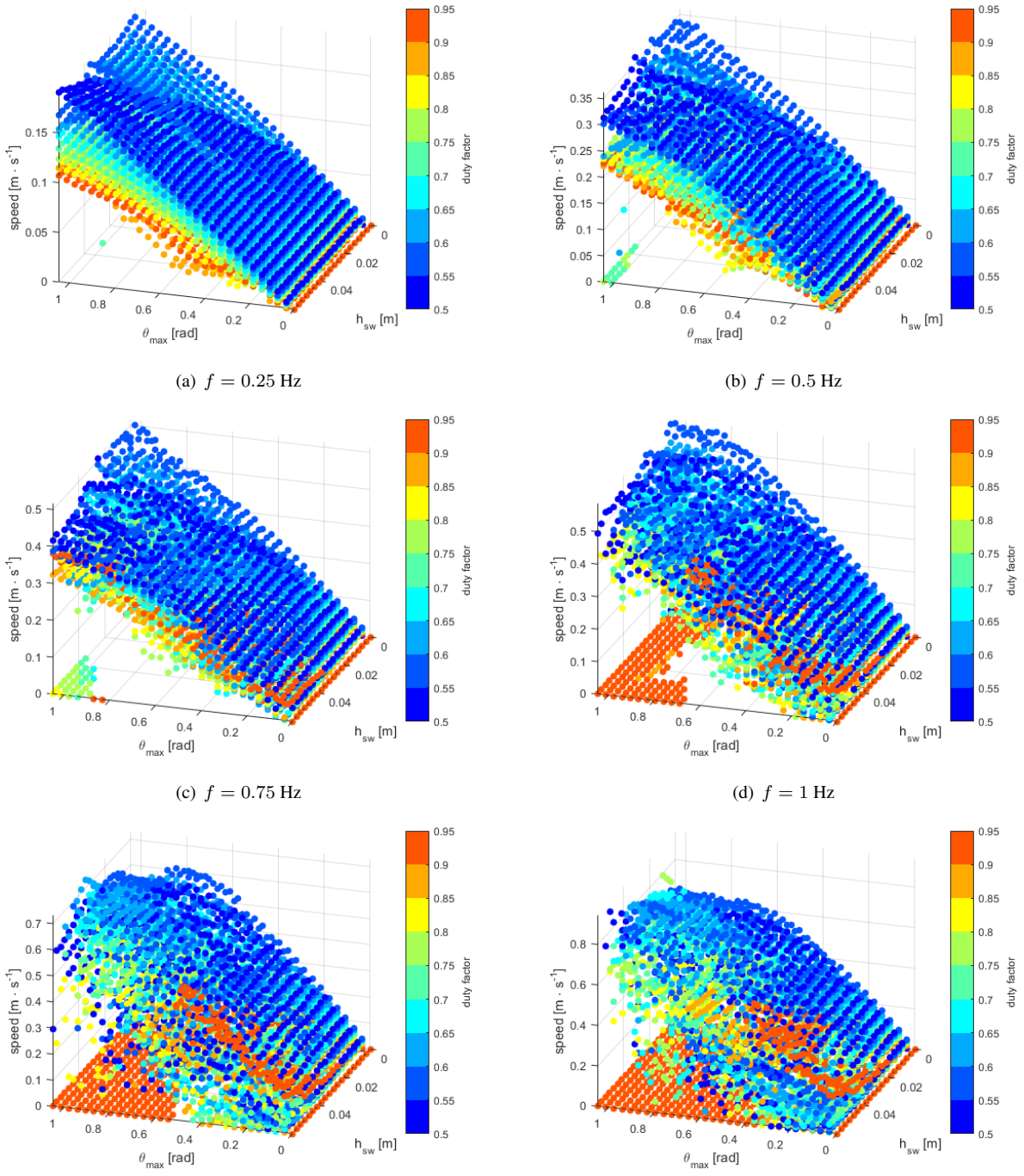
Table C.2: Results of two-stage PSO: energy minimization for desired speed v_d.

