UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Gait modulation of a humanoid robot, using bio-inspired mechanisms

Dissertation presented by
Philippe GREINER

for obtaining the Master's degree in
Electro-mechanical Engineering
Option: Mechatronics

Supervisors
Renaud RONSSE, Nicolas VAN DER NOOT

Readers
Auke IJSPEERT, Paul FISETTE, Philippe CHATELAIN

Academic year 2016-2017

Acknowledgements

This Master Thesis was carried out both at the Biorobotics Laboratory of the Ecole Polytechnique Fédérale de Lausanne (Lausanne, Switzerland) and at Université catholique de Louvain (Louvain-la-Neuve, Belgium). I would like to express my gratitude to the people who helped me along the way; without them, this work would never have been possible.

I would like to thank *Professor Renaud Ronsse*, my supervisor, for his continuous support throughout the academic year, and for always making time for (Skype) meetings when needed. His advice and experience were extremely useful during the whole work, bringing new ideas and providing useful feedback. Thanks to him, I was also able to access the Biorobotics Laboratory in Lausanne, which was very useful for my work.

I also express my thanks to *Nicolas Van der Noot*, my co-supervisor. Nicolas' insight and expertise concerning the CoMan and the controller he has developped can't be overstated, and he has always been available and responsive for any question, even the most obvious ones. Throughout the academic year, our weekly meetings have helped me set the pace for my work, clarify my objectives and solve day-to-day problems in a breeze.

I would like to express my sincere gratitude to *Professor Auke Jan Ijspeert*, for welcoming me at the Biorobotics Laboratoy in Lausanne. His sincere interest in the project, and very on point advice during the first semester, were of great help.

Philippe Greiner

Louvain-la-Neuve, June 2017

Abstract

Gait modulation of a humanoid robot, using bio-inspired mechanisms

by Philippe GREINER

Today, more and more bipedal humanoid robots are being used, for a variety of purposes. These can range from helping out in everyday situations, to looking through debris in devastated areas. However, bipedal robots are still far from perfect: they lack versatility and robustness to be used more readily in more wide-ranging situations. Two main approaches exist to control the gait of bipedal robots: top-down approaches concentrate on mathematically proven methods to design stable gait controllers; these approaches often result in high computational costs, low energy-efficiency and low versatility. The second kind of approaches are bottom-up approaches. These make use of human motion analysis, biomechanics and neuro-scientific research and try to transfer those to the control of bipedal robots. Such an approach is used in this work, trying to make an existing bio-inspired gait controller more versatile by enabling precise control of the robot's gait.

The existing bio-inspired controller is first introduced, as well as all the tools necessary for the development and simulation. Then, the optimization procedure used to obtain satisfying gaits throughout this work is introduced, together with an example of 'natural gait' obtained with the controller. After that, the work is presented as *4 successive experiments* to adapt the gait controller and enable the control of gait features. In the first experiment, a sensitivity analysis is performed to help guide the development of the next experiments. In the second experiment, a first controller is developed enabling to control two gait features (step height and step length) by continuous modulation. In the third experiment, a second way to develop a controller for step height and step length is explored. In the last experiment, new modulations are tested, allowing to modify the gait for a few steps only, enabling obstacle avoidance. Finally, the main results and take-aways from all the experiments are summarized in the conclusion, and future perspectives are discussed.

Contents

List of Abbreviations

CPG	**C**entral **P**attern **G**enerator
DOF	**D**egree **o**f **F**reedom

Leg Sagittal Muscles

HFL	**H**ip **Fl**exors
GLU	**Glu**teus
HAM	**Ham**string
BFSH	**B**iceps **F**emoris **S**hort **H**ead
RF	**R**ectus **F**emoris
VAS	**Vas**tus
GAS	**Gas**trocnemius
SOL	**Sol**eus
TA	**T**ibialis **A**nterior

Chapter 1

Introduction

Bipedal locomotion in humanoid robots

Today, more and more bipedal humanoid robots are being developed. This kind of robots can be used in a wide variety of situations, ranging from looking through debris in devastated areas [Wag+15] to helping out in everyday situations [Sak+02]. Because bipedal humanoid robots have a locomotion system so similar to humans, they have a big advantage over wheeled or four-legged robots: they can integrate into our normal environment in a more seamless way. To achieve this however, bipedal robots still have a long way to go. For now, their gait is often energy inefficient, they can handle few perturbations and they are often not able to recover after a loss of balance [Ijs08]. These limitations hinder existing robots from being used to their full potential. There is hence a growing need for more human-like bipedal robots, both in terms of performances, energy-efficiency and gait modularity.

Top-down versus bottom-up approach Bipedal locomotion in humanoid robots can be achieved through many ways. The two fundamentally different approaches could be called the *top-down* and *bottom-up* approaches. In this case, top-down approaches are understood as methods using mainly insights from industrial robotics. They make use of mathematical concepts like multi-body dynamics, (non-)linear control theory, forward and inverse kinematics and/or dynamics [Luk10]. Bottom-up approaches on the other hand use insights from human motion analysis, biomechanics or neuro-scientific research and try to transfer those to technical systems.

The advantages of a top-down approach, most often based on the use of the Zero-Moment Point (ZMP) [VB04], are mainly found in its ability to predict exact motions, thanks to its use of mathematically sound concepts and multi-body modeling techniques. The ZMP criterion is an indicator of dynamic stability [VB04] which can be seen as a generalization of the concept of center of mass for dynamic conditions. Indeed, for static bodies, the straightforward stability criterion is that the vertical projection of the center of mass must stay within the support polygon formed by the convex hull of the feet in contact with the ground. For moving bodies, accelerations have to be taken into account. This is where the ZMP is needed: it is the point on the ground where the tipping moment (i.e. the component of the moment that is tangential to the supporting polygon) acting on the biped, due to gravity and inertia forces, equals zero [SB04].

Such top-down approaches however also face many drawbacks. These approaches present high computational costs and do not make use of the natural dynamics of the human body, hence producing slow, unnatural movements that present low energy-efficiency [NB14]. Another drawback is the poor resistance against external perturbations, and no appropriate reaction when balance is lost [Dal11]. In particular, ZMP-based methods rely on full local controllability (i.e. each point of the gait cycle

is stable), which is not necessary to ensure a stable walking gait, then leading to a higher energy consumption [Dal11].

The emerging concept called 'Limit Cycle Walking' considers the gait as a limit cycle whose global stability is prevalent to the local stability [HW07]. Bio-inspired bottom-up approaches are promising in this regard, by making use of passive dynamics, providing local torque controls instead of providing fixed trajectory controls to the joints. Because they do not require full-featured dynamic models, biological approaches are also less computationally heavy [Luk10]. One of these approaches is the muscle-reflex approach developed by Geyer & al. [GH10]. Their approach is based on the use of local torque control through modelisation of muscles.

Biological approaches are often based on a central pattern generator (CPG) [Ijs08; DKI14] or on the use of reflexes [GH10]. A CPG is composed of coupled neural oscillators and generates spatio-temporal patterns of activity without external inputs. Neurophysiological studies of animals revealed that rhythmic movements are controlled by such rhythm generating networks in nervous systems [Del80; Tag94].

Other advantages of CPGs mentioned by [Ijs08] are the handling of perturbations inherent to stable rhythmic patterns and the few control parameters needed to modulate the gait with this approach. CPG-based controllers also present disadvantages: as of today, an adequate design methodology for CPGs is yet missing. Moreover, a solid mathematical foundation for describing CPGs does not exist, making it difficult to prove the stability of a particular CPG-controller.

In this project, the controller used and developed is a bio-inspired controller, striving to achieve modular, energy-efficient locomotion with limited computational costs. The model used for this controller is based on the bio-inspired model presented in [GH10]. In this work by Geyer & al., a robot is controlled using only reflexes i.e. stimulations computed using the current state of the joints. The controller presented in this master thesis is based on the controller developed in [NRI15], and runs on the CoMan robot. The CoMan robot is a humanoid robot with compliant joints, modelled to the size of a 5-year-old child.

In this controller, Geyer's model is combined with a CPG. The CPG is mainly used to control the proximal muscles (close to the body torso, i.e. the muscles close to the hip in this case) while the reflexes are kept for the control of mainly the distal muscles (muscles further out on the limbs, i.e. the muscles closer to the ankle in this case).

Gait modulation in humanoid robots Like previously stated, one of the major flaws in existing bipedal robots is their limited capacities in terms of gait adaptation and modularity. However, gait modularity is crucial to achieve more reliable locomotion and seamless integration into human environment, hence increasing usability of bipedal robots overall.

Like stated by Matthis & al. [MBF15], humans are able to adapt their gait to achieve precise foot placement in no more than two steps. However, humanoid walkers still struggle to achieve this. Several approaches to modulate the gait of human walkers exist, but most concentrate on the modulation of the speed of the robot, without being concerned about precise gait features such as step height, step length or step period and precise foot placement [SG15b; SG12; Sap+16; NRI15]. The applications of a speed-modulated human walker are vastly different than those that can be associated with a human walker with precise foot placement, with adaptation of step height and step length. Moreover, the approaches that exist concerning precise foot placement are based on the control of the trajectories of the feet [Bru04], and generally use stiff motors to control the joints [Cec11].

Indeed, when trying to navigate complex terrain or to avoid obstacles, it is of course necessary to have a robust controller [Her15], but purposeful gait modulation is sometimes necessary for bigger perturbations. This is the goal of this project: modulate the gait features to achieve precise control of the gait features, hence enabling precise foot placement and purposeful obstacle avoidance.

Objectives of the thesis

In this Master Thesis, the goal is to modify an existing bio-inspired controller to enable gait modulation. In particular, the goal is to achieve the computation of high-level commands to drive the precise robot behavior, like foot landing location and obstacle avoidance.

The **primary objective** of this project is hence to develop high-level commands allowing for precise control of the robot behavior.

A **secondary objective** of this project is to develop commands enabling obstacle avoidance. Two different types of commands will be considerd : firstly, a continuous modulation command which will generate stable gait patterns. Secondly, a command enabling next-step obstacle avoidance, i.e. modulate the robot behavior for a brief amount of time only, to avoid an obstacle, before going back to a continuous walking pattern.

The **third and last objective** of this project is to apply all these developments in a simulator in the context of teleoperation – with this simulator predicting the robot behavior, allowing a human operator to adjust the robot goals if necessary. This last objective, realized in collaboration with Nicolas Van der Noot, will not be tackled in this report.

Decomposition into precise objectives

To correctly understand the different steps undertaken to successfully carry out this project, it is necessary to start by decomposing the objectives into a road-map. We hence define here intermediary objectives needed to obtain the desired results. First of all, in order to drive the precise robot behavior, the chosen criterion was the ability to control the **step height and step length**. Those two parameters are enough to fully determine the foot landing locations and avoid obstacles; the speed could be controlled by adding the step period as a third parameter. In this report, the control of step height and step length is thus tackled, leaving the third parameter (step period) for further work.

The first objective, which can now be understood as the development of high-level commands to control *step height and step length*, can be split into the following steps :

1. Analysis of step height and step length depending on muscle activations. This will help to determine which parameters of the control have a notable influence on the considered gait parameters.

2. Add the necessary inputs, allowing modulation of the previously found control parameters, hence enabling modulation of step height and step length.

The second objective can also be divided into two steps : firstly, the commands developed for the first objective have to be precisely controlled, allowing to predict the steady-state behavior of the robot as a function of the commands sent to it. Secondly, discrete – instead of the steady-state commands previously developed – commands

will be developed, that are able to modify the robot behavior for one or two steps only, enabling obstacle avoidance.

These two objectives, and more precisely the four sub-objectives defined in this section, will pave the way to a successful – and useful – modulation of the robot gait.

Executive summary

This report is split up into 6 chapters. Because different ways to achieve the previously mentionned objectives are successively explored, and with varying results, the different approaches are labeled as *experiments*. Different experiments hence also make different assumptions, which will be detailed in the corresponding chapters. The structure of the work is presented in Figure 1.1, and the content of each chapter shortly detailed hereafter.

Chapter 2 describes the robotic platform used to develop the controller, and the robot being simulated. It also describes the existing controller used as a starting point for this project, and the modifications applied to it in the context of this project.
Chapter 3 describes the optimization procedure used in the whole master thesis, and presents and analyzes one resulting gait in details. This gait, refered to as *natural gait*, is obtained with the controller developed in chapter 6. It is analyzed and compared to human walking data, in order to give the reader a good insight into the different aspects of the gait, and the possibilities of the controller.

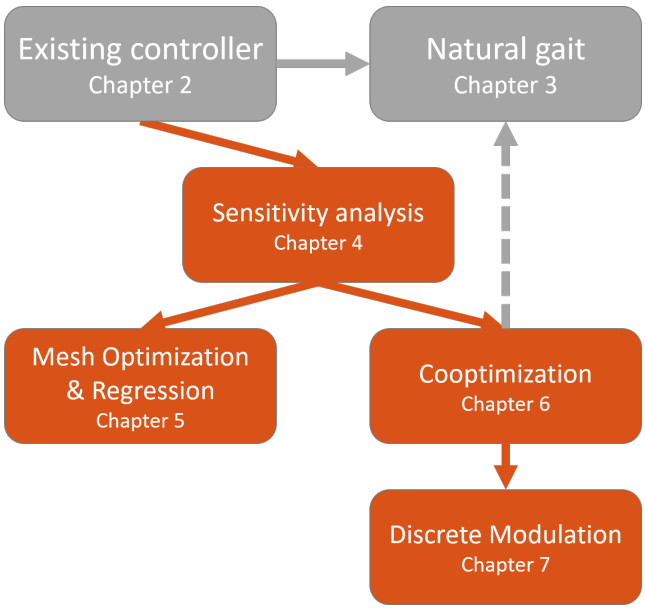


FIGURE 1.1: Structure of the master thesis, with the 'development' chapters in grey and the experiments in red.

Chapter 4 describes the first experiment: a sensitivity analysis done on the reference gait of this controller. The goal of this sensitivity analysis is to determine the controller parameters (defined in chapter 2) that have an important influence on the step height and step length. At the end of this chapter, the parameters having the most impact on step height and step length are selected to be modulated in the next experiment.

In chapter 5, a first controller of step height and step length is developed. This controller is obtained by performing a mesh optimization – computing controller parameters for a range of different gait characteristics – and then performing a regression to continuously modulate between all the points of the mesh. These modulation rules are described and tested.

A second continuous modulation controller is developped in chapter 6, which uses a method of cooptimization instead of a mesh optimization: the coefficients of the modulation function, obtained by regression in chapter 5, are here optimized instead. A more detailed and quantitative analysis of this controller is then performed,

including an analysis of the kinematic and dynamic variations resulting of the step height and step length modulations and a comparison with human data.

Finally, in chapter 7, the development of discrete adaptations to the stable walking gait is tackled. The resulting modulations are described, and the usefulness and performances are discussed.

Two electronic addendums are submitted with this report. The two first addendums contain additional material for chapters 3 to 7: videos of the simulations and all the figures presented in this report, as well as the results files and the matlab scripts to handle those. The third addendum contains the whole project's code.

Chapter 2

Robotic platform and tools, useful definitions and existing controller

Because the CoMan robot was not available for experiments, this master thesis is achieved completely in simulation, with the robot being simulated in the Robotran simulation environment [CER16]. This chapter presents the CoMan robot, the simulation environment and the existing controller, as well as some definitions useful for the understanding of this report.

2.1 CoMan

As explained before, the robot used for this project is the CoMan (**Co**mpliant Hu**man**oid Platform), developed by the Italian Institute of Technology (IIT). This section, adapted from [NB13], concisely describes the robot as well as its simulation environment. More information and detailed explanations as to the development of this simulation environment can be found in [NB13].

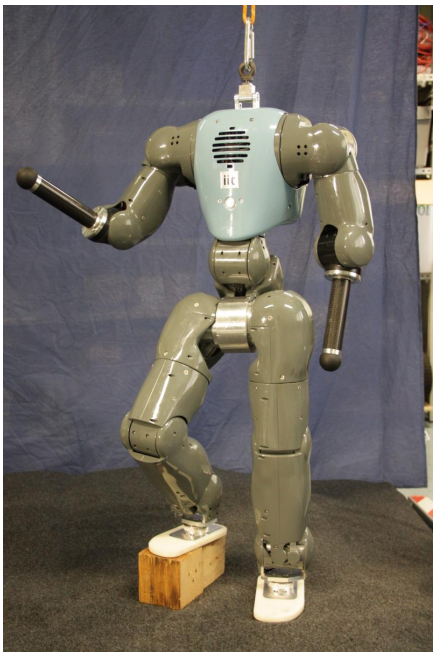
2.1.1 The robot

The CoMan Robot is represented Figure 2.1a. It is modeled on a 5-year-old child and is based on the iCub robot (pictured in Figure 2.1b) which was also developed at IIT. The CoMan however, is different from the iCub in a few important ways. The most important difference is that CoMan is equipped with compliant joints, implemented using series elastic actuators[1] (SEA).

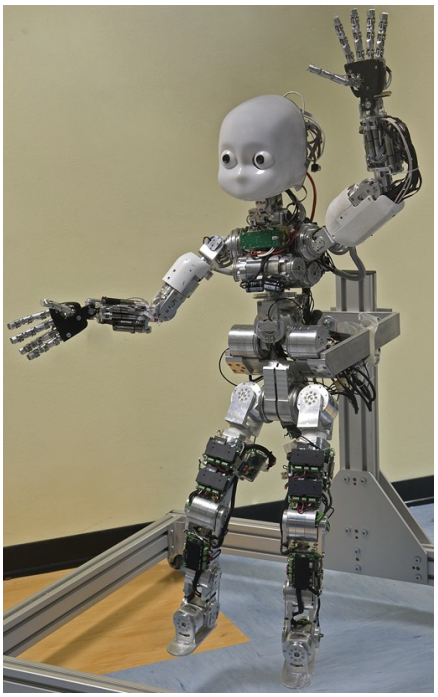
The CoMan is also fully covered (i.e. no wires exposed), is equipped with a battery and features a redesigned torso joint. The version of CoMan discussed here is however not equipped with hands nor a head; this does not matter as the focus of this project is solely on the gait of the robot. Other versions of the robot have been developed in projects with other focuses, for example with hands or a camera at the head location.

Technical specifications This paragraph, adapted from [AMA16], details the technical specifications of the CoMan. The CoMan robot is 95cm tall, weighs 31kg and has 25 DOF (2 of which are in the neck, and not controlled at all in the context of this project), represented in Figure 2.2. Its mechanical components are made from titanium alloy, stainless steel and aluminum alloy, giving it good physical robustness. Its modular joint design uses brush-less, frame-less DC motors, Harmonic Drive

[1] A series elastic actuator is an actuator incorporating series elasticity as a purposeful element. This brings greater shock tolerance, accurate and stable force control and a safer interaction with the environment.



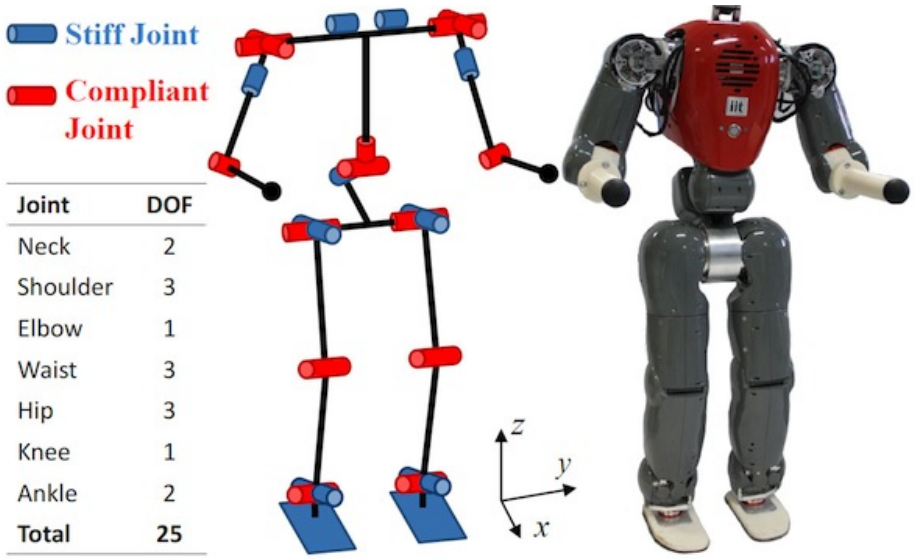
(A) The CoMan robot. Courtesy from [AMA16].



(B) The iCub robot. Courtesy from www.icub.org.

FIGURE 2.1: The CoMan robot and the iCub robot.

gears and series elastic elements. Leg, waist and shoulder joints have a peak torque capability of 55Nm. Custom torque sensors are integrated into every joint to enable active stiffness control and 6-DOF sensors are included at the ankle to measure ground reaction forces. CoMan can walk and balance using inertial sensors in the pelvis and chest, and its series elastic joint design makes it robust against impacts and external disturbances. CoMan is fully power autonomous. The torso contains a dual core Pentium PC104, on-board battery and battery management system giving up to 2.5 hours of continuous operation.



Joint	DOF
Neck	2
Shoulder	3
Elbow	1
Waist	3
Hip	3
Knee	1
Ankle	2
Total	**25**

FIGURE 2.2: Representation of the 25 Degrees of Freedom of the CoMan. Courtesy from [Fal13].

Further information about the CoMan can be found in [NB13], [IIT16], [AMA16] and [Kor11].

2.1.2 Robotran simulation environment

Robotran is a general purpose multibody simulation environment developed by the Multibody research group of the Center for Research in Mechatronics (CEREM) which is part of institute of Mechanical, Material and Civil engineering (iMMC) institute within Université catholique de Louvain (UCL). Robotran software is intended to deal with any kind of multibody dynamics application. Its main features are the use of relative joint coordinates and the symbolic generation of the equations of motion[2].

Using the Robotran model developed by Nicolas Van der Noot and Allan Barrea ([NB13], further refined in [NRI15]), the CoMan is entirely simulated using the same outputs as those sent to the real robot. As Robotran also allows to place *sensors*, the inputs available to the controller in simulation are the same as the inputs available on the real CoMan using its sensors described above.

The model developed by Van der Noot & Barrea uses parameters based on measurements realized on the CoMan robot. More information on the model development, the pros and cons of using Robotran and the Ground Contact Model used can be found in [NB13].

2.2 Some definitions

To understand this report correctly, it is necessary to define some terms and underlying concepts before.

Step height and step length

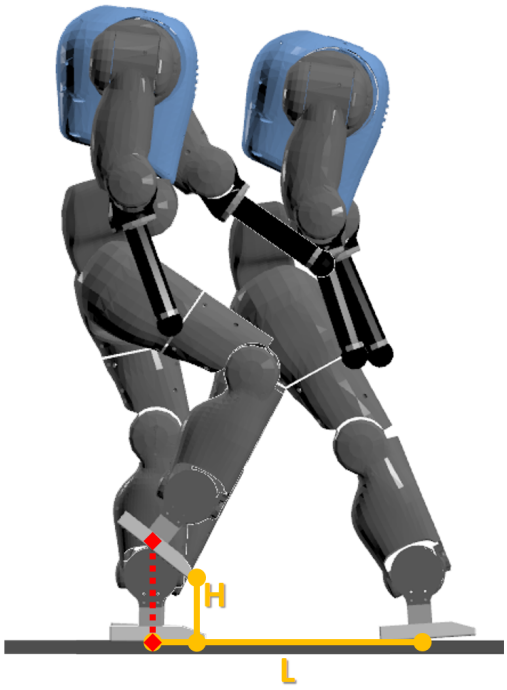


FIGURE 2.3: Definition of step height and step length as used in this report.

The **step height**, or ground clearance, is understood as the ground clearance of the swing foot at the moment when the heels of the two feet cross during the gait cycle. This moment is represented by the red dashed line in Figure 2.3. The resulting height corresponds to the height of the **toes** at that moment, as represented by the 'H' in Figure 2.3. If the foot was oriented in the other direction (with the toes pointing upwards), it would correspond to the height of the back of the heel.
The **step length** is understood as the distance between two consecutive foot strikes. Important to note here is that this distance, in the context of the 2D walk studied here, thus corresponds to the distance along the direction of the walk from one foot strike to the opposite foot strike, where both foot strikes are projected onto one direction: the walking direction. The step length is represented by the 'L' in Figure 2.3.

[2] Adapted from [CER16].

Gait features vs Controller parameters In the context of this report, it is important to understand the difference between gait features and controller/gait parameters. Gait features will refer to the actual results obtained, usually in terms of step height and step length. Gait parameters on the other hand will refer to the parameters of the controller or the gains applied on the control of the muscles (see section 2.3).

Gait cycle

According to [Whi07], the gait cycle is defined as the time interval between two successive occurrences of one of the repetitive events of walking. In the context of this master this thesis, the chosen event is the initial contact of the right foot, as shown in Figure 2.4. Then, the cycle will continue until the right foot contacts the ground again. The left foot, of course, goes through exactly the same series of events as the right, but displaced in time by half a cycle. Throughout this thesis, percentages of a gait cycle will hence correspond to a percentage of time of the gait cycle as defined here.

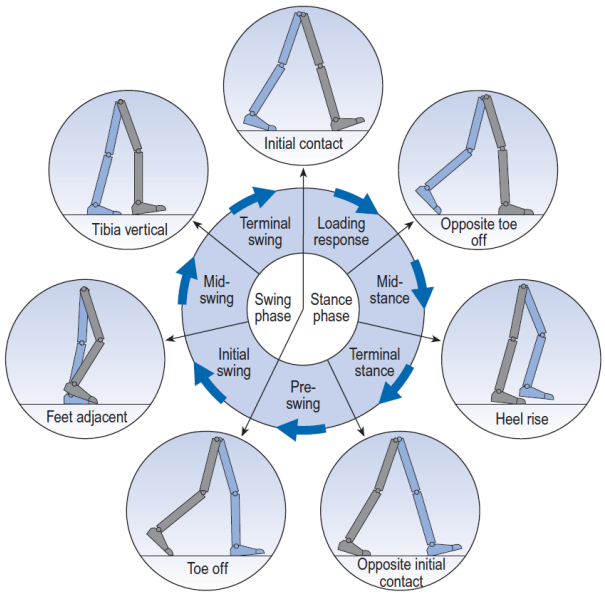


FIGURE 2.4: A normal human gait cycle, split into its different phases. Image courtesy from [Whi07].

2.3 Bio-inspired controller overview

As mentioned in chapter 1, the robot is controlled using a bio-inspired controller. This controller, originally developed in [NB13], is the one represented in the blue box in Figure 2.5. This section will explain the working of the existing controller, and is hence greatly inspired from [NB13] and the new version of the controller presented in [NRI15].

2.3.1 General working principle

The controller pictured in Figure 2.5 shows the signal paths and the way the control is applied to the robot or its simulation. The body state, measured by the sensors described in subsection 2.1.1, is sent as inputs to the controller. This controller has separate behavior for the various parts of the body, and is hence represented as split up between Leg sagittal DOFs controller, Leg non-sagittal DOFs controller and the Arms and trunk DOFs controller. Referring to Figure 2.2, the sagittal DOFs correspond to the joints in the y-axis direction.

After the different controllers have generated the control signals (either position or torque references, see Figure 2.5), they are sent to the impedance controllers of the robot, developed by the IIT. The references are used to compute the voltages that are sent to the motors.

In the context of this project, only the leg **sagittal** DOFs are of interest. The leg non-sagittal joints are given constant zero-position references.

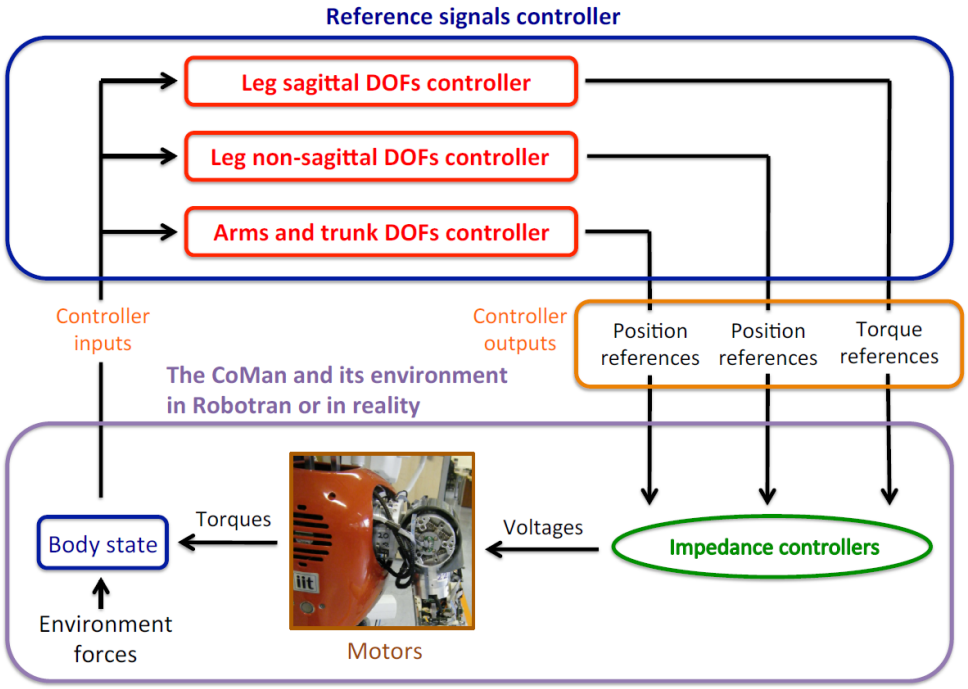


FIGURE 2.5: Path of the signals and application of torque and position references to the CoMan. Adapted from [NB13]. In this controller, the arms and trunk sagittal DOFs are also controlled with torque references.

