# POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master of Science in Automation and Control Engineering


POLITECNICO
MILANO 1863

# Introducing a proprio-ceptive feedback in the bio-inspired Tegotae control approach: enhanced learning and energy efficiency


Supervisor: Dercole Fabio

Co-supervisor: Hayashibe Mitsuhiro

Master Thesis dissertation of:

Zamboni Riccardo Matr. 893484



Accademic year 2018-2019

語りえぬことについては，沈黙するしかない。

# Contents

**Abstract** **1**

**Introduction** **3**

**1 CPGs Architectures** **7**

1.1 Introduction . . . . . 7

1.2 Neuro-biological Models of CPGs . . . . . 8

1.3 CPGs for Robot Locomotion . . . . . 9

**2 Neural Oscillator Models** **13**

2.1 Introduction . . . . . 13

2.2 Derivation of Kuramoto Oscillator Model . . . . . 14

2.3 Kuramoto Oscillator Model . . . . . 16

2.4 Couplings and External Fields . . . . . 17

2.5 Neural Oscillator Models: an Extension . . . . . 20

**3 The Group Formalism for Pattern Generation** **23**

3.1 Introduction . . . . . 23

3.2 Group Theoretic Framework . . . . . 24

3.3 Gaits and Symmetries of Periodic Solutions . . . . . 32

3.4 Algebraic Restrictions and Characterization . . . . . 35

3.5 Applicative Examples: Rings of Neurons . . . . . 36

**4 The Tegotae Approach** **41**

4.1 Introduction . . . . . 41

4.2 Mathematical Formulation . . . . . 42

4.3 Stability Analysis . . . . . 44

4.4 Dimensional Extension . . . . . 45

4.5 Tegotae and Learning . . . . . 46

4.6 Tegotae Control Policy: Preliminary Design and Extensions . . . 51

**5 Case Analysis: Monoped** **55**

5.1 Introduction . . . . . 55

5.2 Mathematical Formulation . . . . . 55

5.3 Simulations: Adaptation Transient and Energy Efficiency . . . . . 56

5.4 Simulations: Robustness and Adaptivity . . . . . 58

**6 Case Analysis: Biped** **61**

6.1 Introduction . . . . . 61

6.2 Mathematical Formulation . . . . . 61

6.3 Simulations: Gaits . . . . . 63

6.4 Simulations: Robustness and Adaptivity . . . . . 65

**7 Optimal Control: Theory and Results** **67**

7.1 Introduction . . . . . 67

7.2 Optimal Control: Formulation and Methods . . . . . 67

7.3 Direct Methods: Single Shooting . . . . . 70

7.4 Direct Methods: Multiple Shooting . . . . . 72

7.5 Simulations and Results . . . . . 75

7.5.1 Optimal Controller with Multiple Shooting Only . . . . . 75

7.5.2 Optimal Controller with Multiple Shooting and Single Shooting . . . . . 81

**8 Lyapunov Exponents: Theory and Results** **85**

8.1 Introduction . . . . . 85

8.2 Lyapunov Exponents . . . . . 86

8.3 Filippov Systems . . . . . 88

8.4 Numerical Computation of Lyapunov Exponents . . . . . 92

8.5 Simulations and Results . . . . . 94

**Conclusions** **101**

**Bibliography** **107**

# List of Figures


1.1 Frequency of papers containing CPG-related topics . . . . . 8
1.2 CPG scheme for salamander-robot locomotion . . . . . 11
1.3 CPG scheme with rhythm generator and pattern formation . . . . 12

2.1 Three types of phase coupling function . . . . . 18

3.1 Differentiable manifold and atlas . . . . . 29
3.2 Rings of neurons . . . . . 37
3.3 Four-neurons network . . . . . 39

4.1 Trend in the local oscillator and Tegotae interpretation . . . . . 43
4.2 Phase oscillator stability . . . . . 45
4.3 Block Diagram of the controller for tacit learning . . . . . 47

5.1 Feedback dynamics over the phase $\phi$ . . . . . 57
5.2 Dynamic environment and adaptation process . . . . . 59

6.1 Neuro-mechanical structure of the mono-dimensional hopper with two legs . . . . . 62
6.2 In-phase hopping . . . . . 63
6.3 Anti-phase hopping . . . . . 64
6.4 Gait transition . . . . . 65
6.5 Adaptation to lower step with in-phase hopping . . . . . 66
6.6 Adaptation to lower step with anti-phase hopping . . . . . 66

6.7 Adaptation to higher step with anti-phase hopping . . . . . 66

7.1 Simplified Optimal Control Problem . . . . . 68
7.2 Numerical methods for optimal control . . . . . 70
7.3 Illustration of Single Shooting Method . . . . . 71
7.4 Illustration of Multiple Shooting Method . . . . . 73
7.5 Multiple Shooting only: $m = 0.1$ Kg (Case MS 1) . . . . . 78
7.6 Multiple Shooting only: $Q_1 = 0$ (Case MS 2) . . . . . 78
7.7 Multiple Shooting only: $R_1 = 0$ (Case MS 3) . . . . . 79
7.8 Multiple Shooting only: $L_1 = 0$ (Case MS 4) . . . . . 79
7.9 Multiple Shooting only: adaptation to $m = 0.3$ Kg (Case MS 5) . 80
7.10 Multiple Shooting only: adaptation to $m = 0.6$ Kg (Case MS 6) . 80
7.11 Multiple Shooting and Single Shooting similarities: $m = 0.3$ Kg (Case MS-SS 1) . . . . . 83
7.12 Multiple Shooting and Single Shooting similarities: adaptation to $m = 0.3$ Kg (Case MS-SS 2) . . . . . 83
7.13 Multiple Shooting and Single Shooting differences: $m = 0.4$ Kg (Case MS-SS 3) . . . . . 84
7.14 Multiple Shooting and Single Shooting differences: adaptation to $m = 0.6$ Kg (Case MS-SS 4) . . . . . 84

8.1 A three-dimensional example of the deformation of the sphere into an ellipsoid . . . . . 87
8.2 Projections of derivatives on the normal of the hyper-surface . . . 90
8.3 LEs trend without hysteresis approach . . . . . 95
8.4 Trajectory sample along the forward continuation: in-phase hopping . . . . . 97
8.5 Trajectory sample along the forward continuation: anti-phase hopping . . . . . 97
8.6 Trajectory sample along the forward continuation: gait transition 97

8.7 Trajectory sample along the backward continuation: hysteresis effect 98
8.8 Trajectory sample along the backward continuation: in-phase hopping regeneration . . . . . 98
8.9 Trajectory sample along the backward continuation: gait transition at different $\omega_c$ . . . . . 98
8.10 Hysteresis effect example: gait transition in forward continuation . 99
8.11 Hysteresis effect example: anti-phase hopping in backward continuation . . . . . 99
8.12 Hysteresis effect of the Lyapunov exponents . . . . . 100
8.13 Test Function trend . . . . . 100

# List of Tables

3.1 Gait symmetries . . . . . 33
3.2 Oscillations in three neurons bidirectional ring . . . . . 38

5.1 Simulation results . . . . . 58

7.1 Weights Values for MS Only . . . . . 77
7.2 Weights Values for MS-SS Comparison . . . . . 82

# Abstract

In bio-inspired robotic control, CPGs (Central Pattern Generators) architectures have been extensively adopted to generate periodic patterns in the actuation, due to the interesting properties of nonlinear oscillators which they make use of [1]. Although sensory feedback in CPGs is not necessary for the generation of the patterns, it plays a central role in guaranteeing adaptivity to the environmental conditions [2]. Moreover, feedbacks permit to derive information about the state of the system, but their inclusion do not usually lead to a different formulation of the control policy. This may remain unvaried with respect to the open-loop case. Finally, their inclusion greatly modifies the dynamics of the CPGs architecture [3], often leading to bifurcations and other interesting dynamical phenomena, which remain mostly untreated. A study about a novel proprio-ceptive feedback design is performed, together with the definition of a control policy directly based on it. The additional effects in terms of learning and energy-efficiency are shown, and the occurring dynamical phenomena are also outlined.

# Introduction

## Scope of Interest

The main scope of interest of this work is to embrace the embodied intelligence theory in the specific case of CPGs framework. Specifically, this means assuming that the emergence of any sort of intelligence in embodied and situated agents can not be tackled without the consideration of the strict coupling between the agent and its environment, mediated by the constraints of the agent's own body, perceptual and motor system, and brain [4]. In the CPGs framework, we think that this consists in firstly making use of the mediation of the body as a source of information about the environment, and secondly in designing a control policy directly based on this information, in a reflex-like manner. Then our scope of interest is to study how a smart design based of such assumptions is able to provide further positives such as adaptivity, learning, energy efficiency, and how this is realized de facto, in terms of dynamical phenomena. Following the embodied intelligence framework, a body is mainly an heuristic device for intelligence emergence. Due to this, the mechanical structures we take into account are extremely minimalist, their aim being to guarantee a sufficiently complex behavior for our scopes. We finally hope that, being other researchers convinced or not of the positives of our framework, nonetheless this work will help setting a standard evaluation method in the CPGs architectures framework. This is the very first comprehensive research about the Tegotae approach in both feedback and con-

trol policy design. With the aim of setting the directions of analysis about this approach, we believe that touching these topics from multiple aspects will ease further researches in the future.

## Main Contribution

Our main contribution is a novel design of a feedback law and a related control policy based on the Tegotae approach [5]. We then apply optimal controllers based on shooting methods and Lyapunov exponents computation in order to study the results of the feedback inclusion and the specific control policy. While the analysis methods we performed on these are not fresh-new, their application in this framework is novel as well. Finally, some more theoretical studies are performed, mainly about other learning theories and the relations between CPGs and the group theoretic framework. These did not produced direct results, also due to our simple case studies, yet it have been done to underline how, in a purely theoretic way, more comprehensive studies are also possible, on the basis of these introductory intuitions and similarities.

## Structure of the Thesis

Our study is constructed as follows.
In **Chapter 1** we briefly introduce the main ideas underlying CPGs control, showing one typical and effective example. In **Chapter 2**, we analyze the oscillator models often used in such architectures, with an emphasis on Kuramoto oscillator models and how these well represents neural population dynamics. In **Chapter 3**, we further discuss the formalism able to describe how CPGs networks generate periodic outputs, that is the group theory of [6]. We try to analyze the possible effects of the inclusion of a feedback through such framework. In **Chapter 4**, the main contribution of this work is introduced, that is the construction of a

specific proprio-ceptive feedback law through the so-called Tegotae approach [5], together with a specific control policy to exploit it fruitfully, based on the concept of embodied intelligence. We extensively analyze the approach and we connect it with introductory considerations to learning and energy efficiency [7, 8], to better justify our research path. Then, in **Chapter 5** and **Chapter 6** we apply our feedback to some mechanical systems, i.e. hopping systems, firstly considered in the simplest case of one leg, then extended to the case of two legs. In such circumstances the sensory feedback plays an important role in shaping the rhythmic patterns and keeping CPGs and body movements coordinated. We show the adaptation processes occurring and we show how different gates are obtained, also in relation to the group theoretic formalism. In **Chapter 7**, we compare the analytical solutions for the single-leg case with an optimal controller solution based on direct methods such as Single and Multiple Shooting methods, confirming the intuitions on the energetic efficiency of our control policy. Finally, in **Chapter 8** we analyze the dynamical phenomena occurring via the study of the Lyapunov exponents of the system, in order to better clarify the process occurring in the generation of different gaits.

# Chapter 1

# CPGs Architectures

## 1.1 Introduction

The ability to efficiently move in complex environments is a key property of animals and their survival. This means that many aspects of morphology and central nervous systems in animals have been shaped by constraints related to locomotor skills. By focusing on the control aspects, we tackle in particular the periodic pattern generation by means of central pattern generators (CPGs). Central pattern generators are neural circuits easily found in invertebrate animals that are able to produce rhythmic patterns of neural activity without receiving rhythmic inputs. The term *central* indicates that the sensory feedback from the *peripheral* nervous system is not needed for generating the rhythms [9]. Biological CPGs underlie many fundamental rhythmic activities such as chewing, breathing and digesting, but they are also fundamental building blocks for the locomotor neural circuits. From the control point of view, they present several interesting properties such as distributed control, the ability to deal with redundancies, the presence of fast control loops and the modulation of locomotion by simple control signals. For these properties, CPGs as transferred mathematical models are interesting building blocks for locomotion controllers in robots as well as they are

increasingly used in the robotics community, as illustrated in Figure 1.1.
The interest for CPGs in this work does not rely on the fact that broad architectures will be implemented. Being the original approach presented unknown in terms of its effects on the dynamics, the CPGs models will be limited to minimalist structures. Nonetheless, once demonstrated the effectiveness of the approach, further extensions will be the natural continuation of the current study. For the sake of completeness, the founding theory of CPGs was supposed to be explained, at least in an introductory manner. In this chapter, we briefly introduce CPGs theory and show one of the most renowned applications.

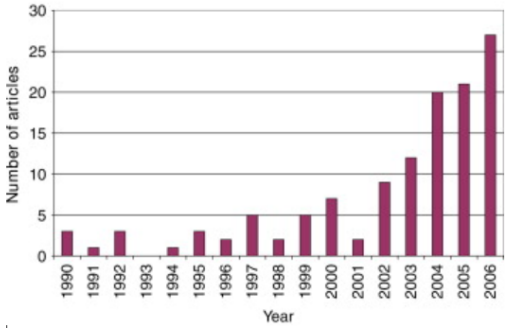


Figure 1.1: Frequency of papers containing CPG-related topics

## 1.2 Neuro-biological Models of CPGs

Depending on the phenomena under study, CPGs models have been designed at several levels of abstraction, from detailed biophysical models, to connectionist models, to abstract systems of coupled oscillators.
Detailed biophysical models are constructed on the basis of Hodgkin-Huxley type of neuron models. They are neuronal models that compute how ion pumps and ion channels influence membrane potentials and the generation of action potentials. These models tenderly investigate the problem of rhythmogenesis, i.e. generation of rhythmic activity, in small circuits of neurons.

Connectionist models uses simplified neuron models such as leaky-integrator neurons or integrate-and-fire neurons. The focus of these models is on how rhythmic activity is generated by network properties and how different oscillatory neural circuits get synchronized via interneuron connections.
Finally, oscillator models are based on coupled nonlinear oscillators and each oscillator represents the activity of a complete oscillatory center and not of a single neuron or of a small circuit. The purpose of these models is not to explain rhythmogenesis, since the oscillatory mechanisms is assumed to exist. Thus, the aim is to study how inter-oscillator couplings and differences of intrinsic frequencies affect the synchronization and the phase lags within the population of oscillatory centers. The motivations for this type of modeling comes from the fact that the dynamics of populations of oscillatory centers depends mainly on the type and topology of couplings rather than on the local mechanisms of rhythm generation [10]. In particular, the importance of the topology means that, from a robotics point of view, there is not much to gain from using too complicated oscillators as building blocks of a CPGs model, which is one more positive aspect of such architectures.

## 1.3 CPGs for Robot Locomotion

The type of CPGs models implemented in robots include connectionist models and systems of coupled oscillators. Virtually all implementations involve sets of coupled differential equations to be numerically integrated on a micro-controller or on a processor. As far as ground locomotion is concerned, quadruped locomotion using CPGs has been extensively tackled. Among other results, it was found that sensory feedback modulating CPGs activity tends to lead to the most stable locomotion in complex terrain, as opposed to feedback that is independent of the CPGs activity [2].
Finally, there are at least five properties for which CPGs models are useful in

terms of locomotion control [1]:

**(i)** The purpose of CPGs models is to exhibit limit cycle behavior, i.e. to produce stable rhythmic patterns. In this case, the system returns to its normal rhythmic behavior after transient perturbations of the state variables. This provides robustness against perturbations.

**(ii)** CPGs are well suited for distributed implementation, which might be interesting for modular robots and re-configurable robots.

**(iii)** CPGs models have a few control parameters (e.g. drive signals) that allow modulation in the locomotion, for instance the speed or the direction or the type of gait. It follows that a properly implemented CPGs model reduces the dimensionality of the control problem so that higher level controllers or learning algorithms do not need to directly produce multidimensional motor commands but only higher level control signals. In biological CPGs, typically, low-level stimulation leads to a slow (low frequency) movements, and high-level stimulation to faster (higher frequency) movements. In the biological framework, stimulations are pulses of electric current. The level of stimulation can be changed by changing either the frequency of the pulses or their current. Related to this, CPGs models typically produce smooth modulations of the produced trajectories even when the control parameters are abruptly changed, since the differential equations typically act as first or second order filters.

**(iv)** CPGs are ideally suited to integrate sensory feedback signals, which can be directly added as coupling terms in the differential equations.

**(v)** CPGs models usually offer a good substrate for learning and optimization algorithms. In particular, it is possible to construct a general framework based on the concept of motion primitives, in which dynamical systems and optimal control can be efficiently unified [11].

One of the most performative results has been developed in [12], whose architecture is reported in Figure 1.2. In this study the bursting properties of an oscillatory center, the oscillations between bursts of motoneuron activity and periods

of rest, are modeled by means of a phase oscillator with controlled amplitude:

$$\dot{\theta}_i = 2\pi\nu_i + \sum_j r_j w_{ij} \sin(\theta_j - \theta_i - \psi_{ij})$$
$$\ddot{r}_i = a_i \left( \frac{a_i}{4}(R_i - r_i) - \dot{r}_i \right)$$
$$x_i = r_i \left(1 + cos(\theta_i)\right)$$

Where $\theta_i$ and $r_i$ are the state variables representing the phase and the amplitude of the $i$th oscillator, $\nu_i$ and $R_i$ determine its intrinsic frequency and amplitude and $a_i$ is a positive constant. Couplings between oscillators are defined by weights $w_{ij}$ and phase biases $\psi_{ij}$. Finally, a positive oscillatory signal represents the burst produced by the center. The output of the oscillators are used to determine the set-points $\phi_i$ (desired angles) provided to proportional-derivative (PD) feedback controllers that control the motor torques (through their voltages $V_i$) given the actual angles $\tilde{\phi}_i$. The CPGs model receives left and right drives $d$ representing descending signals from the central nervous system. The velocity, direction, and type of gait exhibited by the robot can be adjusted by modifying these two signals.

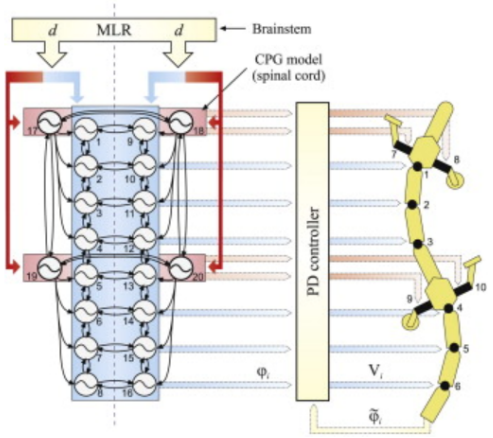


Figure 1.2: CPG scheme for salamander-robot locomotion

Finally, several are the models representing the CPGs architecture. The most

recent ones are more structured, as presented in [13]. The locomotor CPG consists of two levels: a half-center rhythm generator (RG) and a pattern formation (PF) network, with reciprocal inhibitory interactions between antagonist neural populations at each level, as shown in Figure 1.3. The whole structure will then be able to provide a new time frame, not affected by sensory feed-backs, together with the actual patterns, from which the name of these architectures.

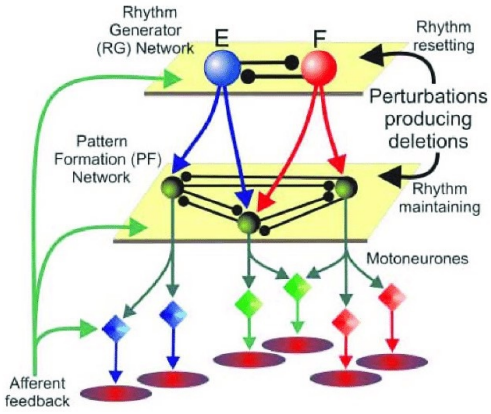


Figure 1.3: CPG scheme with rhythm generator and pattern formation

# Chapter 2

# Neural Oscillator Models

## 2.1 Introduction

Some aspects of neural functions are apparently independent from the neurological details of the individual neuron and of the interconnections in the neural population. Models of neural oscillatory behaviors rely on this anticipation. In particular, many models take into account the spatial information in threshold elements. It is yet possible to recover the phase information, which means the genre of information encoded in the form of specific temporal structures of sequence of neuronal spiking that a real brain should make full use of. In particular, the Kuramoto model [10] captures essential features of synchronization phenomena in large populations of coupled oscillators, but also it is an effective framework in which to study oscillator models, due to its simplicity for mathematical treatment [14].

In this chapter, the Kuramoto model is briefly derived and it is explained how couplings and feedbacks can be included, leading to the final form which will be actually used in the case analysis. Finally, it is shown how Kuramoto model is equivalent to a more common model, the leaky integrate-and-fire model, underlining how the mathematical simplicity does not affect the descriptive capabilities.