PSO Characteristics			Top Fitness					Best Particle Parameters							
v_d	IT Nr.	Best IT	E_t	ϵ_t	$\overline{v_{CM}}$	S_t	Gait	f	θ_{max}	h_{st}	h_{sw}	df	ψ_{12}	ψ_{23}	ψ_{34}
0.25	300	273	196.2	0.079	0.257	11.8	rotary gallop	3.132	0.155	1.34	17.14	0.21	2.362	0.402	2.043
0.25	300	296	53.13	0.295	0.262	33.19	trot	1.399	0.3385	0.68	8.5	0.74	-2.2777	2.3839	-1.996
0.25	400	310	77.18	0.2	0.257	2.54	trot	0.875	0.6023	3.1	26.9	0.77	2.3306	-2.1973	2.1988
0.25	400	385	169.5	0.089	0.252	68.25	rotary gallop	1.113	0.328	0.93	13.4	0.44	1.6142	1.163	1.987
0.25	600	415	159.9	0.094	0.251	5.6792	rotary gallop	2.184	0.2184	1.4	38.5	0.18	3.0207	0.8269	2.4381
0.25	680	317	172.7	0.089	0.255	7.611	pace	2.148	0.2095	1.63	37.7	0.22	-2.571	-1.177	-2.521
0.5	300	299	253.7	0.12	0.51	8.8997	L-S walk	1.895	0.4548	4.4	59.4	0.3	-1.5644	-0.8953	-2.4599
0.5	300	284	339.4	0.089	0.504	1.778	rotary gallop	1.609	0.4824	10.2	47.96	0.39	1.4821	0.9352	-3.0533
0.5	400	373	274.4	0.125	0.57	1.774	pace	2.516	0.334	13.7	36.17	0.35	-2.209	-0.9087	-2.2677
0.5	400	397	117.8	0.261	0.513	2.822	trot	1.491	0.6452	3.07	21.47	0.65	-3.0611	3.0632	-3.0645
0.5	600	583	139.4	0.225	0.522	7.8542	D-S walk	1.16	0.8017	3.2	37.1	0.67	3.1413	2.6758	-3.1093
0.5	700	579	131	0.23	0.501	21.262	D-S walk	1.399	0.6942	4.39	27.73	0.68	-2.8927	2.8913	-3.0235
0.75	300	296	217	0.207	0.75	11.968	D-S walk	1.732	0.7081	3.4	27.6	0.59	-3.1063	2.3567	-3.1308
0.75	400	380	337.9	0.139	0.783	3.1801	rotary gallop	2.411	0.4923	16.7	51.9	0.34	-1.5073	-1.0858	-1.9772
0.75	600	591	390.4	0.12	0.775	3.9524	rotary gallop	2.33	0.4885	7.05	53.5	0.38	1.6322	2.1716	1.4061
0.75	700	410	207.9	0.217	0.753	10.1817	D-S walk	1.578	0.7749	4.76	34.57	0.58	-2.7079	2.399	3.0419
1	300	294	479.3	0.13	1.035	2.2923	bound	2.743	0.5101	6.2	56.7	0.4	-0.6652	-1.95	0.0177
1	400	361	320	0.19	1.01	18.962	D-S run	2.31	0.6998	8.2	26.78	0.53	2.831	2.6956	-3.1034
1	400	266	632.3	0.099	1.046	4.6482	rot. gallop	3.955	0.4244	14.3	30	0.31	1.554	1.8795	1.6095
1	600	593	317.9	0.189	1.002	1.7754	D-S run	2.186	0.7234	6.8	25.4	0.55	3.0952	2.4665	-3.0087
1	700	521	649.4	0.098	1.063	1.2427	bound	3.174	0.4851	11.3	48.4	0.34	0.3377	-2.133	0.037

Appendix D

Systematic Search Results

D.1 Open Loop Systematic Search



(a) $f = 0.25$ Hz

(b) $f = 0.5$ Hz

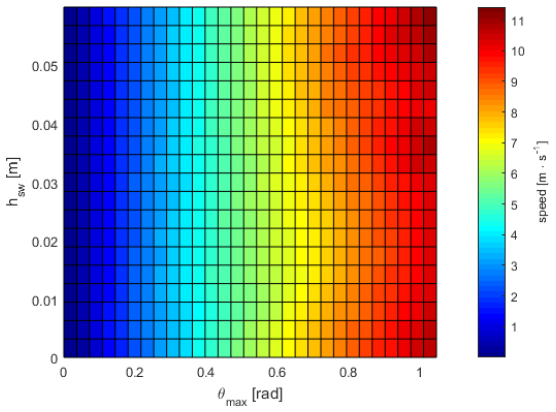
(c) $f = 0.75$ Hz

(d) $f = 1$ Hz

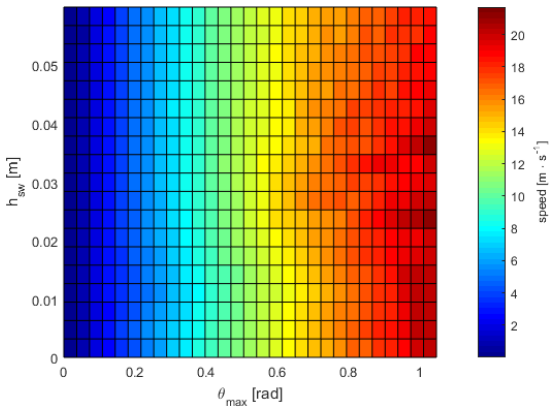
(e) $f = 1.25$ Hz

(f) $f = 1.5$ Hz

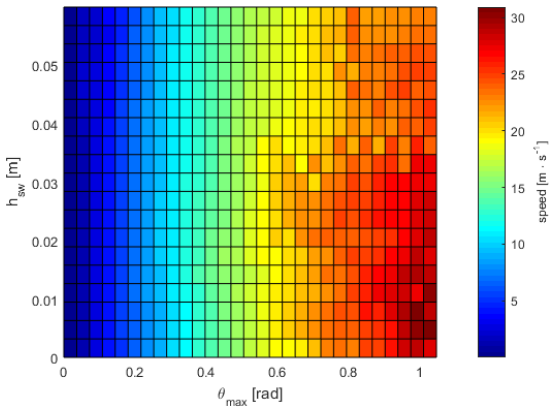
Figure D.1: Average speed analysis: **X** – h_{sw}, **Y** – θ_{max}, **Z** – $\overline{v_{CM}}$, **Color** – df.