2.3.2 Leg sagittal DOFs

The controller for the leg sagittal DOFs is based on Hartmut Geyer's work ([SG15a]). An earlier version of that model ([GH10]) was adapted to take various changes into account. The key principle of this approach is to model 9 leg (sagittal) muscle groups to simulate the behavior of each leg. These muscle groups are stimulated, causing them to react by contracting, which causes the leg to move.

As there are no muscles on the real CoMan, virtual muscles are simulated, considering they are attached to the legs as they would be on a human leg. Consequently, these muscles generate forces and torques on the joints when contracting. The torque reference sent to each joint of the CoMan (or its simulation) is the total torque on a joint, computed by taking into account the torque due to each muscle group.

An overview of this approach is represented in Figure 2.6. Using the Hill-type muscle model, the stimulation sent to each muscle and the muscle state are used to compute the resulting force of the muscle. This force is then used to compute the resulting torque due to the muscle and the lever arm on the joint(s) it is connected to.

Muscle-tendon Unit

Like stated previously, each leg is actuated by 9 Hill-type muscles, each represented as a muscle-tendon unit (MTU) (represented in Figure 2.7). All the equations related to the implemented model can be found in [GSB03; GH10]. This paragraph is adapted from [NB13].

The MTU is composed of a contractile element (CE) and a series elastic element (SE). These form the MTU in normal operation. A parallel elastic element (PE) is used when the CE stretches beyond its optimum length and a buffer elastic element (BE) prevents the active CE from collapsing when CE is under its minimal length. If we

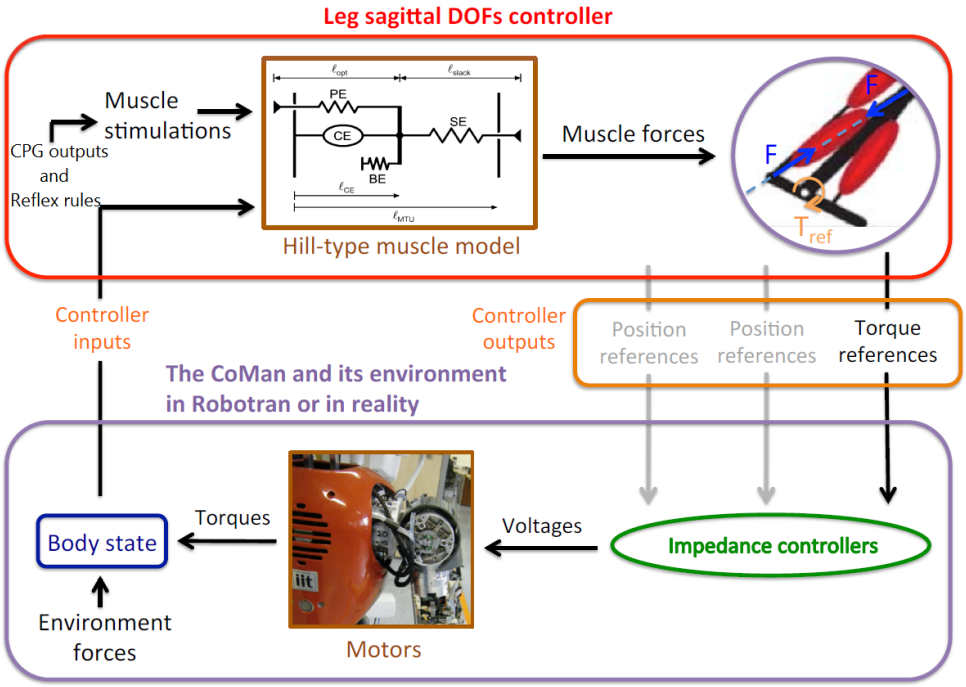


FIGURE 2.6: Leg Sagittal DOFs controller, adapted from [NB13].

assume that we know all the joint positions of the robot, only one state variable is needed to recover the full state of each MTU: the length of the contractile element l_{CE}, represented in Figure 2.7.

Indeed, the length of the MTU l_{MTU} can be computed from the joint angles and the length of the series elastic part l_{SE} can be computed using $l_{SE} = l_{MTU} - l_{CE}$. Knowing the state l_{CE} of all muscles is thus enough to compute the forces developed in the contractile element (F_{CE}) and in the series elastic element (F_{SE}). The force F_m applied by a muscle on the leg equals F_{SE}.

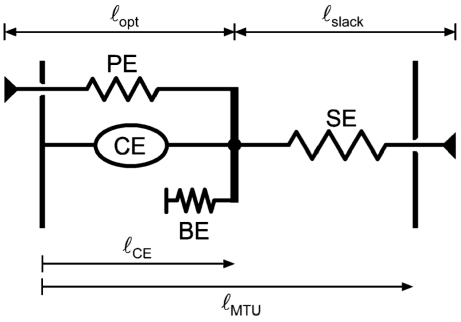


FIGURE 2.7: Muscle-tendon unit from the Hill-model. Adapted from [GH10].

Each sagittal joint depends on several MTUs, presented later in this section. Using the force produced by each muscle, and knowing the points where the muscles are attached to the body, it is possible to compute the torque generated by each muscle on the joints. If we sum the contribution of all the muscles for one joint, we get the reference torque that needs to be produced in a joint. More information related to the MTU state iteration scheme, state initialization and integration time step can be found in [NB13].

An important last note concerning the MTUs is the **joint soft limits** applied to them: like human joints, the joints of CoMan have range limits. As the Hill-model

does not take this into account, an additional torque is applied on each joint that stretches beyond its limit, as suggested in [GH10]. Additionally, a low-pass filter is applied on this torque, as explained in [NB13]. In the controller developed here, only the knee soft limit is usually engaged.

Dynamic scaling The muscle parameters (muscle maximal forces, the neutral length of the contractile elements (CE) and of the series elastic elements (SE), their maximal velocities, etc.) provided in Geyer's works [GH10; SG15a] were gathered on an adult-like model of 80 kg. When compared to this model, CoMan has different weights and lengths: this is logical, as CoMan is modeled on a 5-year-old child. All the parameters provided by Geyer do not fit this model. To ensure that the dynamic behavior of the CoMan matches the one of Geyer's model, we use dynamic scaling to adapt these parameters.

The concept of dynamic scaling is to use fundamental physical variables and relationships (mass, length, time) to obtain system quantities at different scales. To do this, we focus on the length and mass ratios. Even if these ratios depend on the different parts of the body, we take a single ratio for all lengths and another one for all masses. This is of course not strictly exact but provides a good approximation. As the relationships are not exact, and because the morphology can also change from one human to another, this means that we can also adapt the muscle parameters as we see fit. This has a direct impact on the robot gait. For example, a robot that is especially designed to walk with high steps can be expected to have stronger muscles to bend its knees.

The ratios used for the dynamic scaling of the CoMan are given in Equation (2.1). The full equations used for the scaling of the dynamic parameters, as well as the mathematical developments to obtain the ratios, are given in Appendix B (written by Nicolas Van der Noot).

$$\frac{\text{CoMan Lengths}}{\text{Geyer's model lengths}} \approx 0.427 \qquad \frac{\text{CoMan masses}}{\text{Geyer's model masses}} \approx 0.354 \tag{2.1}$$

Virtual Hill-type muscles

As explained previously, 9 Hill-muscles are used to simulate the working of the sagittal DOFs of a human leg. When compared to [GH10], two muscles groups were added in the latest version of the controller ([SG15a]): the Biceps Femoris short head (BFSH) and the Rectus Femoris (RF). These muscles both act on the knee joint (the BFSH is mono-articular, the RF also acts on the hip joint). Hence, we assume they will allow for better control of the knee bending to modulate the gait: adding muscles acting on a joint gives more degrees of freedom to act on the aforementioned joint, and increases the maximum torque that can be applied to it.

In total, each leg is hence controlled by 9 muscle groups: Hip Flexors (HFL), Gluteus (GLU), Hamstring (HAM), Biceps Femoris Short Head (BFSH), Rectus Femoris (RF), Vastus (VAS), Gastrocnemius (GAS), Soleus (SOL) and Tibialis Anterior (TA). These muscle groups are represented in Figure 2.8, and Table 2.1 shows a summary of the articulation(s) that each muscle group acts on.

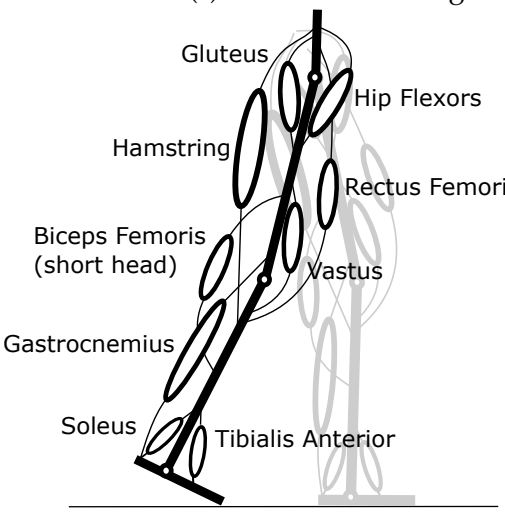


FIGURE 2.8: The 9 muscle groups acting on leg sagittal DOFs.

Muscle Group	Articulation(s) acted on
GLU	Hip
HFL	Hip
HAM	Hip and Knee
RF	Hip and Knee
BFSH	Knee
VAS	Knee
GAS	Knee and Ankle
SOL	Ankle
TA	Ankle

TABLE 2.1: The 9 muscle groups of the leg sagittal DOFs, with the articulations they act on.

CPG model

For the previously described muscle groups to contract, they have to be **stimulated**. The objective of the controller is hence to send stimulations to the 9 muscle groups. The biggest part of these stimulations is generated thanks to a Central Pattern Generator (CPG), described in this paragraph. Like explained in the introduction of this report, the principle of a CPG is that it is a neural circuit capable of producing rhythmic patterns without receiving rhythmic inputs. The CPG model used is adapted from a later version of the CPG model described in [NRI15], and uses 10 neurons to generate its outputs. It is represented in Figure 2.9. The 4 numbered neurons N_1, N_2, N_3 and N_4 are called Rhythm Generation (RG) neurons [MR08] and are used as the rhythm generators of the CPG, not influenced by the 6 other neurons. The 6 other neurons, denoted with letters, are called Pattern Formation (PF) neurons [MR08]. These neurons are used to form the stimulation patterns for the muscles. The explanations concerning the CPG model and the muscle stimulations are adapted in large part from [NRI15].

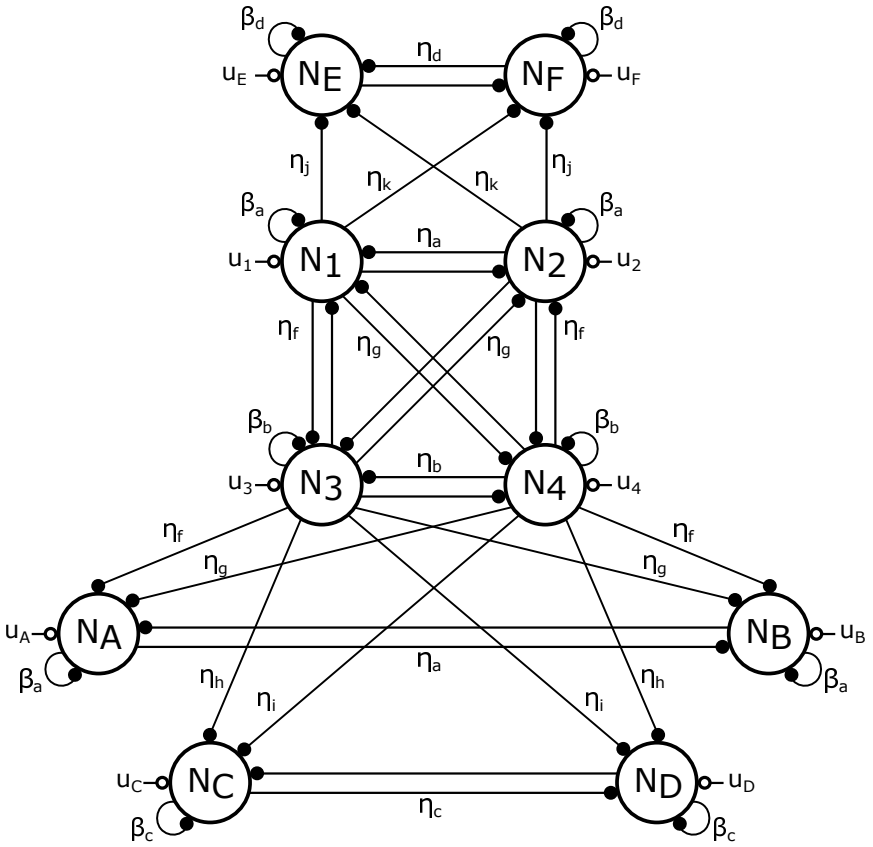


FIGURE 2.9: CPG scheme used for this controller. The 10-neurons oscillator network sends stimulations to the HFL, GLU, HAM, BFSH and RF muscle groups represented in Figure 2.8. Figure adapted from [NRI15].

The firing rate x_i of each neuron N_i is computed according to Equation (2.2), with the indices corresponding to the scheme in Figure 2.9. The 10 equations for the firing rates are fully given in section A.1, in appendix.

$$\dot{x_i} = \frac{1}{\tau}(-x_i - \beta_j \nu_i \ - \sum_{1}^{3} \eta_k [x_l]^+ + u_i) \tag{2.2}$$

In these equations, ν_i is the self-inhibition, modulated by a constant β_j to be optimized. u_i is an external excitation associated to each neuron and τ is the time constant of the CPG. The connection strengths η_k, to be optimized, tune the mutual inhibitions. The function $[\bullet]^+$ takes the positive part of its argument, and saturates to zero if the argument is negative (i.e. $[x]^+ = max(0, x)$). It thus captures the fact that the activation of a given neuron decreases when another is active: mutual inhibition. Thanks to this saturation to zero when the argument is negative, neurons can only inhibit (and not excite) each other, keeping the firing rates under control.

The self-inhibitions ν_i are computed using Equation (2.3), where γ_i are constants to be optimized. The full equations, available in section A.2 in appendix, can be obtained by applying Equation (2.3) with the correct γ_j, depending on the desired symmetries as explained below.

$$\dot{\nu_i} = \frac{1}{\gamma_j \tau}(-\nu_i + [x_i]^+) \tag{2.3}$$

This network obeys a mirror symmetry to reproduce the symmetry of the right and

left legs of the robot. This symmetry between neurons N_1, N_3, N_A, N_C and N_E and neurons N_2, N_4, N_B, N_D and N_F can be observed in the mutual inhibition strengths η_i as well as in the self-inhibition strengths β_i and the γ_i coefficients. Equations (A.1) show this, as well as Figure 2.9. Neurons N_1, N_2, N_3 and N_4 form a fully-connected network, where each neuron fires alternatively over the cycle. The other neurons, $N_{A\to F}$, are connected to them but do not interfere. In this way, the β_i, η_i and γ_i used for these neurons give more flexibility to modify the outputs. Using the outputs computed as described in (2.5), it is possible to stimulate the various muscles with different intensities at different times.

To keep the CPG in synchronization with the robot behavior (and hence obtaining the robustness described in chapter 1), the CPG can also be modulated by the interactions between the robot and its environment. This is done by modulating the neuron excitations u_i of each neuron, to which the outputs are proportional, depending on when the foot strikes happen and on the current status of the legs (swing or stance).

The inputs u_i first consist of a tonic excitation u, kept equal to 1 for simplicity – the oscillator outputs will rather be modulated by external gains applied on the outputs. Some terms are added to u_i to achieve synchronization with the walking gait. The firing rate x_1 is expected to switch from negative to positive at the moment of the right foot strike. Similarly, x_2 should become positive after the left foot strike. These conditions lead to Equations (2.4). The function $[\bullet]^-$ takes the absolute value of its argument if it is negative and saturates to zero otherwise. The functions $[\bullet]_{SR}$ and $[\bullet]_{SL}$ keep their argument intact during the right (respectively, left) leg support phase and saturate to zero otherwise. Thanks to these functions, each neuron firing rate is able to synchronize with the appropriate leg.

$$
\begin{aligned}
u_1 &= u - [x_1]^+_{SL} + [x_1]^-_{Str,R} & \qquad u_B &= u - [x_B]^+_{SR} + [x_B]^-_{Str,L} - [x_B]^+_{\Theta+}\\
u_2 &= u - [x_2]^+_{SR} + [x_2]^-_{Str,L} & u_C &= u - [x_C]^+_{SL}\\
u_3 &= u - [x_3]^+_{SL} - [x_3]^+_{Str,L} & u_D &= u - [x_D]^+_{SR}\\
u_4 &= u - [x_4]^+_{SR} - [x_4]^+_{Str,R} & u_E &= u - [x_E]^+_{SL}\\
u_A &= u - [x_A]^+_{SL} + [x_A]^-_{Str,R} - [x_A]^+_{\Theta+} & u_F &= u - [x_F]^+_{SR}
\end{aligned}
\tag{2.4}
$$

Then, the functions $[\bullet]_{Str,R}$ and $[\bullet]_{Str,L}$ are used for synchronization when the CPG lacks behind: $[\bullet]_{Str,R}$ keeps its argument intact as long as x_1 is not the only positive firing rate out of the $x_{1\to4}$. $[\bullet]_{Str,L}$ does the same for x_2. Thanks to these functions, the neurons are more (or less) excited depending on whether they should become positive (or negative) compared to the moment of the foot strike. Finally, the function $[\bullet]_\Theta$ keeps its argument intact if the difference between the reference torso pitch angle and the actual torso pitch angle is positive. This allows to reduce the excitations u_A and u_B to help track the reference pitch angle more accurately (see later in the stimulations).

All the terms added to the excitations until now were only used to accelerate the firing rates of some neurons while slowing down some others to achieve synchronization. If the CPG is **ahead** of the walking gait however, the network has to be slowed down in order for the gait to catch up. This is done by forcing to zero **all the excitations** if the firing rate x_1 becomes positive before the right foot strike or if the firing rate x_2 becomes positive before the left foot strike. All these mechanisms, together, achieve the synchronization between the different neurons and the walking

gait. In steady state, the contribution of all the synchronization terms described here is very limited.

To initiate the walk, an excitation of 1 is sent to three neurons: N_2, N_B and N_E to initiate the right leg swing or N_1, N_A and N_C to initiate the left leg swing. After 0.2 $[s]$, all excitations are activated as described in Equation (2.4).

Finally, to use the firing rates as stimulations for the muscles, 6 different outputs are computed, as defined in Equations (2.5)[3]. Each output is computed by taking the positive part of the firing rate of one neuron. These outputs are designed to feed the appropriate stimulations to the muscles at different times.

$$\begin{aligned} y_1 &= [x_C]^+ & y_4 &= [x_B]^+ \\ y_2 &= [x_D]^+ & y_5 &= [x_E]^+ \\ y_3 &= [x_A]^+ & y_6 &= [x_F]^+ \end{aligned} \tag{2.5}$$

Muscle Stimulations

The muscle stimulations are, depending on the muscle, a combination of CPG output signals, reflex rules and pre-stimulations S_0. The stimulations are always bounded between a maximum stimulation of 1 and a minimum of $0.01 (= S_0)$. The stimulations of the HFL, GLU, HAM, BFSH and RF muscles ($S_{HFL,R/L}$, $S_{GLU,R/L}$, $S_{HAM,R/L}$, $S_{BFSH,R/L}$, $S_{RF,R/L}$ for the right/left leg muscles, respectively) are linear combinations of the CPG output signals. The full equations giving the stimulations are given in appendix, in section A.4 and section A.5. All the coefficients multiplying the CPG outputs are coefficients to be optimized.

On top of that, reflexes are also used to stimulate the muscles. All the reflexes used come from [GH10]. For HFL, an extra stimulation is added to the stance leg to help maintain the trunk at a desired reference angle Θ. This stimulation is computed as

$$S^{ext}_{HFL} = \left[\frac{F_z}{w}(K_{p,\Theta} * (\Theta - \Theta_t) - K_{d,\Theta} * \dot{\Theta}_t)\right]^+$$

where $K_{p\Theta}$, $K_{d,\Theta}$ and Θ are constants to be optimized, F_z is the vertical force under the foot of the corresponding leg, Θ_t is the trunk absolute angle, Θ_t its derivative and w the total robot weight. For the right leg, if y_2 and y_3 are negative (y_1 and y_4 for the left leg), the negative part of this same term is added to the GLU muscle of the stance leg. This is to prevent contradictory signals between CPG and reflexes.

The other reflex rules can be split between the reflexes applied to the stance leg and those applied to the swing leg. For the stance leg, the reflexes are those visible in Equations (2.6), where G_{SOL}, G_{VAS}, G_{GAS}, G_{SOL-TA} and $G_{TA,Sw/St}$ are parameters to be optimized and $\frac{F_\bullet}{F_{\bullet,max}}$ are the normalized forces produced by the muscles. $l_{CE,TA}$ is the length of the contractile element of TA, $l_{opt,TA}$ its neutral length and $l_{TA,St/Sw}$ are parameters to be optimized. The equations related to the stimulations during the swing phase are given in (2.7).

[3] In this work, two different sets of outputs are used. The outputs used for chapter 4 and chapter 5 are given in section A.3.

$$
\begin{aligned}
S_{SOL,St} &= S_0 + G_{SOL} * \frac{F_{SOL}}{F_{SOL,max}} \\
S_{TA,St} &= S_0 + G_{TA,St} * \left(\frac{l_{CE,TA}}{l_{opt,TA}} - l_{TA,St} \right) - G_{SOL-TA} * \frac{F_{SOL}}{F_{SOL,max}} \\
S_{GAS,St} &= S_0 + G_{GAS} * \frac{F_{GAS}}{F_{GAS,max}} \\
S_{VAS,St} &= S_0 + G_{VAS} * \frac{F_{VAS}}{F_{VAS,max}}
\end{aligned}
\tag{2.6}
$$

Finally, S_{VAS} is set to S_0 when the corresponding leg is the trailing leg during the double support phase or if the corresponding knee angle ϕ_k exceeds an over-extension threshold ϕ_{off} (to be optimized) while $\dot{\phi}_k$ is positive (see [GH10]). This prevents knee over-extension.

$$
\begin{aligned}
S_{SOL,Sw} &= S_0 \\
S_{TA,Sw} &= S_0 + G_{TA,Sw} \left(\frac{l_{CE,TA}}{l_{opt,TA}} - l_{TA,Sw} \right) \\
S_{GAS,Sw} &= S_0 \\
S_{VAS,Sw} &= S_0
\end{aligned}
\tag{2.7}
$$

With these last equations, all the stimulations sent to the muscles have been defined. A lot of parameters have to be optimized to obtain a stable gait; this process is described in section 3.1.

2.3.3 Upper-body control

The control of the upper body is less critical during walking. However, [Wan+12] presents rules that allow to balance the trajectories of the arms, reducing the total energy consumption as a result. The rules used for the control of the arms is similar to those rules.

For all joints except for the shoulder sagittal DOF, constant position references are tracked. The position references are 25° for the elbow sagittal DOF, 5° for the shoulder lateral DOF and 7.5° for the shoulder transverse DOF. These position references are used to stimulate the elbow flexion and extenstion muscle groups, the shoulder adduction and abduction muscle groups and the shoulder internal and external ones, using a simple feedback mechanism.

Finally, the shoulder sagittal DOF is controlled by shoulder flexion and extension muscles, stimulated by the appropriate CPG neurons to be in phase with the gait cycle. More precisely, the stimulations are designed to be in phase with the opposite (contralateral) leg motion. Such control leads to balancing arm trajectories, reducing the total energy consumption during the walking gait.

The focus of this project, as explained in chapter 1, is to modify the leg sagittal DOFs controller to obtain a more adaptable gait. For this reason, the upper-body controller is not explained in more details in this work.

Chapter 3

Optimization and analysis of natural gait

This chapter can be seen as a complement to the introduction of the tools presented in chapter 2, and serves two purposes. The first objective is to describe the *optimization procedure* used in this work. Indeed, the previous chapter introduced numerous coefficients without assigning values to all those coefficients. The values are found by particle swarm optimization, which will be explained here.

The second objective of this chapter is to present and analyze a working gait of the robot. This working gait comes from the controller developed in chapter 6. It is presented here to give the reader an insight into the characteristics of the attainable gaits, and the similarities with the human gait.

3.1 Step height and step length optimization

To obtain gaits with various features (desired step lengths and step heights), the optimization procedure developed in [NRI15; NB13] was used. The optimization uses a particle swarm optimization (PSO) framework in order to optimize on large sets of parameters. The goal is to maximize an objective function (called fitness function). This fitness function was modified to serve the needs of this project, i.e. optimize to obtain desired performances in terms of step heights and step lengths. This chapter explains the computation of the fitness function, describes the obtained reference gait and the mesh optimization that was performed to obtain a broad set of gaits with the desired features.

Particle Swarm Optimization and fitness function

In short, a PSO algorithm starts by generating a set of N random solutions, where each parameter is taken at random (inside a user-defined range) for each solution. Then, the fitness of each generated solution is computed according to criteria defined by the user. To generate the next iteration of N solutions, each solution is attracted by both its own best fitness and by other solutions with higher fitnesses. This heuristic algorithm and the way it is implemented in this case are further described in [Sof; NB13].

To use the algorithm, the design of an effective fitness function is fundamental. This project uses a fitness function that works by **stages**, where the next stage can only be unlocked if the necessary performances are obtained in the previous stage (although some stages where run in parallel). Each stage computes a fitness in a range from 0 to 100. The different fitness stages used for this project are graphically represented in Figure 3.1.

The first stage is the **minimum distance stage**. It guarantees that the evaluated set of parameters satisfies a minimal distance condition $d_{min} = 0.25 \cdot T_s \ [m]$ where T_s is the objective simulation time (the time where the simulation ends if the robot does not fall before). This fitness is computed as $Fitness = max\left(100,\ 100 \cdot \frac{d}{d_{min}}\right)$ where d is the actual distance walked. This stage makes sure that the robot actually moves, and does not stay in place without leaving its initial position. If the d_{min} condition is satisfied, the next stage is unlocked.

The second stage is the **walking time stage**. This stage is placed here to ensure that the robot walks for at least a certain amount of time. This fitness is, like the previous one, computed linearly by $Fitness = 100 \cdot \frac{t}{T_s}$ where t is the actual end time of the simulation (the simulation stops when the robot falls). These two first stages, together, make sure that the next criteria are evaluated only for parameter sets that enable the robot to walk for the full time of the simulation. If the robot walks for the full time of the simulation, the next staged is unlocked.

The third and fourth stages, computed in parallel, are the **step height and step length stages**. The previous steps only made sure the robot walked, without any other constraint. Now, these stages further analyze the gait features to assess them with respect to the objective step height and step length criteria. These two fitnesses are computed using Gaussian fitness functions: $Fitness = 100 \cdot e^{-\Delta_l^2 \cdot k}$ where Δ_l (or Δ_h) represents the difference between measured mean step length (or step height) and objective step length (or step height) and k is the exponential gain, controlling the standard deviation of the Gaussian. A higher value of k results in a narrower Gaussian, as represented in Figure 3.2. To unlock the next stages, the requirements are $\Delta_l < 3 \cdot 10^{-2} \ [m]$ and $\Delta_h < 1 \cdot 10^{-2} \ [m]$. If both conditions are fulfilled, the next stages are unlocked.

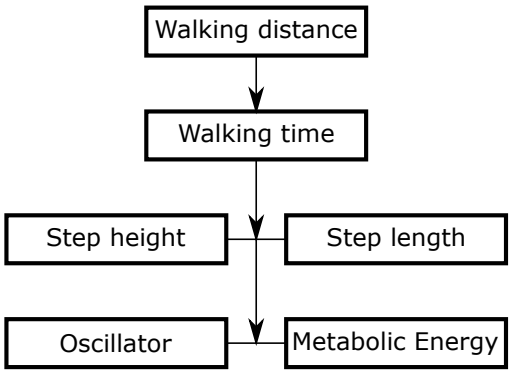


FIGURE 3.1: The stages of the fitness function. Stages represented on the same level are computed in parallel; stages represented on different levels are computed sequentially, if unlocked by the stages above.

Figure 3.2 represents the sum of the step length and step height fitnesses, for an objective step length of $0.4 \ [m]$ and objective step height of $0.065 \ [m]$. The result is two Gaussian 'tunnels', situated in the two different dimensions (step height and step length). The objective for each dimension is represented by a dark blue horizontal plane situated under the Gaussian. To fulfill both conditions at the same time (i.e. be above both horizontal planes), the gait features necessarily have to be situated near the center of the represented field, at the crossing of the two Gaussian 'tunnels', resulting in the highest total fitness.

The exponential gain k will be different for the step length fitness and for the step height fitness: $k = 200$ for the step length fitness, and $k = 2000$ for the step height fitness. This can be explained by the amplitude of each feature: as the amplitude is bigger for the step length (≈ 0.2 to $0.5 \ [m]$) than for the step height (≈ 0.01 to $0.1 \ [m]$), the width of the respective Gaussian has to be adapted accordingly. With the values

of k, Δ_l and Δ_h chosen, the condition to unlock the next stage corresponds to a fitness ≈ 80 when computing the $100 \cdot e^{-\Delta_l^2 \cdot k}$ function described previously for each of the two stages discussed here.