## 2.2 Derivation of Kuramoto Oscillator Model

To better understand the nature of Kuramoto oscillator [15] is followed. Suppose to have $N$ differential equations undergoing a *Hopf Bifurcation* to stable oscillations and that these all are linearly coupled. It is possible then to write in normal form coordinates for the $k$th oscillator:

$$\dot{z}_k = (\lambda + i\omega_k)z_k - |z_k|^2 z_k + \frac{\epsilon}{N}\sum_{j=1,j\neq k}^{N} z_j$$

with $\lambda > 0$. It is possible then to convert to polar coordinates by using the transformation $z_k = r_k e^{i\phi_k}$:

$$\begin{aligned}\left(\dot{r}_k + r_k i\dot{\phi}_k\right) e^{i\phi_k} = \\ (\lambda + i\omega_k)r_k e^{i\phi_k} - r_k^3 e^{i\phi_k} + \\ + \frac{\epsilon}{N}\sum_{j=1,j\neq k}^{N} r_j e^{i\phi_j}\end{aligned}$$

And by dividing each term by $e^{i\phi_k}$ the following equation:

$$\begin{aligned}\dot{r}_k + r_k i\dot{\phi}_k = \lambda r_k + i\omega_k \\ r_k - r_k^3 + \\ + \frac{\epsilon}{N}\sum_{j=1,j\neq k}^{N} r_j e^{i(\phi_j - \phi_k)}\end{aligned}$$

Next, the Euler formula for complex numbers is applied:

$$\begin{aligned}\dot{r}_k + r_k i\dot{\phi}_k = \lambda r_k + i\omega_k r_k - r_k^3 + \\ + \frac{\epsilon}{N}\sum_{j=1,j\neq k}^{N} r_j \cos(\phi_j - \phi_k) \\ + i r_j \sin(\phi_j - \phi_k)\end{aligned}$$

This leads to the set of equations:

$$\dot{r}_k = \lambda r_k - r_k^3 + \frac{\epsilon}{N} \sum_{j=1, j\neq k}^{N} r_j \cos(\phi_j - \phi_k)$$
$$\dot{\phi}_k = \omega_k + \frac{\epsilon}{N} \sum_{j=1, j\neq k}^{N} \frac{r_j}{r_k} \sin(\phi_j - \phi_k)$$

Now, due to the week-coupling assumption on $\epsilon$, which requires this quantity to be negligible with respect to the others, the first equation leads to $r_k = \sqrt{\lambda}$, which means that for any $k$th oscillator, $r_k$ remains almost constant. This means that the actual dynamics is determined by the phase equation only:

$$\dot{\phi}_k = \omega_k + \frac{\epsilon}{N} \sum_{j=1, j\neq k}^{N} \sin(\phi_j - \phi_k) \tag{2.2.1}$$

And we obtain the actual form of the Kuramoto oscillator.
On the other hand, following [16], it is possible to construct the Kuramoto model as in order to directly guarantee convergence on limit cycles. In fact, according to

**Theorem 2.2.1** ( Convergence Theorem for Oscillatory Neural Networks)**.** *Consider the oscillatory neural network defined on the monodimensional sphere* $\mathbb{S}^1$*:*

$$\dot{\phi}_i = \omega_i + \sum_{j=1, j\neq k}^{N} H_{ij}(\phi_j - \phi_k)$$

*and suppose that* $\omega_1 = \cdots = \omega_n = \omega$ *and for* $\chi \in \mathbb{S}^1$ *the connection function is such that:*

$$H_{ij}(-\chi) = -H_{ji}(\chi), \quad \forall i, j = 1, \ldots, n \tag{2.2.2}$$

*Then the network dynamics converges to a limit cycle attractor. On the limit cycle, all neurons oscillate with equal frequencies and constant phase deviations. This corresponds to the synchronization of the neural activity.*

When all connection functions in the neural network are equal, namely, $H_{ij}(\chi) = H_{ij}(\chi) = H(\chi)$ for some function $H(\chi)$, then Eq. (2.2.2) implies that $H(\chi)$ is an odd function. Any odd function on $\mathbb{S}^1$ can be represented as a Fourier series:

$$H(\chi) = \sum_{k=i}^{\infty} a_k \sin k\chi$$

The first term of the serie can be considered dominant in first approximation, leading to the standard representation of Eq. (2.2.1), out of proportionality terms. Moreover, guaranteeing the conditions required by the Theorem 2.2.1 it is possible to guarantee the convergence to a limit cycle.

## 2.3 Kuramoto Oscillator Model

The above case can be easily generalized to systems of slightly nonidentical oscillators with phase $\phi_i$ by replacing $\omega$ with $\omega + \delta\omega_{1,2} \triangleq \omega_{1,2}$. In general, $\omega_i$ can follow a certain distribution $g(\omega)$, unimodal and symmetric about $\omega = \Omega$. Thus, a generic network of N similar oscillators with pairwise coupling with a coupling coefficient $K$ reduces to:

$$\dot{\phi}_i = \omega_i + \frac{K}{N} \sum_{j=1, j\neq i}^{N} \sin(\phi_i - \phi_j) \tag{2.3.1}$$

This model and its generalization offer a canonical model for oscillator networks, not limited to neural networks, as for phase synchronization in electrical power distribution networks [17, 18, 19]. Other typologies of oscillators are usually Hopf oscillators, as implemented in [3, 20]. They are not explicitly phase oscillators, even though they can be reconducted to such framework via a proper diffeomorfism. A single oscillator, without any coupling or external fields, is generally in the form of:

$$\dot{x} = (\mu - r^2)x - \omega y$$
$$\dot{y} = (\mu - r^2)y + \omega x$$

with $r = \sqrt{x^2 + y^2}, \mu > 0$. By setting $x = r\cos\phi$, $y = r\sin\phi$ the equivalent oscillator in phase notation is:

$$\dot{r} = (\mu - r^2)r$$
$$\dot{\phi} = \omega$$

It follows that the only difference between this two notations is how the couplings, the feedbacks and the external fields are included in the dynamics.

## 2.4 Couplings and External Fields

Phase coupling functions can be classified into a few basic types and they lead to qualitatively different dynamics of a coupled pair. Assuming a symmetric coupling, that is $\Gamma_{12}(x) = \Gamma_{21}(x) = \Gamma(x)$, it is possible to derive a global equation for the phase difference $\phi \triangleq \phi_1 - \phi_2$ in the form:

$$\dot{\phi} = \Delta\omega + \Gamma_a(\phi)$$

with $\Delta\omega \triangleq \omega_1 - \omega_2$ and $\Gamma_a(\phi)$ being the twice anti-symmetric part of $\Gamma(\phi)$, i.e., $\Gamma_a(\phi) = \Gamma(\phi) - \Gamma(-\phi)$. It follows that $\Gamma_a(x)$ satisfies $\Gamma_a(0) = \Gamma_a(\pm\pi) = 0$. The stability of the system is determined by the frequency difference $\Delta\omega$, which determines the position of the equilibrium point, and by the slope of $\Gamma_a$ in such point, which defines the nature of the equilibrium.
For instance, for identical oscillators ($\Delta\omega = 0$) there are three typical situations, as shown in Figure 2.1. On the basis of the type of coupling, the resulting dynamics may be completely different. Respectively, for type A coupling (Top),

$\Gamma_a'(0) < 0$ and $\Gamma_a'(\pm\pi) > 0$, so that the phase-synchronized state $\phi = 0$ is stable, while the anti-phase state $\phi = \pm\phi$ is unstable. This form of coupling is called *in-phase type*. For type B coupling (Center), $\Gamma_a'(0) > 0$ and $\Gamma_a'(\pm\pi) < 0$, so that the coupling is called *anti-phase type*. For type C coupling (Bottom), both the in-phase and anti-phase states are unstable, while the phase difference is locked to some intermediate value. This type is called *out-of-phase type*.

More complicated situations are possible where multiple values of $\phi$ become stable. When the oscillators are nonidentical, each curve will be shifted upward or downward, so that the stable value of $\phi$ also changes. Clearly, outside a certain range of $\Delta\omega$ no fixed point can exist. Then the oscillators fail to synchronize and the system as a whole exhibits quasi-periodic motion with two independent frequencies, their difference being given by the long-time average of $\dot{\phi}$.

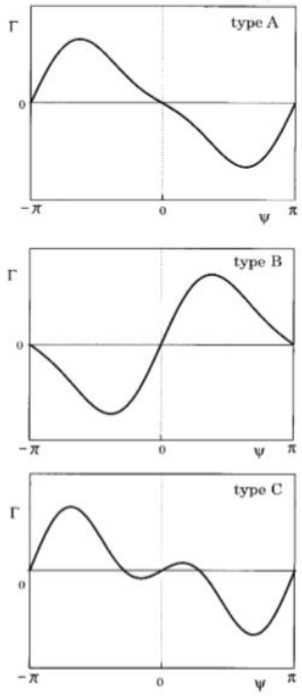


Figure 2.1: Three types of phase coupling function

Moreover, interesting dynamics can be obtained by enriching the coupling term and adding an external field, as presented in [14], which introduces the following equation for a N-dimensional oscillator network:

$$\begin{aligned}\dot{\phi}_i = \omega_i + a\sin(\phi_i) + \\ \frac{1}{N}\sum_{j=1,j\neq i}^{N} a_{ij}\sin(\phi_i - \phi_j)\end{aligned} \tag{2.4.1}$$

In Eq. (2.4.1) the mutual coupling strengths $a_{ij}$ are not assumed to be positive as in the case of Kuramoto [15], and neither the same for all the oscillators, yet they are free to assume both positive and negative values. The oscillators with positive coupling tend to fall in line with neighboring oscillators in favor of the in-phase relationship with them, while the ones with negative coupling drive the oscillators apart to align anti-phase with each other. Since positive and negative communication coexists in biological systems, as for excitatory and inhibitory neurons, this generalization seems to be appropriate.
Finally, the external fields taken into account so far are pinning-forces with an effect of intensity $a$, which mimics the dynamics of excitable limit-cycle oscillator [21]. The main difference will be that, in case of taking into account coupling terms:

$$\begin{aligned}\dot{\phi}_i = \omega_i + a_i\cos(\phi_i) + \\ \frac{1}{N}\sum_{j=1,j\neq i}^{N} a_{ij}\sin(\phi_i - \phi_j)\end{aligned} \tag{2.4.2}$$

This equation is particularly complex due to the pinning-force terms $a_i\cos(\phi_i)$, related to each oscillator and not even monotonic on $[-\pi, \pi]$, property used to guarantee the existence of a Lyapunov function in systems like the ones of Eq. (2.4.1). The final case of analysis can be reconducted to this case.

## 2.5 Neural Oscillator Models: an Extension

So far, phase oscillators have been taken into account to reproduce the behavior of the oscillatory centers in a CPGs architecture. Nonetheless, more adequate models of the oscillatory behavior of the neuronal signal have been developed. Neural oscillators are often modeled with the so-called leaky integrate-and-fire (LIF) neurons. It is possible to show a strong connection between such oscillators and Kuramoto models, as presented in [10].

Let a LIF neuron be described by a variable $u$ ( $0 \leq u \leq 2\pi$ ), and let identify the states $u = 0$ and $u = 2\pi$ just as it was done for the phase variables. The intrinsic dynamics of this neuron is such that u is monotone, increasing with $t$, so that when $u$ reaches the level $u = 2\pi$, it is immediately reset to the zero value. This instant is interpreted as the time of firing. Specifically, $u$ is supposed to obey to the equation:

$$\dot{u} = -u + a \tag{2.5.1}$$

Obviously, the neuron repeats firing if $a > 2\pi$, while it is non-oscillatory but only excitable when $a < 2\pi$. As for coupling with another LIF neuron, the usual assumption is that $u$ changes by a small amount $\epsilon$ each time $t_n$ ($n = 1, 2, \dots$) in which the second neuron fires. The coupling is excitatory if $\epsilon > 0$ and inhibitory otherwise. This dynamical rule is conveniently represented by the term $\epsilon \sum_n \delta(t - t_n)$ added to the right-hand side of Eq. (2.5.1).

In the self-oscillatory regime ($a > 2\pi$), the natural frequency is given by $\omega = 2\pi/|\ln(1 - 2\pi/a)|$. It can be shown that the network of self-oscillatory LIF neurons with weak coupling is equivalent to that of phase oscillators given the form of Eq. (2.3.1) with a coupling function defined over $\psi \in \mathbb{S}^1$:

$$\Gamma(\psi) = \epsilon\omega a^{-1} \exp(\psi/\omega)$$

Note that $\Gamma(\psi)$ has a discontinuity at $\psi = 0$. It is easy to check that for

positive (negative) $\epsilon$, the in-phase (anti-phase) state gives a unique and strongly stable state.

# Chapter 3

# The Group Formalism for Pattern Generation

## 3.1 Introduction

In the CPGs architectures framework, of particular relevance is how the dynamics can be studied merely in terms of the neural topology and interaction [10], that is the magnitude and structure of the network and the nature of the couplings between each element. Here, a substantial contribution was offered by [22, 6, 23, 24] with the introduction of the group formalism for such networks. In particular, group theory has been extensively adopted in order to guarantee a priori the patterns generated by a specific architecture, as in [20], where for patterns we mean gaits in the movements. In fact, this theory provides conditions on the existence of symmetric periodic solutions in networks of coupled dynamical systems. In this chapter, group theory will be briefly introduced and applied to CPGs architectures. Finally, a case leading to the final architecture which will be actually used in the case analysis will be shown. It is important to specify how this framework is essential to understand the standard control policy in CPGs control previously described in chapter 1, but it still needs to be applied to the

extended case of feedback inclusion. On the other end, the results derived consist in an insight to such theory, mainly being a motivation for further expansions of the CPGs architecture to check their effectiveness, rather then being actually applied to the current case study.

## 3.2 Group Theoretic Framework

The interest of the use of group theory is that the design of the network structure can be made independent of the internal dynamics of the single oscillators. The design of a network then relies only on algebraic arguments, which make it easier and salable to more complex networks. Moreover, it is possible to calculate the possible periodic solutions of the network and make sure that only the desired ones are stable. The symmetries present in a network of coupled oscillators induce the existence of periodic solutions possessing the same symmetries, but these symmetries are not limited to that. The first step in understanding the above phenomena is to formalize the meaning of symmetry of a dynamical system, and the meaning of symmetry of a solution, following the steps presented in [6] and using the definitions provided in [25, 26]. The symmetries of a systems of ODE (or indeed PDEs) are specified in terms of a group of transformations of the variables that in some sense preserves the structure of the equations, in particular, its solutions. It follows the need to define:

**Definition 3.2.1.** *A* ***group*** $(G, \star)$ *is a set* $G$ *together with a binary operation* $\star : G \times G \to G$ *satisfying the following conditions:*

1. *Associativity:* $\forall x, y, z \in G,\ (x \star y) \star z = x \star (y \star z)$
2. *Identity element:* $\exists e \in G : \forall g \in G : e \star g = g \star e = g$
3. *Inverse element:* $\forall g \in G, \exists h \in G : g \star h = h \star g = e$

Often, the notation $(G, \star)$ is simply written as $G$, and basically the binary operation is called multiplication, with the symbol $\star$ being omitted. The power of the group definition is that by removing any non-empty subset of these conditions, the resulting class of algebraic objects in largely uninteresting.

**Definition 3.2.2.** *If $(G, \star)$ is a group and $H \subseteq G$ is a subset such that $(H, \star)$ satisfies the Def. (3.2.1), then H is called a* ***subgroup*** *of G, which is written as $H \leq G$*

**Definition 3.2.3.** *For any subset S of a group G, the* ***subgroup generated by S*** *is defined as the smallest subgroup of G containing S. This subgroup is denoted as $\langle S \rangle$. By smallest, it means that if $S \subseteq H \leq G$ then $\langle S \rangle \leq H$.*

**Definition 3.2.4.** *A group is called* ***abelian*** *(or commutative) if $g\, h = h\, g\ \forall g, h \in G$. A group is called* ***cyclic*** *if it is generated by a single element, that is $\exists\ g \in G : G = \langle g \rangle$. In general if $S \subset G, \langle S \rangle = G$, G is said to be generated by S.*

**Definition 3.2.5.** *Let $G = (G, \cdot)$ and $G' = (G', \star)$ be groups, and let $\phi : G \to G'$ be a map between them. Then $\phi$ is called an* ***homomorphism*** *if for every pair of elements $g, h \in G$, it is valid:*

$$\phi(g \cdot h) = \phi(g) \star \phi(h) \tag{3.2.1}$$

*If $\phi$ is a bijective homomorphism it will be called* ***isomorphism****, in which case we say the groups G and G' are* ***isomorphic****, which is written as $G \cong G'$.*

It is important to notice how from a group-theoretic perspective, isomorphic groups are considered the same exact group.

**Definition 3.2.6.** *Given an homomorphism $\phi : G \to G'$, its* ***kernel*** *$ker\phi$ is defined to be the set of $g \in G$ that get mapped to the identity element in G' by $\phi$. Its* ***image*** *$\phi(G) \subset G'$ is just its image as a map on the set G.*
*If $\phi$ is a group homomorphism, then $ker\phi \leq G$, $\phi(G) \leq G'$.*

**Definition 3.2.7.** *Given a subgroup $H \leq G$ and an element $g \in G$, the corresponding* ***left coset*** *of $H$ in $G$, written $gH$, is the set:*

$$gH \triangleq \{gh : h \in H\} \tag{3.2.2}$$

*Notice that since $e \in H$, it will always be true that $g \in gH$, and thus:*

$$\bigcup_{g \in G} gH = G \tag{3.2.3}$$

**Proposition 3.2.8.** *An element $x$ belongs to a coset $gH$ if and only if $g^{-1}x \in H$, this happens if and only if $gH = xH$. Thus, distinct cosets are disjoint. It follows that the distinct cosets of any subgroup for a partition of the whole group. The number of cosets of $H$ in $G$ is then called the* ***index*** *of $H$ in $G$, written $|G : H|$.*

**Definition 3.2.9.** *A subgroup $N \leq G$ is called* ***normal*** *(written $N \trianglelefteq G$), if $\forall\ g \in G$:*

$$gN = Ng \tag{3.2.4}$$

*that is the equality of cosets, which is often expressed equivalently as:*

$$gNg^{-1} = N \tag{3.2.5}$$

This property is important because when right and left cosets coincides, it is actually possible to turn the set of cosets of a subgroup into a group itself.

**Definition 3.2.10.** *If $N$ is a normal subgroup of $G$, then we define the* ***quotient group*** *$G/N$ (read $G$ mod $N$) to be the set of cosets $gN$ of $N$ in $G$ with the group law:*

$$(gN)(hN) = (gh)N \tag{3.2.6}$$

*if $N \leq G$ is not normal, then $G/N$ still denotes the set of cosets of $N$ in $G$, although the above operation is no longer well defined.*

In order to better understand this notation, let us consider $G = \mathbb{Z}$ and $N = n\mathbb{Z}$ being $n\mathbb{Z}$ the subgroup consisting of integers divisible by n. It is possible to form the quotient group $G/N = \mathbb{Z}/n\mathbb{Z}$ of integers modulo n. In this quotient group, some information is lost, about the integers by zooming out to a coarser view: everything divisible by n becomes indistinguishable from the identity element of $\mathbb{Z}$ which is 0, and consequently the only thing known about the other integers is their reminder upon division by n. This concepts of re-conducting any element outside the quotient group to the identity element of the group is a constant factor of this operation.
Finally, of particular interest is to consider *group actions*, which are a formalization of certain external properties often associated to groups. A group action is a formal way of interpreting the manner in which the elements of a group correspond to transformations of some space in a way that preserves the structure of that space. For example, the symmetric group $S_m$ as a group of permutations of the set $\{1, \ldots, m\}$, or the dihedral group $D_{2m}$ of plane symmetries of the regular n-gon:

$$D_{2m} \triangleq (r, s) \tag{3.2.7}$$

where $r$ is a rotation about the n-gon's center by an angle $2\pi/n$ and $s$ is a mirror-flip about a diameter. These concepts of permutations or symmetries are not actually contained in the mathematical structure of a group, but are rather their actions on a specific set.

**Definition 3.2.11.** *Given a set X, let*

$$Sym(X) \triangleq (\{f : X \to X : f \ bijective\}, \circ) = S_{|X|} \tag{3.2.8}$$

*then an* ***action*** *of a group $G$ on $X$ is a homomorphism*

$$\rho : G \to Sym(X) \tag{3.2.9}$$

It is the homomorphism $\rho$ that encodes all the geometric and combinatorial information we associate with certain groups. For example, once taken $D_{10}$ and labeled the five vertices of the pentagon $1, 2, 3, 4, 5$ consecutively, $X = 1, 2, 3, 4, 5$. Since the rotation $r$ just shifts the vertex $i$ to $(i+1)/\text{mod}5$, it is possible to write $\rho(r)$ as the 5-cycle (1 2 3 4 5) $\in S_5$. The flip symmetry $s$ swaps vertices 2 and 5, and 3 and 4 while leaving 1 fixed, thus $\rho(s) = (1)(25)(34) \in S_5$. In this case, the cycle notation for permutations has to be interpreted like this: each number is mapped to the number on its right, except the last one before the parenthesis, which is sent to the first umber in that same set of parenthesis. So that, for example $(1)(25)(34)$ is the permutation which sends $1 \to 1, 2 \to 5, 5 \to 2, 3 \to 4, 4 \to 3$.

The concept of a Lie group arises naturally by merging the algebraic notation of a group with the geometric notation of a differentiable manifold. This leads to a significant extension of linear algebra and analytic geometry. First,

**Definition 3.2.12.** *A n-dimensional* ***(differentiable) manifold*** $M^n$ *is a Hausdorff topological space with countable basis, together with a maximal differentiable atlas. This atlas consists of a family of charts* $h_\lambda : U_\lambda \to U'_\lambda \subset \mathbb{R}^n$*, where the domains of the charts* $\{U_\lambda\}$ *form an open cover of* $M^n$*, the* $U'_\lambda$ *are open in* $\mathbb{R}^n$*, the local coordinated* $h_\lambda$ *are homeomorphisms and every change of coordinated* $h_{\lambda\mu} = h_\mu \circ h_\lambda^{-1}$ *is differentiable on its domain of definition.*

The atlas is maximal in the sense that it cannot be enlarged to another differentiable atlas by adding more charts, so any chart which could be added to the atlas in a consistent way is already in the atlas.
A continuous map $f : M \to N$ of differentiable manifold is called differentiable if, after locally composing with the charts of M and N, it induces a differentiable map of open subsets of Euclidean spaces. These concepts are visualized in Fig. (3.1).

**Definition 3.2.13.** *A* ***Lie group*** *is a differentiable manifold G which is also a*

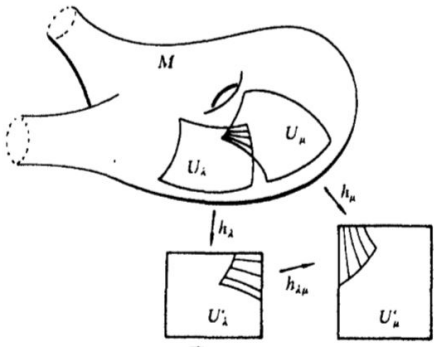


Figure 3.1: Differentiable manifold and atlas

*group such that the group multiplication:*

$$\mu : G \times G \to G \tag{3.2.10}$$

*is a differentiable map, where here it means directly* $\mu \in C^\infty$

Now, a finite group is a group, of which the underlying set contains a finite number of elements, while a continuous group is a group having continuous group operations. A continuous group is necessarily infinite.
The main compact finite Lie groups that shall be considered here are:

- $D_{2m}$, the dihedral group of order $2m$, that is the rotations and reflections in the plane that preserve a regular m-gon.
- $Z_m$, the cyclic group of order m, with rotations only.
- $S_m$, the symmetric group consisting of all permutations on m symbols, with order m!.

The main compact continuous Lie groups are:

- $\mathbb{S}^1$, the circle group.
- $SO(2)$, the special orthogonal group in $\mathbb{R}^2$.

- $O(n)$, the orthogonal group in $\mathbb{R}^n$, consisting of all $n \times n$ matrices A such that $AA^T = I$.

- $SO(n)$, the special orthogonal group, where in addition $\det A = 1$.