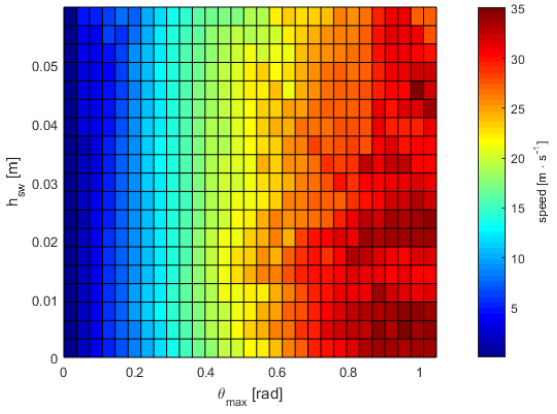
(a) $f = 0.25$ Hz



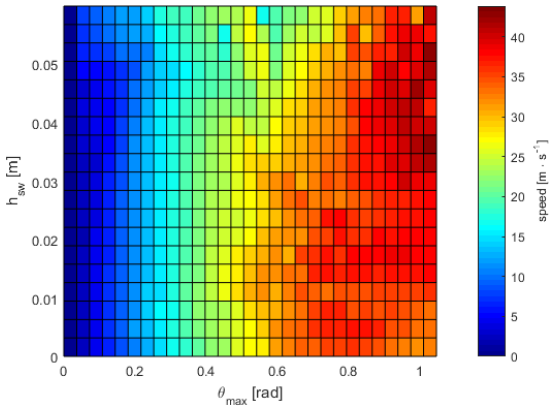
(b) $f = 0.5$ Hz



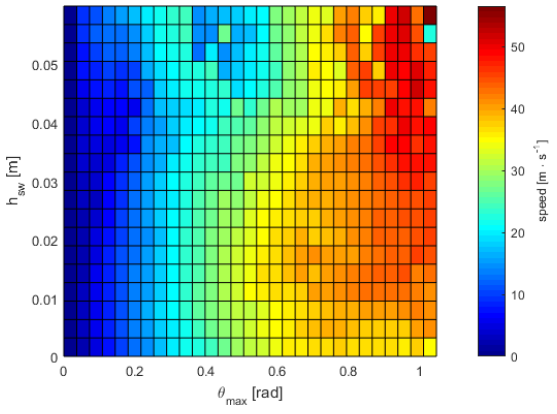
(c) $f = 0.75$ Hz



(d) $f = 1$ Hz

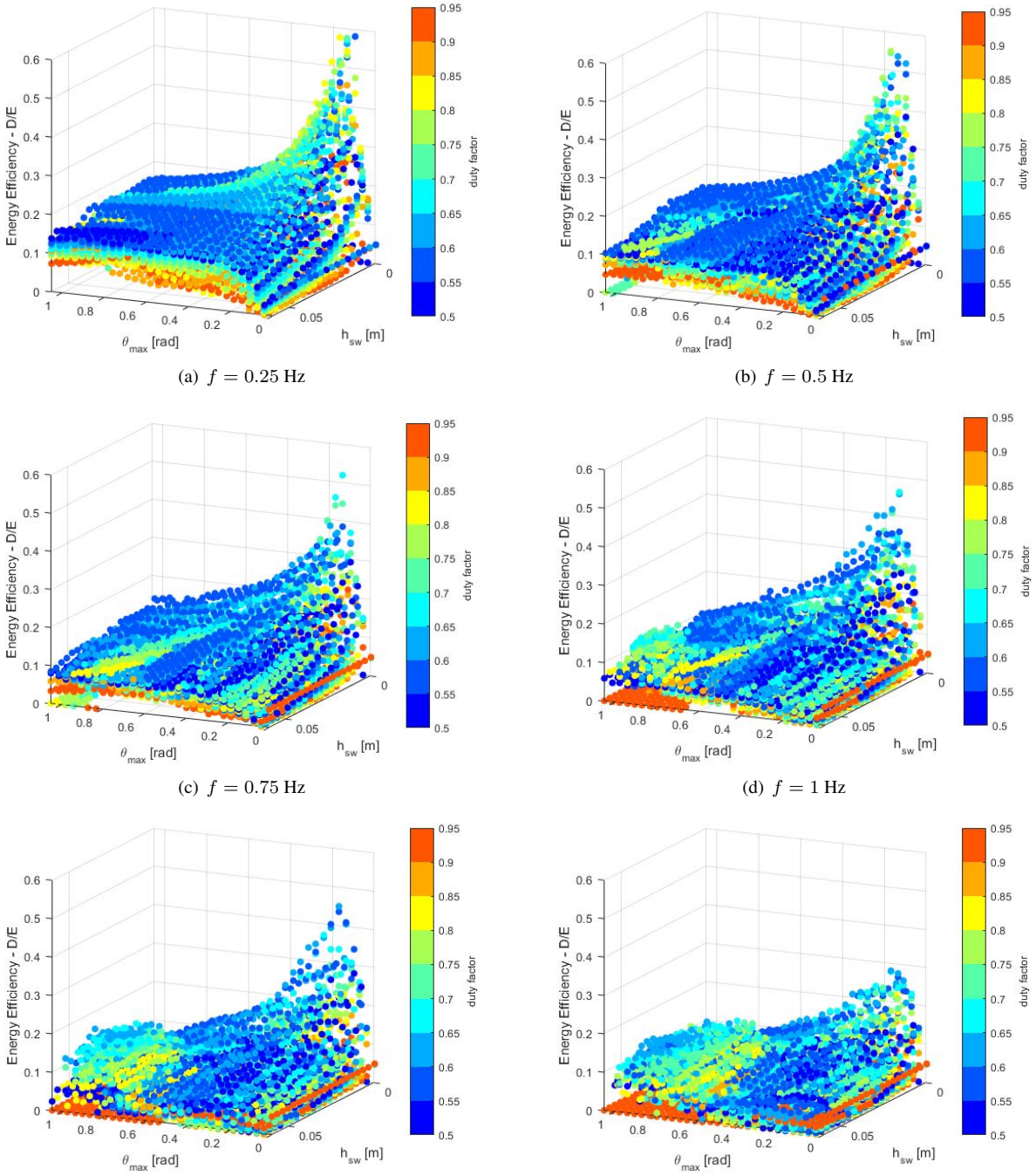


(e) $f = 1.25$ Hz



(f) $f = 1.5$ Hz

Figure D.2: Top average speed of each trajectory set: **X** – θ_{max}, **Y** – h_{sw}, **Color** – $\overline{v_{CM}}$.