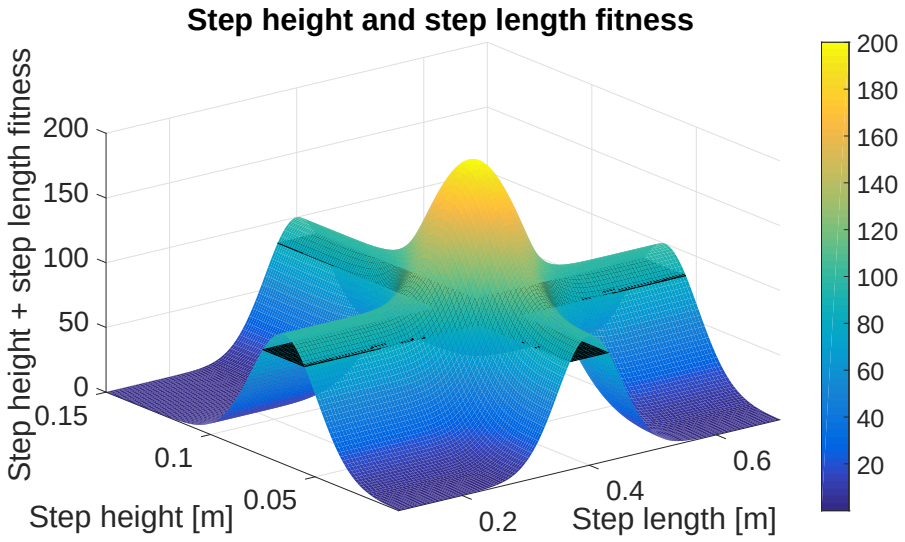


FIGURE 3.2: Graphical representation of the sum of step height and step length fitness, for an objective step length of 0.4 $[m]$ and objective step height of 0.065 $[m]$. The two dark blue horizontal planes represent the two conditions to be fulfilled simultaneously to unlock the next stage.

The fifth and sixth stages, also computed in parallel, correspond to the **minimization of metabolic energy consumed** (in muscle contraction per unit distance walked) and the **oscillator phase prediction error**. The purpose of the computation of the metabolic energy is to favor solutions minimizing metabolic energy consumption. The purpose of the oscillator phase prediction optimization is to encourage the emergence of solutions minimizing the phase error between the oscillator and the gait – hence minimizing the need for the additional synchronization inputs described in Equations (2.4). For both stages, the same function is used as for the step length and step height fitness. In this case, the objective is 0 for both metabolic energy consumption and oscillator prediction error, as the goal is to minimize those. More information about the computation of metabolic energy in muscle contraction, as well as more extensive descriptions of these two fitness functions, can be found in [NB13; NRI15].

3.2 Analysis of a natural gait

This section describes a gait obtained in chapter 6, which will be considered as the natural gait. The reference gait features for this natural gait are a step length of 40 $[cm]$ and a step height of 6.5 $[cm]$. As the optimization used for chapter 6 optimizes on the whole domain of gait characteristics, and not only on one reference set, the fit with the reference is not perfect: there is a small offset error on the mean of step height and step length. The measured mean step height with the reference parameter set is 6.19 $[cm]$. For the step length, the measured mean step length is 42.4 $[cm]$. The objective of this analysis is mainly to show the main characteristics of the controller presented in chapter 2: the kinematics and dynamics of the main joints, the outputs of

the CPG, the generated muscle activations and the interactions with the environment (the ground reaction force).

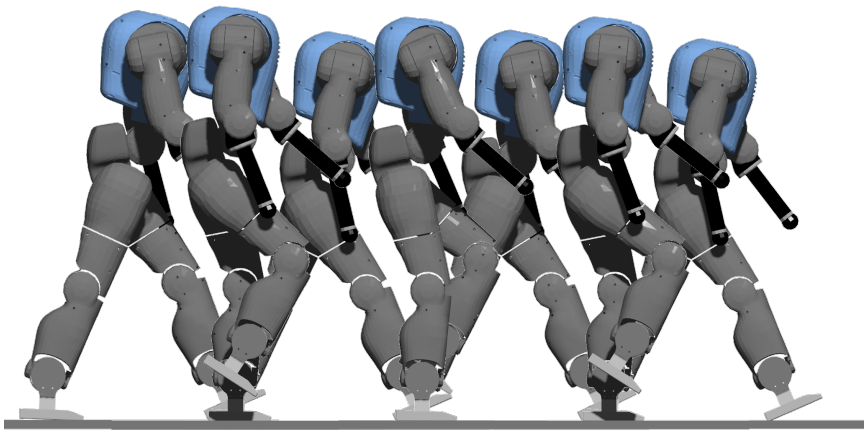


FIGURE 3.3: Snapshots of the natural gait, taken at foot strike and at step height measurement.

Firstly, Figure 3.3 shows snapshots of a few consecutive steps, taken at foot strike and at the step height measurement[1]. Figure 3.4 represents the gait characteristics corresponding to the natural gait, during a period of $60\ [s]$. As stated previously, the step length is mostly above its reference while the step height is mostly under it. It is however also interesting to note the **regularity** of the observed gait. Indeed, the mean step length difference between two consecutive steps is $1.72\ [cm]$, and the maximum is $5.36\ [cm]$. For the step height, the mean difference between two consecutive steps is $0.27\ [cm]$ and the maximum is $0.7\ [cm]$. These characteristics give very small standard deviations for both gait features, at $1.3\ [cm]$ for the step length and $0.21\ [cm]$ for the step height. We can hence consider that plotted results are representative and reproducible, although they present an offset error with respect to the reference.

[1] A full video of the natural gait, called *naturalgait.webm*, is available is the extra material.

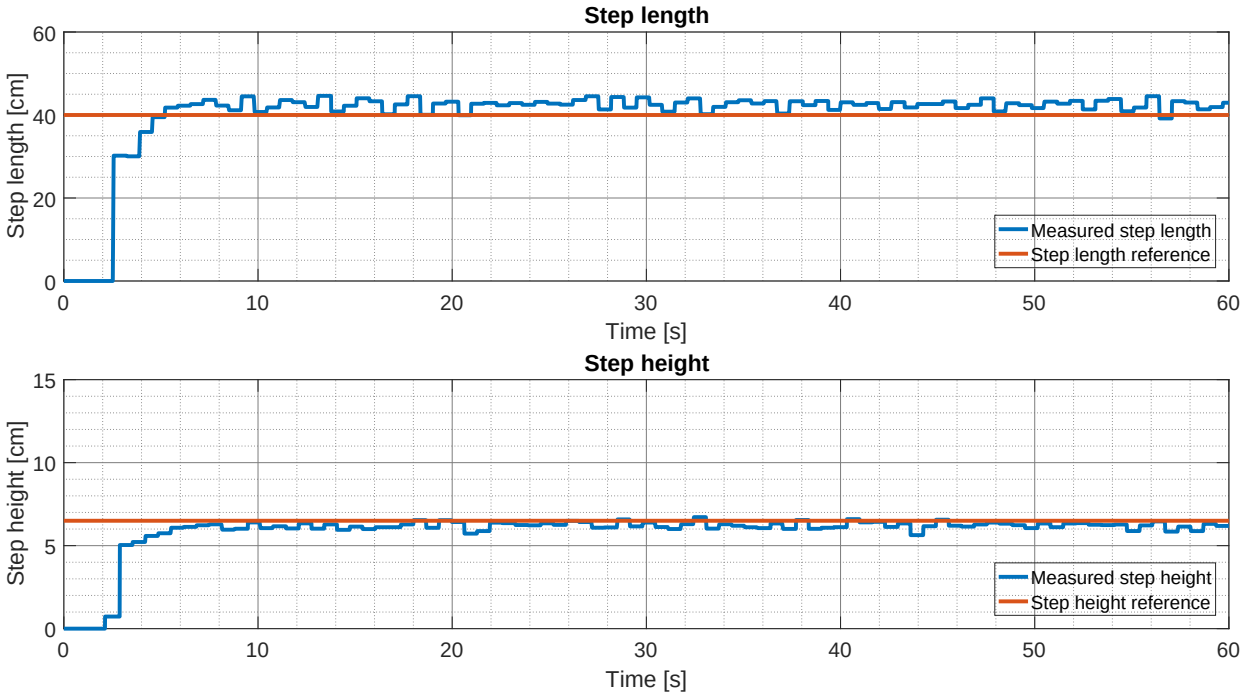


FIGURE 3.4: Gait characteristics of the obtained natural gait, observed during 60 seconds of simulation.

Another interesting aspect of the natural gait are the outputs of the CPG. For the natural gait, the mean step period if $0.657\ [s]$. One gait cycle hence lasts an average of $1.31\ [s]$. Figure 3.5 represents the 6 outputs of the CPG. As visible in this figure, the signals come by pairs, corresponding to the scheme presented in Figure 2.9. Indeed, as the pattern formation neurons inhibit each other by pairs, they cannot activate simultaneously. The symmetry introduced when describing the CPG scheme can thus be retrieved in the output signals plotted here. Signals 2, 3 and 5 mostly correspond to the stimulations sent to the right leg while signals 1, 4 and 6 mostly stimulate muscles from the left leg.

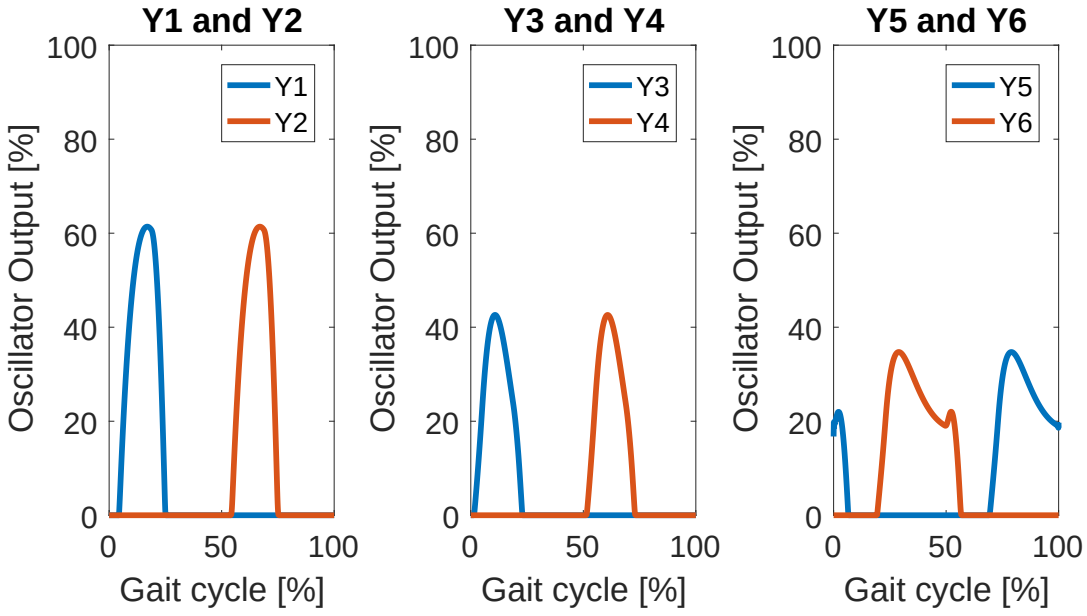


FIGURE 3.5: Outputs of the CPG for the natural gait. The mean (on 174 gait cycles) of each signal is displayed.

Looking back at the equations generating the stimulations sent to the muscles (Equation (A.5), page 70), and knowing the action of each muscle (Figure 2.8, page 14), we can conclude a few things:

- The gait cycle begins at right foot strike, which means that the most important muscles just after strike are those stimulated by y_3. For the right leg, those are the GLU, HAM and BFSH muscles.
- The swing is initiated at about 60% of the gait cycle. We can observe that for the right leg, y_2 is instrumental in this swing initiation, by stimulating the HFL, BFSH and RF muscles.
- Finally, y_5 is mostly necessary during the swing phase and to prepare for strike by stimulating all the right leg muscles except for the HFL.

It is also interesting to notice the *amplitude* of the various signals. As the HFL muscle is (almost) the only muscle pulling the leg forward to initiate the swing, we can notice that the signal that stimulates this muscle (y_1 or y_2, depending on the leg) is of great amplitude compared to the other signals. The other signals are less specific: they are used for multiple muscles, or multiple muscles accomplish the same movement together.

Having looked at the CPG outputs, it is now interesting to visualize the resulting muscle activations in Figure 3.6. In this figure, human data is also represented, allowing to observe similarities in muscle activations throughout the gait cycle. Important to keep in mind for all comparisons to human muscle activation data is that the human data are not measured the same way that our simulated data is: the human data corresponds to EMG data collected by probes on subjects' legs, while neuromuscular data corresponds to the actual activation on the simulated muscles. Moreover, the muscles simulated in this model often correspond to muscle groups and not to individual muscles.

We can see similarities between the muscle activations and the EMG data for some muscles only. Interestingly, we notice these similarities only for muscles stimulated with the **reflex rules**: GAS, SOL, TA. Indeed, for these three muscles, we observe increases in muscle activation that follows and increase in EMG measurement. The signal of the muscle activations seems to be delayed of about 10 $\%$ when compared to the EMG data. As we will see in figure Figure 3.7, this can be observed at the joint level too. The fact that the activation of only these three muscles follow that of the EMG data is interesting, because these are the three muscles that act on the ankle: the only part of the leg that is in direct contact with the environment. These muscles are thus activated as a direct **reaction** to the contact with the environment and the positions of the joints, while most other muscles are activated (only) by the CPG outputs, which merely synchronize with the gait – without having their amplitude directly influenced by the joint positions.

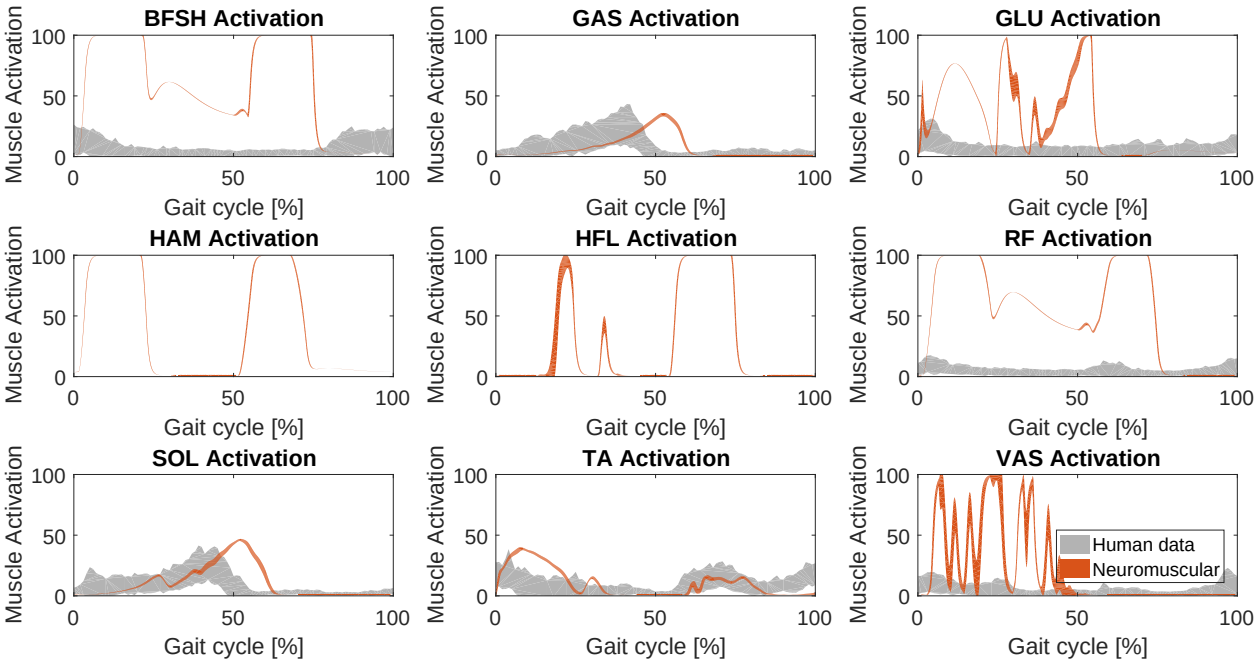


FIGURE 3.6: Activation profiles of the 9 considered leg muscles. Human EMG data from [Bov+11] (Adult, Natural Speed), when available, are compared to those from the natural gait. For the neuromuscular data from the CoMan, the average over 174 gait cycles (starting at right foot strike) is shown, augmented by its standard deviation (shaded areas).

The muscle activations described here then cause the torques and thus movements at the articulations. These positions and torques are represented in Figure 3.7. Going from top to bottom of the leg, we can first observe the ankle on the left of the figure. A first observation, concerning the angle of the ankle, is that the movement obtained using our controller is very similar to the movement measured on human subjects. We can however observe a small lag between the human and robotic data, both on angle and torque of the signal. This lag was already visible in the muscle activation signals for the muscles acting on the ankle.

Looking more into the ankle motion, we can observe that the foot strike occurs when the ankle is almost perpendicular to the leg (corresponding to zero-angle). Immediately after strike, the ankle experiences plantarflexion (corresponding to a positive angle; the leg bending backwards relative to the foot). This is logical, as the weight of the robot is being transferred to the stance foot, and the foot thus laying flat on the ground while its leg is still behind. We can observe that this plantarflexion is more important on our neuromuscular model than in human data. A possible explanation is the lack of compliance of the simulated foot, that results in this increased plantarflexion. Then, as the leg moves forward and surpasses the foot, the ankle transitions from plantarflexion to dorsiflexion. The ankle stays in dorsiflexion until toe-off, at about 60 % of the gait cycle. At this point, it goes into plantarflexion again – due to the torque generated by the activation of the TA muscle. The plantarflexion during swing is important to keep the foot up, preventing it from hitting potential obstacles or the ground.

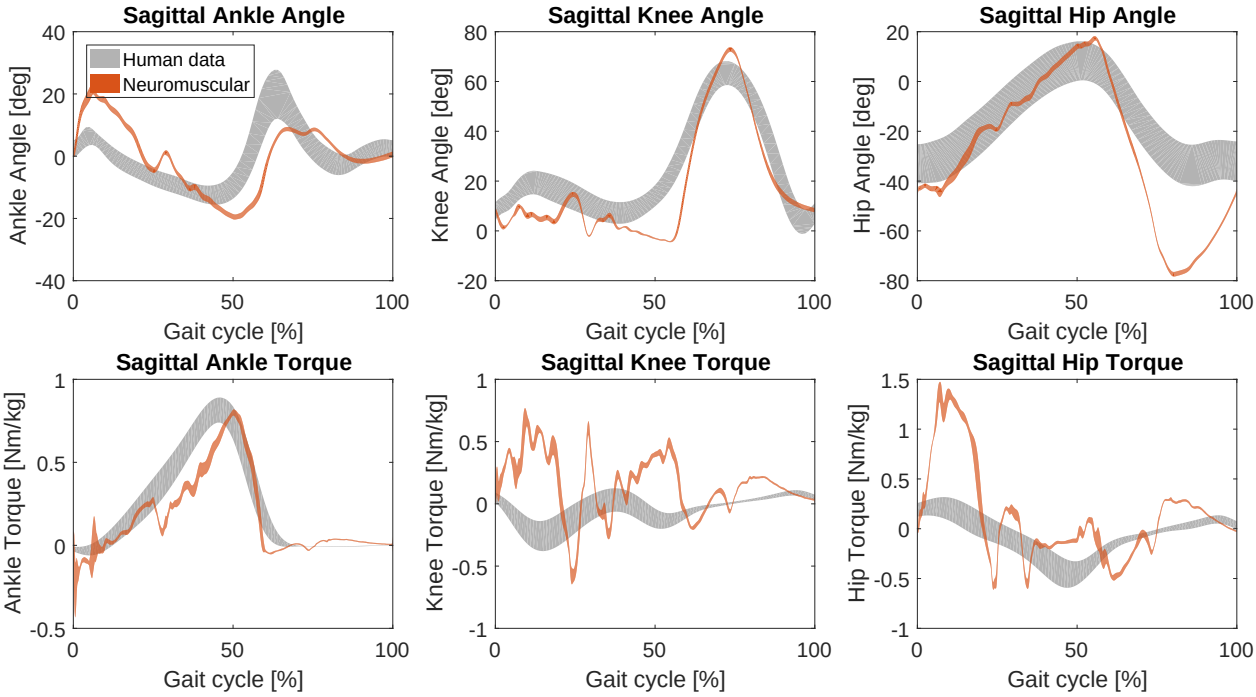


FIGURE 3.7: Kinematic and dynamic profiles of the three main sagittal articulations. Human data from [Bov+11] (Adult, Natural Speed) are scaled to the CoMan and compared to those from the natural gait. For the neuromuscular data from the CoMan, the average over 174 gait cycles (starting at right foot strike) is shown, augmented by its standard deviation (shaded areas).

Concerning the knee articulation, we can observe good similarities between the motion from our controller and the human data. The main difference can be observed in the stance phase, more precisely just after strike: while the human data shows increasing flexion just after strike, the neuromuscular model results in quite the opposite, with the knee showing close-to-zero angle – corresponding to a straight leg. This opposite motion after strike can also explain the differences in torque that are observed for the knee: because the human and neuromuscular model's knees show opposite motions, the torques are in anti-phase. Still, both data show a knee torque oscillating around zero during the stance phase.

For the hip articulation, the stance phase of the human data and neuromuscular model show the same kinematics and dynamical behavior, with the torque of the neuromuscular model being of much greater amplitude than the torque shown in the human data. This extension torque is necessary to keep the torso from falling forward. Different causes are possible for this increased hip extension torque: the knee flexion torque, generated in part by the HAM muscle, is partially responsible for this torque because the HAM also acts on the hip. This torque is also likely highly dependent on the hip center of rotation.

A last comparison that can be done between human data and the results on our neuromuscular controller concerns the interactions between the walker and the environment: the ground reaction forces. Figure 3.8 shows the vertical ground reaction force, both for human data and the natural gait. Both for human data and neuromuscular measurements, the force shows a characteristic double hump, which results from an upward acceleration of the center of gravity during early stance, a reduction in downward force as the body 'flies' over the leg in mid-stance and a

second peak due to deceleration, as the downward motion is checked in late stance [Whi07]. We can however notice an extra hump on the neuromuscular data, which can be explained by various elements: the lack of compliance of the feet results in a lack in damping of the impact of the foot; the difference in knee motion between human data and neuromuscular data could also be in cause. Indeed, at the same moment as the extra hump appears (25-30% of the gait cycle), we observe an oscillation – from flexion to extension – of the knee that is not present in human data.

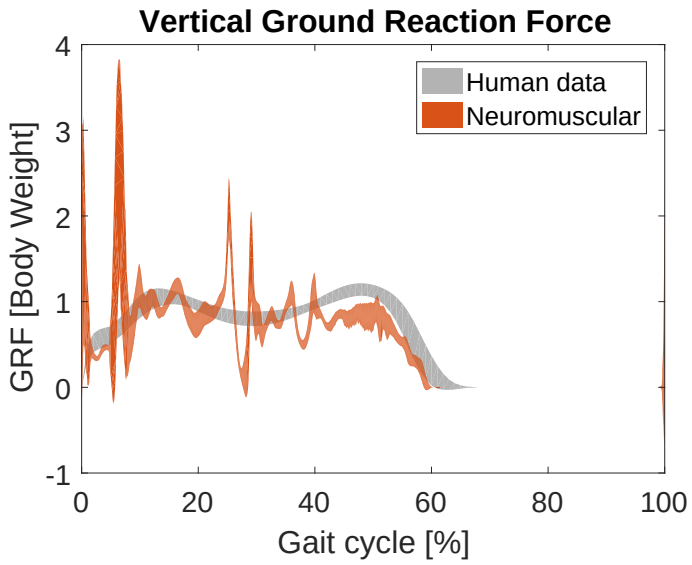


FIGURE 3.8: Vertical Ground Reaction force for the CoMan compared to human data from [Bov+11] (Adult, Natural Speed). For the CoMan, the average over 174 gait cycles (starting at right foot strike) is shown, augmented by its standard deviation (shaded areas).

Chapter 4

Experiment: Sensitivity analysis

As stated in the introduction of this report, the goal of this project is to modify the existing controller to be able to modulate the gait of the robot. To modulate the gait while keeping the working principle of the controller and without making the CPG-model more complex, only some parameters will be allowed to be modulated. Among those are Θ, the trunk reference angle, and τ, the time constant of the CPG. The **gains** applied on the CPG outputs to compute the muscle stimulations will be allowed modulation as well. To obtain an understandable model of this modulation, only a reduced set of these parameters will be modulated. This section details the sensitivity analysis that was performed to determine and select the relevant parameters for this modulation.

4.1 Defining range and reference values of the gait features

4.1.1 Length and height reference

To perform a sensitivity analysis, the first step in this case is to define a reference gait, which will be the center of the sensitivity analysis. We will define the reference in the middle of the chosen range, for both step height and step length. To define the objective range, it is important to take into account the characteristics of the real robot (its size) and the characteristics of our bio-inspired controller (large step-to-step variations inherent to the controller and maximum muscle forces).

As the robot is modeled on the body of a 5-year-old child, the goal range for step length should be certainly smaller than $50\ [cm]$ when looking at data from [bKb93; Scr69; BA08]. Although no exact data concerning average step length depending on the height of a child were found, a range of $\mathbf{[30;\ 50]}\ [\boldsymbol{cm}]$ with reference of $\mathbf{40}\ [\boldsymbol{cm}]$ was chosen using these sources and initial optimization trials, $50\ [cm]$ being an ambitious upper bound (it corresponds to a normal step length for adult humans according to [bKb93]). Indeed, $50\ [cm]$ is longer than the leg length and was not obtained in the initial optimization trials. For humans, performing steps longer than the length of their legs is only possible for punctual variations according to [BA08].

Concerning the step height, [Dad+13] presents measurements of foot clearance for able-bodied human adults. Depending on the measurement set (experiments on large groups were performed in 2010 and 2011), the average foot clearance is $6\ [cm]$ or $7\ [cm]$. These measurements correspond to **adult humans**, and we can thus consider that again, smaller values should be considered for our $95\ [cm]$-high robot. However, we decide to take $\mathbf{6.5}\ [\boldsymbol{cm}]$as reference step height, for two reasons:

- Due to our bio-inspired control model using the CPG outputs as main muscle activation, the gait features tend to show significant variations from step to step. Low steps heights, when subject to such a variation, can result in the robot tripping if the swing foot hits the ground in the middle of the swing phase.

- Because our objective is to modulate the gait in order to avoid obstacles, relatively large step heights will be necessary. Such large step heights will be more unstable if they are further away from the reference height, hence making a relatively large reference step height desirable.

The objective range chosen for the step height is $[\mathbf{1;\ 12}]$ $[\boldsymbol{cm}]$. Both boundaries are really extreme values of what the Co-Man should be able to accomplish. For the lower boundary, the large variations of the gait that are inherent to our CPG-based control will make this step height difficult. Concerning the upper boundary, a step height of 12 $[cm]$ is very ambitious considering the height of the robot.
The reference gait features are represented in Figure 4.1.

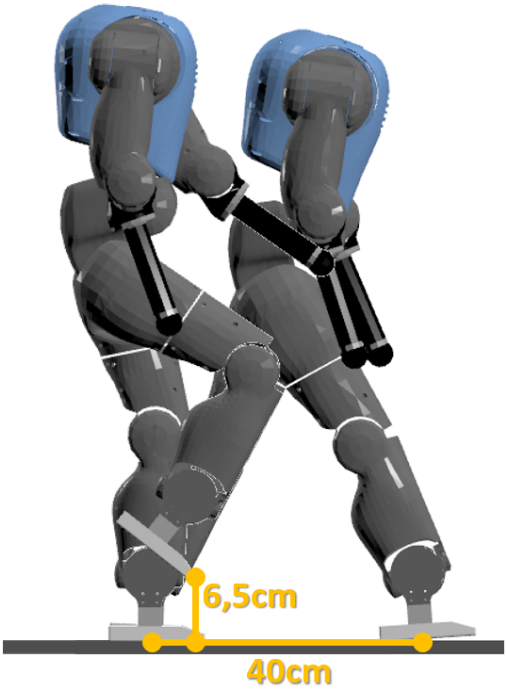


FIGURE 4.1: Reference gait features: step height of 6.5 $[cm]$ and step length of 40 $[cm]$.

4.1.2 Reference parameters set

Now that the reference gait features are defined (as a reminder: step length of 40 $[cm]$ and step height of 6.5 $[cm]$), optimizations were performed on these features. The optimizations were performed on **all** the open parameters of the gait. In total, 42 open parameters are concerned: 11 parameters related to the reflex rules, 10 parameters related to the modulation of the outputs of the CPG, the trunk reference angle Θ, 19 parameters related to the CPG itself (the neuron equations and the time constant) and 1 parameter related to the impedance-based initialization of the walk. The optimization procedure itself is explained in section 3.1. All these parameters, as well as their bounds, are given in appendix, section D.1.

The bounds were found by starting with large ranges for each value, and looking at the values obtained when performing multiple optimizations. This allowed to reduce the range to a narrower set, thus producing optimizations that converged faster. This is important, because PSO is a heuristic process: the solution obtained after a predefined number of iterations is not the mathematical optimum. It is thus crucial to narrow-down the range of the parameters so that the algorithm does not spend a lot of iterations evaluating solutions that do not give a stable gait before finding the area in which the solutions are interesting.