**Definition 3.2.14.** *Let $\Gamma$ be a Lie group and $V$ a vector space. An **action** of $\Gamma$ on $V$ is a smooth homomorphism $\rho : \Gamma \to GL(V)$. To simplify the notation, it is usually written $\gamma v = \rho(\gamma)(v)$. Then the condition that $\rho$ is an homomorphism implies the following identities:*

1. $(\gamma\delta)v = \gamma(\delta v)$

2. $1v = v$ *where 1 is the identity element of* $\Gamma$

The group $\rho(\Gamma) = \tilde{\Gamma} \subseteq GL(V)$ is a subgroup. $GL(V)$ is the general linear group of the vector space V, that is the set of invertible matrices of dimension equal to the dimension of V together with the operation of ordinary matrix multiplication.

Without loss of generality we may assume that a compact Lie group acts by orthogonal transformations, and this is a useful simplification:

**Theorem 3.2.15.** *Every compact Lie group $\Gamma$ acting on $\mathbb{R}^n$ may be identified with a subgroup of $O(n)$, the group of orthogonal matrices together with the operation of ordinary matrix multiplication.*

Now, suppose to have a parametrized system of ODEs:

$$\dot{x} = f(x) \tag{3.2.11}$$

where $f : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$, $f \in C^\infty$.

**Definition 3.2.16.** *The group element $\gamma \in O(n)$ is a symmetry of Eq. (3.2.11) if for every solution $x(t)$ of such equation, $\gamma x(t)$ is also a solution.*

**Definition 3.2.17.** *Since the arbitrarity to initial conditions and the constant nature of $\gamma$ along the trajectory, $f$ is said to be $\Gamma$-**equivariant** if, for $\Gamma$ acting on $\mathbb{R}^n$:*

$$\forall \gamma \in \Gamma, x \in \mathbb{R}^n \ f(\gamma x) = \gamma f(x) \tag{3.2.12}$$

Suppose that x is a solution of Eq. (3.2.11). Then it is possible to express the symmetries of this solution of the ODE system as opposed to the symmetries of the ODE system itself, as follows:

**Definition 3.2.18.** *Let $v \in \mathbb{R}^n$. The **isotropy subgroup** of $v$ is:*

$$\Sigma_v = \{\gamma \in \Gamma : \gamma v = v\} \tag{3.2.13}$$

Generally, the isotropy subgroup of a solution of an equivariant system of ODEs provides useful information about the form of that solution. More importantly,

**Definition 3.2.19.** *Let $\Sigma \leq \Gamma$ be a subgroup. Then the **fixed-point subspace** of $\Sigma$ is :*

$$Fix(\Sigma) = \{v \in \mathbb{R}^n : \sigma v = v, \ \forall \in \Sigma\} \tag{3.2.14}$$

**Theorem 3.2.20.** *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be $\Gamma$-equivariant and let $\Sigma \leq \Gamma$ be a subgroup. Then:*

$$f(Fix(\Sigma)) \subseteq Fix(\Sigma) \tag{3.2.15}$$

An important consequence is that:

**Proposition 3.2.21.** *Let $x(t) \in \mathbb{R}^n$ be a solution trajectory of an equivariant ODE. Then:*

$$\Sigma_{x(t)} = \Sigma_{x(0)}, \ \forall t \in \mathbb{R} \tag{3.2.16}$$

*That is, isotropy subgroups remain constant along trajectories.*

## 3.3 Gaits and Symmetries of Periodic Solutions

These concepts are useful to study steady-state bifurcations, as expressed in [6]. Yet, our main objective is to analyze time periodicity and spatio-temporal symmetries, and in particular which spatio-temporal symmetries are possible in a $\Gamma$-equivariant system of ODEs, given a group $\Gamma$ acting on $\mathbb{R}^n$.
To better understand these concepts, the gaits of quadrupeds provide an excellent introduction to the world of space-time symmetries. In fact, there is one feature in common to all gaits: they are repetitive, that is, they are time periodic. In the pace, trot, and bound gaits the animal's legs can be divided into two pairs, the legs in each pair moving in synchrony, while legs in different pairs moving with half-period phase shift. The two pairs in a bound consist of the fore legs and the hind legs; the two pairs in a pace consist of the left legs and the right legs; and the two pairs in a trot consist of the two diagonal pairs of legs.
Each of these gaits can be distinguished by symmetry in the following sense. Spatio-temporal symmetries are permutations of the legs coupled with time shifts. So interchanging the two fore legs and the two hind legs of a bounding animal does not change the gait, while interchanging the two left legs and the two right legs leads to a half-period phase shift. Spatio-temporal symmetries of many gaits are listed in Table 3.1, where 1 stands for the left hind leg, 2 for the right hind leg, 3 for the left fore leg, 4 for the right fore leg.

Now, taken into account CPGs, these systems are described by systems of ODEs. It follows that such a system needs to have some symmetries in order to guarantee some spatio-temporal symmetric solutions, i.e., some gaits.

Suppose to have a system similar to Eq. (3.2.11) with a symmetry group $\Gamma$. Suppose then that $x(t)$ is a T-periodic solution and that $\gamma \in \Gamma$. It follows that $\gamma x(t)$ is another T-periodic solution of Eq. (3.2.11). Should the two trajectory intersect, then the common point of intersection would be the same initial point

Table 3.1: Gait symmetries

| Gait | Symmetries (leg permutation, phase shift) | | |
|---|---|---|---|
| pace | $((1\ 3)(2\ 4), 0)$ | $((1\ 2)(3\ 4), \frac{1}{2})$ | $((1\ 4)(2\ 3), \frac{1}{2})$ |
| trot | $((1\ 3)(2\ 4), \frac{1}{2})$ | $((1\ 2)(3\ 4), \frac{1}{2})$ | $((1\ 4)(2\ 3), 0)$ |
| bound | $((1\ 3)(2\ 4), \frac{1}{2})$ | $((1\ 2)(3\ 4), 0)$ | $((1\ 4)(2\ 3), \frac{1}{2})$ |
| walk | $((1\ 3\ 2\ 4), \frac{1}{4})$ | $((1\ 2)(3\ 4), \frac{1}{2})$ | $((1\ 4\ 2\ 3), \frac{3}{4})$ |
| rotary gallop | $((1\ 4)(2\ 3), \frac{1}{2})$ | | |
| transverse gallop | $((1\ 3)(2\ 4), \frac{1}{2})$ | | |

for the two solutions. Uniqueness of solutions then implies that the trajectories of $\gamma x(t)$ and $x(t)$ must be identical. So either the two trajectories are identical of they do not intersect at all. Suppose that two trajectories are identical. Then this implies that there exists $\theta \in \mathbb{S}^1 = [0, T]$ such that:

$$\gamma x(t + \theta) = x(t) \tag{3.3.1}$$

$(\gamma, \theta) \in \Gamma \times \mathbb{S}^1$ is called a *spatio-temporal symmetry* of the solution $x(t)$. A spatio-temporal symmetry of $x(t)$ for which $\theta = 0$ is a *spatial symmetry*, since it fixes the point $x(t)$ at every moment of time. The group of all spatio-temporal symmetries of $x(t)$ is denoted as:

$$\Sigma_{x(t)} \subseteq \Gamma \times \mathbb{S}^1 \tag{3.3.2}$$

It follows that the symmetry group $\Sigma_{x(t)}$ can be identified with a pair of subgroups H and K of $\Gamma$ and a homomorphism $\Theta : H \to \mathbb{S}^1$ with kernel K. Define:

$$\begin{aligned} K &= \{\gamma \in \Gamma \ : \ \gamma x(t) = x(t)\ \forall t\} \\ H &= \{\gamma \in \Gamma \ : \ \gamma\{x(t)\} = \{x(t)\}\} \end{aligned} \tag{3.3.3}$$

The subgroup $K \subseteq \Sigma_{x(t)}$ is the group of spatial symmetries and the subgroup $H$ consists of the symmetries that preserve the trajectory of $x(t)$, in short, the

spatio-temporal symmetries. Indeed, the groups $H \subseteq \Gamma$ and $\Sigma_{x(t)} \subseteq \Gamma \times \mathbb{S}^1$ are isomorphic; the isomorphism is just the restriction to $\Sigma_{x(t)}$ of the projection of $\Gamma \times \mathbb{S}^1$ onto $\Gamma$. The group $\Sigma_{x(t)}$ can be written as:

$$\Sigma_{x(t)} = \{(h, \Theta(h)) \ : \ h \in H\} \tag{3.3.4}$$

To consider an example, suppose the model of the CPGs of a quadruped to be a system of four identical subsystems of ODEs, coupled in some manner. Suppose in addition that the system has periodic solutions corresponding to walk, referring to Table 3.1. For example, if $((1\ 2\ 3\ 4), \frac{1}{4})$ is a spatio-temporal symmetry of the 1-periodic solution $x(t)$, it follows that:

$$\begin{aligned} x_2(t) &= x_1\left(t + \frac{1}{2}\right) \\ x_3(t) &= x_1\left(t + \frac{1}{4}\right) \\ x_4(t) &= x_1\left(t + \frac{3}{4}\right) \end{aligned}$$

and the solution sends the pattern of a walk to the four legs. As a further example of application, in a quadruped the trot gait is a gait in which the diagonal legs move in synchrony and half a period out of phase with the contralateral legs. Thus, using permutation notation, if the limbs are numbered as in Table 3.1, there exists one spatial symmetry $((14)(23), 0)$, where the 0 means no phase shift after the permutation, and 2 additional spatio-temporal symmetries $((12)(34), \frac{1}{2})$ and $((13)(24), \frac{1}{2})$. The symmetry group generated by these symmetries is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$. It can be easily calculated that the pace and bound gaits have the same group of symmetries and that generically in a system of 4 coupled oscillators having the same symmetry all the 3 gaits will co-exist, since the subgroups of the network are $\{((12)(34)), I\}$, $\{((13)(24)), I\}$, $\{((14)(23)), I\}$.

## 3.4 Algebraic Restrictions and Characterization

There are algebraic restrictions on the pair H and K defined in Eq. (3.3.3) in order for them to correspond to symmetries of a periodic solution. In particular:

**Lemma 1** (Algebraic Restrictions). *Let $x(t)$ be a periodic solution of Eq. (3.2.11) and let $H$ and $K$ be the subgroups of $\Gamma$ defined in Eq. (3.3.3). Then:*

1. *$K$ is an isotropy subgroup for the $\Gamma$-action*
2. *$K$ is a normal subgroup of $H$ and $H/K$ is either cyclic or $\mathbb{S}^1$*
3. *dim $Fix(K) \geq 2$. If dim $Fix(K) = 2$, then either $H = K$ or $H = N(K)$*

With $\Gamma$ be a finite group acting on $\mathbb{R}^n$ and let $x(t)$ be a periodic solution of a $\Gamma$-equivariant system of ODEs. Being K the subgroup of spatial symmetries and H the subgroup of spatio-temporal symmetries defined as in Eq. (3.3.3).

It is possible to claim one additional proposition leading to:

**Theorem 3.4.1** (H / K Theorem). *Let $\Gamma$ be a finite group acting on $\mathbb{R}^n$. There is a periodic solution to some $\Gamma$-equivariant system of ODESs on $\mathbb{R}^n$ with spatial symmetries $K$ and and spatio-temporal symmetries $H$ if and only if*

1. *$K$ is an isotropy subgroup for the $\Gamma$-action*
2. *$H/K$ is cyclic*
3. *dim $Fix(K) \geq 2$. If dim $Fix(K) = 2$, then either $H = K$ or $H = N(K)$*

*Moreover, when these conditions hold, hyperbolic asymptotically stable limit cycles with the desired (H,K) symmetry exist, and these are robust in $\Gamma$-equivariant systems of ODEs.*

It is important to notice how these theorems provide the methods to characterize possible patterns of oscillation, permitted by the network symmetries.

But there is no theorem so far which actually guarantees the existence of all the possible patterns, which has to be verified with other methods such as bifurcation techniques, in this case Hopf bifurcations.

## 3.5 Applicative Examples: Rings of Neurons

For the purpose of our analysis, a system of N identical coupled neurons or oscillatory centers is defined as a system of differential equations of the form:

$$\frac{dx_j}{dt} = f(x_j) + \sum_{i \to j} \alpha_{ij} h(x_i, x_j) \tag{3.5.1}$$

It is immediately apparent how Eq. (3.5.1) describes dynamics similar to Kuramoto model of Eqs. (2.4.1) and finally (2.4.2), which is the case of our study. Examples of such rings can be seen is Figure 3.2, where it is shown respectively: (Left) Unidirectional ring with $Z_3$ symmetry, (Middle) Asymmetric bidirectional ring with $Z_3$ symmetry, (Right) Bidirectional ring with $D_6$ symmetry. In the case when the internal dynamics of a coupled neurons system is $k \geq 2$, such as in the Kuramoto model, it is possible to introduce the following corollary:

**Corollary 3.5.1.** *Let $\Gamma$ be a finite group acting on $V$ and $W = V^k$ for some $k \geq 2$ be the state space of the coupled neurons system. Then there is a hyperbolic periodic solution to some $\Gamma$-equivariant system of ODEs on $\mathbb{R}^n$ with spatial symmetries $K$ and and spatio-temporal symmetries $H$ if and only if*

1. *$K$ is an isotropy subgroup for the $\Gamma$-action*
2. *$H/K$ is cyclic*
3. *If dim $Fix(K) = 2$, then either $H = K$ or $H = N(K)$*

To clarify some of the definitions done so far, lets consider a *bidirectional rings*, that is coupled neurons systems with N neurons and $D_{2N}$ symmetry with

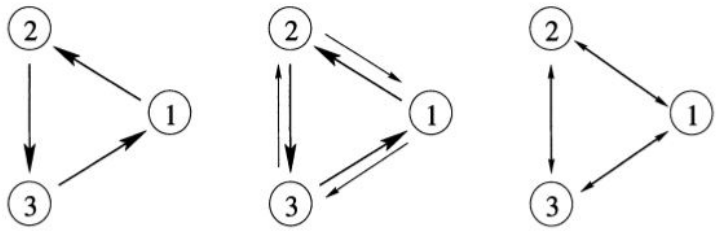


Figure 3.2: Rings of neurons

the same coupling between two cells.

First, a *bidirectional ring of two neurons* is taken into account. Its dynamics can be represented by:

$$\begin{aligned} \dot{x}_1 &= f(x_1, x_2) \\ \dot{x}_2 &= f(x_1, x_2) \end{aligned} \tag{3.5.2}$$

All two neurons systems following Eq. (3.5.2) have a permutation symmetry $\sigma(x_1, x_2) = (x_2, x_1)$, inherited directly from the symmetry of the network. The system is $\sigma$-equivariant according to Eq. (3.2.12). One consequence of the $Z_2$ symmetry of this system is then the existence of solutions in which $x_1(t) = x_2(t)$, $\forall t$. This happens because the subspace $\Delta = \{(x_1, x_2) : x_1 = x_2\}$ is a flow-invariant subspace for all g. In particular, this corresponds to saying that the fixed-point subspace $\mathrm{Fix}(\sigma)$ is a flow-invariant subspace, as defined by Eq. (refeq:fixed). It follows that synchrony is a direct result of the symmetry of the system. On the other hand, using the notation of Eq. (3.3.4), assume to have applied the permutation $\sigma$ twice on a T-periodic solution of Eq. (3.5.2), then it implies that $2\theta = 0 \ (\mathrm{mod} T)$. Hence either $\theta = 0$ or $\theta = \frac{T}{2}$. Since $\sigma$ is the permutation (1 2) it follows that $x_2(t) = x_1(t)$ for $\theta = 0$ (synchrony case ), and $x_2(t) = x_1(t + \frac{T}{2})$ when $\theta = \frac{T}{2}$ (phase-locked case).

Indeed, it is an important feature of identical two-neurons models that these systems naturally produce in-phase periodic solutions and half-period out-of-

phase periodic solutions. Biped gaits provide another excellent illustration: in-phase periodic states correspond to two-legged hopping and half-period out-of-phase periodic states correspond to walking.

Next, lets consider *bidirectional ring of three neurons*, as presented in Figure 3.2. In this case, let the action of $\Gamma = D_6$ on $V = \mathbb{R}^3$. The subgroups of $D_6$ are 1, $D_2$, $Z_3$, $D_6$. All but $Z_3$ are isotropy subgroups, which then cannot be selected as K. Then, the pairs $(H, K)$ are $(1, 1)$, $(D_2, 1)$, $(Z_3, 1)$, $(D_2, D_2)$, $(D_6, D_6)$, leading to the results in Table 3.2:

Table 3.2: Oscillations in three neurons bidirectional ring

| H | K | Patterns |
|---|---|---|
| $D_6$ | $D_6$ | $(x(t), x(t), x(t))$ |
| $D_2$ | $D_2$ | $(x(t), x(t), y(t))$ |
| $Z_3$ | 1 | $(x(t), x(t+\frac{1}{3}), x(t+\frac{2}{3}))$ |
| $D_2$ | 1 | $(x(t), x(t+\frac{1}{2}), y(t) = y(t+\frac{1}{2}))$ |
| 1 | 1 | $(x(t), y(t), z(t))$ |

Just two periodic solutions have non-trivial spatio-temporal symmetries for which $H/K$ is cyclic and $K$ is an isotropy subgroup: $(D_2, 1)$, $(Z_3, 1)$. The symmetry associated to $(D_2, 1)$ is: interchange neurons 1 and 2 and phase shift them by half a period. Thus, the third neuron is fixed by the symmetry and $y(t)$ is forced to satisfy $y(t) = y(t+\frac{1}{2})$, that is, $y(t)$ oscillates with half the period of the other neurons. The pair $(Z_3, 1)$ forces the three neurons to oscillate with the same wave-form but with a one-third period phase shift between the neurons. Finally, the implications of Corollary (3.5.1) include some surprises. For example, lets consider a four-cell network with two distinct *bidirectional ring of two neurons* as shown in Figure 3.3. The equations would be:

$$\begin{aligned}\dot{\phi}_1 &= f(\phi_1, \phi_2, y_3, y_4)\\ \dot{\phi}_2 &= f(\phi_1, \phi_2, y_3, y_4)\\ \dot{y}_3 &= g(\phi_1, \phi_2, y_3, y_4)\\ \dot{y}_4 &= g(\phi_1, \phi_2, y_3, y_4)\end{aligned} \tag{3.5.3}$$

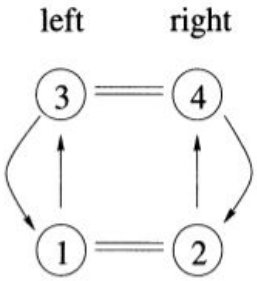


Figure 3.3: Four-neurons network

In case two dynamics are identical, such a network has been shown in [6] to be able to guarantee bipedal locomotion with all the possible patterns of symmetry. In case the dynamics differ, the symmetry group of this system is $\Gamma = Z_2 \times Z_2$. Now, suppose that a 1-periodic solution:

$$X(t) = (\phi_1(t), \phi_2(t), y_3(t), y_4(t)) \tag{3.5.4}$$

to this coupled neuron system exists. Suppose that this solution has two spatio-temporal symmetries $((1\ 2), \frac{1}{2})$ and $((3\ 4), \frac{1}{2})$. The product of the two symmetries is:

$$\gamma = ((1\ 2)(3\ 4), \frac{1}{4}) \tag{3.5.5}$$

explicitly exhibiting the isomorphism $Z_2 \times Z_2 \cong Z_4$. Thus $X(t)$ actually has the form:

$$X(t) = (\phi_1(t), \phi_2(t + \frac{1}{2}), y_3(t), y_4(t + \frac{1}{2})) \tag{3.5.6}$$

This solution actually exists due to Corollary (3.5.1), at least if all non-linearities consistent with $Z_4$ symmetry are permitted to be present. This solution corresponds to the anti-phase gait for the biped locomotion. Additionally, also the solution with the spatio-temporal symmetries $((1\ 2), 1)$ and $((3\ 4), 1)$ is permitted, and it leads to the in-phase gait. It will be shown that this network is equivalent to the CPGs network with the feedback inclusion in the biped case, being $\phi_1, \phi_2$ the phases of the Kuramoto oscillators and $y_3, y_4$ the states of the mechanical sections, out of other additional symmetries which would actually enrich the dynamics. This proves two important facts: not only the our framework is able to support the gaits but also it seems to be able to decrease the number of oscillators required by the group formalism, using the mechanical oscillators as an actual section of the CPGs network. For the biped case, this leads to a limited improvement, but the same considerations may be extended to quadruped case.

# Chapter 4

# The Tegotae Approach

## 4.1 Introduction

The inclusion of a feedback in the CPGs architecture is the natural extension of these control structures. As previously mentioned, any modification of the canonical form for the Eq. (2.3.1) leads to a modification in the main dynamics. A particular family of feedback functions as introduced in [5] is taken into account, and the way this inclusion locally affects the dynamics of a neural oscillator is outlined. In particular, a Kuramoto oscillator is taken as example, and the resulting synchronization phenomena are introducted. The approach used is called Tegotae approach.

Tegotae is a novel concept describing the extent to which a perceived reaction matches an expectation, or intention of a controller. Tegotae stems not only from the reaction received from the environment, but also from the consistency between the perceived reaction and the intention or expectation of the controller, i.e., what the controller wants to do. In case of matching, it is said that good Tegotae is obtained, otherwise, bad Tegotae is obtained. In this way, a cognitive meaning is added to the control framework, denoting some actions as positive and others as negative. The objective is then to maximize the Tegotae function. The

strongest assumption here is that the controller is able to generate some sort of expectations. In order to do this, some morphism between the controller and the controlled object, i.e., the body, is necessary, for instance via an internal model. Yet, how such internal model is generated is not the point of this discussion, so far. In this section, we introduce such formalism.