(a) $f = 0.25$ Hz

(b) $f = 0.5$ Hz

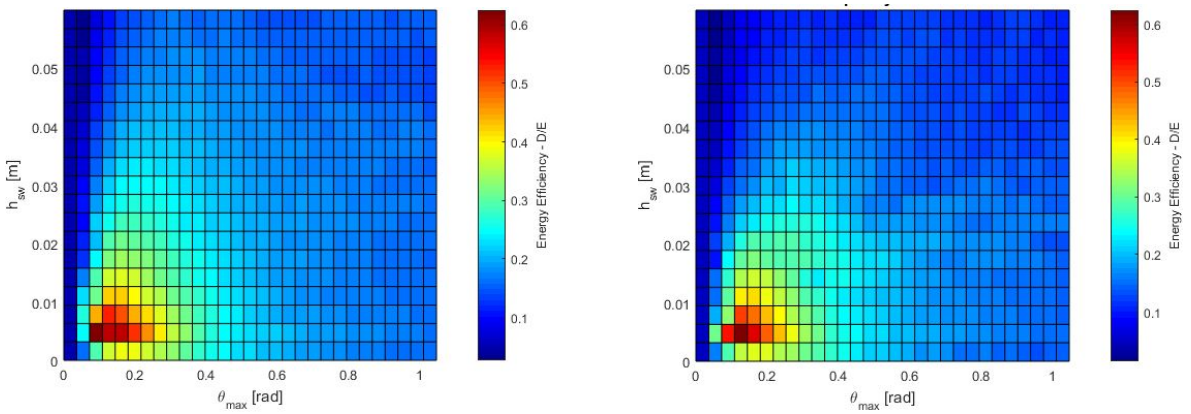
(c) $f = 0.75$ Hz

(d) $f = 1$ Hz

(e) $f = 1.25$ Hz

(f) $f = 1.5$ Hz

Figure D.3: Energy efficiency analysis: $\mathbf{X}$ – h_{sw}, $\mathbf{Y}$ – θ_{max}, $\mathbf{Z}$ – $\epsilon_t[m \cdot J^{-1}]$, **Color** – df.