Using the parameters and bounds shown in section D.1, the PSO was run 5 times. This resulted in 5 different parameters sets, with the same gait features. By comparing the five different optimization results, it was possible to choose the less 'local' one.

This was done by computing, for every solution, the total distance to the mean of the solutions, according to Equation (4.1).

$$Distance(j) = \sum_{i=1}^{n} |param(i)_j - \overline{param(i)}| \tag{4.1}$$

Where $Distance(j)$ represents the total distance between solution j and the mean, n is the number of optimized parameters (42), $param(i)_j$ represents the value of the ith parameter in solution j and $\overline{param(i)}$ represents the mean value of parameter i across the five solutions. The solution with the smallest '$Distance$' was kept as reference gait. The advantage of choosing the reference solution by this method is that it helps to obtain a more general solution for this reference gait, instead of a special case that would less easily be modulated to obtain gait variations.

Figure 4.2 shows the features obtained during a simulation when using the reference set chosen by the method described above. As we can see, the features do oscillate around their reference position. This is inherent to the use of a CPG-based control, as we do not control the precise joint angles at any time.

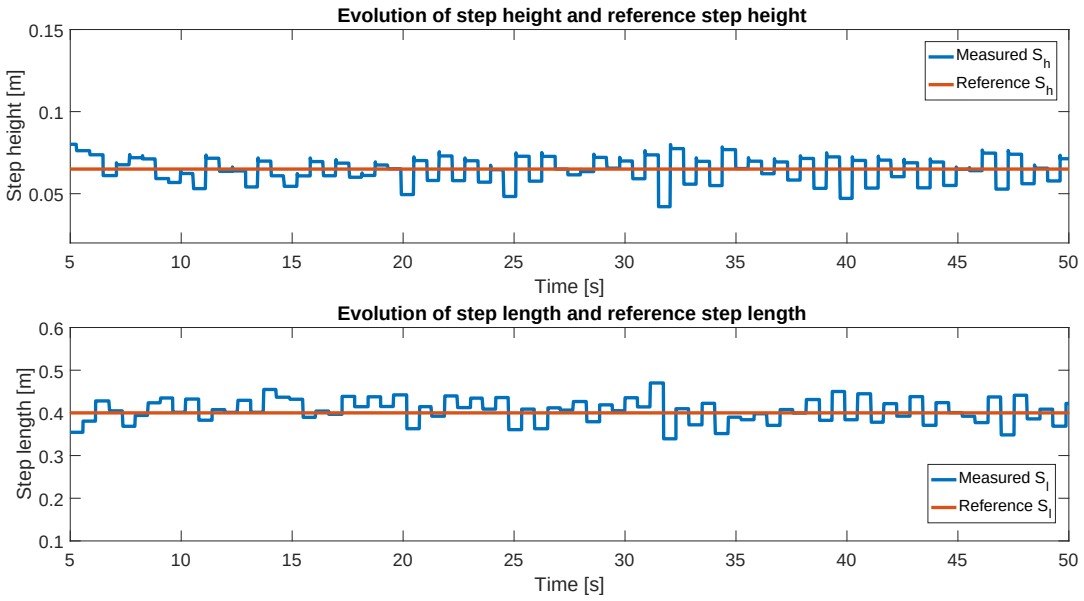


FIGURE 4.2: Reference gait features using the chosen reference parameter set.

4.2 Procedure for sensitivity analysis

Starting from the reference solution found in subsection 4.1.2, the objective is now to analyze the influence of the parameters on the gait features. As mentioned before, only the gains applied to the CPG outputs will be modulated, as well as τ, the time constant of the CPG and Θ, the trunk reference angle. As a reminder, the equations governing the muscle stimulations by CPG are given by Equations (A.4), in Appendix A.

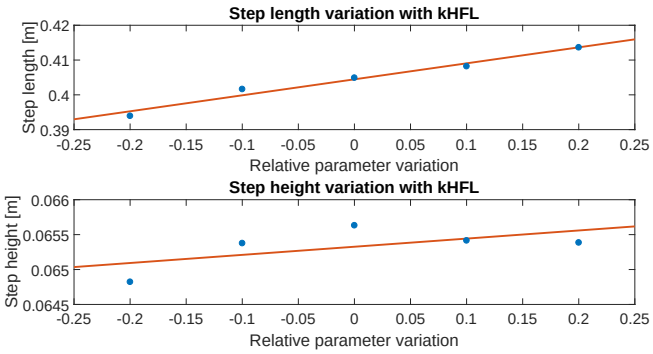
In total, the sensitivity analysis will hence be performed on 12 parameters (see section 2.3.2 for their introduction): τ, Θ, k_{HFL}, k_{GLU1}, k_{GLU2}, k_{HAM1}, k_{HAM2}, k_{HAM3}, k_{BFSH1}, k_{BFSH2}, k_{RF1} and k_{RF2}.

The objective is thus to determine the most important parameters out of this parameters set, thereby obtaining a reduced set of parameters to modulate while

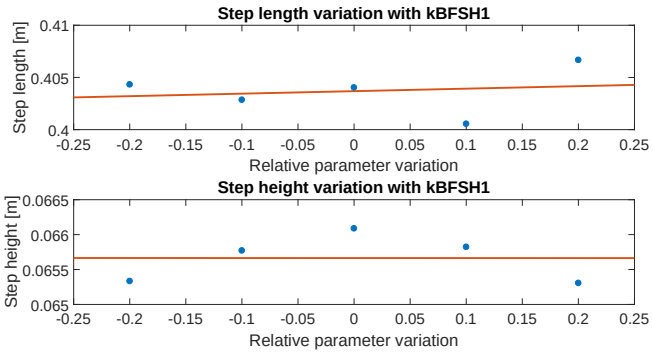
keeping the others at their reference value. This will help generate simpler control rules. To perform this analysis, different modifications are applied to each parameter while keeping the other parameters constant. These modifications are -20%, -10%, $+10\%$ and $+20\%$. For each parameter, one of these modifications is applied while freezing the other parameters to their reference value. The simulation is run 3 times and the average step length and step height are extracted. Then, the next modification is applied on the same parameter following the same procedure. When all modifications have been applied to all the parameters, we have a total of 4 (modifications) $\times$ 12 (parameters) $+$ 1 (reference) $=$ 49 step length/height measurements. Using these values, we want to estimate the influence of each parameter on every feature.

4.3 Results of analysis

To select the relevant parameters, the best linear fit (in a least-squares sense) is computed for both gait features for each parameter. From these 2×12 fits, the slopes are extracted and compared. Figures showing the fits obtained for every one of the 12 parameters, as well as all the slope coefficients, are available in Appendix C; for clarity, only a selection is shown here. The Matlab script used to compute all the slope coefficients and generate the figures shown here, is available in the digital matlab appendix.



(A) Sensitivity analysis for parameter k_{HFL}.



(B) Sensitivity analysis for parameter k_{BFSH1}.

FIGURE 4.3: Continuous lines represent the least-squares fit, blue dots the means of the different measurements.

Looking at Figure 4.3a gives an example of a parameter that does have an observable influence on only one of the two gait features of interest. Indeed, we can observe that the **step length** varies by almost 2 $[cm]$ when modifying k_{HFL} alone. However, this parameter does not seem to have a notable impact on step height, as the step height variation is less than 1 $[mm]$ between the two extreme measurements. Looking at Figure 4.3b, we can observe that this parameter does not have an influence on any of the two features: both variations are very small, if observable at all. For this reason, k_{HFL} is a parameter that will be kept and modulated to control the gait while k_{BFSH1} will stay at its reference value at all times.

The same analysis could be done for each one of the 12 parameters. The objective however is to compare the importances of the different parameters, and not to look at each parameter separately.

To compare more easily the slopes of all the parameters at once, Figure 4.4 and Figure 4.5 display the values of theses slopes, relatively to the maximum slope for

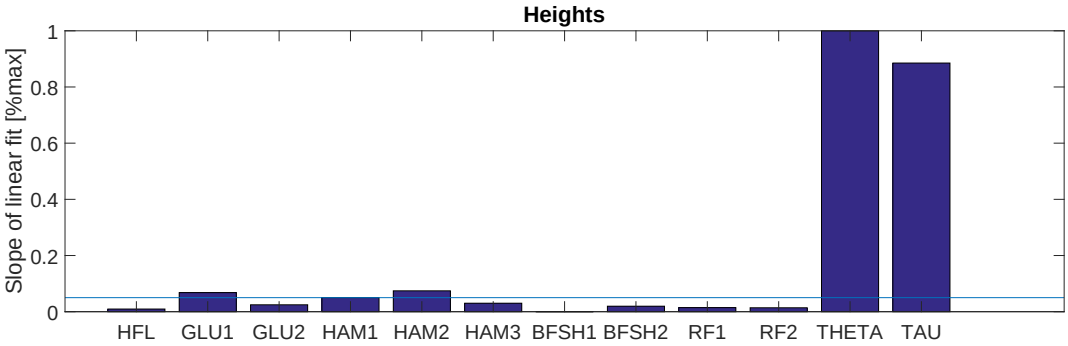


FIGURE 4.4: Comparison of slope coefficients for linear fits of height variations, displayed relatively to the maximum slope.

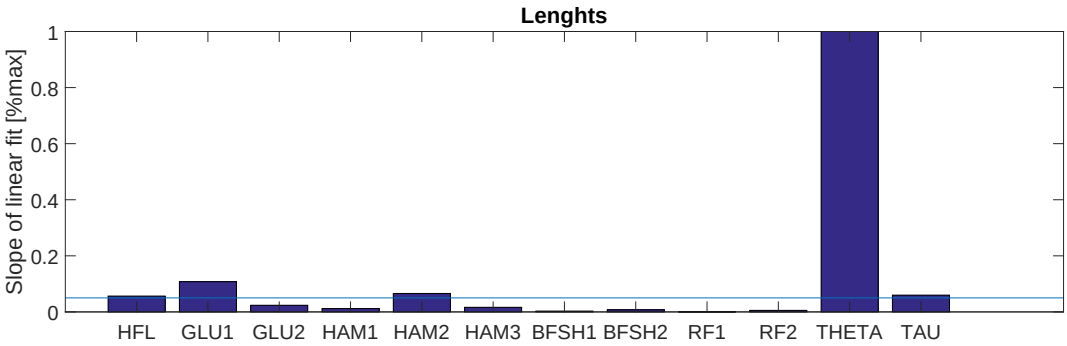


FIGURE 4.5: Comparison of slope coefficients for linear fits of length variations, displayed relatively to the maximum slope.

height and length, respectively. The first observation that can be done is that Θ has an influence that is significantly bigger than the influence of most other parameters, for both the step height and the step length features: Θ shows the maximum slope for both features, with no other parameter showing more than 10% of its slope (except for τ). This is also true for τ in the case of the step height. This is not surprising, as it is known that the trunk angle is a very important factor in gait modulation [NRI15; GH10]. This is especially true in our case, as the trunk angle is the control input for the reflex rules partially controlling the GLU and HFL muscle groups.

Concerning the oscillator time constant τ, its importance can also be expected to be big for both features [Mur+66; SG12], which is not the case for the step length here. This result should be taken with a grain of salt, given that the gait was too unstable to produce results in two out of the 4 variations applied to τ. This is represented in Figure 4.6, where the blue dots that would correspond to the results of the -10% and -20% do not exist.

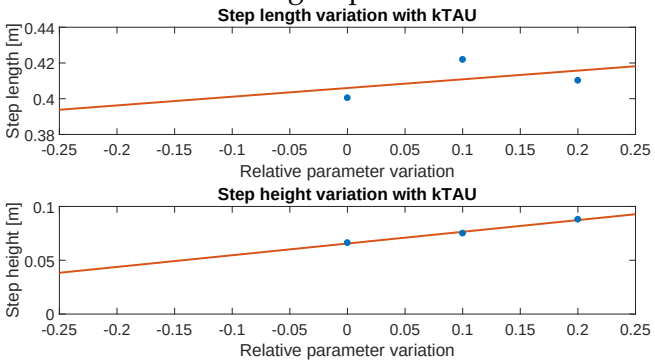


FIGURE 4.6: Sensitivity analysis for parameter τ. The continuous line represents the least-square fit while the blue dots represent the means of the different measurements.

The light blue reference lines drawn in Figure 4.4 and Figure 4.5 represent the arbitrary value chosen to determine the parameters to keep for modulation. This line, placed at 0.05, gives a uniform way to choose the parameters to keep: any parameter that has at least 5% the influence of the most important parameter is selected.

For Figure 4.4, the parameters that are selected by this criterion are Θ, τ, k_{GLU1}, k_{HAM1} and k_{HAM2}. In Figure 4.5, the selected parameters are Θ, τ, k_{HFL}, k_{GLU1}, and k_{HAM2}. Although chosen arbitrarily, this reference gives a total of 6 parameters to keep: the same parameters as [NRI15]. In section 5.1 and chapter 5, the parameters that will be modulated will thus be $\boldsymbol{\Theta}$, $\boldsymbol{\tau}$, $\boldsymbol{k_{HFL}}$, $\boldsymbol{k_{GLU1}}$, $\boldsymbol{k_{HAM1}}$ and $\boldsymbol{k_{HAM2}}$.

Chapter 5

Experiment: Mesh optimization regression and results

The goal of this chapter is to develop a controller able to modulate the gait on the ranges defined in chapter 4. To achieve this, a mesh optimization was performed: several optimizations were run, covering the whole objective range. For these optimizations, only the parameters selected in section 4.3 are optimized, while the others are kept at their reference value selected in subsection 4.1.2. This mesh optimization procedure, described in section 5.1, leads to 360 parameter sets throughout the objective space, each parameter set corresponding to another combination of step length and step height. We then use these 360 results to obtain a continuous modulation of the gait. To do this, polynomial regression models are computed for each of the 6 parameters selected in section 4.3 and optimized in section 5.1 (Θ, τ, k_{HFL}, k_{GLU1}, k_{HAM1} and k_{HAM2}). These regressions are then tested to verify that the gait features adapt correctly when the parameters are modulated using the computed models.

Another method to modulate the gait could have been to use look-up tables to pick the correct parameters depending on the high-level commands. However, this approach is less interesting, for two main reasons:

- An approach using look-up tables is less bio-inspired than a continuous modulation using adaptable gains, as it is unlikely that human muscles use look-up table to store low-level control parameters corresponding to all the possible situations.
- As stated in [SG15b], look-up tables gives less consistent properties across different situations because the parameters used result from independent optimizations.

5.1 Mesh optimization

The step lengths were discretized every 5 $[cm]$ while the step heights were discretized every 1 $[cm]$. This resulted in an optimization on 5 different step lengths and 12 different step heights. An extra streak of optimizations was performed afterwards, corresponding to a step length of 0.25 $[m]$, when looking at the results of the first optimizations. The step length/step height pairs optimized are graphically represented in Figure 5.1. For each length/height pair for which the gait was optimized, **5** optimizations were run, giving 5 resulting parameter sets. This gives a total amount of **5 (optimizations) $\times$ 6 (different lengths) $\times$ 12 (different heights) $=$ 360** resulting parameter sets to analyze.

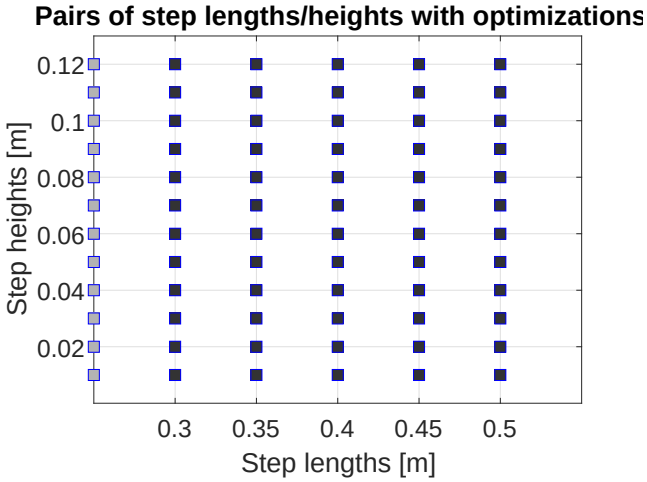


FIGURE 5.1: Pairs of step length/step height with which an optimization was performed. The light-gray column corresponding to a step length of 0.25 $[m]$ was added later on.

As said when choosing the step length and step height ranges in subsection 4.1.1, the chosen extreme values are ambitious. This is confirmed when looking at the results of the optimizations, as not all optimizations converge. For the purpose of this next part, where the optimization results are analyzed, an optimization is considered as 'successful', or converging, if the desired step length and step height were obtained. This is verified by a very simple criterion: if the total fitness of an optimization is > 400, the optimization is considered successful. This criterion is relatively easy to explain, referring to the fitness stages described in section 3.1: for an optimization to be successful, the two first stages will necessarily be fulfilled, resulting (together) in a fitness of 200. Then, the condition is that the required step height and step length are obtained. For these stages to be fulfilled, only ≈ 80 of fitness is required for each. However, the next two stages always give (together) > 60 fitness. We can sum this up as follows: $Fitness = 200 + 2 \cdot 80 + 60 = 420$, which shows that requiring a minimal fitness of 400 comes down to requiring the last two stages to be activated. When using this criterion to exclude the optimizations that did not converge, we obtain the result visible in Figure 5.2.

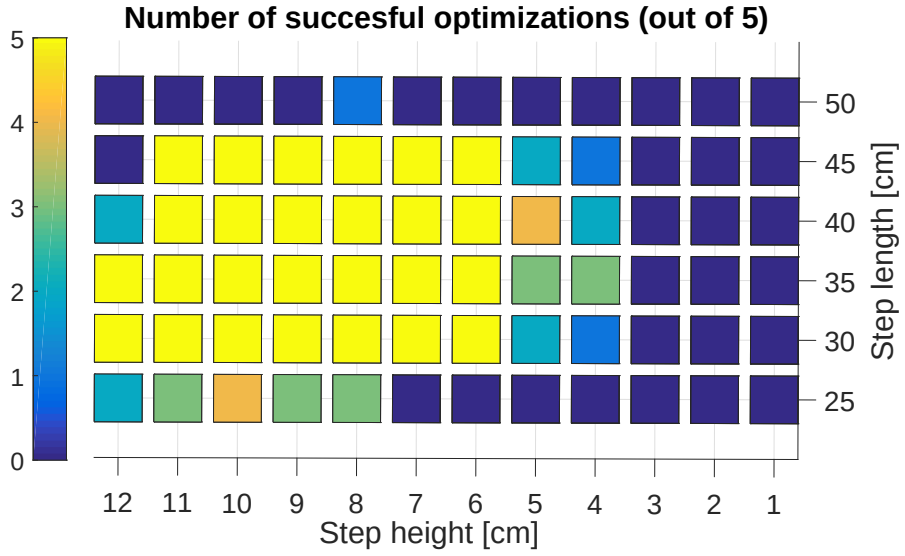
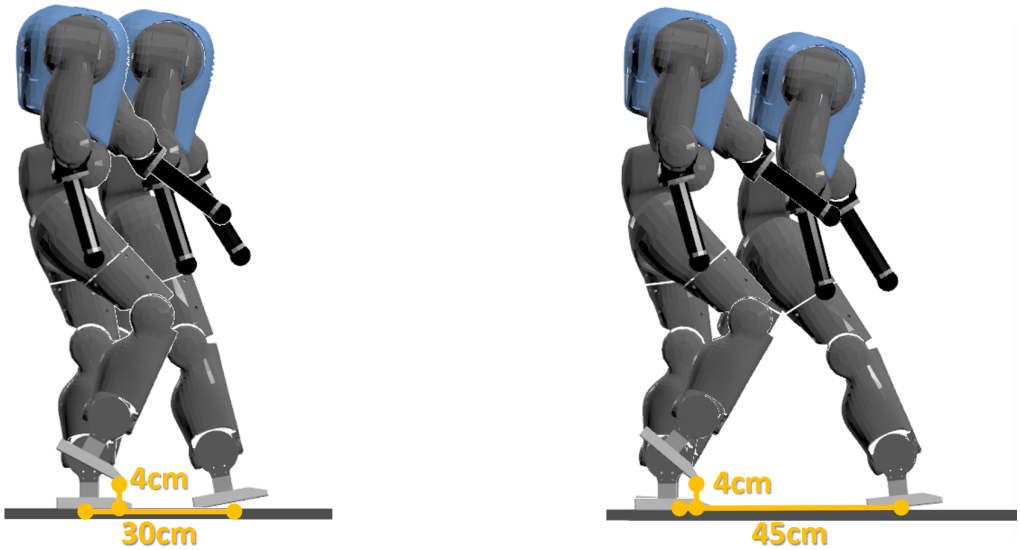


FIGURE 5.2: Representation of the number of converging optimizations for each optimized step length/step height pair.

As we can see in this figure, we get 5 successful optimizations out of 5 only in a reduced range. We can conclude from this graph that the objective of $50\ [cm]$ steps was too ambitious, as only one optimization converged for that step length (the one that converged was for a step height of $8\ [cm]$). The optimizations with a step length of $25\ [cm]$ were added after visualizing the rest of the results. We can observe that such a small step length is only obtained with relatively high step heights. Another conclusion we can have is that not a single optimization succeeded for a mean step height lower than $4\ [cm]$. This can be explained by the reasons given in subsection 4.1.1.

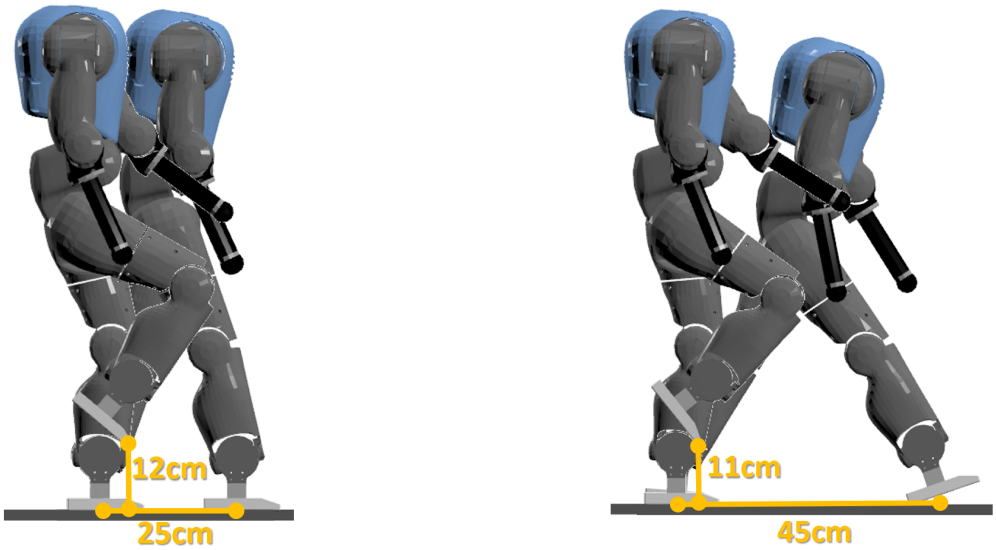
This paragraph presents the results corresponding to the 4 extremes reached here. S_l represents the step length, S_h represents the step height. A step length/step height pair will consistently be noted as $\{S_l,\ S_h\}$, with both measurement in **centimeters**. The 4 extreme cases represented here are $\{30,\ 4\}$ in Figure 5.3a, $\{45,\ 4\}$ in Figure 5.3b, $\{25,\ 12\}$ in Figure 5.4a and $\{45,\ 11\}$ in Figure 5.4b. Snapshots are shown in the report, but videos[1] are available as well and will be submitted with the electronic archive. When multiple optimizations converged for one step length/step height pair, the gait shown corresponds to the solution with the highest fitness.



(A) $S_l = 30$ (B) $S_l = 45$

FIGURE 5.3: Two gaits with $S_h = 4$. Snapshots taken at step height measurement and at the next strike.

[1]The name convention for videos is *meshN_sh_sl* where *meshN* represents which one of the 5 optimizations gave the result ($N \in [1;\ 5]$), *sh* is S_h in mm and *sl* is S_l in cm. Slow-motion versions of a few steps are available with *_slow* appended to the name.



(A) $S_l = 25$ (B) $S_l = 45$

FIGURE 5.4: Gaits with high step height ($S_h = 12$ and 11). Snapshots taken at step height measurement and at the next strike.

5.2 Regression

The regression chosen is a second-degree polynomial regression, for both dimensions. This allows to have a reasonably good fit (see the errors mentioned later in this report) while avoiding to overfit. The regression function used is of the form written in Equation (5.1),

$$f_i(S_l, S_h) = K_i + L_{10,i} \cdot S_l + L_{01,i} \cdot S_h + L_{11,i} \cdot S_l \cdot S_h + M_{20,i} \cdot S_l^2 + M_{02,i} \cdot S_h^2 \quad (5.1)$$

where i corresponds to one of the 6 parameters and the coefficients K_i, $L_{10,i}$, $L_{01,i}$, $L_{11,i}$, $M_{20,i}$ and $M_{02,i}$ are the coefficients of the polynom. The fit was done using the *Curve Fitting Tool* from *Matlab*.

To obtain good fits, the optimizations which did not converge (i.e. with $Fitness < 400$, see section 5.1) were excluded from the regression and the optimizations that did converge were weighted by their fitness when fitting the data. When several optimizations converged for one $\{S_l, S_h\}$ pair, the mean of their fitnesses and resulting parameters were used.

As an example of the resulting fits obtained, the fit for k_{HAM2} is represented in Figure 5.5. The excluded points represented correspond to the $\{S_l, S_h\}$ pairs that did not produce any converging solution, like represented in Figure 5.2. The data points represented in this figure correspond, for each data point, to the mean of the k_{HAM2} values obtained for all the converging optimizations on the corresponding $\{S_l, S_h\}$ pair.

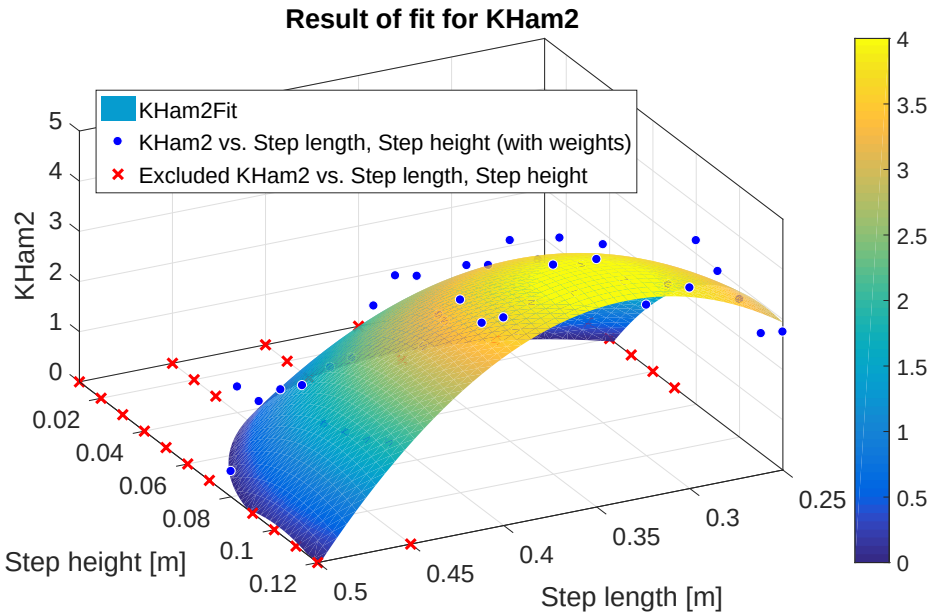


FIGURE 5.5: 3-dimensional representation of the polynomial fit obtained for k_{HAM2}, with blue points representing the mean of the optimization results and red crosses representing the excluded results.

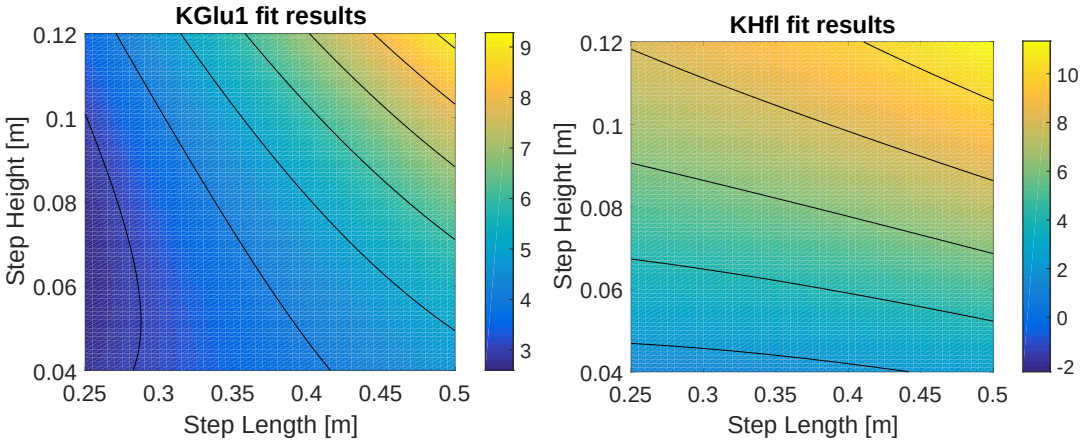
5.3 Results

Figures 5.6, 5.7 and 5.8 display colormaps of the parameters evolution on the selected step length and step height ranges, using the polynomial fit obtained.