## 4.2 Mathematical Formulation

In the initial steps of the investigation, Tegotae is quantified in the simplest mathematical form, i.e., a function based on the type of separation of variables as follows:

$$T(u, e) = C(u)S(e) \tag{4.2.1}$$

Hereafter, the function $T$ is referred as the Tegotae function (T-function), i.e., a function that quantitatively measures Tegotae. In Eq. (4.2.1) $u$ is a control variable and $e$ is the sensory information obtained from multiple sensors embedded in the body. Note that the T-function $T$ is expressed as the product of two functions $C(u)$, $S(e)$. The former is a function expressing the intention of the controller, the latter denotes the reaction obtained from the environment.
$T$ is designed such that it becomes more positive when enhanced Tegotae is detected. Therefore, given that the T-function is defined a priori, the local sensory feedback $f$ is designed in such a way that the control system modulates $u$ in order to increase the amount of Tegotae received. Thus, since a continuous system is used, $f$ is expressed simply as a mono-dimensional gradient system of the T-function $T$ with respect to the control variable $u$, as follows:

$$f = \frac{\partial T(u, e)}{\partial u} \tag{4.2.2}$$

Note that with this formulation, it is possible to systematically design decen-

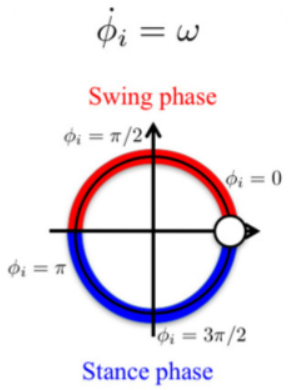


Figure 4.1: Trend in the local oscillator and Tegotae interpretation

tralized controllers by only designing the T-functions required.
Considering the CPGs framework, the $i$th controller can be firstly defined as a generic Kuramoto oscillator of phase $\phi_i$ without coupling terms but with a specific external field $f_i$ consisting in the local sensory feedback:

$$\dot{\phi}_i = \omega_i + f_i(\phi_i, e) \tag{4.2.3}$$

It follows that in this formualtion:

$$f_i = \frac{\partial T_i(\phi_i, e)}{\partial \phi_i} \tag{4.2.4}$$

In [5] the T-function was supposed to satisfactorily reproduce the hexapedal interlimb coordination observed in insect locomotion by using Kuramoto oscillators. For this reason, it was generally defined in the first case as:

$$T_i(\phi_i, N) = (-\sin\phi_i)N_i^V \tag{4.2.5}$$

Where in Eq. (4.2.5) the sensory information $e$ consisted of vertical ground reaction forces $N_i^V$ acting on each leg. In this formulation, $T_i$ quantifies Tegotae on the basis of the information that is only locally available at the corresponding leg. As explained graphically in Figure 4.1, when the local controller intends to

be in the stance phase ( $-\sin\phi_i > 0$ ) and results in receiving a ground reaction force ( $N_i^V > 0$), $T_i$ evaluates the situation as good Tegotae, and vice versa.
As said above, the reaction in Eq. (4.2.1) is generic, and other sorts of reactions may be taken into account. In particular, in our study the force passing through the body is taken into account, i.e., an elastic force. Such a definition is inspired by the muscle spindle, which is a proprio-ceptive sensory receptor organ that senses changes in muscle tension due to external factors. The T-function is then defined in our for a generic $i$th phase oscillator and consequently the feedback signal as well:

$$T_i(\phi, F) \triangleq (-\sigma \sin(\phi_i))F \tag{4.2.6}$$

$$f_i(\phi, F) = \frac{\partial}{\partial \phi_i} T(\phi, F) = -\sigma \cos(\phi)F \tag{4.2.7}$$

with $\sigma$ being a proportionality factor and $F$ being the force passing through the body. By the nature of Eq. (4.2.6), it follows that this sensory feedback will be absent if there is no contact with the ground.

## 4.3 Stability Analysis

As shown in Figure 4.2 and further analyzed in [21], the behavior of the Kuramoto oscillator of Eq. (4.2.3) can be classified into two states on the basis of the value $a = \sigma F$. When $a < \omega$, i.e. $\omega/\sigma F > 1$ an oscillatory behaviour is exhibited, whereby the oscillator rotates periodically around the unit circle. In this case there will be an oscillation with frequency:

$$\tilde{\omega} = 2\pi \left[ \int_{-\pi}^{\pi} \frac{d\phi}{\omega - a \sin\phi} \right]^{-1} \tag{4.3.1}$$

In contrast, when $a > \omega$, i.e. $\omega/\sigma F < 1$ there exists a pair of equilibria at which the equilibrium condition is satisfied: one point is stable and the other is

unstable. For $\omega/a \cong 1$ the two equilibria are relatively close to each other. As a result, the system behaves very differently in its course back to the resting state (i.e. stable equilibrium) depending on whether or not the initial disturbance from the resting state is strong enough to force the system beyond the potential barrier peaked near the unstable equilibrium. This feature is known to be the most basic to excitable systems. In such a case, the oscillator exhibits excitatory behavior. Thus, the value of $\omega/\sigma F$ is one index that determines the property of the phase dynamics in the proposed model, yet the resulting dynamics will be extremely complex due to the time varying value of $F$.

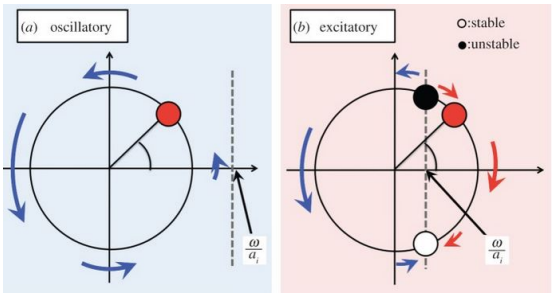


Figure 4.2: Phase oscillator stability

## 4.4 Dimensional Extension

Finally, as presented in [5], it is possible to further extend the Tegotae feedback in case of several forces, in case we think each force as related to one oscillator. By writing:

$$T_i(\phi, F) \triangleq -\sigma_i \sin(\phi_i) F_i + \tag{4.4.1}$$
$$+ \frac{K}{N} \sum_{j=1, j\neq i}^{N} \sigma_{ij} \sin(\phi_i) F_j$$
$$f_i(\phi, F) = \frac{\partial}{\partial \phi_i} T_i(\phi_i, F) = -\sigma_i \cos(\phi_i) F_i + \tag{4.4.2}$$
$$+ \frac{K}{N} \sum_{j=1, j\neq i}^{N} \sigma_{ij} \cos(\phi_i) F_j$$

It is apparent how this formulation asks to the system to show an anti-phase gait in the movement in case of couplings $\sigma_{ij} > 0$, an in-phase movement otherwise. It is still an open question how to formalize the T-function in terms of a global coupling matrix $\Sigma$ in order to permit several gaits. This is not the main objective of the stuy and this issue will not be tackled. In general, it is sufficient to design the values $\sigma_{ij}$ in order to match the particular expectations defined in the Tegotae framework.

## 4.5 Tegotae and Learning

The Tegotae approach shows interesting similarities with different learning frameworks which motivates some of the intuitions on its energy efficiency as well. In fact, the adaptivity in learning processes is usually defined for the parameters/weights of the controller/learning agent. In the Tegotae framework, even though a further adaptation of the feedback coefficients $\sigma$ may be included, the main adaptation is induced via a modification of the dynamics of the oscillators. This factor will be the one taken into account in the comparison, since an eventual adaptation of the parameters is straightforwardly implementable.

Firstly, it is interesting to notice how the Tegotae approach shares some similarities with the *tacit learning*, a learning framework introduced in [27, 28], whose

feedback loop is shown in Figure 4.3.

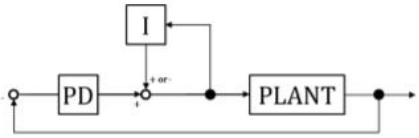


Figure 4.3: Block Diagram of the controller for tacit learning

In tacit learning, the control law consists in an extension of the PD controller via a tacit learner block in the time frame (Lt). Considering the scalar case for simplicity without loss of generality:

$$\begin{aligned} u &= kx_c^T + q \\ q &= \int f(e)dt \quad \text{(Lt)} \end{aligned} \tag{4.5.1}$$

where $u$, $x_c$, $k$, $e$ are respectively the control, the state variable expressed in the control space, the proportional and derivative gain and any kind of quantity to be minimized, where the use of the letter $e$ will be later clarified. The actual learning process is obtained in the (Lt) block via the accumulation in the integral over the time of the quantity to be minimized. Due to this, the proportional and derivative terms will be neglected now on, obtaining:

$$\begin{aligned} u &= q \\ q &= \int f(e)dt \quad \text{(Lt)} \end{aligned} \tag{4.5.2}$$

The function $f(q)$ is required to be of the form $f(e) = p(\xi)\ a(e)^T$. In the mono-dimensional case $a(e)$ can be a simple linear transformation $a(e) = ae$ and $p(\xi)$ is a periodic function of a virtual variable $\xi$. Both of these additional terms have to be selected in order to guarantee:

$$\begin{aligned} \frac{p(\xi)\ x_e}{\|p(\xi)\|\ \|x_e\|} &= 1 \quad \text{if } \alpha = 0 \\ \frac{p(\xi)\ x_e}{\|p(\xi)\|\ \|x_e\|} &= -1 \quad \text{if } \alpha = \pi \end{aligned} \tag{4.5.3}$$

In Eq. (4.5.5) $x_e$ is the state variable expressed in the task space for which the error $e$ is minimized. On the other hand, $\alpha$ is generically defined as the angle between $\dot{e}$ and $D(e)$, the latter being the direction towards which $e$ is minimized. In the mono-dimensional case $\alpha = 0 \wedge \pi$. This formulation guarantees the fact that $\min(f(e)) = \min(e)$.

Now lets take into account the Tegotae framework. The objective is to construct a feedback, and not a feed-forward controller. To do this, lets consider the factor to be minimized corresponding exactly to $e = -F_k$, the virtual variable $\xi$ to the phase variable $\phi$ and the error function $a(e) = \sigma e$. Neglecting constant terms due to the integration, the feedback over the oscillator results in:

$$\begin{aligned} u &= q \\ e &= -F_k \\ p(\phi) &= -\sin(\phi) \\ q &= \int f(e) d\phi = \int \sigma \sin(\phi) F_k d\phi \\ &= -\sigma \cos(\phi(t)) F_k \quad (\mathrm{L}\phi) \end{aligned} \tag{4.5.4}$$

In the Tegotae framework, it is possible to set $x_e = \dot{\Delta} l$, that is the speed of elongation of the spring length. In fact, this variable is pointing to the direction of minimization of the value of $e = -F_k$, it follows that:

$$\begin{aligned} \frac{-\sin(\phi)\ x_e}{\|\sin(\phi)\|\ \|x_e\|} &= 1 \quad \text{if } \alpha = 0 \\ \frac{-\sin(\phi)\ x_e}{\|\sin(\phi)\|\ \|x_e\|} &= -1 \quad \text{if } \alpha = \pi \end{aligned} \tag{4.5.5}$$

This shows how the Tegotae approach is de facto realizing a tacit learning block similar to the (Lt) previously described. Yet, this is done via the accumulation of the quantity to be minimized in the integral of the state space variable directly (L$\phi$). The integration over the state space $\phi$ is actually coherent with the CPGs framework. In fact, the role of the oscillators is to provide a different time frame to the mechanical dynamics, which is reproduced via the linear transformation $\phi = \omega t$. Thus, in the CPGs framework the integration/derivation over the state variable of the oscillator $\phi$ is conceptually equivalent to the integration over the time in standard tacit learning. Interestingly it has been shown in [29] that such a controller is somehow able to guarantee energy efficiency in the task realization in case the quantity to be minimized is the actuation torque.

Finally, another significant learning framework which can lead to interesting interpretations is Hebbian learning. According to the original formulation by Hebb [30], such learning principle consists in:
*When an axon cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*
The Hebbian framework shows two important aspects, as analyzed in [31]. First, the learning rule is local: only the information that is available at the location of the synapse can be used to change the weight of that synapse. Second, the learning rule must be sensitive to the correlations between the action potentials of the two neurons. These two aspects can be summarized in the concepts of locality and cooperativity. Interestingly, the Tegotae approach as it has been defined in in Eq. (4.2.1) so far shares these two aspects, since each Tegotae feedback $f_i$ depends on the state of the related oscillator $\phi_i$ (locality) and by definition it contains the correlation between the phase oscillator and the mechanical oscillator corresponding to the relation between action and expectation (cooperativity). The most coherent framework with the Tegotae can be found considering the *dif-*

*ferential extrinsic plasticity* (DEP) proposed in [7], an extension of the classical Hebbian learning rule. In Hebbian laws such as in [30], the rate of change of the synapse weight $\dot{c}_{ij}$ between the $i$th and $j$th neuron is proportional to the input $x_j$ into the synapse of the $i$th neuron multiplied by its activation $y_i$, i.e., $\dot{C}_{ij} \propto y_i x_j$. However, this rule produces typically fixed-point behaviors, leading to a cease of the evolving dynamics. Then, time can come into play if the so-called *differential Hebbian learning* (DHL) is used, i.e., replacing neuronal activities by their rates of change, so that $\dot{C}_{ij} \propto \dot{y}_i \dot{x}_j$. This rule focuses on the dynamics because there is only a change in the behavior if the system is active, interacting with the environment. Yet, this learning rule as well shows limits in the richness of the behavior. Trivially, once $\dot{y} = 0$, learning and any change in behavior stop altogether. Now, the idea is to lift this structure to the level of the environment, enriching learning by the reactions of the physical system to the controls. Assuming that the agent has a basic understanding of the relations between actions and reactions represented in the sensor values, this is realized by an inverse model which approximately relates the current sensor values $x'$ back to its causes, the motor command $y$, with a certain time lag w.r.t. $x'$. The model will reconstruct y with a certain mismatch $\delta y$. By formulating in terms of the rated of change, it follows:

$$\dot{y} + \delta\dot{y} = F(\dot{x}') \tag{4.5.6}$$

Finally, it is possible to rewrite the Hebbian rule as a DEP:

$$\dot{C}_{ij} \propto F(\dot{x}')\dot{x}_j \tag{4.5.7}$$

This is coherent with the Tegotae framework of using the reaction from the environment, while on the other hand it might suggest a further generalization of the Tegotae in terms of the rate of changes of the phases of the oscillators and of the proprio-ceptive information. It is important to notice that the link shown

has the aim to further extend the qualitative understanding of what a feedback designed in a Tegotae manner might be actually doing, and it is far from being a rigorous evidence of equivalence between the DHL framework and the Tegotae one. Once again, this first research finds its aim in widing the understandings of the Tegotae approach, also in view of subsequent analysis.

## 4.6 Tegotae Control Policy: Preliminary Design and Extensions

In the majority of CPGs controllers, as previously described in chapter 1, the actuator is driven by a PID control scheme, which compares the actual state of the physical system with the reference signal originated by the CPGs network. One of our main contribution is to try to maintain the model-free control policy while taking into account some of the most recent considerations above embodied intelligence and control by the use of neural-like dynamical systems, as extensively analyzed in [32]. The main point introduced by their research group is to interpret neural dynamics as a cognitive dynamics, in which particular events in the first, for example bifurcations, have a direct equivalent from the cognitive point of view, for example decisions. The dynamics introduced by the specific neural system taken into account, let it be Kuramoto oscillators, generic Hopf oscillators, or neuronal dynamics according to the Amari equations, as in [32], has then to be somehow interpreted in terms of decisions, categorical representations or other cognitive objects, based on the application. For example, a pitchfork bifurcation, leading to a couple of stable and unstable attractors, will correspond to a specific tendency to follow some decision, i.e. the stable attractor. On the other hand, specific dynamics such as the ones obtained vie the Amari equations may be able to support several attractors, these attractors corresponding to specific categories formed from an interaction with the environment. This last, the environment,

is a central aspect of embodied intelligence theory, since in this framework any intelligent behavior can be obtained only once the agent interacts with the environment.

Our control policy has been selected as being more coherent with the interpretation of neural dynamics as actual cognitive processes. Moreover, the Tegotae approach is intrinsically linked to this framework, since the feedback itself has been designed taking into account the interaction with the environment and sorts of expectations from the controller to maximize the Tegotae.

Now, it has been showed how the neuro-mechanical coupling given by the feedback will force a secondary dynamics in the phase oscillator [33] and the current objective is to analyze and possibly exploit such effect in terms of cognitive objects. This will be done by using a critical point of the feedback dynamics, such as a minimum, or a specific section of it, in order to control the system. The evolution of the Tegotae control policy towards its current form will be described.

**Policy 1.1** In the first control policy law, a constant actuation force of value $A$ is taken into account, and the force is injected only when the phase of the oscillator $\phi$ is inside a certain interval of width $\Delta$ containing the selected critical point of the dynamics $\phi_0$, with $\phi \in \mathbb{S}$:

$$F_a(\phi, \cdot) = A \iff \phi \in (\phi_0 - \Delta/2, \phi_0 + \Delta/2) \tag{4.6.1}$$

It is apparent that a critic factor of this preliminary policy is the on-line adaptation of the values of $\phi_0$ and $\Delta$ according to the evolution of the dynamics from the transient to the steady state, assuming it is reached. This is non-trivial. In a first instance, such values were selected a posteriori once the specific dynamics of the oscillator was studied and they were maintained constant over the whole simulation. The results with this simple control policy will be analyzed in the monoped case study in Chapter 5, showing how even this simple policy is able to guarantee good performances.

**Policy 1.2** Clearly, this policy can be made smoother by substituting the square wave in the input force with other sorts of smoother functions such as Gaussian profiles:

$$F_a(\phi, \cdot) = \frac{A}{\sqrt{2\pi\Delta}} e^{-\frac{1}{2}\frac{(\phi-\phi_0)}{\sqrt{\Delta}}} \tag{4.6.2}$$

Even if leading to an easier actuation and solving the numerical issues introduced by a switching controller, this control policy does not simplify how the specific values of $\phi_0$ and $\Delta$ should be selected.

**Policy 2** Finally, by taking into account the cognitive interpretation of the Tegotae, the whole negative section centered around the minimum of the Tegotae feedback corresponds to a critical phase of the whole dynamics. In fact, in this section:

$$f_i(\phi, F) = \frac{\partial}{\partial \phi_i} T(\phi, F) \leq 0 \tag{4.6.3}$$

This specifically means that the Tegotae is decreasing. By definition, the aim of the controller being to maximize it, then it is clear how this area is the designed area to inject some force. In particular, such force will be required to lead to a maximization of the Tegotae, depending on the case study. In the current problem formulation, a positive force leading to a jump fits the requirements. Due to this, following the Eq. (4.2.6), the final control policy is selected as:

$$F_a(\phi, \cdot) = -\min(0, f_i) \tag{4.6.4}$$

This clearly reintroduces the numerical issues of a switching controller, but on the other hand it directly links actuation and Tegotae feedback, while assuring an online adaptation to the variation of the dynamics, since the Tegotae feedback corresponds to this variation itself.

# Chapter 5

# Case Analysis: Monoped

## 5.1 Introduction

First, a mono-dimensional hopping robot was taken into account, being characterized by a mass connected to a mass-less spring and damper system. A linear actuator was in parallel to the spring and the damper and determines a vertical thrust. The *Kuramoto model* for phase oscillators was used as a model for the CPGs oscillator, simplifying the analysis of the effects of the feedback. The integration of the ODEs was done via a MATLAB routine which automatically stops the integration when the event of switching was detected. The initial step of integration was set to $1e^{-3}$, equal to the maximum step of integration.

## 5.2 Mathematical Formulation

The evolution of a single phase of the oscillator $\phi$ and of the vertical height of the mass $y$ was described by the ODE:

$$\begin{aligned} \dot{\phi} &= \omega + f(\phi, F) \\ \ddot{y} &= \frac{1}{m}\left(F_c(\dot{y}) + F_k(y) - mg + F_a(\phi, \cdot)\right) \\ F_c(\dot{y}) &= -c\dot{y} \\ F_k(y) &= k(l_0 - y) \end{aligned} \tag{5.2.1}$$

In Eq. (5.2.1) $f(\phi, F)$ was the sensory feedback in the Kuramoto model, while $F_k(y)$, $F_c(\dot{y})$ and $F_a(\phi, \cdot)$ were the elastic, the viscous and the generic actuation force due to the spring, the damper, and the actuator respectively. These three components were absent during the flight phase, assuming no forces acting from the environment. The actuation force was selected to be as the policy 1 in Eq. (4.6.1) in first instance. As described in [5] and introduced in chapter 4, the Tegotae sensory feedback $f(\phi, F)$ was defined directly by the T-function $T(\phi, F)$, in our case we selected $F = F_k(y)$:

$$\begin{aligned} T(\phi, F_k) &\triangleq (-\sigma \sin(\phi))F_k \\ f(\phi, F_k) &= -\frac{\partial}{\partial \phi} T(\phi, F_k) = -\sigma \cos(\phi) F_k \end{aligned} \tag{5.2.2}$$

with $\sigma$ being a proportionality factor. By the nature of Eq. (5.2.2), it followed that this sensory feedback was absent during the flight phase as well.

## 5.3 Simulations: Adaptation Transient and Energy Efficiency

The goal of the simulations was to analyze the effects of different feedbacks in terms of stability, transient periods and power injection by the actuator. Four different instances were taken into account as sensory feedbacks, as reported in Figure 5.1. While $f_2$ corresponded to the height of jump, $f_4$ was the force passing

through the spring. Then, $f_1$ and $f_3$ were respectively the Tegotae feedback and the feedback proposed in [33]. Interestingly, both of these shared a neuro-mechanical coupling. It was evident that all of them were introducing a strong polarization with critical points, with which we defined $\phi_0$.

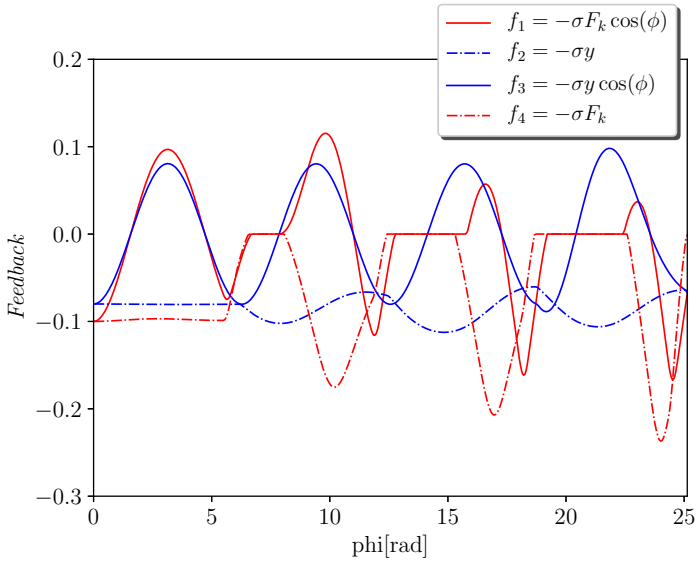


Figure 5.1: Feedback dynamics over the phase $\phi$

The mechanical parameters and the natural length of the spring were $m = 0.1$ kg, $k = 5$ N/m, $c = 0.2$ Ns/m, $l_0 = 1$ m respectively. The parameters of the oscillator were $\omega = 8$ rad/s and $\sigma = 2$, whose dimensionality was determined on the basis of the feedback law. The initial conditions were respectively $y_1 = 0.7$m, null velocity and the angle of oscillator selected randomly, to guarantee a certain robustness with respect to the initial conditions. The actuation parameters and the results of the simulations were worked out on oscillations in the steady state and they are reported in Table 5.1. The transient period $\Delta t$ was defined as when a limit cycle was reached. It followed that the case $f_4$ was not able to provide a stable orbit. Finally, it was evident that the introduction of the Tegotae feedback was optimal in terms both of the transient period of synchronization and of the energy efficiency $E_e$, defined on a limit cycle of period $T^\star$ with an actuation force

$F_{act}$, as:

$$E_e = \frac{h_{max,T^\star} - h_{min,T^\star}}{E},\ E = \int_{T^\star} F_{act}(t)\dot{y}(t)dt \tag{5.3.1}$$

Interestingly, to obtain similar hopping in terms of maximum height of jump the cases $f_2$,$f_3$ required higher magnitude of the actuation force.