(a) $f = 0.25$ Hz

(b) $f = 0.5$ Hz

(c) $f = 0.75$ Hz

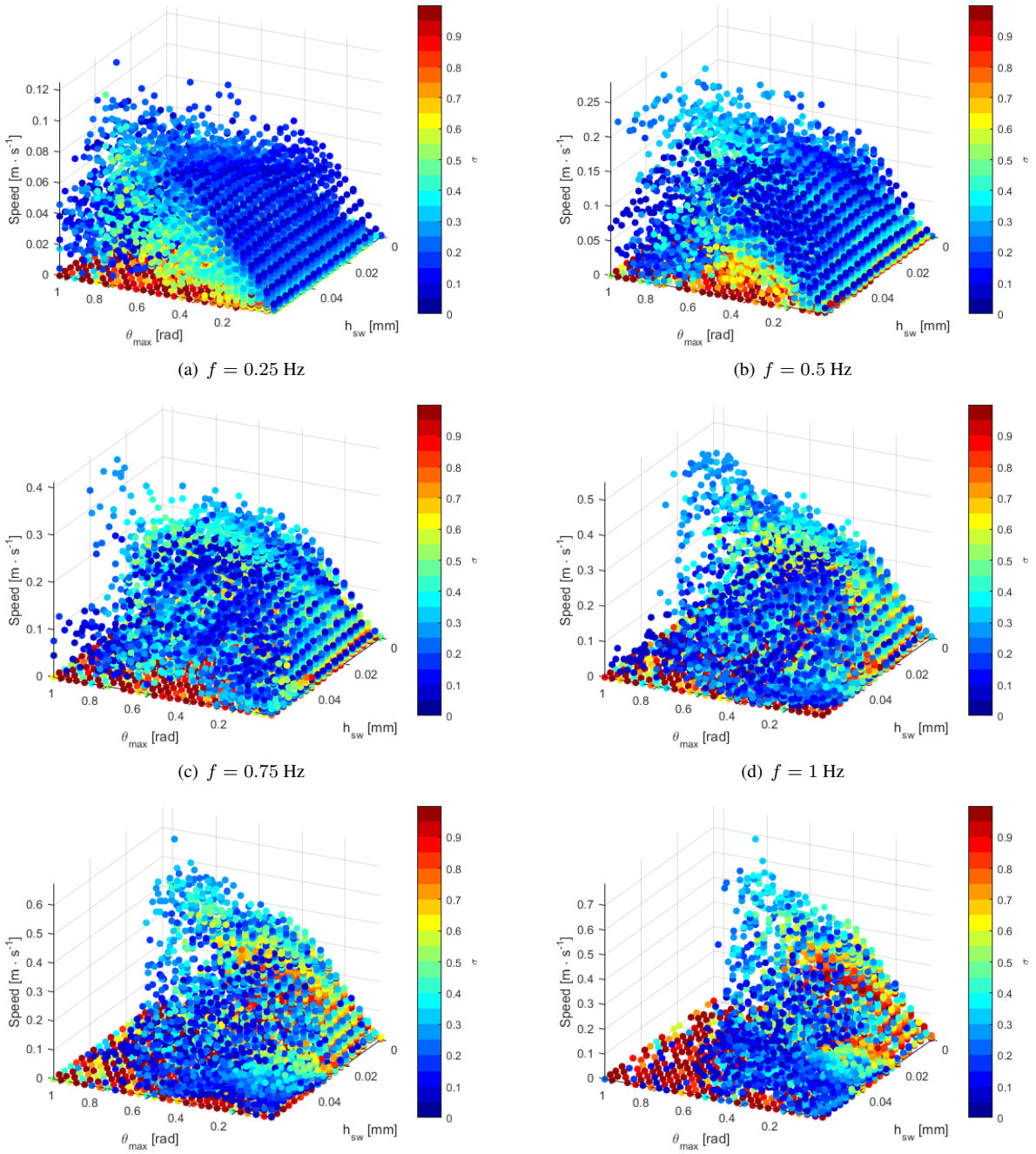
(d) $f = 1$ Hz

(e) $f = 1.25$ Hz

(f) $f = 1.5$ Hz

Figure D.4: Top energy efficiency of each trajectory set: $\mathbf{X}$ – θ_{max}, $\mathbf{Y}$ – h_{sw}, **Color** – $\epsilon_t[m \cdot J^{-1}]$.

D.2 Closed Loop Systematic Search

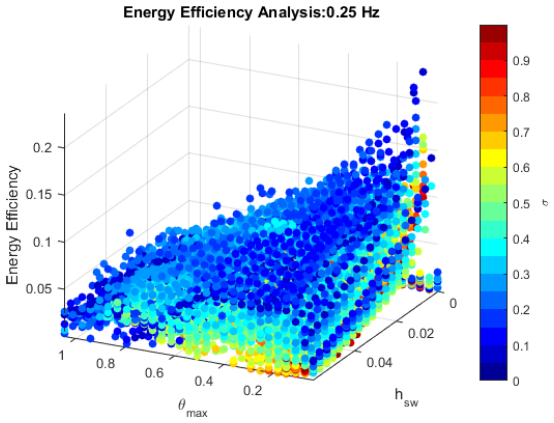


(a) $f = 0.25$ Hz (b) $f = 0.5$ Hz

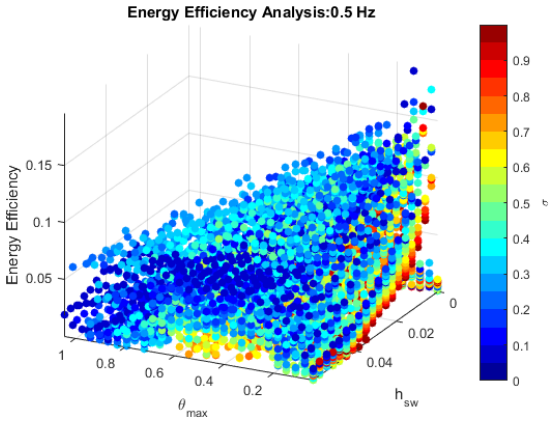
(c) $f = 0.75$ Hz (d) $f = 1$ Hz

(e) $f = 1.25$ Hz (f) $f = 1.5$ Hz

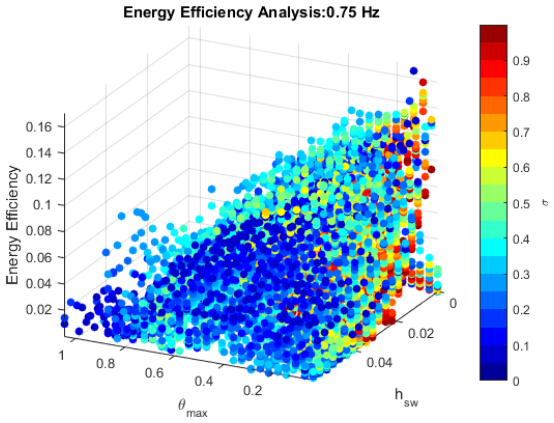
Figure D.5: Average speed analysis: **X** – h_{sw}, **Y** – θ_{max}, **Z** – $\overline{v_{CM}}$, **Color** – σ.