Concerning the fit for k_{GLU1} first, in Figure 5.6a, we notice that (like most other parameters as we will see) this parameter increases with the step length and with the step height, and finds its maximum value with maximum values of step length and step height. As this parameter acts on the activation of the GLU muscle during the stance phase, this is expected: the muscle, situated behind the thigh, pulls the trunk backwards with respect to the stance leg. This helps to pull the swing leg forward, resulting in longer steps if the muscle is more activated.

Concerning the fit for k_{HFL} in Figure 5.6b, we note that this parameter is the one with the biggest range of all the modulated parameters. Like k_{GLU1}, it increases with both step length and step height. However, we notice that the influence of the step height on this parameter is much higher than that of the step length. This is confirmed by looking at Table 5.1 (especially when comparing L_{10} and L_{01}). This is logical, because (looking back at the meaning of this coefficient in section 2.3.2) this coefficient acts on the activation of the Hip Flexors during the swing phase. We can conclude that to increase the step height, a very important factor is to increase the force put into the muscle mainly controlling the swing; for the step length, this is less critical: the trunk angle has more influence on this feature.

When looking at the results for k_{HAM1} and k_{HAM2}, we notice that the amplitude of k_{HAM1} varies on a wider range that of k_{HAM2}. The modulation of the Hamstring is interesting to observe, as it is the only modulated muscle with a direct influence on the knee bending. Specifically, when recalling the equations in section 2.3.2, we know that k_{HAM2} is activated during the opposite stance phase: it is the parameter with the most influence on knee bending during the swing phase. Indeed, we see that this parameter is highest when high step heights are required, with relatively small step

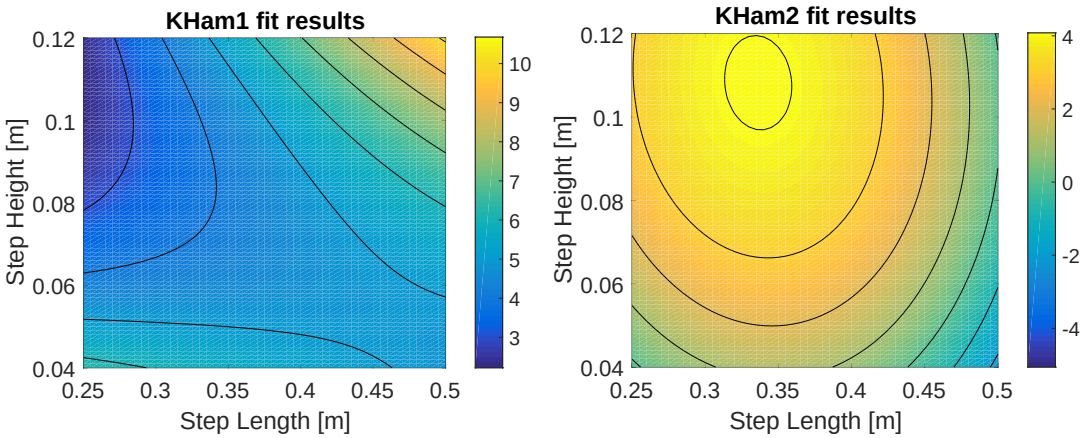


(A) Fit for k_{GLU1}.

(B) Fit for k_{HFL}.

FIGURE 5.6: Colormap representation of the fits obtained for k_{GLU1} and k_{HFL}.

lengths. A possible explanation for this is that the increased knee bending is only critical to obtain a high step height when the step is small: for bigger step lengths, the bigger Θ and stronger swing (thanks to k_{HFL}) already 'throw' the leg higher above the ground.



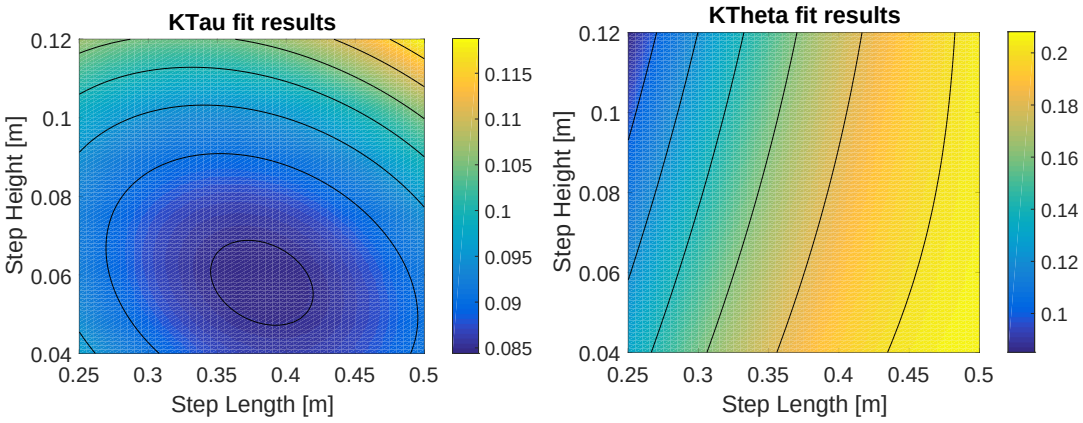
(A) Fit for k_{HAM1}.

(B) Fit for k_{HAM2}.

FIGURE 5.7: Colormap representation of the fits obtained for k_{HAM1} and k_{HAM2}.

Concerning the last two parameters, the only parameters that are not gains applied on the outputs of the CPG, we observe that those are the parameters with the lowest amplitude of variation. This was to be expected, considering the very high sensitivity to those parameters (see section 4.3 and Appendix C). An interesting observation concerning Θ (the trunk reference pitch angle) is that increasing the step height actually **decreases** its value. Biologically, this can be understood by the fact that when leaning forward, it is more difficult to bend the knee. When looking at the model, this makes sense too: an increased solicitation of the Hamstring muscles during the swing will not only bend the knee, but also pull the trunk backwards. As expected from the sensitivity analysis, we observe that the trunk angle has a notable influence on the step length. Concerning the CPG time constant τ, the main conclusion is that it is more important with regards to step height than to step length. The fact that τ is maximum when the step length and step height are maximum can be understood intuitively, as this is the situation in which the step can be expected to take the most

time. If the objective was also to modulate the step period, τ would probably be crucial as it is the time constant of the CPG, synchronizing the stimulations.



(A) Fit for τ.

(B) Fit for Θ.

FIGURE 5.8: Colormap representation of the fits obtained for τ and Θ.

Table 5.1 shows the coefficients obtained for the fit of every parameter. All the figures presented above, as well as the polynomial fits, have been generated and computed using the scripts presented in the digital matlab appendix.

TABLE 5.1: Coefficients of normalized and centered polynomial fits, truncated to 3 digits after the comma. The coefficients used in the code are not normalized nor centered, and are used with all their computed digits.

Coefficient	k_{GLU1}	k_{HAM1}	k_{HAM2}	k_{HFL}	τ	Θ
K	4.399	4.258	3.579	6.032	0.087	0.168
L10	1.078	0.916	-0.421	0.590	0.000	0.029
L01	0.802	0.120	0.797	2.422	0.006	-0.008
L11	0.338	0.809	-0.083	0.183	0.002	0.004
M20	0.005	0.017	-0.714	0.062	0.002	-0.006
M02	0.187	0.558	-0.388	-0.169	0.004	0.000

5.4 Performances

To enable the control of both features in real time during the simulation, two separate commands were added. The command for the step length is limited to the $[0.25;\ 0.45]\ [m]$ range. This is not the same range as originally intended in chapter 4: it has been adapted when analyzing the results of the mesh optimization performed in section 5.1. It takes a minimum of $1.33\ [s]$ to update the command from one extremity of the range to the other, resulting in a maximum step length reference modulation speed of $0.15\ [m/s]$. Similarly, the range for the step height command is fixed to $[0.04;\ 0.12]\ [m]$. To go from one extremity of the range to the other, it takes $1.45\ [s]$ for the step height. This corresponds to a maximum step height reference modulation speed of $0.055\ [m/s] = 5.5\ [cm/s]$.

In this section, a first assertion of the results is done by modulating each feature separately. After that, both features are modulated in one simulation. Videos of the simulations presented here are submitted with this report.

5.4.1 Modulation of step length

In Figure 5.9, the step length is modulated[2]. From this figure, we can first observe that as expected, the step length is modulated by the command sent to the controller. However, the step length measurement oscillates around the command with higher amplitudes than those observed in Figure 4.2.

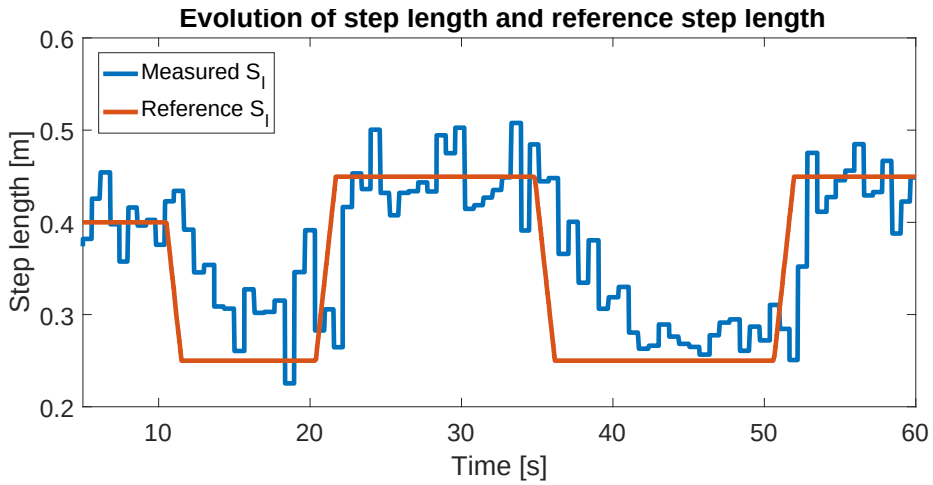


FIGURE 5.9: Modulation of the step length command while keeping the step height command constant at $6.5\ [cm]$.

5.4.2 Modulation of step height

Figure 5.10 shows the successful modulation of the step height[3]. Although, again, oscillations around the command value are present, the amplitude of these oscillations never exceeds $2\ [cm]$, which can be considered reasonable. Moreover, this feature adapts very quickly to new commands, as the oscillations stabilize at their steady-state value only 1 second after the end of the command update.

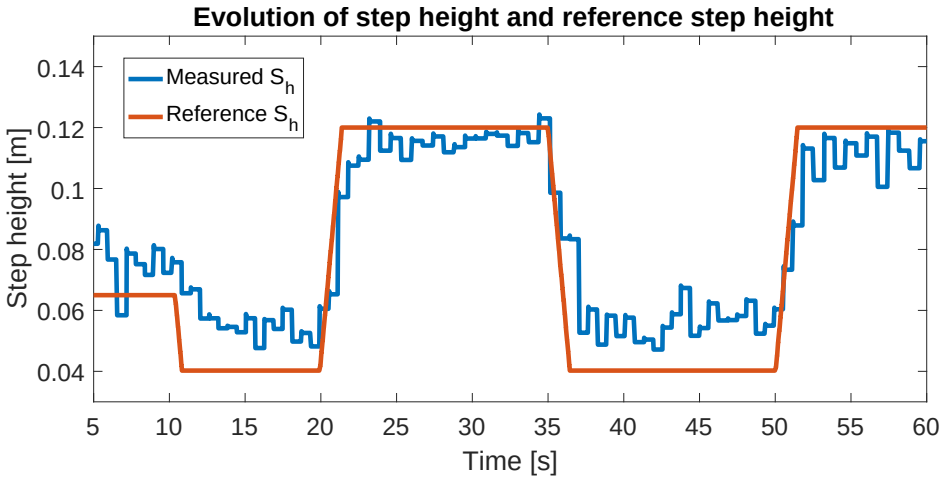


FIGURE 5.10: Modulation of the step height command while keeping the step length command constant at $40\ [cm]$.

[2]The video of this simulation is called *step_length_mod.webm*.
[3]The video of this simulation is called *step_height_mod.webm*.

5.4.3 Modulation of both features

Now that each feature has been shown to be modulated separately, Figure 5.11 shows the results of a simulation when modulating the two features together[4].

A first interesting observation we can make is that both features show oscillations of wider amplitude when the other feature is not set at its center reference. On the step length, these oscillations are of high amplitude (about $10\ [cm]$), which represents $\approx 50\%$ of the considered objective range. The step height on the other hand shows less significant variations: $2-3\ [cm]$, which represents $\approx 30\%$ of the considered range. These oscillations, present on both features together, result in an unstable gait: when modulating the gait, the robot often **falls** when command values at the limits of the ranges are sent. Videos showing such problems are submitted with this report.

Furthermore, although it is difficult to visualize clearly a mean value for each feature during the steady-state references, we can see that *both features are not independent*. For example, this is visible after $t = 20\ [s]$. We can see that increasing the reference step height decreases the measured step lengths. Then, after $t = 40\ [s]$, the measured step length increases following the lower step height reference.

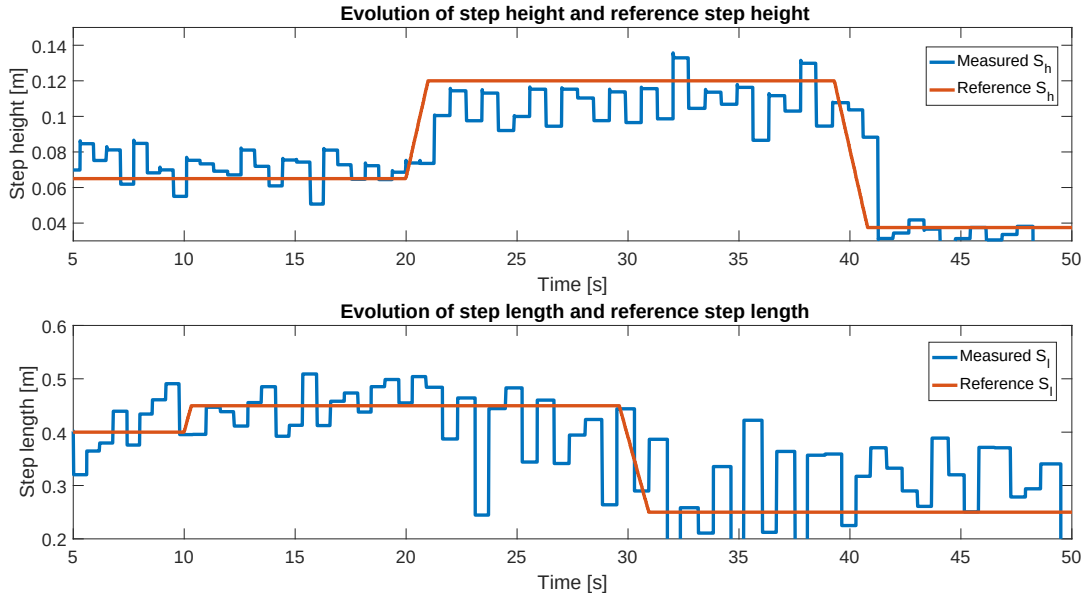


FIGURE 5.11: Modulation of the step length and step height commands.

5.5 Conclusion

Looking back at the initial objectives of this modulation, the first conclusion we can draw is that the primary objectives have been completed: the step height and step length can now be modulated using high-level commands. Although working, this modulation shows mixed results.

First, in terms of reliability: on the step height, the modulated gait shows oscillations of about 30% of the considered range. The step length however shows much bigger oscillations of up to 50% the considered range. Due to these wide oscillations on both gait features, the controller is not reliable and often causes the robot to fall.

[4]The video of this simulation is called *twomodifs.webm*. Simulations ending with the robot falling are called *twomodifs_failN.webm*.

Then, in terms of independence: the step height and step length are not independent, as clearly visible in Figure 5.11. The error on the control of one feature depends on the reference value of the other.

This low reliability is a problem which we will try to solve in another experiment. One method to solve this problem could be to modulate extra muscles: modulating the TA muscle for example could keep the feet more horizontal during swing, preventing the robot from falling if the foot strikes too early. Another method to consider to obtain better results would be to optimize the regression parameters obtained, like in [NRI15]. According to this paper, this kind of optimization can produce results on a wider range.

Chapter 6

Experiment: Cooptimization of the modulation parameters

The results obtained by the 'Mesh optimization and regression' experiment described in chapter 5 are mitigated. Indeed, although the resulting controller enables to control both features, it also presents a lot of disadvantages:

- Step length and step height are not independent; the reference value applied on the first has an effect on the error observed on the second and vice-versa;
- The resulting controller shows oscillations of high amplitude, so the obtained gaits are not predictable;
- As explained at the end of chapter 5, the stability of the controller is not satisfying. The robot often falls when control values close to the limits of their range are applied.

These limitations could be explained by various possible causes. First, the way the regression was performed can be questioned: considering the multiple parameters involved, each $\{S_l, S_h\}$ pair likely has **multiple 'local minima'** of parameter sets that give satisfying results – this is possible due to the optimization procedure being used, see section 3.1. Assuming the different optimization results for one $\{S_l, S_h\}$ pair correspond to different local minima, taking the **mean** of these solutions may result in a parameter set that does not produce satisfying results. Then, performing a regression on the means – each mean corresponding to an unsatisfying mixture of several local minima – will likely not result in a satisfying controller. Another possible explanation raised in chapter 5 is that the number of **modulated muscle stimulations was insufficient** to give enough modulation possibilities to the muscles. Indeed, as most the kinematics of the knee joint are important for both features, it is crucial to have enough DOFs to control the motion of this joint.

The experiment described in this chapter aims to solve these problems by obtaining new continuous modulation rules, using another procedure. This new procedure addresses both problems of the chapter 5 procedure. Firstly, to increase the DOFs on the knee joint modulation, additional muscle stimulation parameters are modulated in this experiment. To achieve this, the stimulations sent to both the BFSH and the RF muscle were modified. The stimulations sent to these muscles for this experiment and the one presented in chapter 7 are detailed in Appendix A; they were developed to generate a pattern similar to the stimulations generated by the reflexes presented in [SG15a]. Two parameters of the RF muscle stimulation are now modulated. As RF acts on the knee joint, the torque applied to the knee joint now has additional adaptability.

Secondly, the coefficients of the polynomial control rules – one control rule for each of Θ, τ, k_{HFL}, k_{GLU1}, k_{HAM1}, k_{HAM2}, $\boldsymbol{k_{RF1}}$ and $\boldsymbol{k_{RF3}}$ – are optimized. It is very

important to notice the difference with the previous experiment: the *coefficients of the polynomials* are optimized, while in chapter 5 the stimulation parameters themselves were optimized, and a regression was then performed to obtain the polynomials. The optimization will in this case give the coefficients of the control rules, instead of giving points in the modulation space between which a regression should be performed.

6.1 Cooptimization of the coefficients

The important question is then to know *how* the polynomial's parameters are optimized, and what fitness function is used to evaluate the parameter sets. The generic equation used for each of the 8 modulation polynomials is given below in Equation (6.1), where $\Delta_l = S_l - 0.40$ and $\Delta_h = S_h - 0.065$. Using Δ_l and Δ_h as arguments for the modulation polynomials allows to center the function on the $\{40, 6.5\}$ pair[1]. This polynomial has 6 coefficients to optimize – K_i, $L_{10,i}$, $L_{01,i}$, $L_{11,i}$, $M_{20,i}$ and $M_{02,i}$. As we have to optimize these 6 parameters for each of the 8 modulated parameters, this results in a total of **48 polynomial coefficients to optimize**.

$$f_i(\Delta_l, \Delta_h) = K_i + L_{10,i} \cdot \Delta_l + L_{01,i} \cdot \Delta_h + L_{11,i} \cdot \Delta_l \cdot \Delta_h + M_{20,i} \cdot \Delta_l^2 + M_{02,i} \cdot \Delta_h^2 \quad (6.1)$$

To evaluate one set of polynomial coefficients, 40 simulations of a maximum of 60 seconds (in case there is no fall) each are run. These 40 simulations correspond to the 40 $\{S_l, S_h\}$ pairs of the height range $[5;\ 12]$ $[cm]$, discretized every 1 $[cm]$ and the length range $[25;\ 45]$ $[cm]$, discretized every 5 $[cm]$. The set of polynomial coefficients is used to compute the values of the 8 parameters for each of these 40 pairs. The resulting fitness at each of these discretized points of the modulation space is then obtained. Finally, the *mean* of the 40 fitnesses is computed. This mean is the fitness of the considered set of polynomial coefficients.

We can see that this method indeed addresses the two possible problems mentioned previously: by optimizing one set of polynomial coefficients on the whole space, the problem of multiple different local minima at different points of the modulation space is avoided. Note however that the solution can still be a local minimum – which is still better than the mean of various local minima. By modulating k_{RF1} and k_{RF3} as additional parameters, the torque applied on the knee joint can now be controlled more precisely.

To obtain satisfying results for this method, the optimization ranges for the 48 polynomial coefficients were chosen in two steps. First, a mesh optimization like the one described in chapter 5 was performed. A regression was then realized on the results of this mesh optimization to obtain an order of magnitude for each of the 48 coefficients. The optimization range for each one of the 48 coefficients was chosen using the results of the regression as an indication: the optimization ranges were centered on the values found by regression. Then, the cooptimization itself was launched. The resulting boundaries for all 48 coefficients are given in appendix, section D.3. In Table 6.1, the resulting coefficients are shown.

6.2 Performances of the resulting modulation

Using the cooptimization procedure described above, a set of 48 polynomial coefficients was obtained. In this section, the performances of the resulting modulations are shown, quantified and discussed. All the figures illustrating the performances of the

[1] As a reminder, the $\{S_l, S_h\}$ notation introduced in chapter 5 is expressed in *centimeters*.

TABLE 6.1: Polynomial coefficients obtained by cooptimization, truncated to 3 digits after the comma.

Coefficient	k_{GLU1}	k_{HAM1}	k_{HAM2}	k_{HFL}
K	1.812	7.219	3.670	4.742
L10	7.467	12.314	-3.773	11.571
L01	-7.465	24.627	-8.266	51.312
L11	103.642	-203.620	-45.721	189.697
M20	53.794	94.483	-19.552	80.765
M02	569.586	-1074.435	8.787	262.557
	k_{RF1}	k_{RF3}	τ	Θ
K	2.230	2.011	0.106	0.128
L10	12.662	16.214	0.038	0.324
L01	-96.920	34.332	-0.147	-0.378
L11	-374.029	125.272	0.281	-3.740
M20	32.925	140.274	0.230	-1.052
M02	904.990	423.944	4.597	-18.810

step height and step length modulations were obtained by performing measurements during simulations of 60 seconds. The space was discretized every 0.5 $[cm]$ for the step height and every 1 $[cm]$ for the step length. For step length, the modulation is analyzed on the $[20;\ 50]$ $[cm]$ range. This is a wider range than the range on which the coefficients were explored, made possible thanks to the better performances obtained. Because of the more limited performances of the step height modulation, the step height range is explored on $[4;\ 12]$ $[cm]$, only expanding the lower bound of the range.

6.2.1 Performance of step height control

We start by measuring the performance of the step height control, shown in Figure 6.1. For a perfect modulation, the figure on the left would show only *vertical* contour lines, corresponding to the varying step height commands. We can notice however, that the step height is not only dependent on the step height command (horizontal axis) but also on the step length command (vertical axis). The most notable influence is witnessed for low step lengths (below 30 $[cm]$). Indeed, for such low step length commands, the step height is not responsive to its command for commands > 8 $[cm]$.

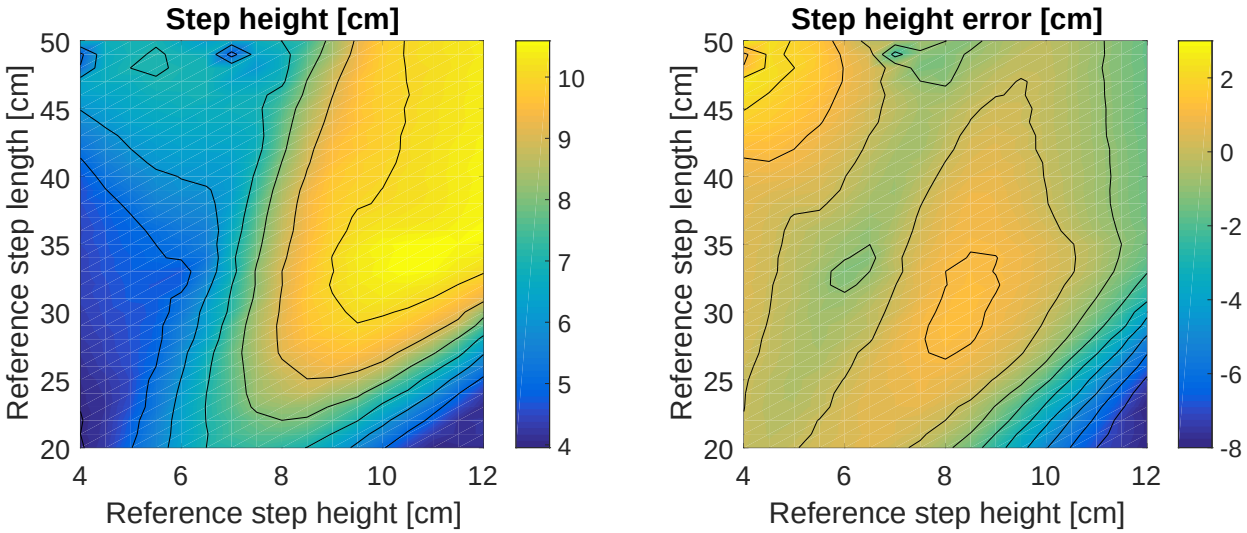


FIGURE 6.1: Performance of the cooptimized modulation parameters for step height control.

In the figure on the right of 6.1, showing the difference between measured step height and reference step height (the command), this behavior is, again, clearly visible: in the corner of the space corresponding to simultaneously low step height reference and high step length reference, the step height error falls almost linearly with the augmenting step height reference. Globally, the measured step height never exceeds $\approx 10.5\ [cm]$; this is in the same range as the observations done on the modulation developed in chapter 5 (as this measurement is a *mean*). Another area of the modulation space where a more significant error can be observed is the area with step length above 40 $[cm]$ and step height below 6 $[cm]$. In this area, the error is *positive*, showing that the measured step height is higher than its reference. This points out that with very high step lengths, it is difficult to keep foot clearance at a low value. We also notice that globally, the problems are located mainly at the boundaries of the range.

Finally, Figure 6.2 (left) shows the standard deviation on the measured step height, for the whole space. Generally speaking, we can observe that this standard deviation is small and does not vary a lot throughout the space. Only two points show higher standard deviations on the step height, both for high step lengths.

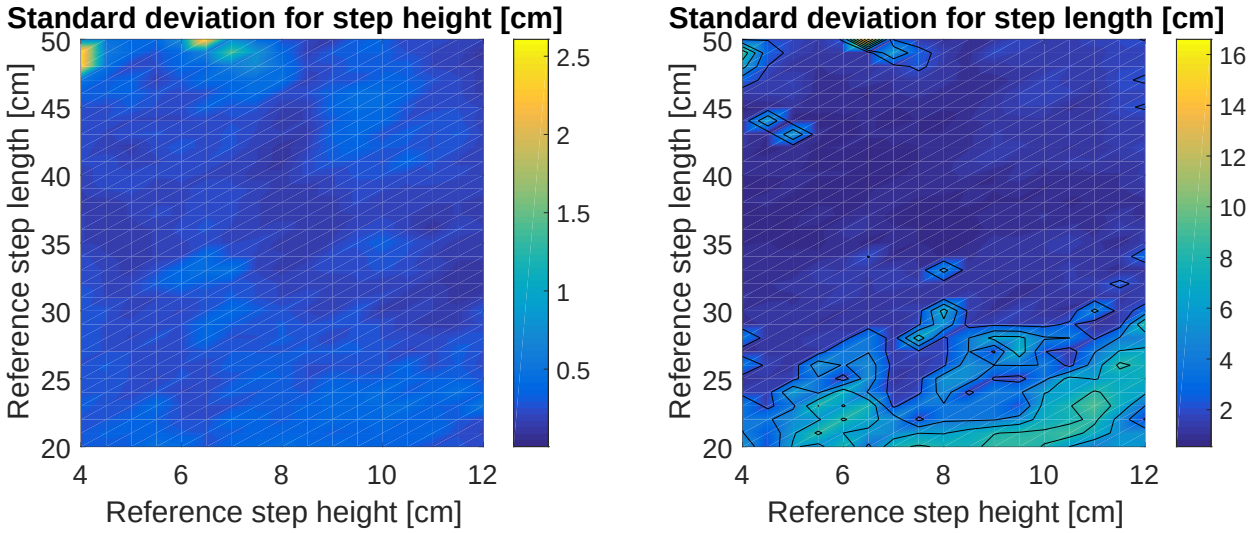


FIGURE 6.2: Standard deviation (in [cm]) of the step height (left) and step length (right).

6.2.2 Performance of step length control

The performance of the step length modulation is represented in Figure 6.3. The representation of the measured step length depending on the two references, in the graph on the left, would show *horizontal* contour lines in the case of a perfect control. From this graph, we see that the step length control shows less obvious tendencies of correlation between step length and step height than the equivalent graph for step height does (Figure 6.1). However, this modulation is by no means perfect either; most notably in the case where step height and step length references are both high. In that situation, the step length shows a negative error (see Figure 6.3, right). This corresponds to *too short steps* for high step height references ($> 9\ [cm]$). We can conclude that with the obtained modulation, *high step height references force the steps to be short, and small step length references force the step heights to be small.* As for the step height control, the problems are located mainly at the boundaries of the range.