Table 5.1: Simulation results

| Feedback | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| A [N] | 4 | 12 | 12 | 4 |
| $\phi_0$, $\Delta$ [rad] | $1.75\pi$,$0.1\pi$ | $1.96\pi$,$0.1\pi$ | $1.96\pi$,$0.1\pi$ | $1.75\pi$,$0.1\pi$ |
| $h_{max}$ [m] | 1.16 | 1.05 | 1.08 | 1.57 |
| $\Delta$t [s] | 3 | 4 | 5 | ∄ |
| $E_e$ [m/Ws] | 1.50 | 1.16 | 1.15 | 1.25 |
| J [W] | 5.49 | 17.69 | 20.15 | 10.56 |

## 5.4 Simulations: Robustness and Adaptivity

Secondly, the case of the Tegotae approach $f_1$ and the case $f_3$ presented in [33] were further compared, testing them with a change in the environment. In particular, at $t = 5$s the ground level was lowered from 0 to $-0.6$ m. The results are shown in Figure 5.2. It was evident that the Tegotae approach was able to cope with these variations by a proper re-polarization of the oscillator, even without an adaptation of $\sigma$, $\phi_0$ or $\Delta$. It was also possible to notice how the Tegotae approach was able to quickly react to such variations, by a modification in the power injection.

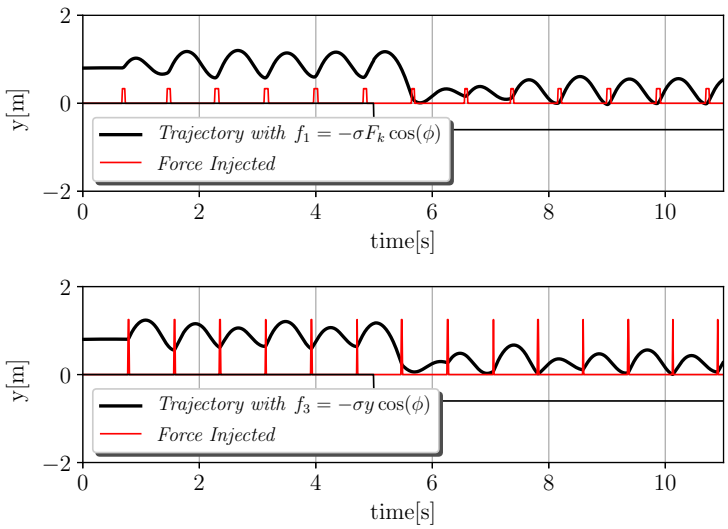


Figure 5.2: Dynamic environment and adaptation process

# Chapter 6

# Case Analysis: Biped

## 6.1 Introduction

The effects of the Tegotae approach on a more complex mechanical and oscillatory system were studied, to prove its effectiveness and ability to sustain different patterns, as already tackled in [5]. An energetic efficiency analysis was not performed with respect to other feedback laws. Since this case was a mere extension from the mechanical point of view, results were assumed to hold and further studies might be performed in the future researches. The integration of the ODEs was done via the same MATLAB routine and integration parameters as in chapter 5.

## 6.2 Mathematical Formulation

The mechanical system was a mono-dimensional hopping robot composed by two legs connected via a spring, as shown in Figure 6.1. The system corresponded to a slight modification of the previous case:

$$\dot{\phi}_1 = \omega_1 - \sigma F_{k1} \cos\phi_1 + \\ +\epsilon_{12} \sin(\phi_1 - \phi_2) \quad (6.2.1)$$

$$\dot{\phi}_2 = \omega_2 - \sigma F_{k2} \cos\phi_2 + \\ +\epsilon_{21} \sin(\phi_2 - \phi_1) \quad (6.2.2)$$

$$\ddot{y}_1 = \frac{1}{m_1}(F_{c1}(\dot{y}_1) + F_{k1}(y_1) - m_1 g + \\ +F_{a1}(t, y_1, \phi_1) + F_{k12}) \quad (6.2.3)$$

$$\ddot{y}_2 = \frac{1}{m_2}(F_{c2}(\dot{y}_2) + F_{k2}(y_2) - m_2 g + \\ +F_{a2}(t, y_2, \phi_2) + F_{k21}) \quad (6.2.4)$$

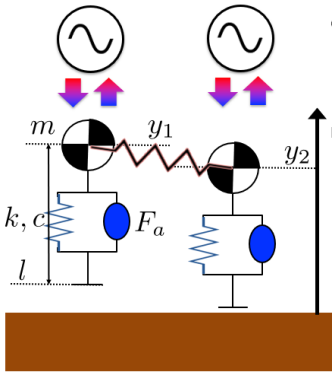


Figure 6.1: Neuro-mechanical structure of the mono-dimensional hopper with two legs

In Eq. (6.2.1,6.2.2) the Tegotae feedback was already taken into account, while the last term on the righthen side represented the weak-coupling between phase oscillators as described in [10]. In Eq. (6.2.3,6.2.4) the components were the same as the one defined in Eq. (5.2.1), out of a simple additional elastic force introduced by the connecting spring $F_{kij} = k_c(y_j - y_i)$. On the other hand, the control policy was selected as the policy 2 of Eq. (). The effect of other Tegotae terms in the actuation were neglected with respect to this main component.

## 6.3 Simulations: Gaits

The objective was to obtain two different gaits, that is in-phase and anti-phase hopping. Considering the cyclic notations and the group formalism, these two gaits were represented respectively by the spatial symmetry $((1)(2), 0)$ and the spatio-temporal symmetry $((1)(2), \frac{1}{2})$. A system of two oscillators was then sufficient to sustain these gaits, since the subgroup of the network is $\{((1)(2)), I\}$. On the other hand, if the whole system of phase and mechanical oscillators was taken into account, then according to the structure of Figure 3.3 previously presented, the system was expected to be able to sustain the two gaits even without a coupling between the oscillators, coupling which would be required to obtain only more complex gaits between the phase and mechanical parts.

As already treated in [5], in CPGs architectures the pulsation of oscillation $\omega$ is a useful control variable which can be exploited to introduce a gait transition in the pattern generation. Such pulsation can be seen as one of the few high-level control variables required by CPGs architectures, as already tackled in [1]. Interestingly, the Tegotae control policy was able to maintain these properties, even without the introduction of any oscillator couplings, i.e.$\epsilon_{12} = \epsilon_{21} = 0$. Two distinct gaits, in-phase hopping and anti-phase hopping, emerged as in Figure 6.2 and in Figure 6.3.

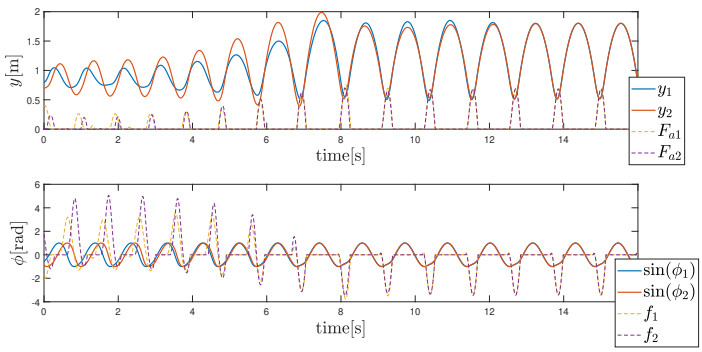


Figure 6.2: In-phase hopping

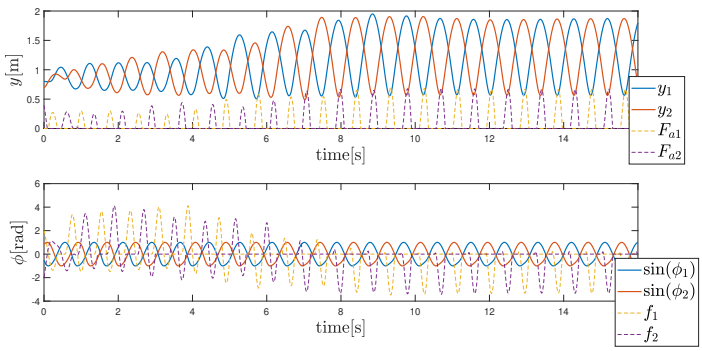


Figure 6.3: Anti-phase hopping

The case of Figure 6.2 was obtained with a pulsation $\omega_{in} = 6$rad/s, while the second case of Figure 6.3 with $\omega_{anti} = 7.5$rad/s, the values of mechanical parameters generally equal to the monoped case, with the addition of a spring constant $k_c = 1$. The feedback strength was set to $\sigma = 2.4$ in order to guarantee higher vertical excursion. The initial conditions were respectively $y_1 = 0.8$m, $y_2 = 0.7$m, null velocities and the angles of oscillators selected randomly, to guarantee a certain robustness with respect to the initial conditions again. These figures represent both the mechanical section of the system and the phase section, with actuation force and Tegotae feedback respectively. Finally, it was evident that by changing the control variable from $\omega_{in}$ to $\omega_{anti}$ it was possible to reproduce a gait transition, as shown in Fig.6.4, where the value was changed at $t = 8$s and the trend of actuation forces and feedbacks have been hidden for clarity reasons, due to the presence of several transient sections. The motivations why these specific gaits were shown for different values of $\omega$ were still an open point so far, considering also the fact that due to the random initialization of the phase angles, other gates are seldom shown.

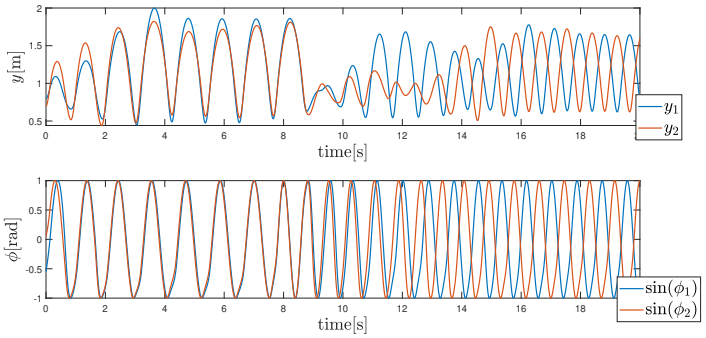


Figure 6.4: Gait transition

## 6.4 Simulations: Robustness and Adaptivity

Finally, equivalently to the monoped case, the way the control policy expressed in Eq. (equ:actuation2) was able to sustain a change in the environmental conditions was studied as well. As reported in Figure 6.5, the ground was lowered for both the legs to $-0.6$m as in the monoped case, while the pulsation was equal to $\omega_{in}$. In Figure 6.6 the ground was again lowered for both the legs to $-0.6$m, while the pulsation was equal to $\omega_{anti}$. Finally, in Figure 6.7 the ground was raised for both the legs to 0.3m, while the pulsation was equal to $\omega_{anti}$. All these cases confirmed a good robustness of the control policy to environmental conditions, in this case the ground level.

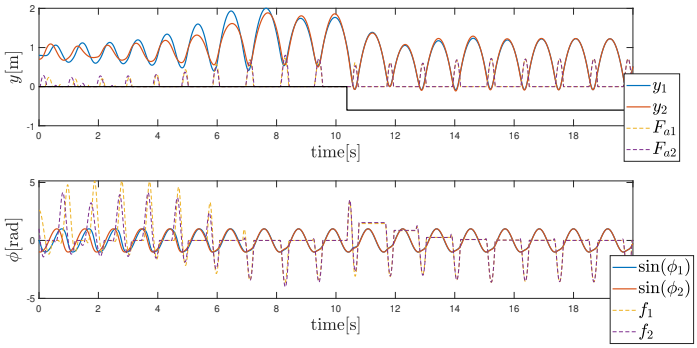


Figure 6.5: Adaptation to lower step with in-phase hopping

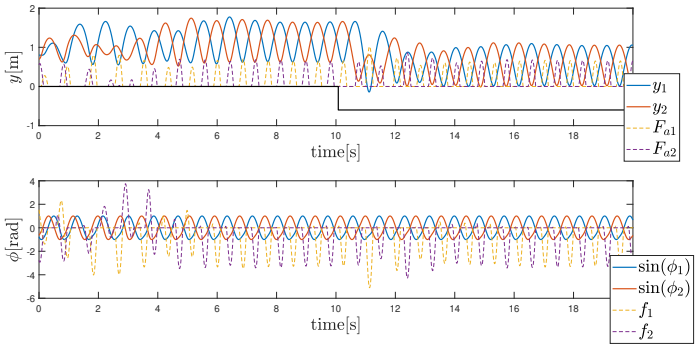


Figure 6.6: Adaptation to lower step with anti-phase hopping

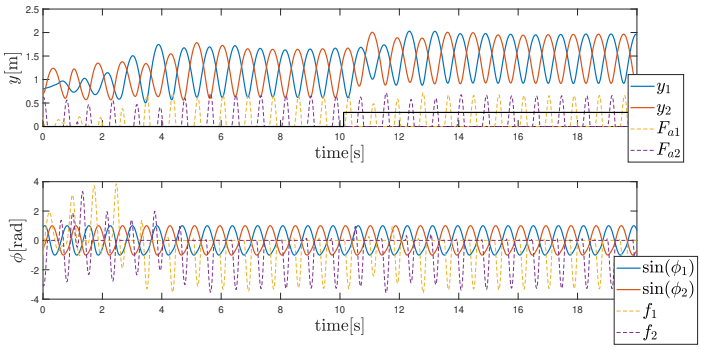


Figure 6.7: Adaptation to higher step with anti-phase hopping

# Chapter 7

# Optimal Control: Theory and Results

## 7.1 Introduction

The energy efficiency of the Tegotae approach is analyzed by the comparison between the Tegotae controller and optimal controller, designed via direct optimal control methods. After a brief introduction to what these methods are and how they can be implemented, following mainly [34, 35] for the contents and [36] for the explanatory illustrations, the optimization algorithms are shown and the results derived, following the algorithms mainly shown in [37, 38].

## 7.2 Optimal Control: Formulation and Methods

The problem will be formulated as an optimal control problem. Then, for explanatory purposes, the following simplified optimal control problem in ordinary differential equations is taken into account:

$$
\begin{aligned}
\underset{x(\cdot),u(\cdot)}{\text{minimize}} \quad & \int_0^T L(x(t),u(t))dt + E(x(T)) \\
\text{subject to} \quad & x(0) - x_0 = 0 \quad \text{(Initial Value Constraints)}, \\
& \dot{x}(t) - f(x(t),u(t)) = 0 \quad t \in [0,T] \quad \text{(ODE Constraints)}, \\
& h(x(t),u(t)) \geq 0 \quad t \in [0,T] \quad \text{(Nonlinear Path Constraints)}, \\
& r(x(T)) = 0 \quad \text{(Terminal Constraints)}
\end{aligned}
\tag{7.2.1}
$$

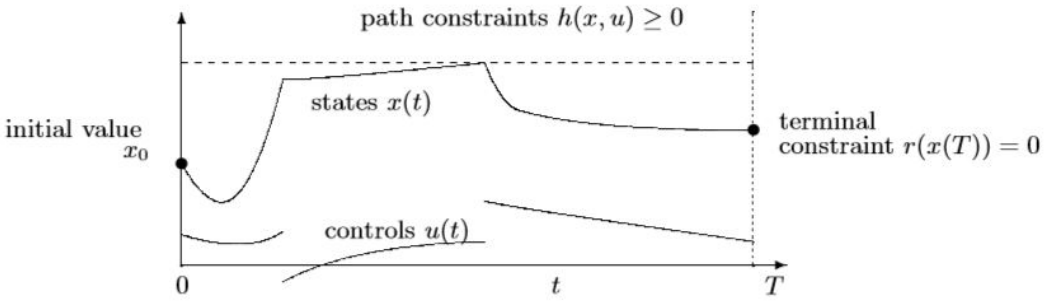


Figure 7.1: Simplified Optimal Control Problem

The problem is visualized in Figure 7.1. The horizon length T may or may not be free for optimization. As an example we may think of a robot that shall move in minimal time from its current state to some desired terminal position, respecting limits on torques.

Generally speaking, there are three basic approaches to address optimal control problems, cf. the top row of Figure 7.2, in particular for Nonlinear Programming Problems (NLP) of the form:

$$
\begin{aligned}
\underset{w}{\text{minimize}} \quad & f(w) \\
\text{subject to} \quad & g(w) = 0, \\
& h(w) \geq 0
\end{aligned}
\tag{7.2.2}
$$

- *Dynamic Programming* uses the principle of optimality of sub-arcs introduced by Bellman [39] to compute recursively a feedback control for all times $t$ and all $x_0$. In the continuous time case, as here, this leads to the Hamilton-Jacobi-Bellman (HJB) equations, a partial differential equation (PDE) in state space. Methods to numerically compute solution approximations exist, but the approach severely suffers from Bellman's "curse of dimensionality" and it is restricted to small state dimensions.

- *Indirect Methods* use the necessary conditions of optimality of the problem to derive a boundary value problem (BVP) in ordinary differential equations (ODE). This BVP has to be solved numerically, and the approach is often sketched as "first optimize, then discretize". The class of indirect methods encompasses also the well known calculus of variations and the Euler-Lagrange differential equations, and the Pontryagin Maximum Principle. The numerical solution of the BVP is mostly performed by shooting techniques or by collocation. The two major drawbacks are that the underlying differential equations are often difficult to solve and that changes in the control structure, i.e. the sequence of arcs where different constraints are active, are difficult to handle. This usually requires a completely new problem setup. In some cases, higher index DAEs arise as well, which necessitate specialized solution techniques.

- *Direct Methods* transform the original optimal control problem into a finite dimensional NLP. This NLP is then solved by variants of state-of-the-art numerical optimization methods, and the approach is therefore often sketched as "first discretize, then optimize". One of the most important advantages of direct methods compared to indirect methods is that they can easily treat inequality constraints, like the inequality path constraints in the formulation above. This is because structural changes in the active constraints during the optimization procedure are treated by well devel-

oped NLP methods that can deal with inequality constraints and active set changes. All direct methods are based on a finite dimensional parametrization of the control trajectory, but they differ in the way the state trajectory is handled.

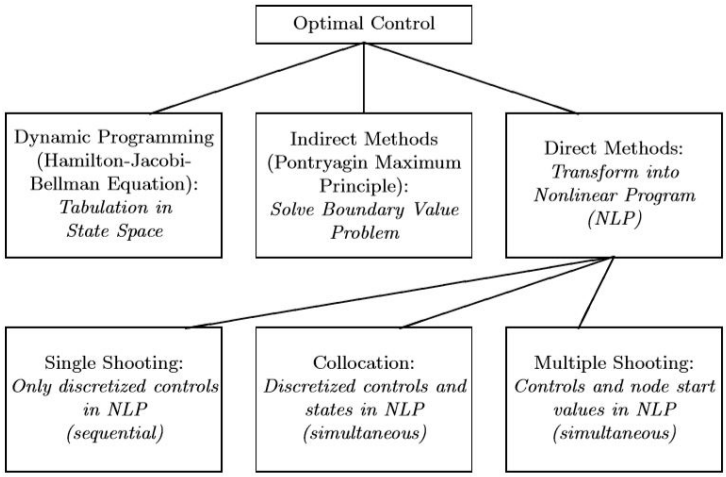


Figure 7.2: Numerical methods for optimal control

## 7.3 Direct Methods: Single Shooting

The single shooting (SS) method starts by discretizing the controls. For instance, grid points on the unit interval might be chosen, $0 = \tau_0 < \tau_1 < \cdots < \tau_N = 1$, and then these grid-points might be re-scaled to the possibly variable time horizon of the optimal control problem $[0, T]$, by defining $t_i = T\tau_i$ for $i = 0, \ldots, N$. The controls $u(t)$ are discretized on this grid, so that $u(t)$ only depends on the finitely many control parameters $q = (q_0, q_1, \ldots, q_{N-1}, T)$ and can be denoted by $u(t, q)$. If the problem has a fixed horizon length $T$, the last component of $q$ disappears as it is no optimization variable. This scheme is visualized in Figure 7.3 Using a numerical simulation routine for solving the initial value problem:

$$\begin{aligned} x(0) &= x_0 \\ \dot{x}(t) &= f(x(t), u(t,q)), \quad t \in [0,T] \end{aligned} \tag{7.3.1}$$

the states $x(t)$ on $[0,T]$ can be regarded as dependent variables and denoted by $x(t,q)$.

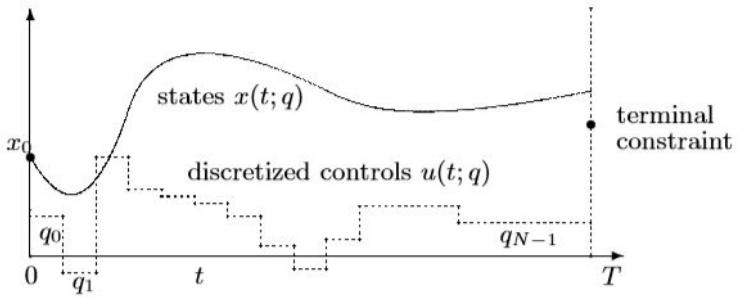


Figure 7.3: Illustration of Single Shooting Method

The question which simulation routine should be chosen is crucial to the success of any shooting method and depends on the type of ODE model. The nonlinear path constraints are discretized as well to avoid a semi-infinite problem, for example by requiring $h(x(t), u(t)) \geq 0$ only at the grid points $t_i$, but also a finer grid could be chosen without any problem. Thus, the following finite dimensional NLP is obtained:

$$\underset{q}{\text{minimize}} \quad \int_0^T L(x(t,q), u(t,q))dt + E(x(T,q))$$

subject to

$$\begin{aligned} &x(0) - x_0 = 0 && \text{(Initial Value Constraints)}, \\ &\dot{x}(t,q) - f(x(t,q), u(t,q)) = 0 && t \in [0,T] \quad \text{(ODE Constraints)}, \\ &h(x(t_i,q), u(t_i,q)) \geq 0 && i = 0, \ldots, N \quad \text{(Nonlinear Path Constraints)}, \\ &r(x(T,q)) = 0 && \text{(Terminal Constraints)} \end{aligned}$$

This problem is solved by a finite dimensional optimization solver, e.g. Sequential Quadratic Programming, as extensively described in [36]. The strong points of SS are:

**(i)** It can use fully adaptive, error controlled state-of-the-art ODEs or DAEs.

**(ii)** It has only few optimization degrees of freedom even for large ODEs or DAEs systems.

**(iii)** Only initial guesses for the controlled degrees of freedom are needed.

On the other hand, the weak points are:

**(i)** It is not possible to use knowledge of the state trajectory $x$ in the initialization (e.g. in tracking problems).