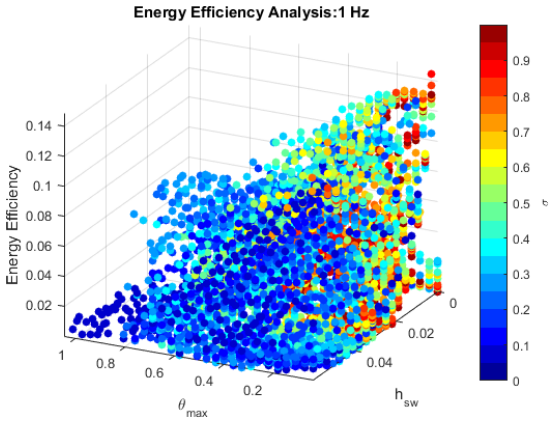
(a) $f = 0.25$ Hz



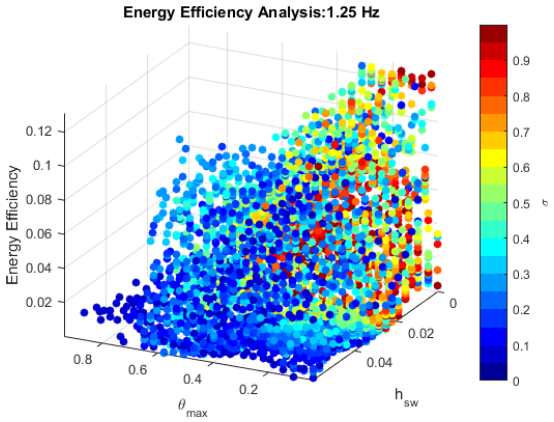
(b) $f = 0.5$ Hz



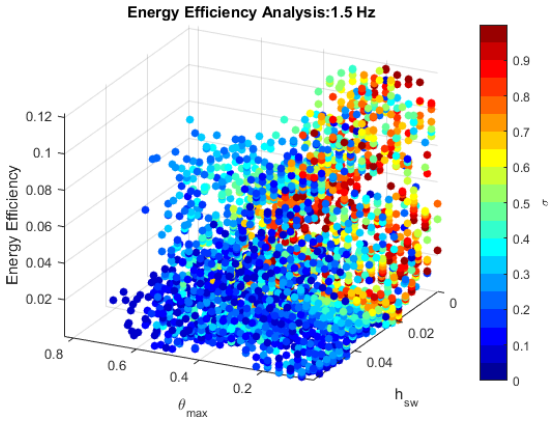
(c) $f = 0.75$ Hz



(d) $f = 1$ Hz



(e) $f = 1.25$ Hz



(f) $f = 1.5$ Hz

Figure D.6: Energy efficiency analysis: **X** – h_{sw}, **Y** – θ_{max}, **Z** – $\epsilon_t[m \cdot J^{-1}]$, **Color** – σ.

Appendix E

Extracting feedback from each module

With the goal of extracting feedback from each module to allow the usage of a local Tegotae approach necessary for crawling movements, compliant elements were studied as possible sensors. The elements under study were cubical with 25 mm (figure E.1) and four feedback candidates were considered:

1. Volume of the compliant element;
2. Volume of half of the compliant element;
3. Variance of distances of vertically coupled vertexes;
4. Mean of distances of vertically coupled vertexes;

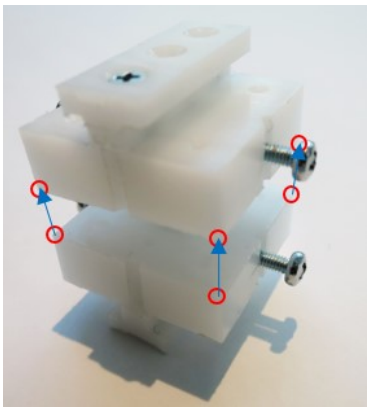


Figure E.1: Compliant element under study for feedback gathering.

This study was composed by a theoretical prediction of how torsion and bending would affect sensor candidates (figure E.2), and by a correlation between ground reaction forces measured in simulation and the performance respective candidates (figure E.3).

To allow the extraction of such correlations and aiming at a real time visualization of the effect of ground reaction force in the behaviour of the 8 compliant elements, 64 GPS sensors were positioned on simulation in the vertexes of these elements. Then, by transformations of the position information in the global reference frame to the local ones, the feedback candidates are computed for every time step.

An example of the visualization method developed on *Matlab* is presented in figure E.4.