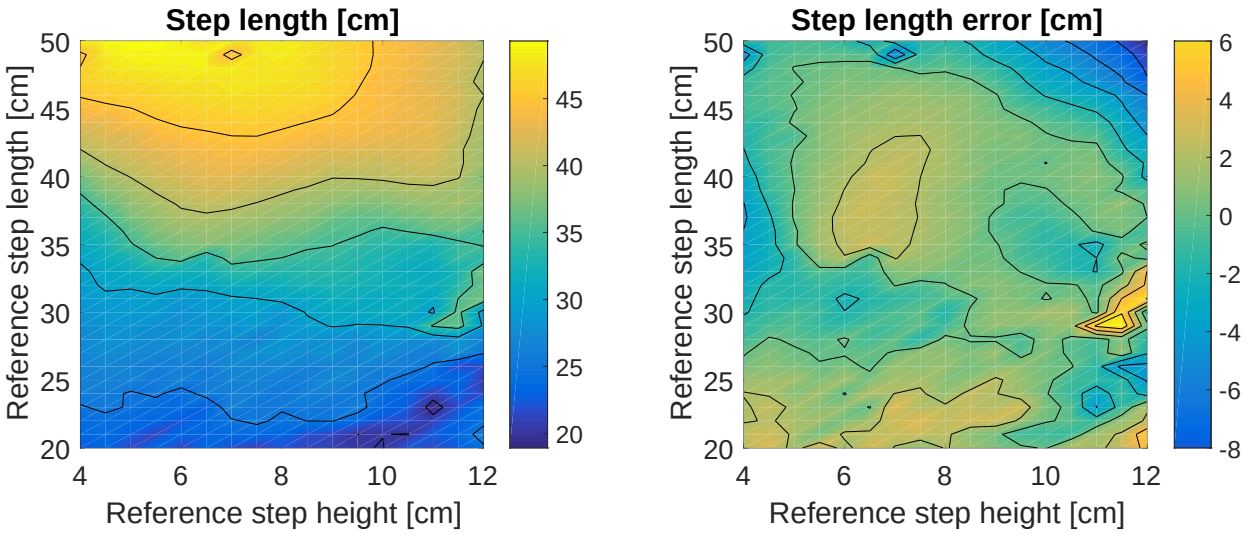


FIGURE 6.3: Performance of the cooptimized modulation parameters for step length control.

When looking at the graph on the right of Figure 6.3 again, we can also see a zone where the step length error is positive of about 6 $[cm]$, for step lengths ≈ 30 $[cm]$ and step heights ≈ 11 $[cm]$. This area also shows a big amplitude for the step height error; these pairs of step length-step height reference should hence be avoided when using this modulation, as the control will not be accurate.

On Figure 6.2 (right), we can see the standard deviation on the step length throughout the modulation space. We can see one main trend: the standard deviation *increases with decreasing step lengths.*

6.2.3 Performance of both controls combined

Having discussed and quantified the control of each feature separately, it is now interesting to visualize the performances of the resulting controller when modulating both features at the same time. Figure 6.4 shows the modulation space categorized in 4 different categories, depending on the accuracy of the control of each feature. For this figure, the threshold for a feature to be accurately controlled was arbitrarily set at 1.5 $[cm]$ error in absolute value for the step height and 3 $[cm]$ error in absolute value for the step length control. Using these criteria, the 4 categories correspond to the areas where no feature is accurately controlled, one of both features, or both.

This accuracy figure is useful to visualize that for the most part of the regression space, both features are accurately controlled. Keeping in mind that the cooptimization was realized on a range of $[25;\ 45]$ $[cm]$ for the step length, and starting from 5 $[cm]$ for the step height, we conclude that most parts where only one of both features is accurately controlled are actually outside of the optimization range.

The biggest difficulty in accurate control corresponds to low step lengths and high step heights. This is logical, as the problems of both features are located in that area:

- The step height shows errors of big amplitude – increasing linearly with the step height reference – in that area;
- The small area where a big *positive* error was observed on the step length is also in this zone;

- Finally, the step length shows a higher standard deviation in that area – hence contributing to the low reliability of the modulation in that area.

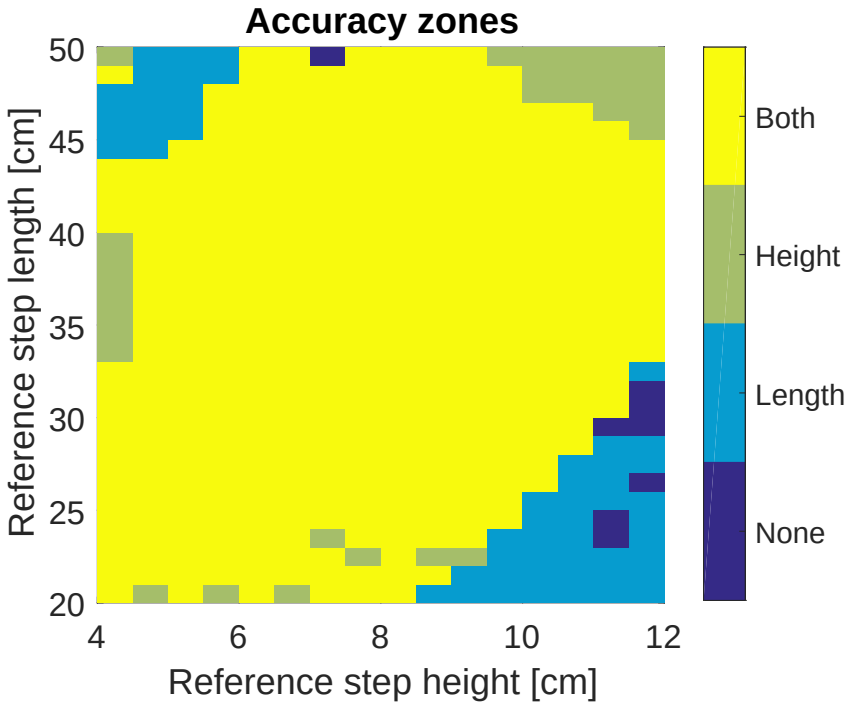


FIGURE 6.4: Graphical representation of zones where the control of step height, step length or both is accurate. Step height and step length are considered as accurate for mean errors smaller than $1.5\ [cm]$ and $3\ [cm]$ respectively, in absolute value.

Knowing the zones in which the modulation should work well, and less well, we can now evaluate the transitional and steady-state performances of the modulation, as represented in Figure 6.5 (left). The measurement of both step length and step height is a discrete measurement: the measurement is realized at foot strike and when both ankles cross, respectively. A big problem with the modulation obtained in chapter 5 were the *step-to-step oscillations* on the gait features. As visible on the graph, these step-to-step variations are now a lot smaller, showing good predictability, and robustness against noise. The adapted accuracy zones figure on the right of Figure 6.5 shows the successive $\{S_l,S_h\}$ references applied to the robot controller. Each reference stays stable for 10 seconds, and each transition lasts 5 seconds[2].

The first reference is $\{40,6.5\}$. This corresponds to the pair taken as basis for the sensitivity analysis of chapter 4 and the gait obtained with the current controller and this reference is the one analyzed in chapter 3. As visible here in Figure 6.5, both step length and step height measurements present small offsets with respect to their reference values; these offsets are however of small magnitude, as are the oscillations.

The second reference is $\{25,6.5\}$, and is applied from $t = 20\ [s]$. As the new reference is applied, we notice that the measured value on step height changes from presenting a negative offset with respect to the reference to presenting a positive offset. This is consistent with Figure 6.1 (figure on the right): the error changes from a negative value to a positive one when transitioning from $\{40,6.5\}$ to $\{25,6.5\}$. Concerning the step length, we notice a comparable (slightly smaller) offset, with wider

[2]The video showing this simulation is called *transitions.webm*.

oscillations. This is also coherent with the individual results presented previously, specifically concerning the increased standard deviation when decreasing the step length reference.

The next transition brings the modulation to reference 3, {25,4}. This corresponds to the lowest step height reference possible, but the resulting reference pair is still a 'yellow-zone' in the accuracy zones figure (Figure 6.4), and indeed we observe that both features correctly follow their respective references. The step length error is slightly smaller.

The fourth reference brings the controller to {45,4}. This reference is in a light blue area in Figure 6.4, meaning only the step length can be accurately controlled there. Indeed, we see very clearly the step height error increasing when the step length transitions to 45 $[cm]$. The step height increases of ≈ 2 $[cm]$ at the same time.

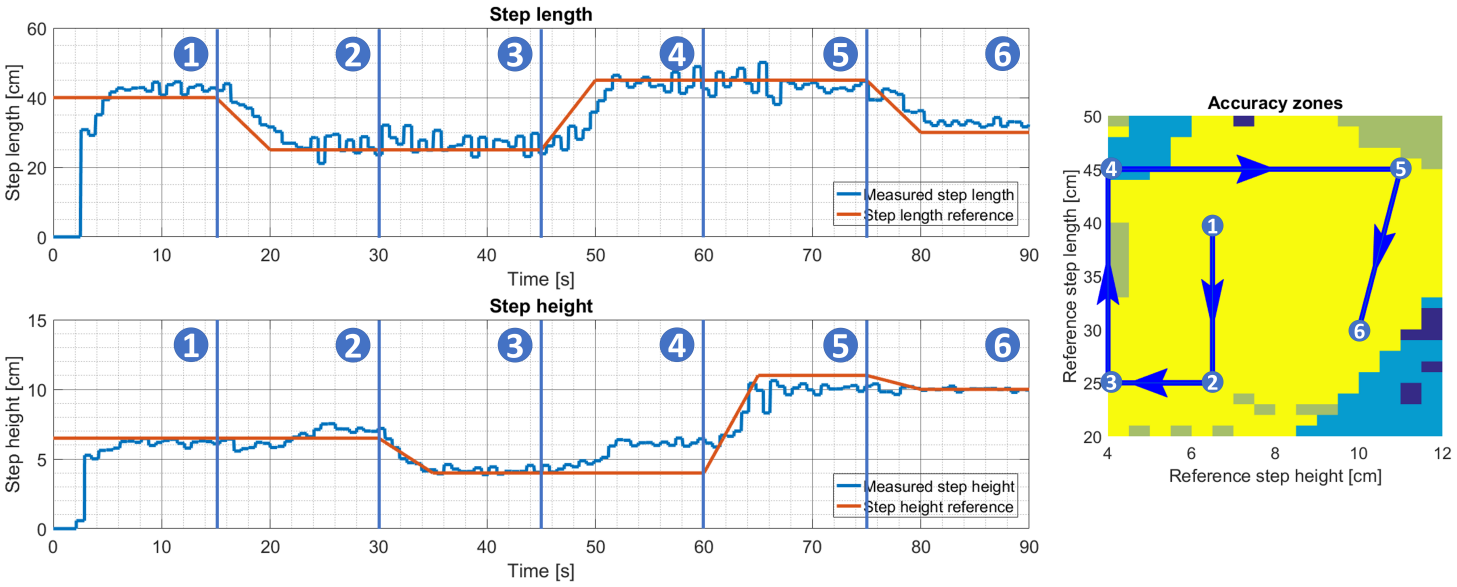


FIGURE 6.5: Transitional performances on control of step length and step height. Left: each measured variable is plotted with its reference. Right: the consecutive references are shown, where each number corresponds to a reference that is applied for 10 seconds and each arrow corresponds to a transition that lasts 5 seconds. The corresponding numbers are shown on the graph (left).

Reference 5 ({45,11}) shows that we approach the limits in step height modulation: although the reference is 11 $[cm]$, the step height oscillates around 10 $[cm]$. In the meantime, the step length is affected by this step height increase and shows a small negative error.

Finally, the last reference, at {30,10} shows a steady-state reference with almost no error, for each of both features. The whole experiment shown in Figure 6.5 also globally shows that the transition from one steady-state reference to another is quite fast. Indeed, the gait stabilizes around its new steady-state values with a delay of about 2 seconds at every transition. This corresponds to a bit more than three steps, as one step lasts about 0.65 $[s]$. The references however are applied after transitions of 5 seconds, so the response to steps on the references would likely take a few more steps to settle.

In Figure 6.6, snapshots of all consecutive foot strikes around the transition from reference 3 to reference 4 are shown. This gives a good insight into the step length evolution after the step length reference modification.

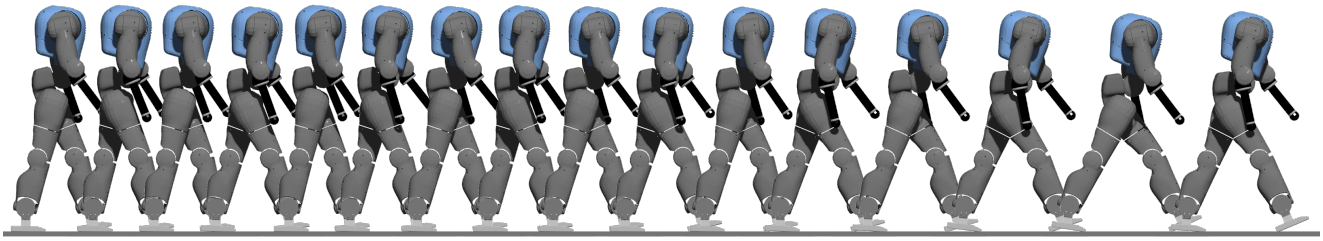


FIGURE 6.6: Transition from Reference 3 to 4, with snapshots taken at each consecutive strike, from $t = 42$ $[s]$ to $t = 52$ $[s]$.

Finally, Figure 6.7 shows two types of snapshots, superimposed. Faded in the background are the snapshots taken at every foot strike during the transition from reference 4 to reference 5. In front, the snapshots are taken at the moment when step height is measured, for every step. Although this feature is less visual than the step length feature, one can still clearly see the evolution in step height from step to step. In particular, it is clearly visible that the knee and hip joints show increased flexion and extension, respectively, when increasing the step height. The specific kinematic (and dynamic) modifications resulting from this modulation will be discussed in the next section.

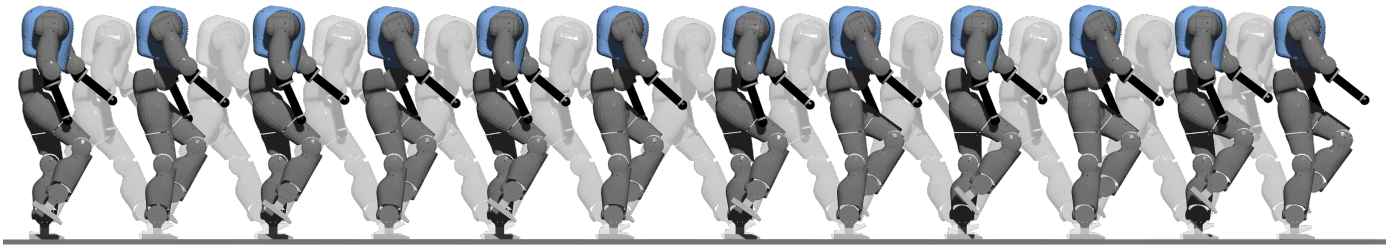


FIGURE 6.7: Transition from Reference 4 to 5, with snapshots in the background taken at each consecutive strike and snapshots in the foreground taken at step height measurement, from $t = 58.5$ $[s]$ to $t = 66$ $[s]$.

Globally, we can conclude that this modulation resulting from the cooptimization of the polynomial coefficients gives far more satisfactory results than the mesh optimization and regression performed in chapter 5. The oscillations are moderate, the offset error small over large part of the modulation space. This modulation is really usable, and predictable, as its performances have been quantified precisely. In the rest of this chapter, the effects of the modulation on the articulations and muscles are analyzed, and compared to human data.

6.3 Analysis of resulting modulations on articular kinematics and dynamics, and muscle activations

Now that the performances of this modulation have been quantified in terms of gait features, it is interesting to analyze the modulation that result from it in terms of joint kinematics and dynamics, as well as muscle activations. In this section, the effects of the modulation of step height and step length are analyzed – separately, for clarity. This help to get an insight into which articulations (and muscles) affect each feature the most.

6.3.1 Analysis of the effects of step height modification

The first analysis is realized by visualizing the kinematics and dynamics of the three sagittal leg joints, for 4 different step height references. For this analysis, the step length reference is always kept at $40\ [cm]$. Figure 6.8 shows the results of this analysis. A first look at the joint kinematics (the three graphs on top) shows clearly that the most affected joint is the knee joint, followed by the hip joint.

The knee joint angle shows a very clear tendency, after toe-off (after 60% of the gait cycle), to *increase flexion* to achieve higher step heights. Between the kinematics resulting from a height reference at $4\ [cm]$ and the kinematics resulting from the a height reference of $11.5\ [cm]$, the maximum knee joint angle is different of more than 30°! Although this modification was already visible in Figure 6.7, this shows that the knee joint angle has a big influence on the step height. Looking at the dynamics of the knee joint, we can again see a different behavior after toe-off. Indeed, we can see the flexion torque becoming an extension torque directly after toe-off for $S_h = 4\ [cm]$ and $S_h = 6.5\ [cm]$ while the torque diminishes more slowly for $S_h = 9\ [cm]$ and $S_h = 11.5\ [cm]$.

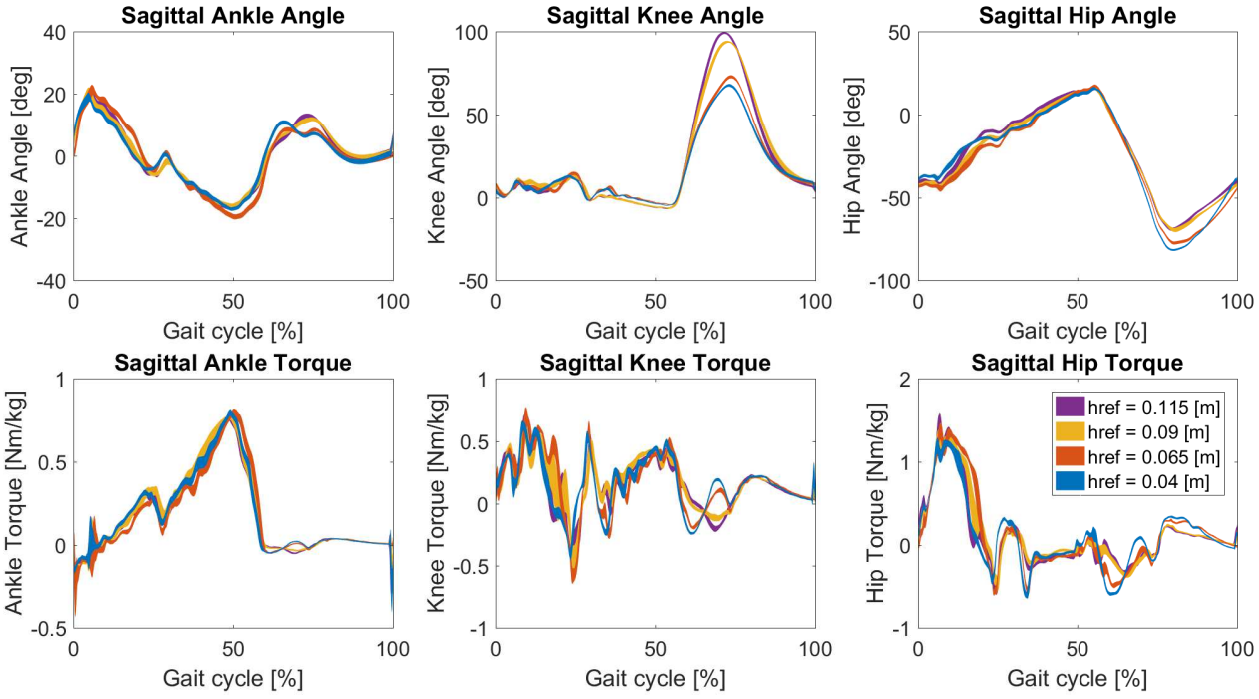


FIGURE 6.8: Kinematic and dynamic effects of changing height reference, with length reference at $40\ [cm]$.

In fact, the sagittal knee torque for the two smallest tested height references is inverted again a bit later in the gait cycle. This can be seen as the knee having to 'slow down' its flexion just after toe-off for small heights, while the flexion has to be encouraged again afterwards to ensure a long enough step length (if the knee joint angle decreases too fast, the foot will strike earlier, resulting in a shorter step length). This may be caused by the moment in the gait cycle when the step height is realized: the step height is measured when the two feet crosses, i.e. quite *early* in the cycle. For this reasons, the knee may be forced to 'brake' early in the cycle, while still applying a flexion torque later on to obtain the needed step length.

Concerning the hip joint kinematics, we can also observe a clear difference – although of less amplitude – on the height reference. This difference comes later in the gait cycle than the main difference observed on the knee joint kinematics, and can help explain the small variations in step length observed for different step height references (see Figure 6.3). Finally, the ankle angle also shows a difference just after toe-off. This difference corresponds to an increased dorsiflexion (flexion towards the dorsum, top of the foot) just after toe-off, for higher step height references. This corresponds to the robot trying to maximize the height of the foot before the step height measurement. After the measurement, the tendency is inversed before the difference is eliminated.

Figure 6.9 shows the muscle activations for the 4 same step height references as in the previous figure, still with $S_l = 40\ [cm]$. Although it is more difficult to distinguish clear tendencies for all muscles, we can still discuss the few muscles that show important differences. The main muscles that seem to be affected by the modification in step height reference are the RF, GLU and TA muscles. The RF and GLU muscles help to explain the difference observed on the knee and hip joint kinematics and dynamics. As the RF muscle is less activated after toe-off for the higher step height references, it applies *less extension torque* on the knee. This is enough to result in the increased knee flexion, as the hip shows increased flexion at that point – the leg is thus pulled to the ground, naturally increasing the flexion.

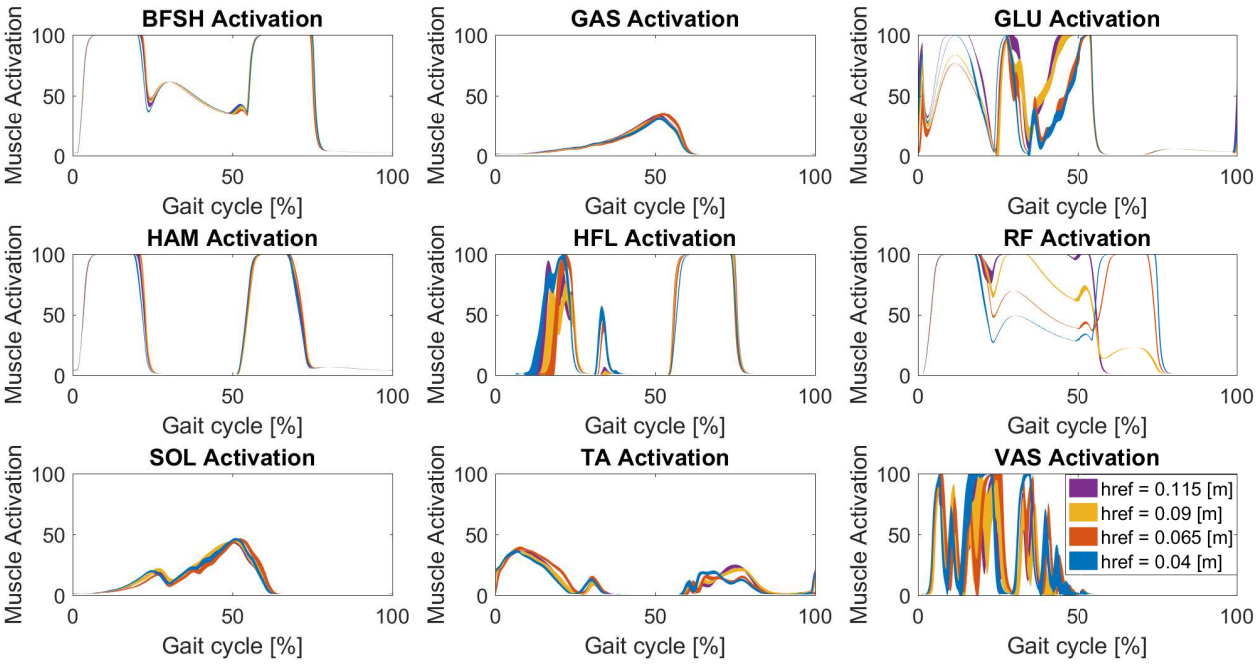


FIGURE 6.9: Effects on muscle activations of changing height reference, with length reference at $40\ [cm]$.

The TA muscle also shows activation profiles in accordance with the ankle kinematics profiles: due to a initially lower activation (after toe-off) for higher step height references, the plantarflexion is favored for these higher step heights, resulting in the anti-phased ankle kinematics between the higher and lower step height references.

6.3.2 Analysis of the effects of step length modification

We now analyze the effects of step length modifications on the articular kinematics and dynamics, as well as on the muscle activations. The articular kinematic and dynamic modifications are compared to human data from [Bov+11]. For this analysis, the step height reference is always kept at $6.5s$ $[cm]$ while the step length references are $S_l = 40$ $[cm]$, $S_l = 30$ $[cm]$ and $S_l = 20$ $[cm]$.

First, we observe Figure 6.10 to see which joints show the biggest variations depending on the length reference. The main kinematic differences are observed on the ankle joint and on the hip joint. For the ankle, we see that longer steps correspond to increased plantarflexion after strike and increased dorsiflexion before toe-off. This is expected, as larger distances between the feet (i.e. longer steps) obviously cause the legs to be more tilted relative to the ground; the ankle hence has more extreme angles at the moments in the gait cycle when the foot is placed flat on the ground. Concerning the hip angle, the same tendency is observed: after strike, longer step lengths correspond to increased flexion. This is justified by the same explanation as for the ankle: the feet being further apart forces the legs to be more tilted. Before toe-off, longer steps cause an increased extension on the hip, although this tendency is less distinct.

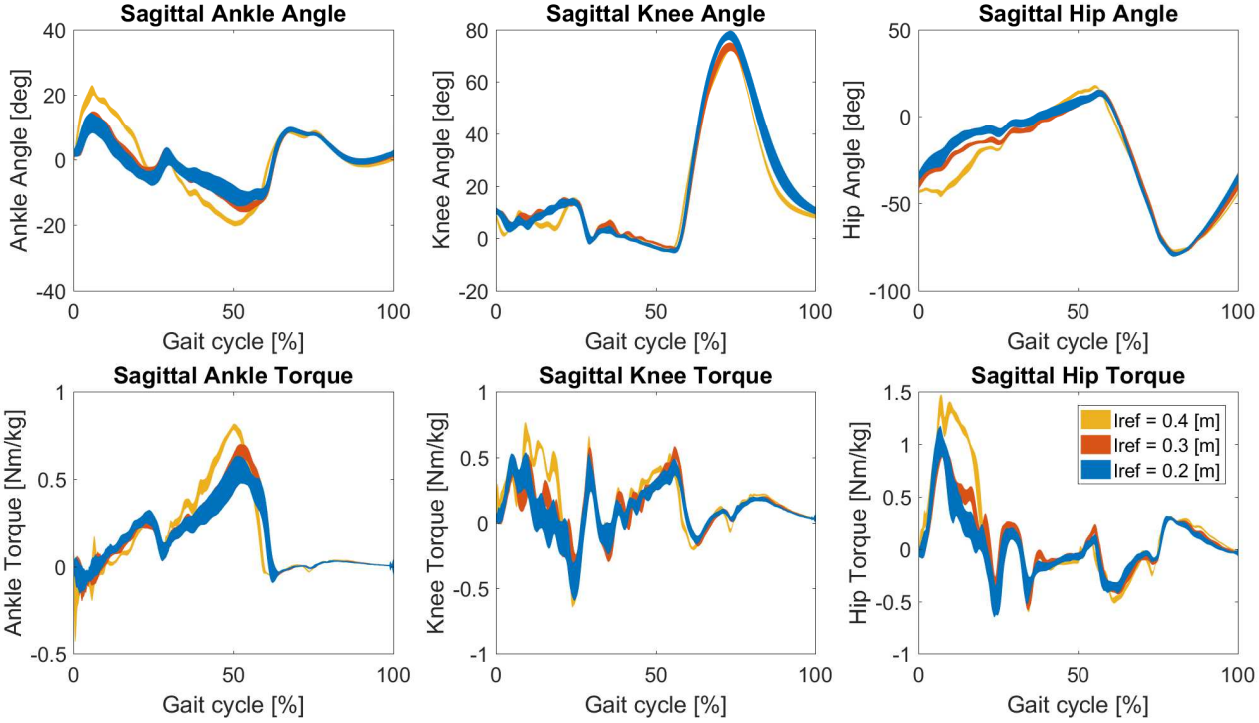


FIGURE 6.10: Kinematic and dynamic effects of changing length reference, with height reference at 6.5 $[cm]$.

In terms of joint dynamics, we can see that the torques tend to compensate for the observed kinematic differences, mainly caused by the environment. Figure 6.11 shows human data for several step sizes. Although some similarities can be observed, the tendencies are not all the same as those noted for the robot. Important to note here is that this human data is obtained *without constraining the step height*, unlike the data measured on the robot. Moreover, the different signals presented in Figure 6.11 were classified according to *walking speed*, and not step length. For the ankle angle

for example, no real difference between step sizes can be observed after strike. Although a difference is observed later during the stance phase, this difference does not correspond to the one observed in the robot. Indeed, in this case, longer steps show increased plantar flexion during late stance phase; this is the opposite as for the robot. During the swing, increased dorsiflexion is clearly visible for larger steps while no difference is noted for the robot.