**(ii)** The ODE solution $x(t, q)$ can depend very nonlinearly on $q$.

**(iii)** Unstable systems are difficult to treat, in some cases numerical issues may arise. This can be partially solved by including a penalty term on input variations in the cost function, as suggested in [36].

## 7.4 Direct Methods: Multiple Shooting

The direct multiple shooting (MS) method tries to combine the advantages of parallel computing with the major advantage of SS, namely the possibility to use adaptive, error controlled ODE solvers. Firstly, the controls piece-wise are again discretized on a coarse grid:

$$u(t) = q_i \quad \text{for } t \in [t_i, t_{i+1}] \tag{7.4.1}$$

where the intervals can be as large as in SS. But secondly, the ODE is solved on each interval $[t_i, t_{i+1}]$ independently, starting with an artificial initial value $s_i$:

$$\begin{aligned} x_i(t_i) &= s_i \\ \dot{x}_i(t) &= f(x_i(t), q_i), \quad t \in [t_i, t_{i+1}] \end{aligned} \tag{7.4.2}$$

By numerical simulation of these initial value problems, trajectory pieces $x_i(t, s_i, q_i)$ are obtained, where the extra arguments are introduced to denote the dependence on the interval's initial values and controls. Simultaneously with the decoupled ODE solution, the following integral is numerically computed:

$$l_i(s_i, q_i) = \int_{t_i}^{t_{i+1}} L(x_i(t_i, s_i, q_i), q_i)dt \tag{7.4.3}$$

Finally, in order to constrain the artificial degrees of freedom $s_i$ to physically meaningful values, continuity conditions $s_{i+1} = x_i(t_{i+1}, s_i, q_i)$ are imposed. The whole problem is graphically represented in Figure 7.4.

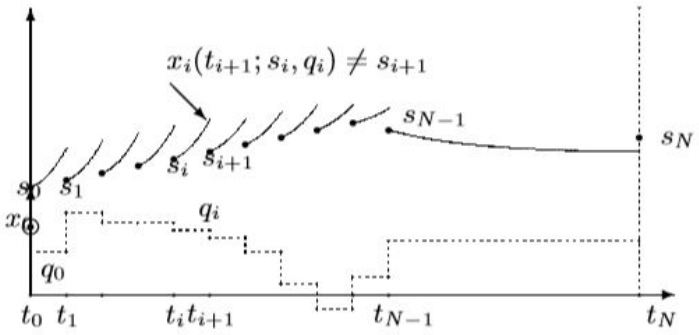


Figure 7.4: Illustration of Multiple Shooting Method

Thus, the result is the following NLP formulation, which is completely equivalent to the SS formulation out of the extra variables $s_i$ and the block sparse structure:

$$\underset{s,\,q}{\text{minimize}} \quad \sum_{i=0}^{N-1} l_i(s_i, q_i) + E(s_N)$$

subject to

$$\begin{aligned}
&s_0 - x_0 = 0 && \text{(Initial Value Constraints)},\\
&\dot{x_i}(t_{i+1}, s_i, q_i) - f(x_i(t_i, s_i, q_i), q_i) = 0 && t \in [0, T] \quad \text{(ODE Constraints)},\\
&h(s_i, q_i) \geq 0 && i = 0, \ldots, N \quad \text{(Nonlinear Path Constraints)},\\
&r(s_N) = 0 && \text{(Terminal Constraints)},\\
&s_{i+1} - x_i(t_{i+1}, s_i, q_i) = 0 && i = 0, \ldots, N-1 \quad \text{(Continuity Constraints)}
\end{aligned}$$

If all variables are summarized as $w \triangleq (s_0, q_0, s_1, q_1, \ldots, s_N)$ an NLP in the form of a finite dimensional NLP is obtained, as in Eq. (7.2.2). The advantages of MS are mostly that the knowledge of the state trajectory can be used in the initialization, and that it robustly handles unstable systems and path state and terminal constraints, in particular it is interesting to note that the terminal constraint is already satisfied in the first iteration, due to its linearity, while on the other hand it needs to be tackled explicitly in SS including in the cost function some penalty terms on terminal conditions in order to obtain a sensible trajectory, as suggested in [36]. The nonlinear effects of the continuity conditions are distributed over the whole horizon, which is seen in the discontinuities. This is in contrast to SS, where the nonlinearity of the system is accumulated until the end of the horizon, and the terminal constraint becomes much more nonlinear than necessary. As said above, MS can combine adaptivity with fixed NLP dimensions, by the use of adaptive ODE/DAE solvers. Within each Sequential Quadratic Programming iteration, the ODE solution is often the most costly part, that is easy to parallelize. The possibility to use efficient state-of-the-art ODE/DAE solvers and their inbuilt adaptivity makes MS a competitive option. From a practical point of view it offers the advantage that the user does not have to decide on the grid of the ODE discretization, but only on the control grid.

## 7.5 Simulations and Results

The construction of an optimal controller based on the previous methods is non-trivial if a discontinuous system is taken into account, the main issue being passing the information of a switching dynamics and a switching controller to the minimizer of the cost function. Due to this, the following scheme have been used. Firstly, the simulation with the Tegotae controller was run until when a steady state was reached. This was done by running the simulation for a sufficiently large time. Secondly, the peak values of the trajectory in the very last jump were taken. Now, in case these corresponded to peaks in flight phase, the dynamics was further cut up to the stance phase of the period only. In fact, this dynamics was the reference dynamics with respect to which the optimal controller was supposed to be constructed. It followed then that the flight phase was not needed to be tackled by the optimal controller, since a control would not be effective in this phase. In case a flight phase was absent, the peak point were directly taken as reference point. Subsequently, the values of position and velocity were stored and a Finite Horizon Optimal Control (FHOC) problem was solved using shooting methods. Firstly the MS only was taken into account, due to its numerical robustness, then the results were compared with a SS method, for the sake of clarity.

### 7.5.1 Optimal Controller with Multiple Shooting Only

The FHOC problem was solved with respect to these initial and final conditions, which were automatically added via the extended formulation above cited. The cost function was constructed by taking into account Eq. (5.3.1). Finally, according to the Tacit Learning formulation, the energy stored in the spring system may have had a leading role. On the other hand, if the original formulation of energy efficiency in chapter 5 was taken into account, the maximum height in the vertical excursion was substituted by the length of the body itself, due

to this considerations about the relevance of the flight phase. The MS routine was solved using again the interior-point method provided by MATLAB built-in function fmincon. The FHOC for the MS method was formulated using the norm notation $\|x\|_Q^2 = x^T Q x$ :

$$\underset{s,\,q}{\text{minimize}} \quad \sum_{i=0}^{N-1} \|q_i \dot{s}_i\|_{Q_1}^2 + \|(F_k + F_c)\dot{s}_i\|_{R_1}^2 + \|(l - s_i)\|_{L_1}^2$$

subject to

$A[s,q] = b$, (Initial Value Constraints),

$C[s,q] \geq d, \quad i = 0,\ldots,N$ (Inequality Constraints),

$s_{i+1} - y_i(t_{i+1}, s_i, q_i) = 0, \quad i = 0,\ldots,N-1$ (Continuity Constraints),

$m\ddot{s}_i(t_{i+1}, s_i, q_i) - (F_c + F_k - mg + q_i) = 0, \quad t \in [0,T]$ (ODE Constraints),

$h(s_i, q_i) \geq 0, \quad i = 0,\ldots,N$ (Nonlinear Path Constraints)

Here the initial value constraints $A[s,q] = b$ coincided with the one in order to have:

$$\begin{bmatrix} s_0 - y_{in} \\ \dot{s}_0 - v_{in} \end{bmatrix} = 0$$

And the inequality constraints $C[s,q] \geq d$ such that:

$$\begin{bmatrix} s_i - 0 + \epsilon \\ -s_i + l + \epsilon \\ \dot{s}_i - v_{min} \\ -\dot{s}_i + v_{max} \\ q_i - 0 - \epsilon \\ -q_i + F_{max} \end{bmatrix} \geq 0$$

The optimization was run over several values of the mass, in order to validate the results for different feedback dynamics, while all the other parameters were

kept at the same value as the monoped case study. On the other hand, the Tegotae controller were taken according to the policy 2 as in Eq. (6.2), in order to exploit the adaptivity of the Tegotae feedback. The values of the weights in the cost functions for each relevant simulation are reported in Table 7.1 with respect to each simulation to check the effectiveness of the weights.

Table 7.1: Weights Values for MS Only

| Simulation Case | $m$ [Kg] | $Q_1$ | $R_1$ | $L_1$ |
|---|---|---|---|---|
| MS 1 | 0.1 | $1e1$ | $-1e1$ | $-1e1$ |
| MS 2 | 0.1 | 0 | $-1e1$ | $-1e1$ |
| MS 3 | 0.1 | $1e1$ | 0 | $-1e1$ |
| MS 4 | 0.1 | $1e1$ | $-1e1$ | 0 |
| MS 5 | 0.3 | $1e1$ | 0 | 0 |
| MS 6 | 0.6 | $1e1$ | 0 | 0 |

It follows that an actual effect of the weights is restricted to the power injection by the controller, while on the other hand the optimal controller has neither access to the energy stored in the spring and damping system nor to the vertical excursion as shown in Figure 7.6, 7.7, 7.8. In fact, no particular effect is shown by removing each weight previously referred to out of the case of $Q_1$, which leads to an utterly different solution. On the other hand, the ability to dynamically adapt to the mass changes proper of the Tegotae controller is verified by the optimal controller as well, as shown in Figure 7.5, 7.9, 7.10, where the control trajectory varies accordingly to the mass change.

Further increments of the mass might require a change of the value of $\sigma$ or the use of a non linear spring in order to avoid negative values of vertical movements: the case of 7.10 was already at the limit of applicability.

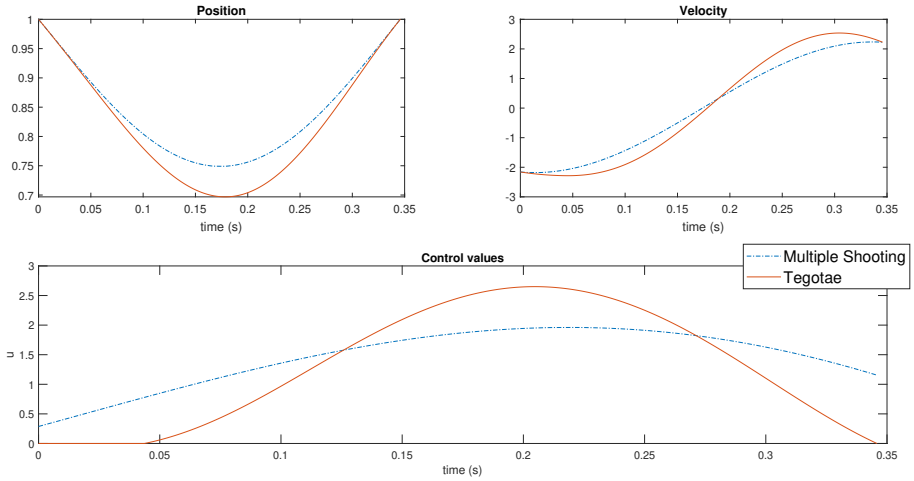


Figure 7.5: Multiple Shooting only: $m = 0.1$ Kg (Case MS 1)

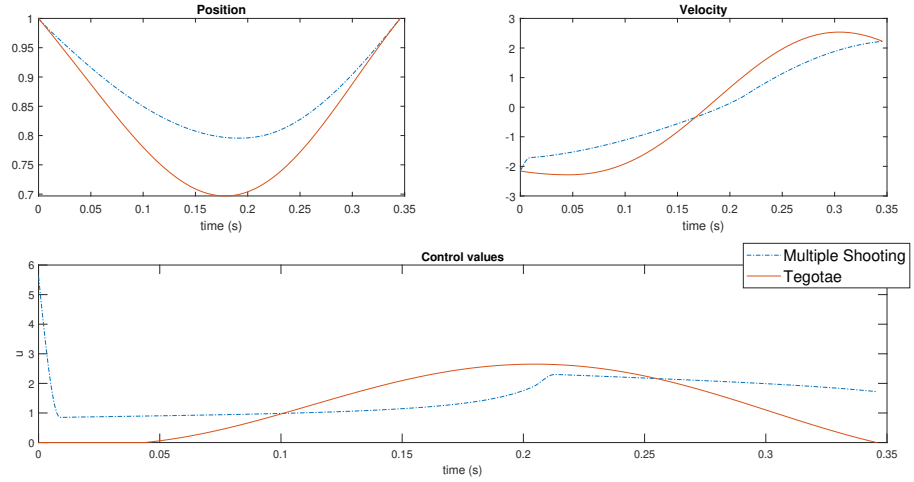


Figure 7.6: Multiple Shooting only: $Q_1 = 0$ (Case MS 2)

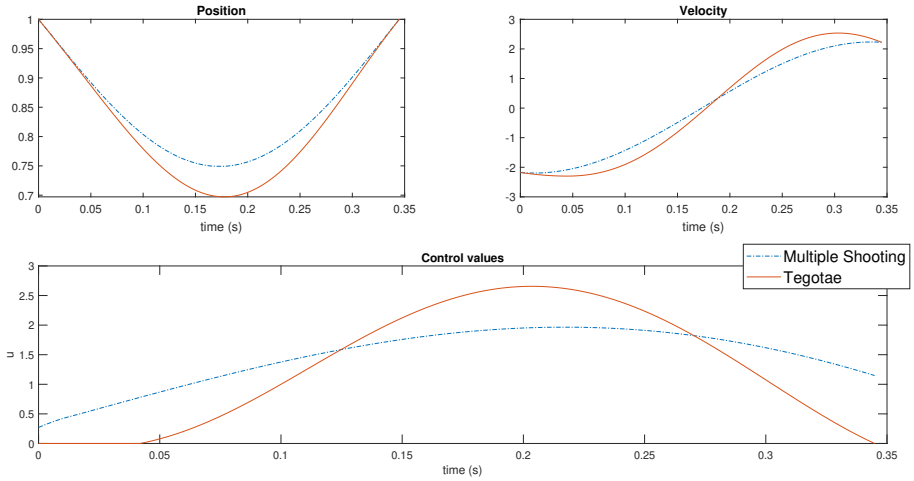


Figure 7.7: Multiple Shooting only: $R_1 = 0$ (Case MS 3)

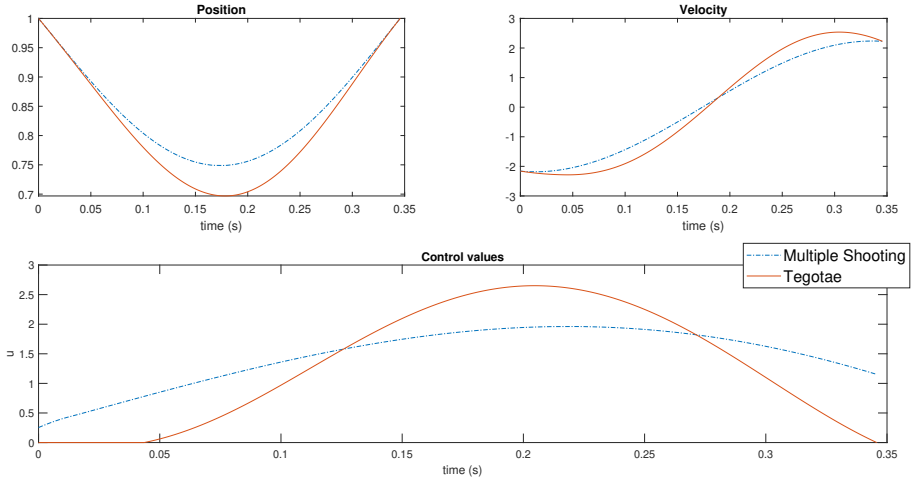


Figure 7.8: Multiple Shooting only: $L_1 = 0$ (Case MS 4)

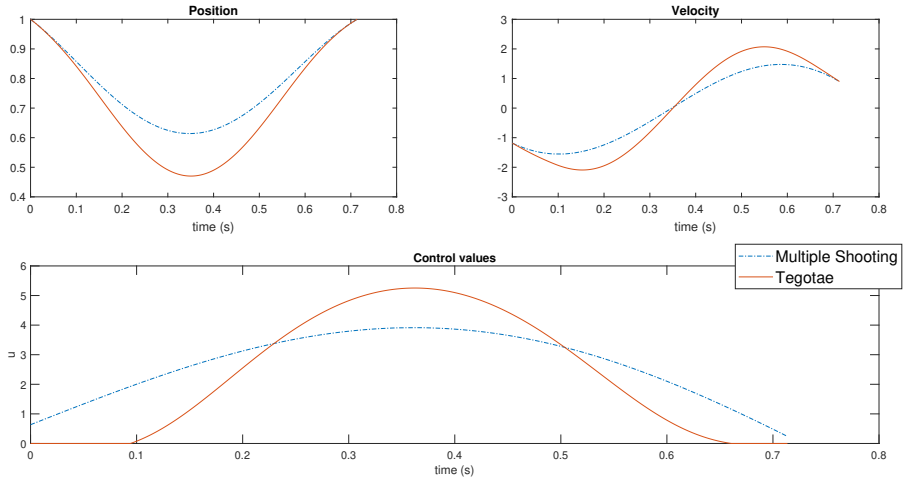


Figure 7.9: Multiple Shooting only: adaptation to $m = 0.3$ Kg (Case MS 5)

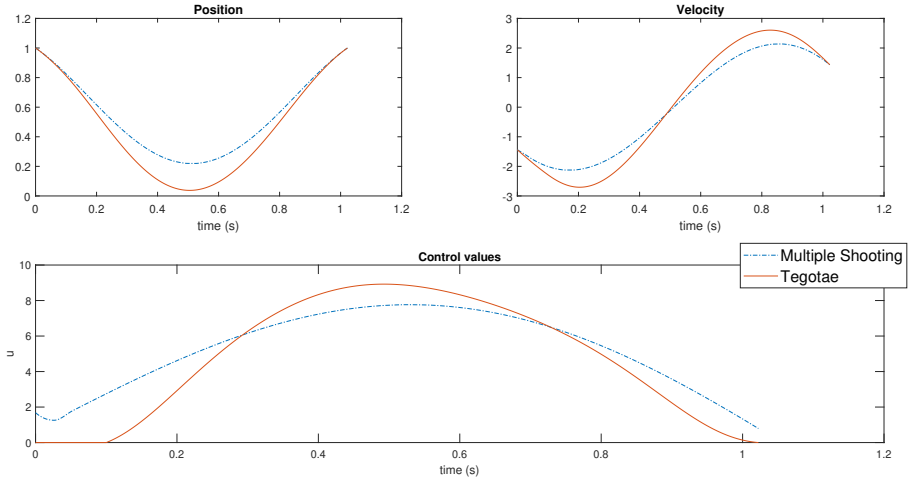


Figure 7.10: Multiple Shooting only: adaptation to $m = 0.6$ Kg (Case MS 6)

### 7.5.2 Optimal Controller with Multiple Shooting and Single Shooting

These results represent merely the MS case. In fact the SS has several practical drawbacks which motivate this choice. Firstly, it requires extremely high weights on the sensitivity function to the final conditions and on the smoothness of the control policy, conditions automatically satisfied by the continuity constraints in the MS. Secondly, its convergence is harder to be obtained. Nonetheless, it was implemented as well for comparison purposes, aiming at obtaining meaningful differences with respect to the Multiple Shooting case. The MS routine was solved using again MATLAB built-in function, on the other hand the SS routine was solved using BFGS method and SQP designed on the material provided by [36]. As far as the integration of the dynamics is concerned, the time interval have been split in 40 nodes with 2 points per sub-interval for the MS case, while a sampling time of $0.01s$ have been used for the SS case. For both cases, the integration of the dynamics have been done via an explicit Runge-Kutta method of order 4, since the restricted dynamics is now non-stiff. The step-size being $0.01s$ for both the methods. The FHOC for the SS method is formulate using the norm notation and additional weights to guarantee sensitivity to final conditions and control policy:

$$\begin{aligned}
\underset{y,\,q}{\text{minimize}} \quad & \int_0^T \|q_i\dot{y}\|^2_{Q_2} + \|F_k\dot{y}\|^2_{R_2} + \|(l-y)\|^2_{L_2} + \|q_i\|^2_{S_2} + \\
& \|q_i - q_{i-1}\|^2_{\gamma_1}\, dt + \|y_{end} - y(T)\|^2_{F_2} + \|\dot{y}_{end} - \dot{y}(T)\|^2_{F_2}
\end{aligned}$$

subject to

$A[y,\dot{y}] = b, \quad$ (Initial Value Constraints),

$C[y,\dot{y}] \geq d, \quad i = 0,\ldots,N \quad$ (Inequality Constraints),

$m\ddot{y}(t,q) - (F_c + F_k - mg + q) = 0, \quad t \in [0,T] \quad$ (ODE Constraints)

Here the initial value constraints formulation $A[y,\dot{y}] = b$ coincided with the

one in order to have:

$$\begin{bmatrix} y_0 - y_{in} \\ \dot{y}_0 - v_{in} \end{bmatrix} = 0$$

And the inequality constraints $C[y, \dot{y}] \geq d$ such that:

$$\begin{bmatrix} y - 0 - \epsilon \\ -y + l + \epsilon \\ \dot{y} - v_{min} \\ -\dot{y} + v_{max} \\ q_i - 0 - \epsilon \\ -q_i + F_{max} \end{bmatrix} \geq 0$$

In this case, $\gamma_1 = 1e4$, $F_2 = 1e10$. As previously noticed, these values resulted to be extremely high compared to the remaining weights of the cost function, shown in Table 7.1, while for the MS case the weights have been maintained the same as for MS 5 in Table 7.2.