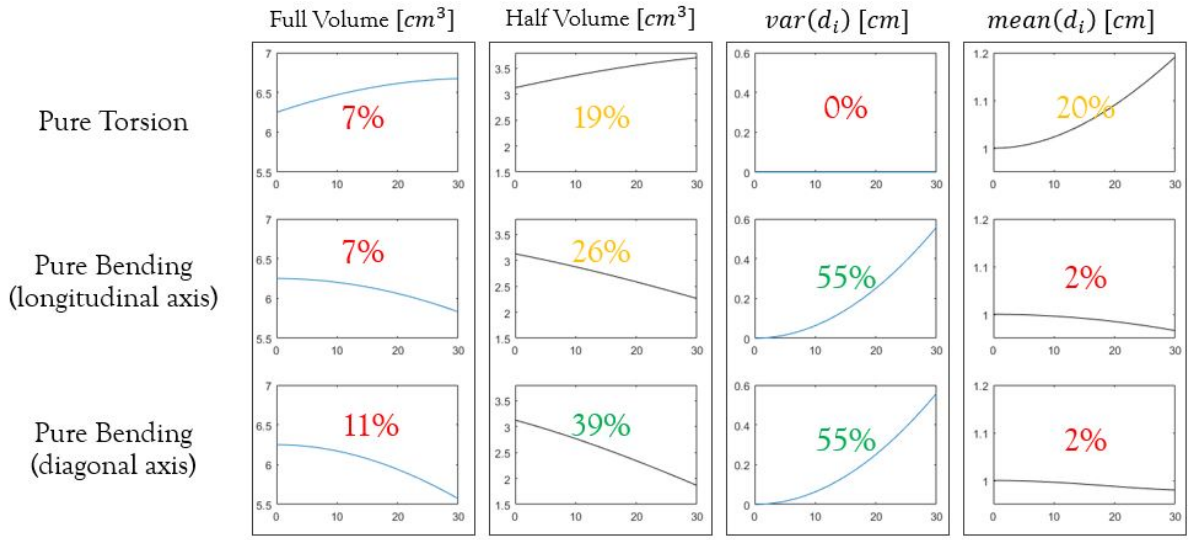


Figure E.2: Theoretical analysis of possible sensory information.