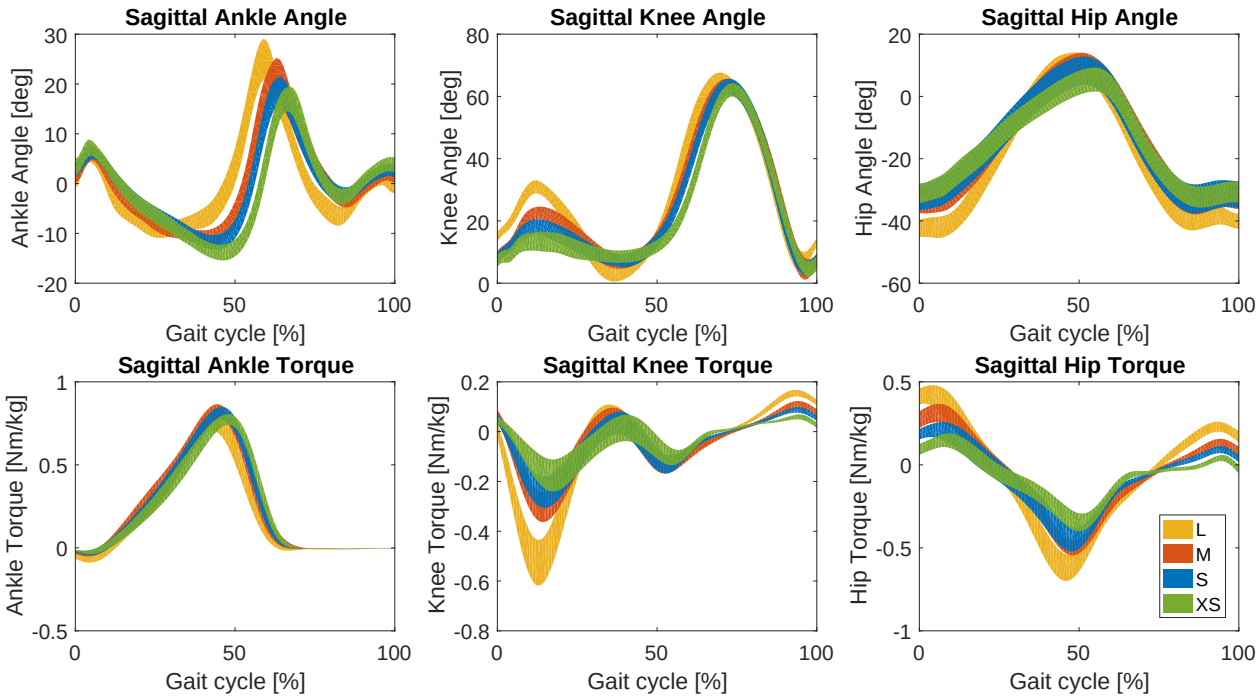


FIGURE 6.11: Human data taken from [Bov+11] of kinematic and dynamic variations for different step sizes. The shaded area represent one half of the standard deviation, for clarity.

Concerning the knee articulation, a clear increase in flexion after strike can be noticed for longer step lengths; again, this is not the case for the robot. This suggests that while the controller developed for the robot prefers to absorb the length difference by increasing the ankle joint angle, humans rather compensate this difference by increasing knee flexion. As for the robot, the human swing phase of the knee shows less differences depending on the step size.

Finally, concerning the hip articulation, the same tendency can be observed between the robot and human data during the whole stance phase. The hip shows an increased flexion angle for increased step sizes. This tendency however is also visible in the human swing phase, which it is not for the robot data.

Having analyzed the kinematic and dynamic effects of the step length adaptations, we can now observe the influence of step length references on the muscle activations in Figure 6.12. Although few muscles show important differences in activation, we can conclude that the RF muscle is the main cause for increased step lengths. During swing, the RF activation profile shows clearly that bigger step lengths correspond to higher RF activation. This increased RF activation resulting in longer step lengths corresponds to the leg being 'pulled' forward to move the next strike further away.

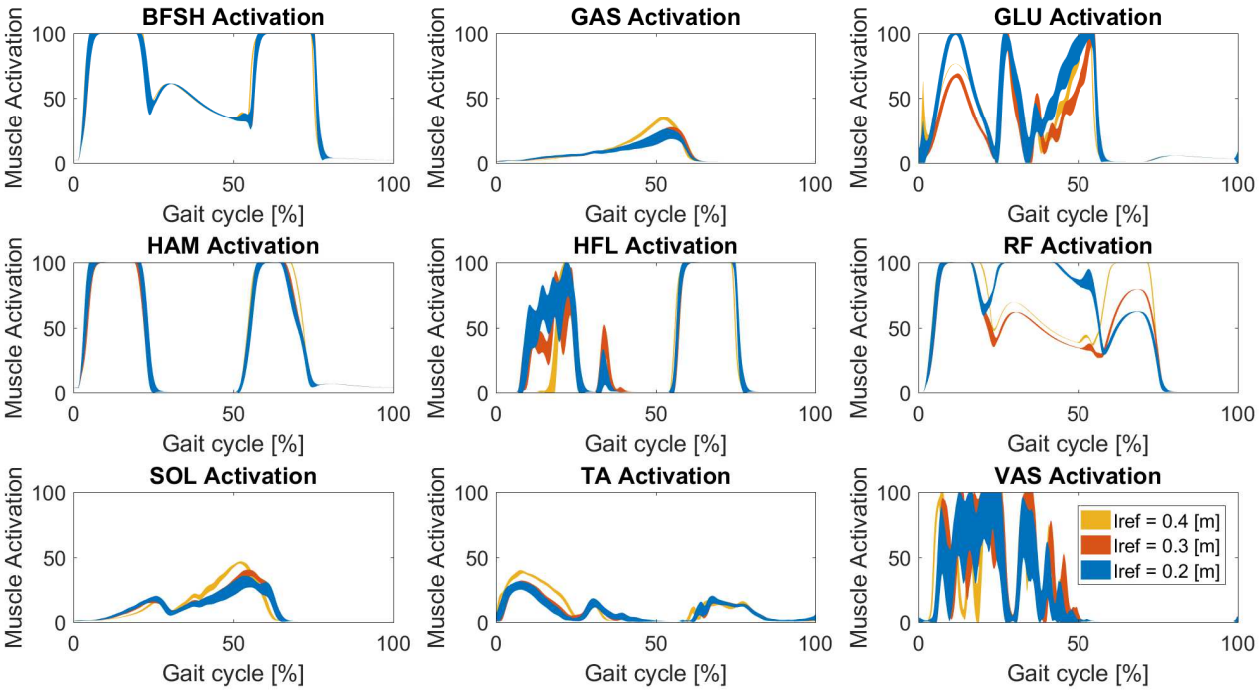


FIGURE 6.12: Effects on muscle activations of changing length reference, with height reference at $6.5\ [cm]$.

6.4 Conclusion

As a conclusion, we can note that this cooptimization method delivered far more satisfactory results than the mesh optimization and regression experiment attempted in chapter 5. The various limitations encountered with that previous method were all either eliminated, or diminished:

- The oscillations are still present – which is logical, as noise is still present – but are of far smaller amplitude;
- The controller is now stable: in the whole modulation space presented in Figure 6.4, the robot walks without falling;
- The maximum step height is not significantly higher, but is more widely attainable (for a larger range of step lengths).

This modulation, although still not perfect by any means, thus allows to control step height and step length in a large modulation space, with predictable results.

Chapter 7

Experiment: Discrete modulations and limits to stability

This last experiment serves a very specific purpose: instead of trying to modulate the steady-state behavior of the walking robot, the objective of this experiment will be to modify its behavior in a discrete manner. The modulation will be applied for a short period of time, in order to avoid an obstacle. The robot should then go back to its steady-state gait modulated by the controller developped in chapter 6.

As presented in chapter 6, the obtained modulations allow to control quite precisely the robot gait, but only in steady-state: the modulations take a few steps to settle. Therefore, this new experiment will try to develop a modulation able to adapt the gait in *less then two steps*. This critical period of two steps is decided in accordance with [MBF15]. According to Matthis & al., the critical period for the visual control of the placement of each step occurs during the preceding step. Indeed, [ZHR15; MBF15] makes the observation that any information available before that does not impact the walking balance and the successful avoidance of an obstacle.

7.1 Discrete modulation procedure and results

The idea of this experiment is to apply discrete modulations to a set of controller parameters, during *one gait cycle* (i.e. two steps). This should be enough to obtain modulations of the gait features sufficient for obstacle avoidance.

Because the objective is to obtain modulations that give the same result starting from anywhere in the modulation space of chapter 6, the idea will be to apply modulations *depending on the current steady-state* references. The modifications applied will hence be relative to the current value of the parameters. This experiment is realized in two steps.

Candidate parameters for modulation First, the parameters to adapt for this discrete modulation must be determined. The initial 'candidate' parameters are the 8 parameters modulated in chapter 6, except for τ, the CPG time constant. For such a short period of time, the CPG would not have the time to adapt. The utility of the CPG will actually be the exact opposite of the modulation: this experiment should take full advantage of the stability of the CPG, so that the gait goes back to its steady-state behavior after the discrete modulation is finished. Some additional parameters will be candidate as well, using the insights gained from the analysis of section 6.3.

One such additional candidate is k_{BFSH1}. Although this parameter did not appear to have an important influence in the sensitivity analysis of chapter 4, it will still be a candidate here for two reasons. As a reminder, the stimulation parameters are given in appendix (section A.5, Equations (A.5)).

- This parameter did not appear as important in the sensitivity analysis probably mainly due to the small force the BFSH muscle can develop relative to other muscles (see Appendix B).
- For this discrete modulation, 'any help' will be useful, and the BFSH muscle acts fully on the knee, which we saw in section 6.3 was crucial in modifying the step height.

Of all the parameters acting on BFSH (see Appendix A), k_{BFSH1} is chosen because it is the one acting at the beginning of the swing phase – when the flexion torque necessary for the step height is applied on the knee.

Another additional candidate is $G_{TA,Sw}$. This parameter is the gain applied on the reflex that acts on the TA muscle during swing. The idea is that by modulating this parameter, additional dorsiflexion could be applied on the foot. This could add the few centimeters height necessary to avoid an obstacle.

Computing the values of the parameters Now that the list of candidate parameters for the modulation is established (Θ, k_{HFL}, k_{GLU1}, k_{HAM1}, k_{HAM2}, k_{RF1}, k_{RF3}, k_{BFSH1} and $G_{TA,Sw}$), the modulations themselves have to be determined. Like stated previously, the modulations will be relative to the current value of each parameter: we will *multiply* each current parameter by a gain to modify in temporarily. The gains are called DIS_{k_i}, where k_i is the name of the parameter they multiply and *DIS* relates to the fact that the resulting modulation is discrete. The resulting new parameters for the time of the discrete modulation were thus computed as in Equation (7.1).

$$k_i \Longleftarrow k_i \cdot DIS_{k_i} \tag{7.1}$$

To determine the value of the gains, a qualitative experimentation is first performed: different values for each gain are tested successively, visualizing the result each time. This allows to get an idea of the necessary amplitude for the gain. In a second step, an optimization on the values of the gains is performed. To run an optimization, a fitness function is necessary. This fitness function is quite straightforward: the *maximum value of step height and step length on the two steps following the application of the discrete modulation*, is maximized.

By running this optimization a first time, a few candidates were removed. The removed candidates were those for which the optimal value of DIS_{k_i} was very close to one, because this corresponds to no modulation at all. The candidates removed at this point were Θ and k_{RF1}. The fact that Θ was not an important parameter to modulate on such a short term is interesting, because it is one of the most sensitive parameters when it comes to steady-state modulations. Concerning k_{RF1} it is less surprising, as this parameters acts for knee extension a the beginning of the swing – acting against step height in fact.

After removing these gains, another optimization was performed, with the boundaries available in appendix, section D.4. Important to note is that this optimization was realized starting always from $\{40, 6.5\}$ as steady-state reference. The final values of the parameters are displayed in Table 7.1.

7.2 Possibilities of the resulting modulation

The modulation gains being computed, we can now compare the gait with and without discrete adaptations. The following experiments were realized *without noise in*

TABLE 7.1: Discrete gains applied on modulated parameters, truncated to 3 digits after the comma.

Gain	Value
$DIS_{k_{BFSH1}}$	2.340
$DIS_{k_{HFL}}$	5.148
$DIS_{k_{HAM1}}$	0.778
$DIS_{k_{HAM2}}$	0.791
$DIS_{k_{GLU1}}$	2.351
$DIS_{k_{RF3}}$	4.008
$DIS_{G_{TA,Sw}}$	2.806

the simulation, to make the experiments reproductible and hence compare adequately the simulations with and without discrete adaptations.

First experiment Figure 7.1 shows a first simulation, with the controller from chapter 6 and $\{40, 6.5\}$ as reference. A hole is added in the ground, that is not visible to the robot until its left foot strikes the ground at the vertical orange line. The hole used for this experiment is 35 $[cm]$ long. Knowing that the feet of the robot are 14 $[cm]$ long, this means that the step length should be at least 49 $[cm]$ to avoid the robot falling into the hole[1]. In this first simulation, the robot does not react to the hole in the ground (the reference is kept at $\{40, 6.5\}$), and the step is consequently *too short* to walk over the hole. As shown[2] by the faded robot in the background, the robot hence falls if no controls are adapted.

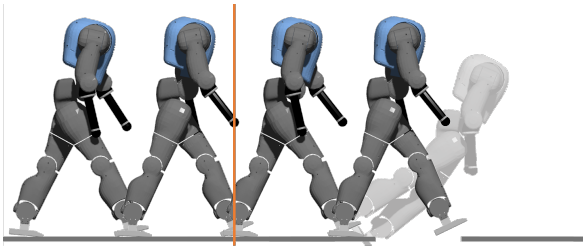


FIGURE 7.1: Without discrete modulation: the robot falls.

Next, the experiment is realized in Figure 7.2. Again, the robot does not see the hole until its left foot strikes at the vertical orange line. Until that point, the simulations realized in Figure 7.1 and Figure 7.2 are thus *exactly the same*, as there is no noise in the simulation. This time, the discrete modulation is activated. This causes the next two steps to have increased step lengths, and the robot is now able to successfully walk over the hole[3].

[1] Indeed, the step length is counted between fixed points on the feet (Figure 2.3, page 9). The distance between the toes of the foot behind and the stub of the foot in front is shorter than the step length, by the length of one foot.

[2] The video, available in appendix, is called *discrete_hole_fall(_slow).webm*.

[3] The video, available in appendix, is called *discrete_hole_avoid(_slow).webm*.

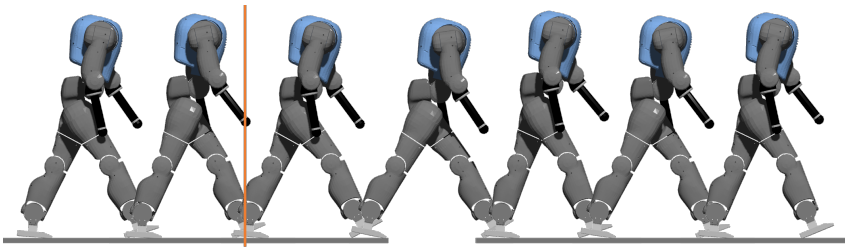


FIGURE 7.2: Discrete modulations activated: the robot avoids the hole.

The measurements corresponding to the experiment described and displayed here, are represented in Figure 7.3. The graph on the left is the one of interest in this case. The discrete modulation is activated at the point where the two step length lines split (and where the blue line hence becomes visible). As we can see, the two steps following the activation of the discrete modulation are longer than all the other steps in the simulation. The longest step, which is the second step of the discrete modulation, is $\approx 52\ [cm]$ long, which is enough to cross the hole.

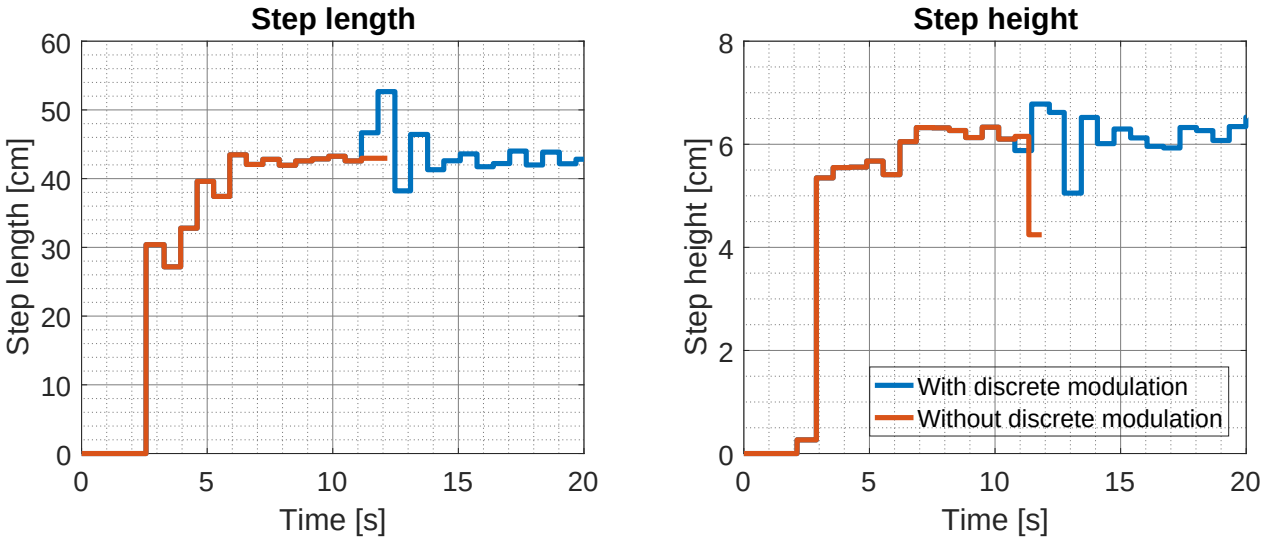


FIGURE 7.3: Effect of discrete modulation on step height and step length. The lines corresponding to the measurements without discrete modulations end when the robot falls.

Second experiment As before, the same situation is simulated twice; once without activating the discrete modulation and once with discrete modulation. Figure 7.4 shows the simulation without activation of the discrete modulation. This time, the obstacle is not a hole but an 18-centimeter-high vertical bar. Keeping in mind that the CoMan is $95\ [cm]$ tall, this would correspond to a $\approx 35\ [cm]$ vertical bar for an adult of about $1.8\ [m]$. In Figure 7.4, we can see that the robot falls because of the obstacle. In the video[4] visible in appendix, it is however visible that the foot actually passes above the obstacle, but collides with it, causing the robot to fall.

[4]The video is called *discrete_obstacle_fall(_slow).webm*.

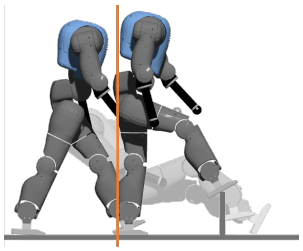


FIGURE 7.4: Without discrete modulation: the robot falls.

Looking now at Figure 7.5, where the discrete modulation is activated, we can see that the robot is able to step above the obstacle and continue its walk normally. Figure 7.3, on the right, shows the adaptation of the step height performed. Although the two steps following the discrete modulation are indeed the highest steps in the whole simulation, the difference in step height is not big. In fact, the real difference is not witnessed on step height like it is measured since the beginning of this work.

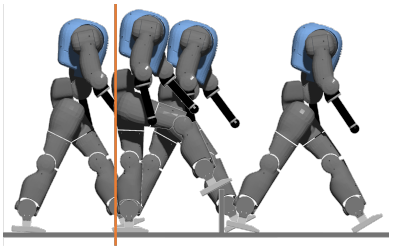


FIGURE 7.5: Discrete modulations activated: the robot avoids the obstacle.

Figure 7.6 shows the foot clearance: the height of the lowest point of the foot, at any time. The step height like it is measured in all the figures up to now, corresponds to a measurement during each of the rising slopes, each corresponding to one step. The blue curve is only very slightly different from the orange one during this rising motion, but shows a much higher maximum ($> 19\ [cm]$ versus $\approx 16\ [cm]$). This is what really helps the robot to avoid the obstacle.

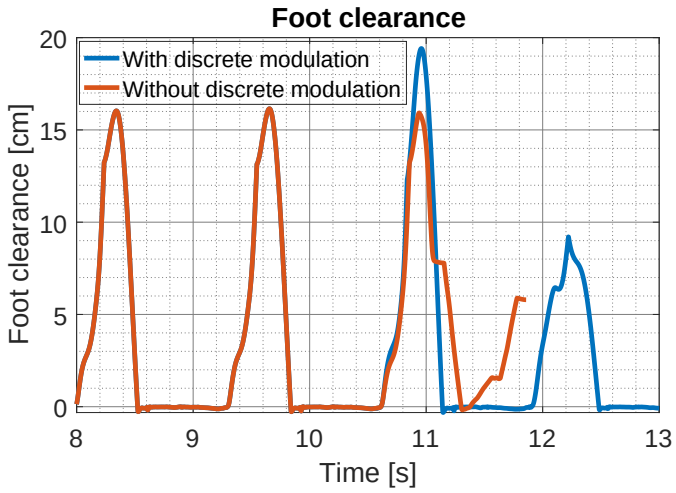


FIGURE 7.6: Effect of discrete modulation on foot clearance. The foot clearance is the height of the lowest point of the foot, here plotted for the right foot of the robot.

7.3 Conclusion and limitations

As a conclusion, we can remember from this experiment that the discrete modulations developed indeed allow us to avoid obstacles, by modifying the gait in a very short time (the next step). This is in accordance with the findings of [ZHR15]. The step length obtained by this technique is noticeably larger than the steady state step length. However, this approach also shows limitations.

Indeed, the obtained step height with this discrete modulation is not remarquably higher than the steady-state step height; the difference in height is much more noticeable when looking at the foot clearance as a continuous signal. Therefore, this discrete modulation could probably be improved by using the maximum foot clearance as a criterion instead of the measured step height.

Chapter 8

Conclusion: take-aways and perspectives

The goal of this chapter is to provide, as a conclusion to this Master Thesis, a summary of the key points the reader should take away from this work. At the same time, the main achievements are underlined and the future research perspectives are discussed. Throughout the 4 different experiments that were conducted, a lot of information about gait modulation can be retrieved.

The title of this work, *Gait modulation of a humanoid robot, using bio-inspired mechanisms*, reminds us of the main objective of this Master Thesis: develop high-level commands allowing for precise control of the robot behavior. In order to fulfill this objective, the sensitivity analysis was realized, giving a good insight into the parameters that would have to be modulated by the high-level commands. Then, the high-level commands were developed first using a mesh optimization and regression procedure, and then enhanced with the cooptimization procedure. A secondary objective of the thesis was to develop commands enabling obstacle avoidance. This, too, was achieved. The discrete modulation obtained at the end of chapter 7 indeed enable obstacle avoidance.

First, in the sensitivity analysis performed in chapter 4, the take-aways are probably the more general ones; they could be applied to other humanoid walkers, or compared to human data. Indeed, the main conclusion from this experiment is that the *trunk angle* and the *time constant* of the oscillator are the most important parameters for this walker. Another important aspect that should be mentioned about the conclusions from chapter 5 is that the most important muscles in the modulation of the gait *all acted on the hip joint*. This tends to show that the modulation of proximal muscles is more important than that of distal muscles. These two elements could also be useful for humans. Indeed, for humans with walking disorders, it could be interesting to analyze the behavior of their trunk angle and the frequency of their steps. Analyzing the activation of the proximal muscles could also be a track.

Concerning the mesh optimization and regression of chapter 5, we remember that the final results were not satisfying. One interesting finding concerned the step height and step length ranges achievable on the CoMan, in steady-state. The optimizations showed that the maximum step height was $12\ [cm]$ and the maximum step length about $45\ [cm]$. Step lengths lower than $20\ [cm]$ were not stable (the robot's foot would

hit the ground mid-swing and make the robot fall), and step heights below $4\ [cm]$ neither (for the same reason). The biggest problem with the modulation obtained in this experiment was its *low reliability*: the gait features showed wide oscillations, and the robot would often fall when the references were close to the range limits. As a result, the modulation obtained by regression can't be used for any further developments. The individual optimization results, in contrast, could be analyzed to obtain a more in-depth look into the importance of different parameters on the gait features. Using these individual results, a more precise version of the analysis realized in chapter 4 could be performed, for the parameters modulated here.

The new version of the modulation, obtained by cooptimization in chapter 6, showed more interesting results. Indeed, the boundaries determined in chapter 5 are now attainable by the controller, without any fall of the robot. The controller is thus able to modulate the gait throughout the modulation space of $[4;\ 12]\ [cm]$ for step height and $[20;\ 50]\ [cm]$ for step length, with bigger errors only near the boundaries. Considering the size of the CoMan, it is not realistic to aim for wider ranges for these features. An interesting perspective regarding this result could be the analysis (but not the control) of the *walking speed* obtained throughout the modulation space. Indeed, the two features currently controlled do not enable any explicit control of the walking speed. Modulating the gait period as third gait feature would allow the control of the walking speed (as step length and step period together determine the speed). This would however be difficult. Indeed, we can see that the two features modulated here already show a certain degree of interdependence. Modulating a third feature while hoping to obtain an independent modulation of the three features will hence give very limited results, at best. Another future perspective regarding this modulation would be to *reduce even more the amplitude of the oscillations*. Oscillations of very low amplitude would enable a new range of possibilities: the precise foot landing locations could be scheduled with confidence.

Finally, the last experiment, in chapter 7, explored the possibilities of discrete modulations. The results of this experiment were mixed. Although the step heights and step lengths obtained were each larger than any other step length or step height obtained with the same references, the differences were small – especially for the step height. An important perspective to explore in this regard is the use of a new metric: by using the *maximum foot clearance* as an optimization criterion, instead of the step height as defined in this work, the obtained results would likely be more useful for obstacle avoidance. Indeed, as obstacles are normally located in front of the walker (and not next to its opposite foot), the step height as computed in this work is less important than the foot clearance further in front of the robot.

From this work, we can see that humanoid robots could eventually achieve efficient human-like gaits, managing precise control of the foot placements at the same time. These achievements would allow the robots to navigate efficiently through complex environments, without assistance. Progressively, humanoid walkers might close the gap with humans and hence be able to achieve more helpful and necessary tasks, taking burdens and dangers away from humans.

Appendices

Appendix A

Equations of the controller

This appendix presents all the equations that are used in the controller, but are not strictly necessary for the understanding of the master thesis.