Table 7.2: Weights Values for MS-SS Comparison

| Simulation Case | $m$ [Kg] | $Q_2$ | $R_2$ | $L_2$ | $S_2$ |
|---|---|---|---|---|---|
| MS-SS 1 | 0.3 | $1e1$ | $-1e1$ | $-1e1$ | $1e3$ |
| MS-SS 2 | 0.6 | $1e1$ | $-1e1$ | $-1e1$ | $1e3$ |
| MS-SS 3 | 0.4 | $1e1$ | $-1e2$ | $-1e2$ | $1e3$ |
| MS-SS 4 | 0.6 | $1e1$ | $-1e2$ | $-1e2$ | $1e3$ |

Interestingly, it has not been a trivial fact to obtain similar results between the two optimal controllers. It was possible to obtain similar control trends with respect to the MS case, as shown in Figure 7.11 and Figure 7.12. On the other hand, cases more similar to the Tegotae controller were reached as well, as shown in Figure 7.14 and Figure 7.13, by appropriately changing the values of the weights to higher weights for the energy stored in the spring system and for

the vertical excursion. For the SS case then, the cost function was more sensitive to the terms proper of the monoped cost function Eq. (5.3.1) and of the spring force. This might be due to the fact that the absence of continuity constraints let higher freedom to the optimizer to follow the constraints.

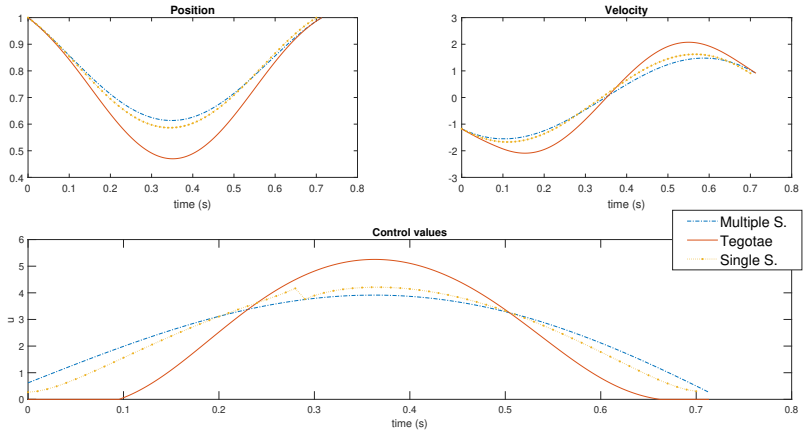


Figure 7.11: Multiple Shooting and Single Shooting similarities: $m = 0.3$ Kg (Case MS-SS 1)

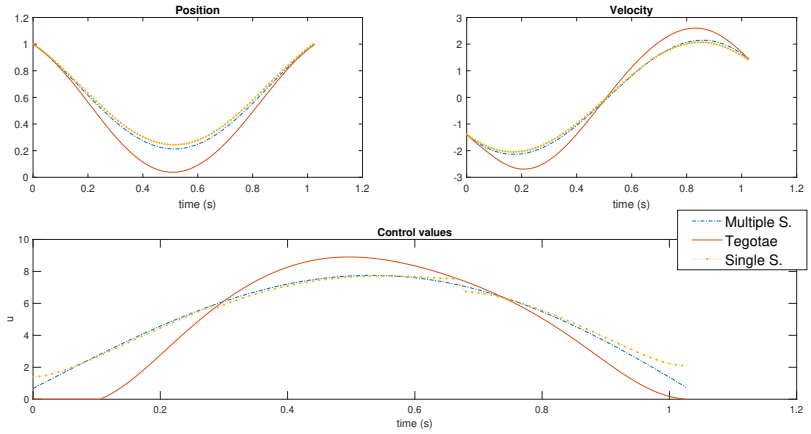


Figure 7.12: Multiple Shooting and Single Shooting similarities: adaptation to $m = 0.3$ Kg (Case MS-SS 2)

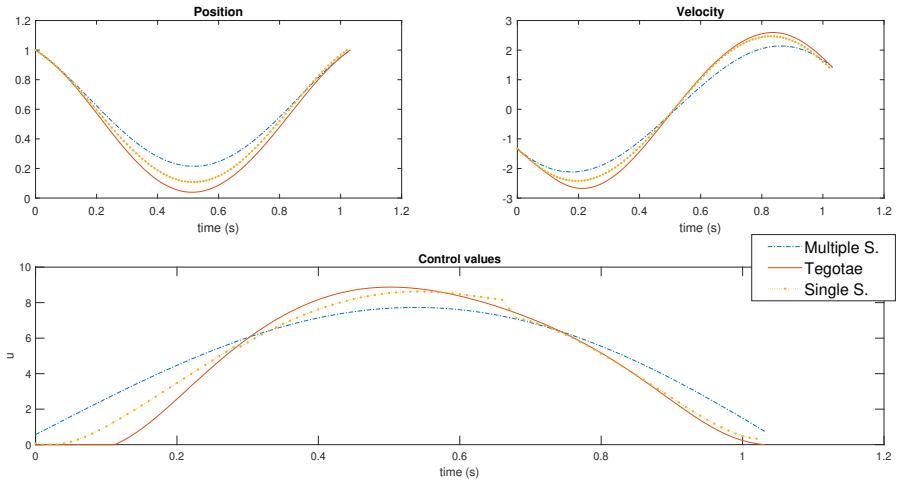


Figure 7.13: Multiple Shooting and Single Shooting differences: $m = 0.4$ Kg (Case MS-SS 3)

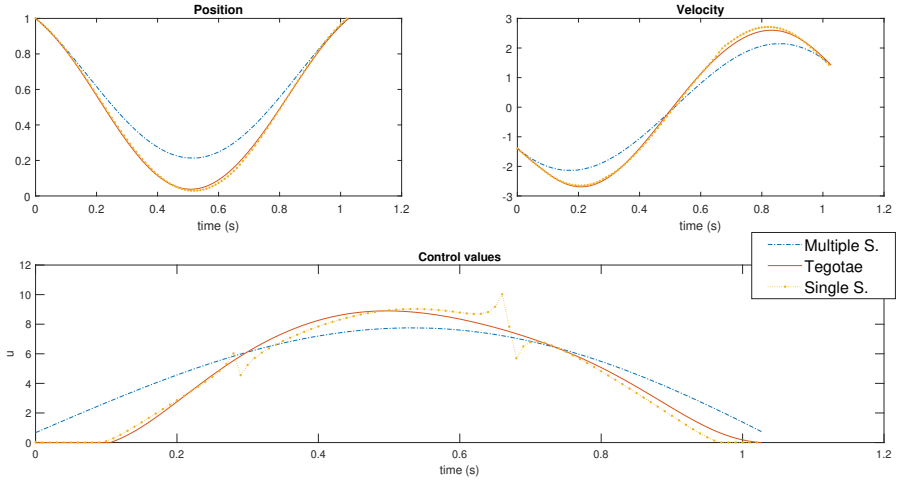


Figure 7.14: Multiple Shooting and Single Shooting differences: adaptation to $m = 0.6$ Kg (Case MS-SS 4)

# Chapter 8

# Lyapunov Exponents: Theory and Results

## 8.1 Introduction

As previously described in the Subsection , the Tegotae feedback leads to a complex dynamics. In order to further extend the stability analysis and possibly define in a rigorous way how to design the values of the parameters in the feedbacks and control laws, a dynamical analysis is performed. Due to the switching nature of both the mechanical system itself and the control law designed, in both cases of the monoped and biped bodies an analysis of bifurcations via MATCONT would not not possible, since these typologies of dynamics are not supported. Then, the analysis is conducted by studying the Lyapunov exponents of the whole system. In fact, these may require a simple dimensional extension of the ODE system, which is able to support discontinuities via an event detection. As far as the theory of Lyapunov exponents is concerned, seminal works such as [40] and [41] has been used, while for the algorithmic computation aspect, [42] and [43] have been the main reference. In our case study, the proper theory of switching systems has been found in Fillipov Systems and Fillippov Convex

Methods, as firstly introduced in [44]. Yet for brevity's sake, these references are reported but the theory itself is not treated extensively.

## 8.2 Lyapunov Exponents

Lyapunov Exponents (LEs) are mostly used to study the nature of a dynamical system, and the way this evolves over time. In particular, they are extremely effective in the study of chaotic systems. To be called chaotic, a system should show sensitive dependence on initial conditions, in the sense that neighboring orbits separate exponentially fast, on average. This means that two trajectories starting very close together will rapidly diverge from each other. A chaotic orbit is one that forever continues to experience the unstable behavior that an orbit exhibits near a source, but that is not itself fixed or periodic. It never manages to find a sink to be attracted to. At any point of such an orbit, there are points arbitrarily near that will move away from the point during further iteration. This sustained irregularity is quantified by Lyapunov numbers and Lyapunov exponents. The Lyapunov number will be the average per-step divergence rate of nearby points along the orbit, and the Lyapunov exponent the natural logarithm of the Lyapunov number. In fact, for a dynamical system whose dynamics is described by the n-dimensional smooth vector field:

$$\dot{x}(t) = f(x(t)) \tag{8.2.1}$$

It is possible to define the T-period map which, at any state x, links the state after T instants, that is:

$$\dot{x}((k+1)T) = F_T(x(kT)), \quad T > 0$$

This being a discrete-time system, it is possible to compute the LEs: $\tilde{L}_1, \ldots, \tilde{L}_n$, defined as the discrete-equivalent of the continuous case $L_1, \ldots, L_n$, $L_i = \tilde{L}_i/T$.

From the numerical point of view, it is crucial that the computation horizon $T$ is little enough to avoid numerical errors and divergence. The discrete case being descriptive of the average separation of perturbed trajectories with respect to the initial conditions, in the mono-dimensional case:

$$|\partial x(t)| \to e^{L_{x(0)}t}|\partial x(0)|$$

Where the $L_{x(0)}$ is the LE of the trajectory starting from $x(0)$.

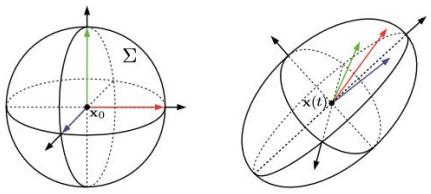


Figure 8.1: A three-dimensional example of the deformation of the sphere into an ellipsoid

In the discrete n-dimensional case, there are n different LEs. Consider the evolution of an infinitesimal sphere of perturbed initial conditions. During its evolution, the sphere will become distorted into an infinitesimal ellipsoid, as in Figure (8.1). The LEs quantifies this phenomenon. Let $\delta_k(t)$ denote the length of the $k$th principal axis of the ellipsoid. Then $\delta_k(t) = \delta_k(0)e^{L_k t}$, where $L_k$ is the $k$th LEs. When a system has a positive Lyapunov exponent, there is a time horizon beyond which prediction breaks down. Such case is, again, called chaotic, where the most universal working definition being found in [40]:

*Chaos is an a-periodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions*

Here "A-periodic long-term behavior" means that there are trajectories which do not settle down to fixed points, periodic orbits or quasi periodic orbits. For

practical reasons, we should require that such trajectories are not too rare. "Deterministic" means that the system has no random or noisy inputs or parameters.

## 8.3 Filippov Systems

The case of study falls in a special class of discontinuous systems known as the Filippov (or piece-wise smooth) systems. A Filippov system is a collection of vector fields $F_q$ defined on domains $D_q \subset \mathcal{R}^n$ separated by a set of (possibly intersecting)$(n-1)$-dimensional smooth manifolds. As also treated in [45], a solution of a Filippov system is any absolutely continuous function that satisfies almost everywhere the differential inclusion:

$$\dot{x} \in \text{co}\{F_{q_1}(x), \ldots, F_{q_n}(x)\}$$

Where co is the convex hull and $D_{q_1}, \ldots, D_{q_n}$ is the set of domains that intersect every arbitrary small neighborhood of $x$. By the above definition, a given initial condition $x$ may admit multiple solutions, whenever the differential inclusion of Eq. (8.3) admits multiple trajectories starting at $x$, and solutions may belong to the interior of a domain or they may slide along a boundary between two or more domains when this is consistent with Eq. (8.3). More precisely, if the convex hull of $F_{q_1}(x), \ldots, F_{q_n}(x)$ has no intersection with the tangent space to the boundary, the only admissible solution is one that crosses the boundary. If the convex hull intersects the tangent space to the boundary, but all vector fields locally point out of their respective domains, then locally solutions of Eq. (8.3) are bound to slide on the boundary. If the convex hull intersects the tangent space to the boundary, and some of the vector fields locally point inside their domain, then Eq. (8.3) admits both solutions that slide along the boundary and solutions that leave the boundary and enter one of the neighboring domains. Discontinuity boundaries are consequently partitioned into crossing, sliding, and escaping re-

gions. Filippov systems are a particular type of hybrid systems, where a value of the discrete variable $q$ is assigned to each of the smooth domains of the Filippov system as well as to each of the sliding regions. For each $q$, the discontinuity set $G_q$ collects all the discontinuity manifolds triggering a switch of the vector field. The switch is defined by the reset map $(R_q(x), q\prime) = R(x, q)$, whose $x$-component $R_q$ is always the identity. Now, given a reference initial condition for Eq. (8.2.1) $x_0 = x(t_0)$, and its forward-time trajectory $x(t)$, the fundamental solution matrix $\psi(t, t_0)$ is defined as the unique matrix that satisfies the relation:

$$x\prime(t) - x(t) = \psi(t, t_0))(x\prime(t_0) - x_0) + O(\|x\prime(t_0) - x_0\|^2)$$

for any possible infinitesimal perturbation $x\prime(t_0)$ of $x_0$. The fundamental solution matrix is therefore associated with the reference trajectory originating at $x(t_0)$. For smooth systems as in Eq. (8.2.1), the fundamental solution matrix can be obtained in an elegant manner by integrating the so-called variational equation as presented in [46]. It is obtained as the solution of the matrix initial value problem:

$$\dot{\psi}(t, t_0) = J(x(t))\psi(t, t_0), \quad \psi(t_0, t_0) = I \tag{8.3.1}$$

Here $J = J(x(t))$ denotes the Jacobi matrix of partial derivatives of the vector field at the point $x(t)$, the initial condition being the identity matrix $I$. Discontinuous systems however exhibit discontinuities, or saltations, in the time evolution of the fundamental solution matrix. The jumps in the fundamental solution matrix can be computed analytically by means of the theory of [47]. Such theory was finally used by [48] to calculate LEs of discontinuous systems, as in the case of analysis. Now, it is derived how the fundamental solution matrix $J$ jumps if the solution $x(t)$ crosses a hyper-surface $\Sigma$, on which the vector

field is discontinuous, using the notation as in [44]. By considering a nonlinear system with discontinuous right-hand side in the Filippov notation starting from

an initial condition $x(t=0)=x_0$:

$$\dot{x}(t) \in F(t,x(t)) = \begin{cases} f_-(t,x(t)), & x(t) \in \mathcal{V}_- \\ \text{co}\{f_-(t,x(t)), f_+(t,x(t))\}, & x(t) \in \Sigma \\ f_+(t,x(t)), & x(t) \in \mathcal{V}_+ \end{cases} \tag{8.3.2}$$

with co being the convex combination.

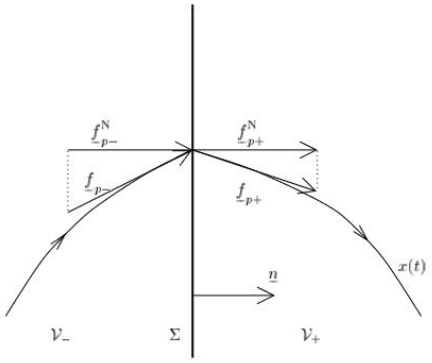


Figure 8.2: Projections of derivatives on the normal of the hyper-surface

Assume that at a certain point in time, say $t_p$, the solution $x(t)$ will cross $\Sigma$. With the definition of an indicator function we obtain:

$$s(x(t_p)) = 0$$

That is the discontinuity set is represented as a zero set of a smooth function as in [45]. At this hyper-surface there are two derivatives $f_{p^+}$ and $f_{p^-}$ which lie in the direction of the solution as denoted in Fig. (8.2). The derivatives have perpendicular components to the hyper-surface with magnitudes $f_{p^+}n^T$ and $f_{p^-}n^T$.

In these conditions the Jacobian of the discontinuity mapping at the point of discontinuity of the reference trajectory is the correction factor resetting the state of the variational system at the time of discontinuity. This Jacobian is

known in the literature of piece-wise smooth systems as the saltation matrix. Complex phenomena arises from the definition of such matrices. In particular, the saltation matrix at the onset of sliding, for example, is singular, and this makes the fundamental solution matrix singular as well. Generally, the matrix $S$ is called the saltation matrix because it describes the jump by mapping $X(t_{p^-}; t_0)$ to $X(t_{p^+}; t_0)$ in the variational equation used to compute the Lyapunov exponents. The saltation matrix can be regarded as a fundamental solution matrix from time $t_{p^-}$ to $t_{p^+}$, omitting the dependence on the state vector:

$$S = \psi(t_{p^-}, t_{p^+})$$

It follows that:

$$\psi(t, t_0) = \psi(t, t_{p^+}) S \psi(t_{p^-}, t_0)$$

The saltation matrix can be derived by inspecting the nonlinear dynamical system in the neighborhood of the occurrence of a discontinuity. It follows that it can be defined as:

$$S = I + \frac{(f_{p^+} - f_{p^-}) n^T}{n^T f_{p^-}} \tag{8.3.3}$$

In our case of analysis, even though the Jacobian in the variational equation is discontinuous, its tangent vectors are locally unvaried through $\Sigma$. This leads to the fact that the saltation matrix will correspond to the identity matrix. Due to this, the computation of the LEs through the discontinuity hyper-surface are de facto unvaried.

## 8.4 Numerical Computation of Lyapunov Exponents

In order to compute the LEs in the continuous-time cases, several ways can be followed. Tenderly, the calculation of the Lyapunov spectrum is based on matrix decomposition techniques. It can in principle be computed by a singular value (SV) decomposition or QR decomposition for a sufficiently large chosen time t. Practically, however, the computation fails for large times, as previously said. This can be avoided with the help of suitable schemes. One approach is to approximate the LEs in a finite number of time steps, for example in the QR decomposition here explained, following [49]. When dealing with continuous systems the associated matrix variational equation previously introduced is again:

$$\dot{\psi} = J\psi, \quad \psi(t_0, t_0) = I \tag{8.4.1}$$

This has to be integrated simultaneously with the Eq. (8.2.1). Taking into account once more the deformation of the initial sphere as in Figure (8.1), the $i$th LE $L_i$ of the reference trajectory measures the average exponential growth of the $i$th principal axis of the ellipsoid:

$$L_i = \lim_{t\to\infty} \frac{1}{t} \ln[\sigma_i], \quad i = 1, \ldots, n$$

with $\sigma_i$ being the $i$th singular value of the fundamental solution matrix $\psi(t, t_0)$. The computation of the LEs is then based on the computation of the fundamental solution matrix $\psi(t, t_0)$. The most popular numerical methods are variations of the discrete QR algorithm. The idea is to work on a set T of time instants $\{t_0, t_1, t_2, \ldots\}$, $t_j > t_{j-1}$, and compute the fundamental solution matrix $\psi(t, t_{j-1})$ in each time interval $[t_{j-1}, t_j], j \geq 1$. This is done by simulating the linearized dynamics:

$$\dot{\psi}^{(j)}(t, t_{j-1}) = J(x(t))\psi^{(j)}(t, t_{j-1})$$

Starting from the initial condition $\psi^{(j)}(t_{j-1}, t_{j-1}) = Q^{(j-1)}$ obtained for $j \geq 2$ from the QR decomposition $\psi^{(j-1)}({}_{tj-1}, t_{j-2}) = Q^{(j-1)}R^{(j-1)}$, with $Q^{(0)} = R^{(0)} = I$. Then, by the initial condition $\psi(t_{j-1}, t_{j-1}) = I$, it follows $\psi(t, t_{j-1})Q^{(j-1)} = \psi^{(j)}(t, t_{j-1})$ for $t \in [t_{j-1}, t_j]$, and the composition of the fundamental solution matrices gives:

$$\psi(t_j, t_0) = \prod_{k=1}^{j} \psi(t_{j-k+1}, t_{j-k}) = Q^{(j)} \prod_{k=1}^{j} R^{(j-k+1)} \tag{8.4.2}$$

The n LEs associated to the reference trajectory $x(t)$ are approximated by:

$$L_i = \lim_{j\to\infty} \frac{1}{t_j - t_0} \sum_{k=1}^{j} \ln[R_{ii}^{(k)}]$$

Accurately performing QR decomposition (typically via the Gram–Schmidt algorithm) is extremely important from a numerical standpoint, since the magnitude of each of the columns of $\psi(t, t_0)$ tends to diverge/vanish if the column is associated to a positive/negative LE. The repeated re-orthonormalization of the fundamental matrix allows us to avoid numerical cancellations. The QR decomposition of a $m \times k$ matrix P consists in a decomposition of the form:

$$P = QR = (Q^1, \cdots, Q^k) \begin{pmatrix} R_{11} & \cdots & \cdots & R_{1k} \\ 0 & R_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \cdots \\ 0 & \cdots & 0 & R_{kk} \end{pmatrix} \tag{8.4.3}$$

being Q an orthogonal $m \times k$ matrix ($Q^{iT}Q^j = \delta_{ij}$), $(1 \leq i, j \leq k)$, and R an upper triangular $k \times k$ matrix with positive diagonal elements $R_{ii} > 0$.

The solution of the LEs computation consists then in the solution of the variational problem together with the computation of the saltation matrix once the

hyper-surface $\Sigma$ is crossed. In our case of analysis, the crossing of the hyper-surface $\Sigma$ corresponding to the transition from stance phase to flight phase happens without a variation of the derivatives, leading to a mere identity matrix in Eq. (8.3.3). Due to this, the problem itself corresponds merely to the integration of the variational equation. Finally, it is important to finally specify how this calculation actually corresponds to the calculation of the discrete-time LEs. As reported in [43] and previously said, the values will have to be further divided by the period of the T-period map.