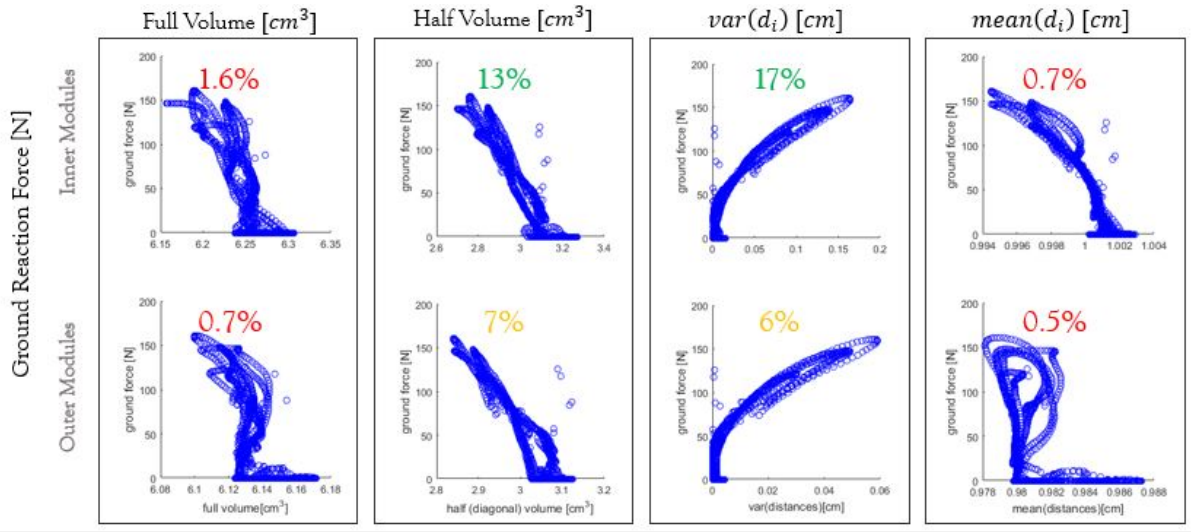
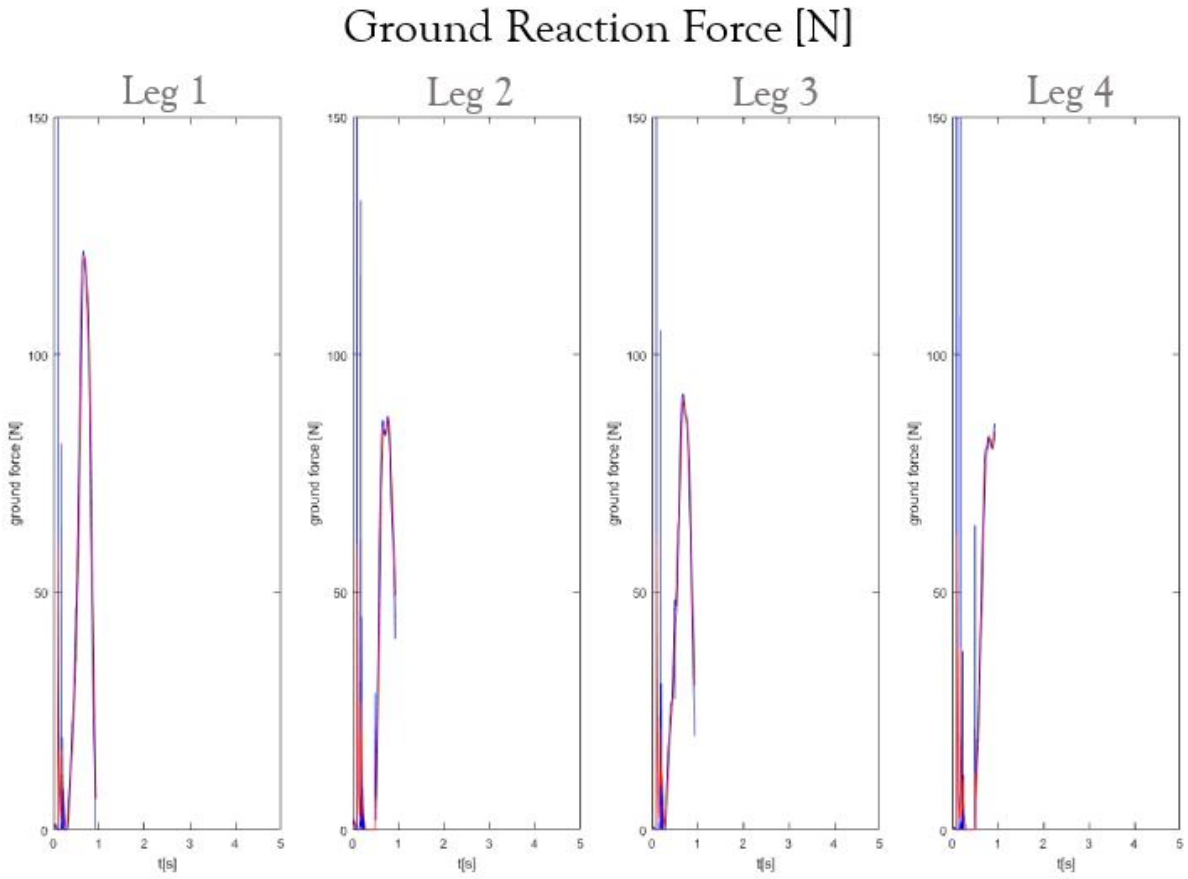
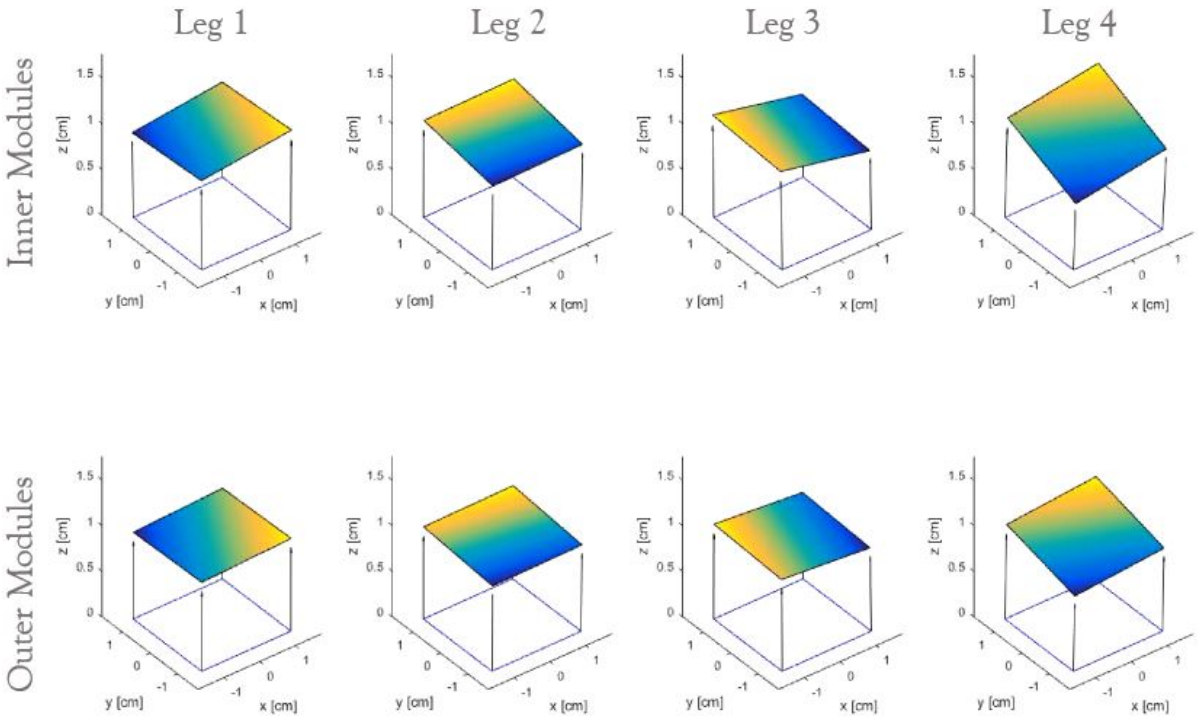


Figure E.3: Correlations extracted in simulation between possible sensory feedback and ground reaction force.



(a) Ground reaction forces.



(b) Compliant elements behaviour.

Figure E.4: Real-time analysis of compliant elements behaviour to be used as sensory feedback from each module.