## 8.5 Simulations and Results

As previously said, the main objective of the current analysis is the study of the dynamics in the transition between one gait and one other in the biped case. These being guaranteed by the group theoretic properties of the system, it is in our interest how this transition is obtained de facto. The calculation of the Lyapunov exponents was done via MATLAB code on the basis of the previous theory. Again, the event detection function was used to avoid numerical errors in the computation of the LEs. The mechanical parameters were left unvaried with respect to the biped case-study. The number of cycles assumed to be sufficient for the convergence of the LEs was $N_{eps} = 10$, due to the relatively simple dynamics. This was done after checking the actual fast convergence on several trials. Finally, the error in this convergence were set to be $\epsilon = 0.01$. The main characteristic of the in-phase and anti-phase trajectories being the relationship between the velocities of the two legs $v_1, v_2$, the gait was studied via the introduction of the following functional test function:

$$G = \frac{1}{T^\star} \int_{t^\star}^{t^\star + T^\star} \frac{v_1 v_2}{\|v_1\| \|v_2\|} dt \tag{8.5.1}$$

The integration of the functional was performed over the period of oscillation

$T^{\star}$, calculated numerically as the mean distance between each peak of oscillation. The relationship between the period of oscillation in the CGPs and the mechanical value was usually almost $1:1$, out of the transition phases here reported, so that numerical errors induced in this procedure were assumed to be negligible. This functional gives values coinciding with $+1$ in the in-phase hopping, with $-1$ in the anti-phase case. Furthermore, the simulation was performed in a continuation-wise manner. That is, after the perturbation of the bifurcation parameter, the initial condition was selected as the final state for the previous value of such parameter. In this way, it was possible to actually follow the attractors over their bifurcation curves and avoid to fall into other basin of attractions.

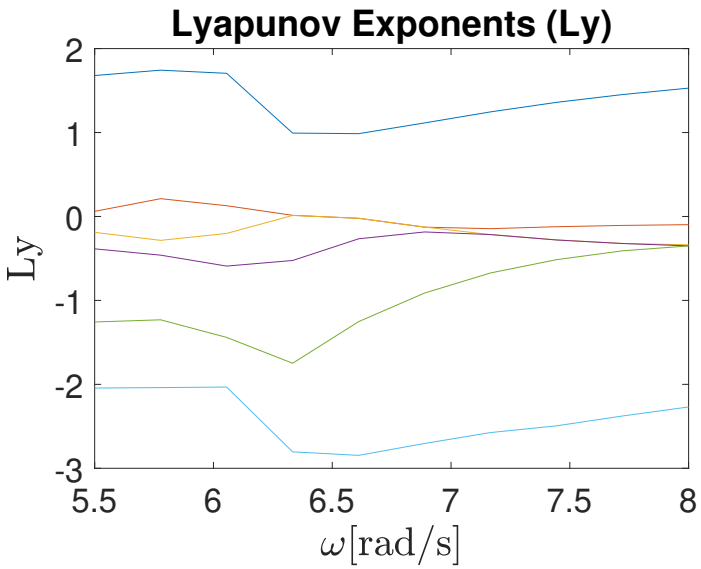


Figure 8.3: LEs trend without hysteresis approach

The initial trend of the LEs is reported in Figure (8.3). It is apparent that there is a critical value $\omega_c$ for which the LEs undergo a qualitative change in their variation. For $\omega < \omega_c$ the greatest Lyapunov Exponent results to be decreasing, while for $\omega > \omega_c$ it increases. In Figure (8.3) the perturbation on the bifurcation parameter corresponded to $\Delta\omega = 0.2$ rad/s.

Finally, in order to determine possible bifurcations, since the first result was not effective, the perturbation of the parameter $\omega$ was performed in a bidirectional manner, from the minimum value to the maximum value and back again to the minimum value. This was done in order to underline possible hysteresis effects. This means that the attractor follows different paths during the perturbation in each direction, and that it settles on the same curves after a while. By doing this, and by reducing the window of perturbation around $\omega_c$ and the perturbation to $\Delta\omega = 0.02$ rad/s, it was possible to notice an interesting behavior. By perturbing $\omega$ in an increasing manner, an example of the resulting trajectories is reported in Figure (8.4), Figure (8.5), and Figure (8.6) where the time is normalized. It is apparent how the trajectory is increasingly perturbed by increasing the values of $\omega$, passing from an in-phase hopping to an anti-phase hopping. Subsequently once $\omega_c$ is surpassed, the direction of perturbation is inverted and the hysteresis effect is noticed. In fact, the value of $\omega$ at which a gait transition is shown changes, as reported in Figure (8.7), Figure (8.8) and Figure (8.9). In particular, for $\omega = 6.27$ rad/s the gait is different on the basis of the direction of perturbation. The same happens for $\omega = 6.3$ rad/s, as reported in Figure (8.10) and Figure (8.11). Finally, the hysteresis is shown for the three biggest exponents in Figure (8.12) together with the test function in Figure (8.13), where the two different lines represent the two different directions. In order to have higher It is clear then in our case of analysis there are two different attractors, which become gradually unstable/stable respectively together with the perturbation of the bifurcation parameter. This justifies the co-existence of both gaits on the basis of the initial conditions, often shown in more complex mechanical cases and already noticed in 6, and it also gives a motivation to the fact that the test function seldom falls in the other gait case, even before the bifurcation.

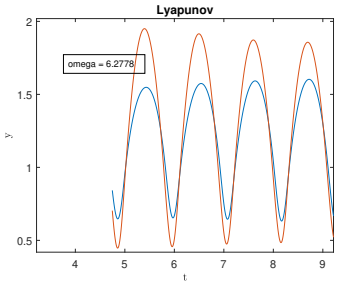


Figure 8.4: Trajectory sample along the forward continuation: in-phase hopping

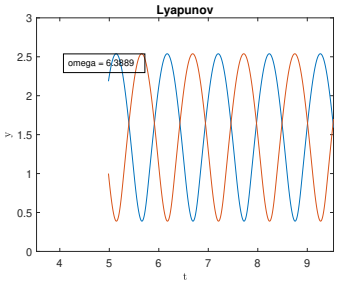


Figure 8.5: Trajectory sample along the forward continuation: anti-phase hopping

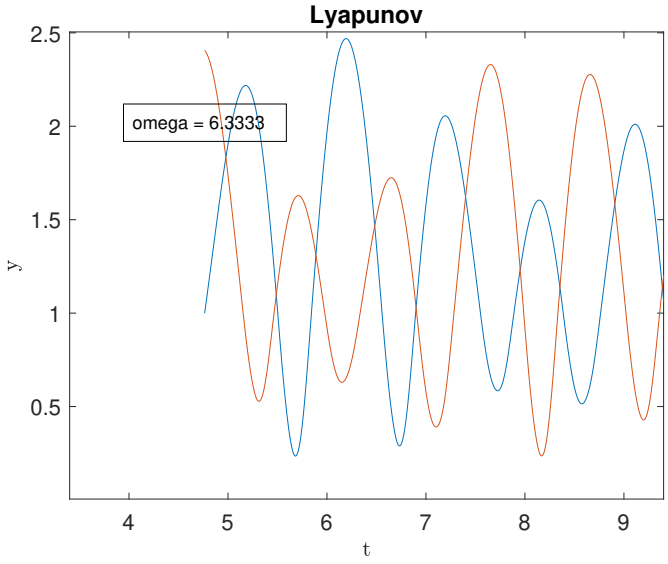


Figure 8.6: Trajectory sample along the forward continuation: gait transition

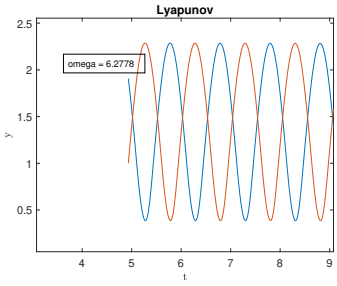


Figure 8.7: Trajectory sample along the backward continuation: hysteresis effect

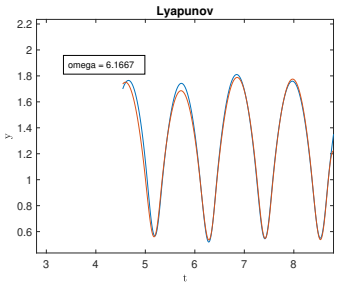


Figure 8.8: Trajectory sample along the backward continuation: in-phase hopping regeneration

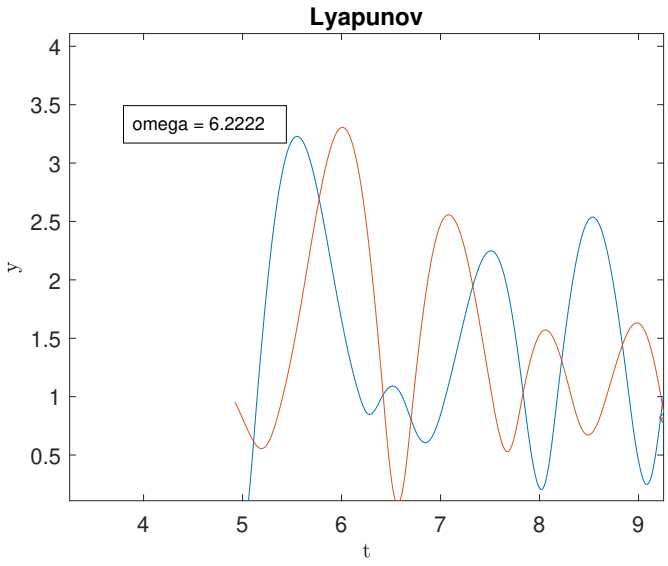


Figure 8.9: Trajectory sample along the backward continuation: gait transition at different $\omega_c$

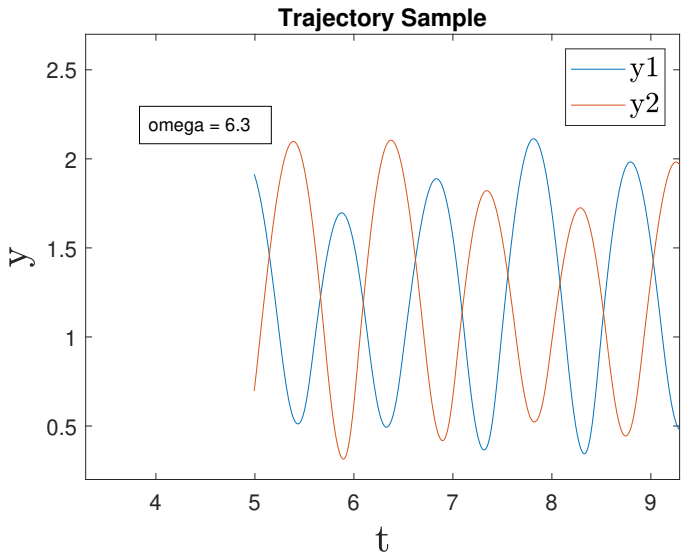


Figure 8.10: Hysteresis effect example: gait transition in forward continuation

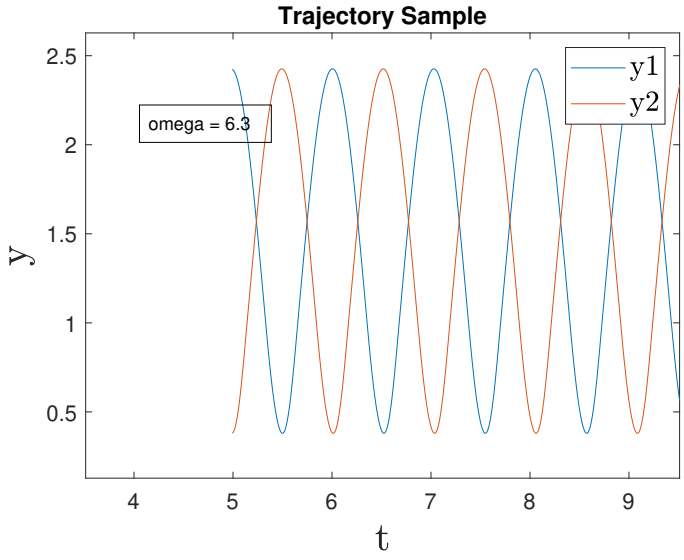


Figure 8.11: Hysteresis effect example: anti-phase hopping in backward continuation

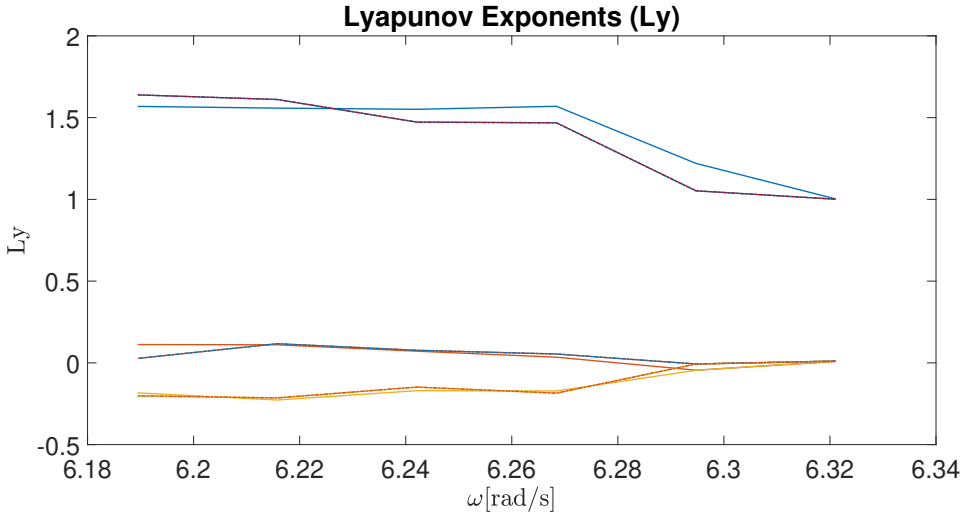


Figure 8.12: Hysteresis effect of the Lyapunov exponents

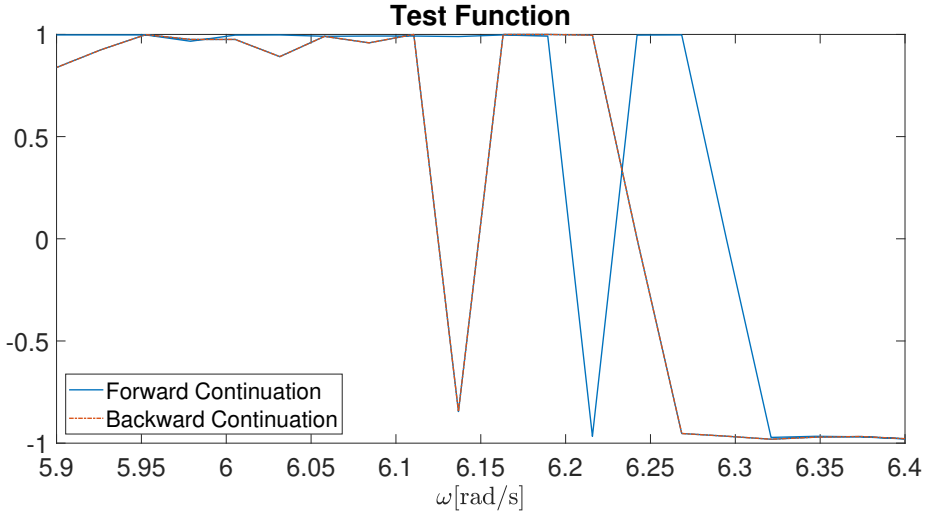


Figure 8.13: Test Function trend

# Conclusions

Central Pattern Generators are still an open issue, both in their biological foundation and in their robotic application. Nonetheless, the deep awareness of the fact that the design of a biological controller, that is the nervous system, happened to evolve in a strict relation to the body itself, this awareness has been proven to lead not only to robust behaviors, but also to enhanced performances in energy efficiency, which is believed to be one of the main leading factors in terms of biological fitness. A smart design with a clear purpose, such as in the Tegotae approach, has been shown to be able to lead to a minimalist architecture with broad positives. The analysis methodologies showed these last, yet also they underlined how little is the knowledge of the complex dynamics of such architectures. Finally, the main theoretical contributions, such as the comparison with other learning frameworks and the group theoretic analysis, were not directly applied to derive novel results, yet they suggested new paths for further research on these topics, and their effective appllication is still to be found.

In conclusion, we hope that showing not only the positives of our research but also the limits will generate new questions. Because this is how science evolves.

# Bibliography


[1] A.J. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural networks : the official journal of the International Neural Network Society*, 214:642–53, 2008.

[2] H. Kimura, Y. Fukuoka, and A.H. Cohen. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *International Journal of Robotics Research*, 26:475–90, 2007.

[3] M. Wang, J. Yu, M. Tan, and G. Zhang. A cpg-based sensory feedback control method for robotic fish locomotion. *Proceedings of the 30th Chinese Control Conference, CCC 2011*, 01 2011.

[4] A. Cangelosi, J. Bongard, M. H. Fisher, and S. Nolfi. *Embodied Intelligence*. Springer-Verlag, 2015.

[5] D. Owaki, M. Goda, S. Miyazawa, and A. Ishiguro. A minimal model describing hexapedal interlimb coordination: The tegotae-based approach. *Frontiers in Neurorobotics*, 11:29, 2017.

[6] M. Golubitsky and I. Stewart. *The Symmetry Perspective: From Equilibrium to Chaos in Phase Space and Physical Space*. Birkhauser Basel, 2002.

[7] G. Martius and R. Der. Novel plasticity rule can explain the development of sensorimotor intelligence. *PNAS*, 2015.

[8] M. Hayashibe and S. Shimoda. Synergetic motor control paradigm for optimizing energy efficiency of multijoint reaching via tacit learning. *Frontiers in Computational Neuroscience*, 8:21, 2014.

[9] E. Marder and D. Bucher. Central pattern generators and the control of rhythmic movements. *Current Biology*, 11:R986–96, 2001.

[10] Y. Kuramoto. Collective behavior of coupled phase oscillators. *The handbook of brain theory and neural networks*, pages 223–26, 2003.

[11] S. Schaal, P. Mohajerian, and A.J. Ijspeert. Dynamics systems vs. optimal control – a unifying view. *Progress in Brain Research*, 165.

[12] A.J. Ijspeert, A. Crespi, D. Ryczko, and J. Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420, 2007.

[13] I.A. Rybak, N. A. Shevtsova, M. Lafreniere-Roula, and D. A. McCrea. Rybak ia, shevtsova na, lafreniere-roula m, mccrea da. modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *The Journal of Physiology*, 2006.

[14] D. Yuan and J. Tian. Periodic synchronization in a system of coupled phase oscillators with attractive and repulsive interactions. *Frontiers of Physics*, 13, 2018.

[15] Y. Kuramoto. *Chemical Oscillations, Waves and Turbulence*. Springer-Verlag, 1984.

[16] F. C Hoppensteadts and E. M. Izhikevich. *Weakly Connected Neural Networks*. Springer-Verlag, 1997.

[17] G. Filatrella, A. H. Nielsen, and N.F. Pedersen. Analysis of a power grid using a kuramoto-like model. *European Physics Journal*, 2008.

[18] M. Rohden, A. Sorge, M. Timme, and D. Witthaut. Self-organized synchronization in decentralized power grids. *Physical Review Letters*, 109.

[19] F. Dorfler, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. *Proceedings of the National Academy of Sciences*, 110.

[20] L. Righetti and A. J. Ijspeert. Pattern generators with sensory feedback for the control of quadruped locomotion. *2008 IEEE International Conference on Robotics and Automation*, May 2018.

[21] S. Shinomoto and Y. Kuramoto. Phase transitions in active rotator systems. *Progress of Theoretical Physics*, 75, 1986.

[22] M. Golubitsky and L. Buono. Models of central pattern generators for quadruped locomotion: I. primary gaits. *Journal of Mathematical Biology*, 42:291–326, 2001.

[23] M. Golubitsky and I. Stewart. Nonlinear dynamics of networks: the groupoid formalism. *Bulletin of the American Mathematical Society*, 43, 2006.

[24] M. Golubitsky, K. Josic, and E. Shea-Brown. Rotation, oscillation and spike numbers in phase oscillators networks. *Journal of Nonlinear Sciences*, To appear.

[25] T. Bröcker and T. Dieck. *Representations of Compact Lie Groups*. Springer-Verlag Berlin Heidelberg, 1985.

[26] G. A. Miller. *Theory and applications of finite groups*. New York, John Wiley & Sons, inc., 1916.

[27] V. Berenz, F. Alnajjar, M. Hayashibe, and S. Shimoda. Generalization of the tacit learning controller based on periodic tuning functions. *International Conference on Biomedical Robotics and Biomechatronics*, 2014.

[28] V. Berenz, F. Alnajjar, M. Hayashibe, and S. Shimoda. Tacit learning for emergence of task-related behavior through signal accumulation. *Emergent Trends in Robotics and Intelligent Systems*, 316:31–38, 2015.

[29] M. Hayashibe and S. Shimoda. Synergetic motor control paradigm for optimizing energy efficiency of multijoint reaching via tacit learning. *Frontiers in Computational Neuroscience*, 2014.

[30] D. O. Hebb. Organization of behavior. *Journal of Clinical Psychology*, 6(3):307–307, 1950.

[31] W. Gerstner. Hebbian learning and plasticity. 2011.

[32] G. Schöner and A. Schutte. *Dynamic Field Theory: Foundations: A Primer on Dynamic Field Theory*. 11 2015.

[33] J. Buchli, F. Iida, and A.J. Ijspeert. Finding resonance: adaptive frequency oscillators for dynamic legged locomotion. pages 3903–09, Oct 2006.

[34] M. Diehl, H.G. Bock, H. Diedam, and P. Wieber. Fast direct multiple shooting algorithms for optimal robot control. *Fast Motions in Biomechanics and Robotics*, 2005.

[35] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings 9th IFAC World Congress Budapest*, 1984.

[36] L. Fagiano. Constrained numerical optimization for estimation and control. *Lecture notes*, 2019.

[37] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambdridge University Press, New York, 2009.

[38] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.

[39] R. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc. 60*, 1954.

[40] S.H. Strongatz. *Nonlinear Dynamics and Chaos.* Addison-Wesley, 1994.

[41] K.T. Alligood and J.A. Y. Sauer. *Chaos - An Introducton to Dynamical Systems.* Springer, 1996.

[42] K. Geist, U. Parlitz, and W. Lauterborn. Comparison of different methods for computing lyapunov exponents. *Progress of Theoretical Physics*, 1990.

[43] F. Dercole and C. Piccardi. Lyapunov exponents. *Lecture notes*, 2012.

[44] A. F. Filippov. Differential equations with discontinous right-hand sides. *Mathematics and Its Applications*, 1988.

[45] F. Bizzarri, A. Colombo, F. Dercole, and G. S. Gajani. Necessary and sufficient conditions for the noninvertibility of fundamental solution matrices of a discontinous system. *Journal of Applied Dynamical Systems*, 2016.

[46] T. S. Parker and L. O. Chua. *Practical Numerical Algorithms for Chaotic Systems.* Springer-Verlag, 1989.

[47] M. A. Aizerman and F. R. Gantmakher. On the stability of periodic motions. *Journal of Applied Mathematics and Mechanics*, 1980.

[48] P.C. Muller. Calculation of lyapunov exponents for dynamc systems with discontinuities. *Chaos, Solitons and Fractals*, 1995.

[49] G. Benettin, L. Galgani, A. Giorgilli, and J. Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems. *Meccanica*, 1980.