A.1 Neuron firing rates

$$
\begin{aligned}
\dot{x_1} &= \frac{1}{\tau}(-x_1 - \beta_a \nu_1 \ - \eta_a [x_2]^+ - \eta_f [x_3]^+ - \eta_g [x_4]^+ + u_1) \\
\dot{x_2} &= \frac{1}{\tau}(-x_2 - \beta_a \nu_2 \ - \eta_a [x_1]^+ - \eta_g [x_3]^+ - \eta_f [x_4]^+ + u_2) \\
\dot{x_3} &= \frac{1}{\tau}(-x_3 - \beta_b \nu_3 \ - \eta_f [x_1]^+ - \eta_g [x_2]^+ - \eta_b [x_4]^+ + u_3) \\
\dot{x_4} &= \frac{1}{\tau}(-x_4 - \beta_b \nu_4 \ - \eta_g [x_1]^+ - \eta_f [x_2]^+ - \eta_b [x_3]^+ + u_4) \\
\dot{x_A} &= \frac{1}{\tau}(-x_A - \beta_a \nu_A - \eta_f [x_3]^+ - \eta_g [x_4]^+ - \eta_a [x_B]^+ + u_A) \\
\dot{x_B} &= \frac{1}{\tau}(-x_B - \beta_a \nu_B - \eta_g [x_3]^+ - \eta_f [x_4]^+ - \eta_a [x_A]^+ + u_B) \\
\dot{x_C} &= \frac{1}{\tau}(-x_C - \beta_c \nu_C - \eta_h [x_3]^+ - \eta_i [x_4]^+ - \eta_c [x_D]^+ + u_C) \\
\dot{x_D} &= \frac{1}{\tau}(-x_D - \beta_c \nu_D - \eta_i [x_3]^+ - \eta_h [x_4]^+ - \eta_c [x_C]^+ + u_D) \\
\dot{x_E} &= \frac{1}{\tau}(-x_E - \beta_d \nu_E - \eta_j [x_1]^+ - \eta_k [x_2]^+ - \eta_d [x_F]^+ + u_E) \\
\dot{x_F} &= \frac{1}{\tau}(-x_F - \beta_d \nu_F - \eta_k [x_1]^+ - \eta_j [x_2]^+ - \eta_d [x_E]^+ + u_F)
\end{aligned}
\tag{A.1}
$$

A.2 Neuron self-inhibitions

$$
\begin{aligned}
\dot{\nu_1} &= \frac{1}{\gamma_A \tau}(-\nu_1 + [x_1]^+) & \dot{\nu_B} &= \frac{1}{\gamma_A \tau}(-\nu_B + [x_B]^+) \\
\dot{\nu_2} &= \frac{1}{\gamma_A \tau}(-\nu_2 + [x_2]^+) & \dot{\nu_C} &= \frac{1}{\gamma_C \tau}(-\nu_C + [x_C]^+) \\
\dot{\nu_3} &= \frac{1}{\gamma_B \tau}(-\nu_3 + [x_3]^+) & \dot{\nu_D} &= \frac{1}{\gamma_C \tau}(-\nu_D + [x_D]^+) \\
\dot{\nu_4} &= \frac{1}{\gamma_B \tau}(-\nu_4 + [x_4]^+) & \dot{\nu_E} &= \frac{1}{\gamma_D \tau}(-\nu_E + [x_E]^+) \\
\dot{\nu_A} &= \frac{1}{\gamma_A \tau}(-\nu_A + [x_A]^+) & \dot{\nu_F} &= \frac{1}{\gamma_D \tau}(-\nu_F + [x_F]^+)
\end{aligned}
\tag{A.2}
$$

A.3 CPG outputs for chapter 4 and chapter 5

The following outputs are used in chapters 4 and 5, coming from the previous version of the controller ([NRI15]).

$$\begin{aligned} y_1 &= [x_C]^+ - [x_3]^+ & y_4 &= [x_B]^+ - [x_4]^+ \\ y_2 &= [x_D]^+ - [x_4]^+ & y_5 &= [x_E]^+ - [x_1]^+ \\ y_3 &= [x_A]^+ - [x_3]^+ & y_6 &= [x_F]^+ - [x_2]^+ \end{aligned} \tag{A.3}$$

A.4 Muscle stimulations for chapter 4 and chapter 5

This first set of muscle stimulations is used in chapters 4 and 5.

$$\begin{aligned} S_{HFL,R} &= k_{HFL} \cdot [y_2]^+ \\ S_{GLU,R} &= k_{GLU1} \cdot [y_3]^+ + k_{GLU2} \cdot [y_5]^+ \\ S_{HAM,R} &= k_{HAM1} \cdot [y_3]^+ + k_{HAM2} \cdot [y_4]^+ + k_{HAM3} \cdot [y_5]^+ \\ S_{BFSH,R} &= k_{BFSH1} \cdot [y_2]^+ + k_{BFSH2} \cdot [y_5]^+ \\ S_{RF,R} &= k_{RF1} \cdot [y_2]^- + k_{RF2} \cdot [y_5]^+ \\ \\ S_{HFL,L} &= k_{HFL} \cdot [y_1]^+ \\ S_{GLU,L} &= k_{GLU1} \cdot [y_4]^+ + k_{GLU2} \cdot [y_6]^+ \\ S_{HAM,L} &= k_{HAM1} \cdot [y_4]^+ + k_{HAM2} \cdot [y_3]^+ + k_{HAM3} \cdot [y_6]^+ \\ S_{BFSH,L} &= k_{BFSH1} \cdot [y_1]^+ + k_{BFSH2} \cdot [y_6]^+ \\ S_{RF,L} &= k_{RF1} \cdot [y_1]^- + k_{RF2} \cdot [y_6]^+ \end{aligned} \tag{A.4}$$

A.5 Muscle stimulations for chapter 3, chapter 6 and chapter 7

This second set of muscle stimulations is used in chapters 3, 6 and 7. It was obtained by modifying the stimulations sent to the BFSH and RF muscle groups, to obtain stimulation more similar to the reflexes presented in [SG15a].

$$\begin{aligned} S_{HFL,R} &= k_{HFL} \cdot [y_2]^+ \\ S_{GLU,R} &= k_{GLU1} \cdot [y_3]^+ + k_{GLU2} \cdot [y_5]^+ \\ S_{HAM,R} &= k_{HAM1} \cdot [y_3]^+ + k_{HAM2} \cdot [y_4]^+ + k_{HAM3} \cdot [y_5]^+ \\ S_{BFSH,R} &= k_{BFSH1} \cdot [y_2]^+ + k_{BFSH2} \cdot [y_5]^+ + k_{BFSH3} \cdot [y_3]^+ + k_{BFSH4} \cdot [y_6]^+ \\ S_{RF,R} &= k_{RF1} \cdot [y_2]^+ + k_{RF2} \cdot [y_5]^+ + k_{RF3} \cdot [y_6]^+ \\ \\ S_{HFL,L} &= k_{HFL} \cdot [y_1]^+ \\ S_{GLU,L} &= k_{GLU1} \cdot [y_4]^+ + k_{GLU2} \cdot [y_6]^+ \\ S_{HAM,L} &= k_{HAM1} \cdot [y_4]^+ + k_{HAM2} \cdot [y_3]^+ + k_{HAM3} \cdot [y_6]^+ \\ S_{BFSH,L} &= k_{BFSH1} \cdot [y_1]^+ + k_{BFSH2} \cdot [y_6]^+ + k_{BFSH3} \cdot [y_4]^+ + k_{BFSH4} \cdot [y_5]^+ \\ S_{RF,L} &= k_{RF1} \cdot [y_1]^+ + k_{RF2} \cdot [y_6]^+ + k_{RF3} \cdot [y_5]^+ \end{aligned} \tag{A.5}$$

Appendix B

Hill muscles characterisation

This document, written by Nicolas Van der Noot, is reproduced – and adapted – here because it has not been published.

B.1 Goal

We want to implement the muscles used in [GH10; SG15a] (partially described in [GSB03]), adapted to the COMAN robot's size and weight.

To this end, we are interested in the following muscles characteristics:

- F_{max}: maximum isometric force [N]
- v_{max}: maximum shortening velocity [l_{opt}/s]
- l_{opt}: optimal fiber length [cm]
- l_{slack}: tendon slack length [cm]
- r_0: lever arm constant [cm]
- φ_{max}: joint [deg] producing the maximum lever arm
- φ_{ref}: joint [deg] at which $l_{mtu} = l_{opt} + l_{slack}$
- ρ: pennation factor[-]
- m: mass [kg]
- λ: the fraction of Type I (slow twitch) fibers [$-$]

The last two parameters are only useful to compute the metabolic energy as in [BPA04].

B.2 Muscles

For all muscles in the sagittal plane (SOL, TA, GAS, VAS, HAM, GLU, HFL, BFSH, RF), we get the characteristics from [GH10; SG15a]. Regarding m and λ, we get the data from [Wan+12].

B.3 Results for adult-like model

These are the results for an adult-like model of 1.8 m and 80 kg, as the one described in [GH10]. Fortunately, this is approximately the same model as the one from [Arn+10] (1.68 m for 82.7kg). The angles are given relative to frames attached to the different bodies. In home position (upright standing), these frames are all oriented in the same way: $\hat{X}$ in forward direction, $\hat{Y}$ on the left and $\hat{Z}$ positively aligned with the gravitational axis. For non-sagittal muscles, the angles are given for the right leg.

	SOL	**TA**	**GAS**	**VAS**	**HAM**	**GLU**	**HFL**	**BFSH**	**RF**
F_{max} **[N]**	4000	800	1500	6000	3000	1500	2000	350	1200
v_{max} $[l_{opt}/s]$	6	12	12	12	12	12	12	12	12
l_{opt} **[cm]**	4	6	5	8	10	11	11	12	8
l_{slack} **[cm]**	26	24	40	23	31	13	10	10	35
r_o **[cm]**	5	4	5 (a) 5 (k)	6	5 (k) 8 (h)	10	10	4	6
φ_{max} **[deg]**	20	-10	20(a) 40 (k)	15	0 (k) - (h)	-	-	-	15
φ_{ref} **[deg]**	-10	20	-10 (a) 15 (k)	55	0 (k) -25 (h)	-30	0	20	55
ρ **[-]**	0.5	0.7	0.7 (a) 0.7 (k)	0.7	0.7 (k) 0.7 (h)	0.5	0.5	0.7	0.5 (k) 0.3 (h)
m [kg]	0.68	0.2	0.32	2.04	1.27	0.7	0.93	0.085	0.34
λ **[-]**	0.81	0.7	0.54	0.5	0.44	0.5	0.5	0.67	0.423

B.4 Dynamic scaling

Dynamic scaling is presented in [BM06] and [STG12]. We present briefly the equations needed for our purpose: scaling the muscles properties for the size and the mass of the COMAN.

Suppose we have a body 1 with length L_1 and mass M_1 for which we know all the muscles characteristics. Now, we want to scale these characteristics for a body 2 having a length L_2 and a mass M_2.

We express the time references for theses two bodies as T_1 and T_2 (the two bodies can indeed have different time references, which might seem counter-intuitive at the beginning).

We will now try to express all the relations using only the international basis units: m, kg, s (A is not used as we do not play with current), so corresponding to L_i, M_i and T_i.

Gravity must be the same for the two bodies (G_1 and G_2):

$$\frac{G_2}{G_1} = 1 = \frac{L_2/T_2^2}{L_1/T_1^2} = \frac{L_2}{L_1} \cdot (\frac{T_1}{T_2})^2 \rightarrow \frac{T_2}{T_1} = \sqrt{\frac{L_2}{L_1}}$$

We can now compute the forces ratio:

$$\frac{F_2}{F_1} = \frac{M_2 \cdot (L_2/T_2^2)}{M_1 \cdot (L_1/T_1^2)} = \frac{M_2}{M_1}$$

We compute then the torques ratio:

$$\frac{\tau_2}{\tau_1} = \frac{L_2 \cdot F_2}{L_1 \cdot F_1} = \frac{L_2}{L_1} \cdot \frac{M_2}{M_1}$$

We then compute the velocity ratio:

$$\frac{V_2}{V_1} = \frac{L_2/T_2}{L_1/T_1} = \frac{L_2}{L_1} \cdot \frac{T_1}{T_2} = \frac{L_2}{L_1} \cdot \sqrt{\frac{L_1}{L_2}} = \sqrt{\frac{L_2}{L_1}}$$

Finally, we compute the velocity ratio normalized by the length:

$$\frac{v_2}{v_1} = \frac{1/T_2}{1/T_1} = \frac{T_1}{T_2} = \sqrt{\frac{L_1}{L_2}}$$

To get the length ratio, we compare the leg length of the human model from [GH10] (L_H) with the one of the COMAN (L_C). To get the mass ratio, we compare the total masses of these two models (M_H compared to M_C).

$$\frac{L_C}{L_H} = \frac{42.7\,cm}{100\,cm} = 0.427 \qquad \frac{M_C}{M_H} = \frac{28.3\,kg}{80\,kg} \approx 0.354$$

Using these two ratios, we get:

$$\frac{F_C}{F_H} = \frac{M_C}{M_H} = 0.354$$

$$\frac{\tau_C}{\tau_H} = \frac{L_C}{L_H} \cdot \frac{M_C}{M_H} = 0.427 \cdot 0.354 \approx 0.151$$

$$\frac{V_C}{V_H} = \sqrt{\frac{L_C}{L_H}} = \sqrt{0.427} \approx 0.653$$

$$\frac{v_C}{v_H} = \sqrt{\frac{L_H}{L_C}} = \frac{1}{\sqrt{0.427}} \approx 1.53$$

B.5 Results for the COMAN

In section B.4, we computed the following ratios between COMAN (C) and the human model (H):

$$\frac{L_C}{L_H} = 0.427 \qquad \frac{M_C}{M_H} = 0.354 \qquad \frac{F_C}{F_H} = 0.354 \qquad \frac{\tau_C}{\tau_H} = 0.151 \qquad \frac{V_C}{V_H} = 0.653 \qquad \frac{v_C}{v_H} = 1.53$$

We now apply them on the muscles characteristics:

	SOL	**TA**	**GAS**	**VAS**	**HAM**	**GLU**	**HFL**	**BFSH**	**RF**
F_{max} **[N]**	1416	283.2	531	2124	1062	531	708	124	425
v_{max} **[**l_{opt}/s**]**	9.18	18.36	18.36	18.36	18.36	18.36	18.36	18.36	
l_{opt} **[cm]**	1.708	2.562	2.135	3.416	4.27	4.697	5.1	3.4	
l_{slack} **[cm]**	11.102	10.248	17.08	9.821	13.237	5.551	4.3	14.9	
r_o **[cm]**	2.135	1.708	2.135 (a) 2.135 (k)	2.562	2.135 (k) 3.416 (h)	4.27	1.708	2.562	
φ_{max} **[deg]**	20	-10	20(a) 40 (k)	15	0 (k) - (h)	-	-	-	15
φ_{ref} **[deg]**	-10	20	-10 (a) 15 (k)	55	0 (k) -25 (h)	-30	0	20	55
ρ **[-]**	0.5	0.7	0.7 (a) 0.7 (k)	0.7	0.7 (k) 0.7 (h)	0.5	0.5	0.7	0.5 (k) 0.3 (h)
m [kg]	0.241	0.071	0.113	0.722	0.45	0.248	0.329	0.03	0.12
λ **[-]**	0.81	0.7	0.54	0.5	0.44	0.5	0.5	0.67	0.423

Of course, these values can be rounded. Indeed, they are based on rounded averages of different muscles, which were evaluated from different people. There is no single possibility, changing a bit the muscles properties (in a realistic range) is like giving the muscles from someone else.

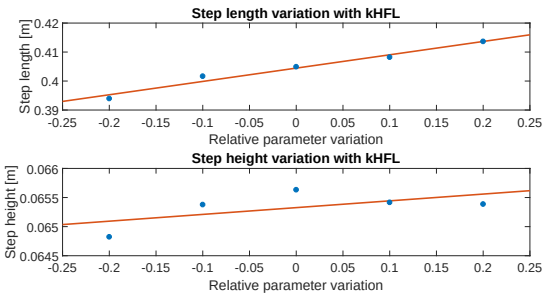
Appendix C

Results of sensitivity analysis

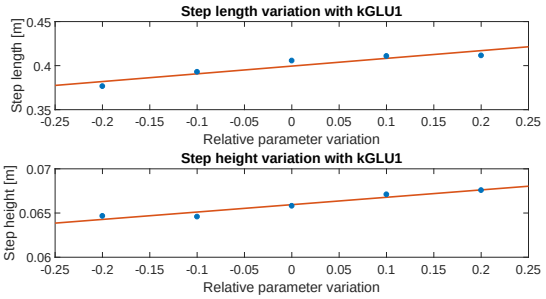
The absolute slope coefficients (i.e. not normalized by the coefficient found for Θ) are given below, in Table C.1. Representations of the linear fits are given for all parameters in Figure C.1 and Figure C.2.

TABLE C.1: Absolute slope coefficients found by sensitivity analysis in chapter 4 (coefficients shown are multiplied by 10^3).

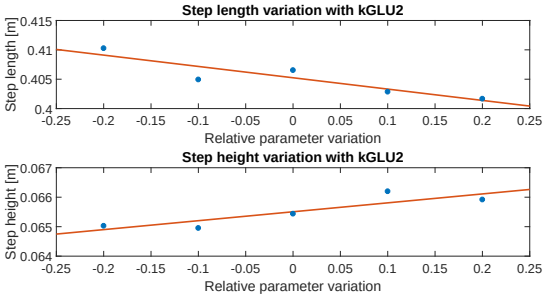
Coefficient	k_{HFL}	k_{GLU1}	k_{GLU2}	k_{HAM1}	k_{HAM2}	k_{HAM3}
Length	45,955	87,742	-19,284	9,6639	-53,476	-13,340
Height	1,1646	8,3462	3,0275	6,1665	9,1388	3,7157
	k_{BFSH1}	k_{BFSH2}	k_{RF1}	k_{RF2}	Θ	τ
Length	2,3980	-5,9990	-0,15400	-4,4460	813,60	48,695
Height	-0,0032000	2,4060	-1,7753	-1,7277	122,92	108,81



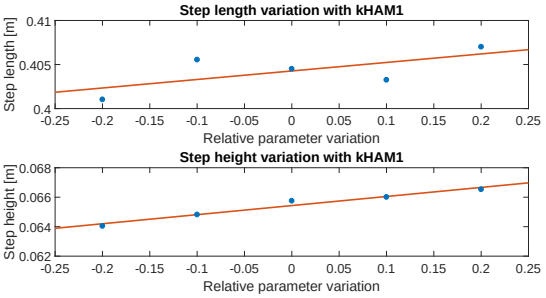
(A) Sensitivity analysis for parameter k_{HFL}.



(B) Sensitivity analysis for parameter k_{GLU1}.

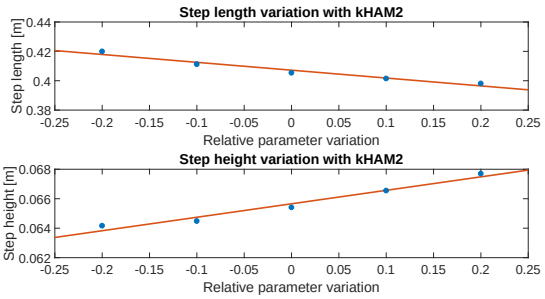


(C) Sensitivity analysis for parameter k_{GLU2}.

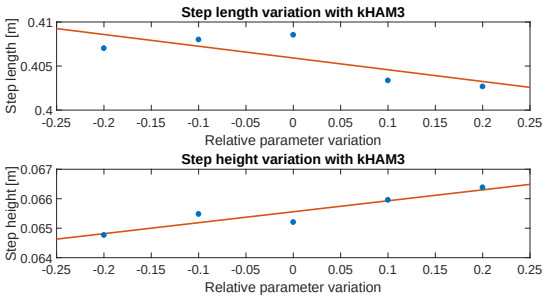


(D) Sensitivity analysis for parameter k_{HAM1}.

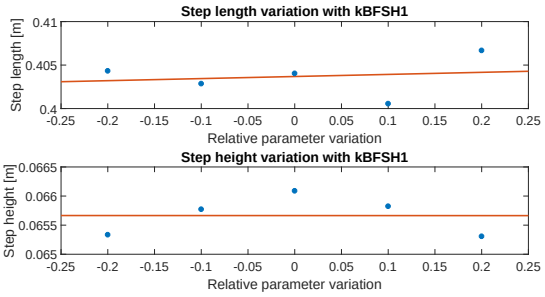
FIGURE C.1: Continuous lines represent the least-squares fit, blue dots the means of the different measurements.



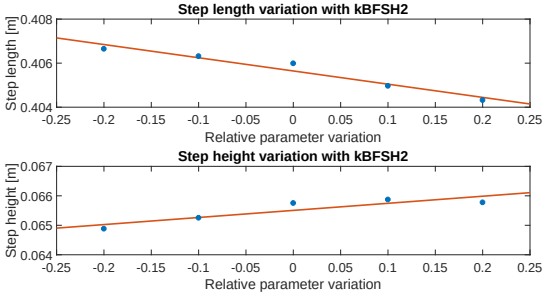
(A) Sensitivity analysis for parameter k_{HAM2}.



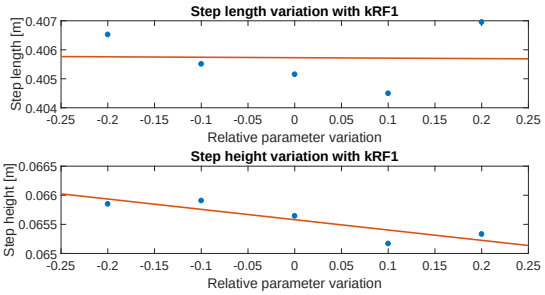
(B) Sensitivity analysis for parameter k_{HAM3}.



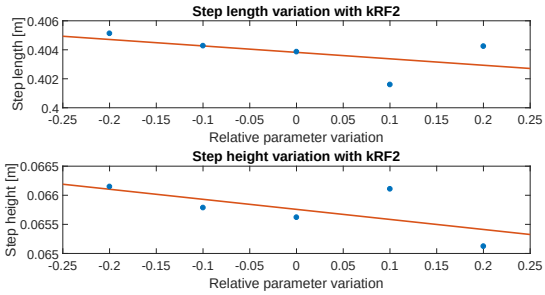
(C) Sensitivity analysis for parameter k_{BFSH1}.



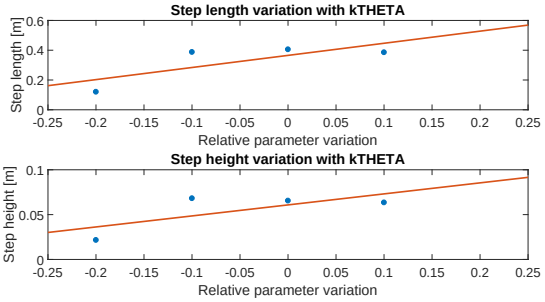
(D) Sensitivity analysis for parameter k_{BFSH2}.



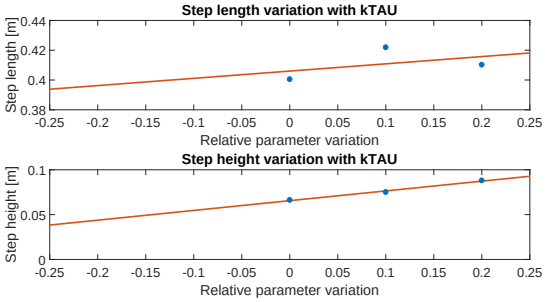
(E) Sensitivity analysis for parameter k_{RF1}.



(F) Sensitivity analysis for parameter k_{RF2}.



(G) Sensitivity analysis for parameter Θ.



(H) Sensitivity analysis for parameter τ.

FIGURE C.2: Continuous lines represent the least-squares fit, blue dots the means of the different measurements.

Appendix D

Bounds for the optimization parameters

D.1 Reference gait

TABLE D.1: Open parameters and their bounds for the optimization of the reference gait.

Reflex rules	min	max	**CPG outputs**	min	max
GSol	0.6	0.9	Θ	0.05	0.3
GSolTa	0.3	1.0	k_{HFL}	5.0	8.0
GTaSw	1.5	4.0	k_{HAM1}	3.0	5.5
GTaSt	1.5	2.5	k_{HAM2}	2.0	3.0
GGas	0.4	0.8	k_{HAM3}	0.0	0.14
GVas	35.0	45.0	k_{GLU1}	2.0	4.5
LTaSw	0.8	0.9	k_{GLU2}	0.0	0.3
LTaSt	0.6	0.75	k_{BFSH1}	4.0	12.0
PhiKT	2.8	3.14	k_{BFSH2}	0.0	0.4
KpTheta	8.0	12.0	k_{RF1}	0.0	2.0
KdTheta	0.2	0.8	k_{RF2}	0.0	8.0
CPG equations	min	max	**CPG equations**	min	max
η_1	4.5	6.0	β_1	4.0	6.0
η_2	4.5	7.0	β_2	3.0	4.5
η_3	2.5	4.5	β_3	2.0	5.0
η_4	4.0	6.5	β_4	3.0	4.5
η_5	2.0	3.0	Γ_1	4.0	6.0
η_6	3.0	4.0	Γ_2	2.0	3.5
η_7	4.0	5.0	Γ_3	1.0	5.5
η_8	3.7	4.8	Γ_4	2.5	3.5
η_9	2.5	3.5	τ	0.082	0.102
η_{10}	1.0	4.0			
Initialization	min	max			
XRef	0.03	0.07			

D.2 Mesh Optimization

TABLE D.2: Remaining open parameters and their bounds for the mesh optimization.

CPG outputs	min	max	**CPG equations**	min	max
k_{HFL}	1.0	10.0	τ	0.06	0.175
k_{GLU1}	0.2	8.0			
k_{HAM1}	1	9			
k_{HAM2}	0.5	5.5			
Θ	0.05	0.3			

D.3 Cooptimization

TABLE D.3: Open parameters and their bounds for the cooptimization.

Coefficient	k_{GLU1}		k_{HFL}	
	min	max	min	max
K	1.3	2.1	4.2	5.0
L10	6.5	8.5	10.0	13.0
L01	-8.5	-6.4	45.0	55.0
L11	95.0	115.0	160.0	210.0
M20	46.0	57.0	75.0	85.0
M02	540.0	600.0	-280.0	-240.0
	k_{HAM1}		k_{HAM2}	
	min	max	min	max
K	6.9	7.6	3.2	4.2
L10	10.0	15.0	-4.8	-2.7
L01	18.0	30.0	-10.0	-7.0
L11	-225.0	-175.0	-55.0	-40.0
M20	90.0	100.0	-21.5	-17.5
M02	-1140.0	-1020.0	8.0	10.0
	k_{RF1}		k_{RF3}	
	min	max	min	max
K	1.8	2.8	1.7	2.7
L10	10.0	14.5	14.0	19.0
L01	-110.0	-85.0	30.0	40.0
L11	-400.0	-350.0	110.0	140.0
M20	30.0	36.5	125.0	155.0
M02	850.0	950.0	390.0	460.0
	Θ		τ	
	min	max	min	max
K	0.12	0.132	0.095	0.125
L10	-1.2	-0.9	-0.05	0.15
L01	-21.0	-16.0	-0.19	-0.13
L11	0.24	0.38	0.26	0.31
M20	-0.42	-0.34	0.2	0.26
M02	-4.1	-3.5	3.9	4.9

D.4 Discrete modulation

TABLE D.4: Open parameters and their bounds for the discrete modulation.

Gain	min	max
$DIS_{k_{BFSH1}}$	1.0	5.0
$DIS_{k_{HFL}}$	3.0	6.0
$DIS_{k_{HAM1}}$	0.0	1.5
$DIS_{k_{HAM2}}$	0.0	1.5
$DIS_{k_{GLU1}}$	1.0	3.5
$DIS_{k_{RF3}}$	2.0	5.0
$DIS_{G_{TA,Sw}}$	0.0	4.0

Bibliography

[Arn+10] Edith M. Arnold et al. "A Model of the Lower Limb for Analysis of Human Movement". In: *Annals of biomedical engineering* 38.2 (Feb. 2010), pp. 269–279. ISSN: 0090-6964.

[BA08] Daryaneh Badaly and Karen E. Adolph. "Beyond the average: Walking infants take steps longer than their leg length". In: *Infant behavior & development* 31.3 (Sept. 2008), pp. 554–558. ISSN: 0163-6383.

[BM06] Adrian Bejan and James H. Marden. "Unifying constructal theory for scale effects in running, swimming and flying". In: *Journal of Experimental Biology* 209.2 (Jan. 15, 2006), pp. 238–248. ISSN: 0022-0949, 1477-9145.

[BPA04] Lindsay J. Bhargava, Marcus G. Pandy, and Frank C. Anderson. "A phenomenological model for estimating metabolic energy consumption in muscle contraction". In: *Journal of Biomechanics* 37.1 (Jan. 2004), pp. 81–88. ISSN: 0021-9290.

[Bov+11] Gabriele Bovi et al. "A multiple-task gait analysis approach: Kinematic, kinetic and EMG reference data for healthy young and adult subjects". In: *Gait & Posture* 33.1 (Jan. 1, 2011), pp. 6–13. ISSN: 0966-6362, 1879-2219.

[Bru04] G. Brunnett, ed. *Geometric modeling for scientific visualization*. Mathematics and visualization. Berlin ; New York: Springer, 2004. 488 pp. ISBN: 978-3-540-40116-2.

[Cec11] Marco Ceccarelli. *Technology Developments: the Role of Mechanism and Machine Science and IFToMM*. p.63. Springer Science & Business Media, May 26, 2011. 474 pp. ISBN: 978-94-007-1300-0.

[DKI14] Florin Dzeladini, Jesse van den Kieboom, and Auke Ijspeert. "The contribution of a central pattern generator in a reflex-based neuromuscular model". In: *Frontiers in Human Neuroscience* 8 (June 26, 2014). ISSN: 1662-5161.

[Dad+13] Farzin Dadashi et al. "Gait and Foot Clearance Parameters Obtained Using Shoe-Worn Inertial Sensors in a Large-Population Sample of Older Adults". In: *Sensors (Basel, Switzerland)* 14.1 (Dec. 27, 2013), pp. 443–457. ISSN: 1424-8220.

[Dal11] Houman Dallali. "Modelling and dynamic stabilisation of a compliant humanoid robot, CoMan". Thesis. 2011.

[Del80] F. Delcomyn. "Neural basis of rhythmic behavior in animals". In: *Science* 210.4469 (Oct. 31, 1980), pp. 492–498. ISSN: 0036-8075, 1095-9203.

[Fal13] Jason Falconer. *This Humanoid Robot Gets Pushed Around But Stays on Its Feet*. IEEE Spectrum: Technology, Engineering, and Science News. Apr. 4, 2013. URL: http://spectrum.ieee.org/automaton/robotics/humanoids/iit-coman-humanoid-robot (visited on 10/28/2016).

[GH10] Hartmut Geyer and Hugh Herr. "A Muscle-Reflex Model That Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18.3 (June 2010), pp. 263–273. ISSN: 1534-4320, 1558-0210.

[GSB03] H. Geyer, A. Seyfarth, and R. Blickhan. "Positive force feedback in bouncing gaits?" In: *Proceedings of the Royal Society B: Biological Sciences* 270.1529 (Oct. 22, 2003), pp. 2173–2183. ISSN: 0962-8452, 1471-2954.

[HW07] Daan G. E. Hobbelen and Martijn Wisse. "Limit Cycle Walking". In: (2007).

[Her15] François Heremans. "Controlling the balance of a humanoid subject to large perturbations". PhD thesis. UCL, 2015.

[Ijs08] Auke Jan Ijspeert. "Central pattern generators for locomotion control in animals and robots: A review". In: *Neural Networks* 21.4 (May 2008), pp. 642–653. ISSN: 08936080.

[Kor11] Petar Kormushev. *COMAN - compliant humanoid robot*. Petar Kormushev. June 30, 2011. URL: http://kormushev.com/topics/coman-robot/.

[Luk10] Tobias Luksch. "Human-like control of dynamically walking bipedal robots". PhD thesis. Technischen Universität Kaiserslautern, 2010.

[MBF15] Jonathan Samir Matthis, Sean L. Barton, and Brett R. Fajen. "The critical period for the visual control of foot placement in complex terrain occurs in the preceding step". In: *Perception and Action Lab, Cognitive Science Dept, RPI, Troy, NY, USA* (2015).

[MR08] David A. McCrea and Ilya A. Rybak. "Organization of mammalian locomotor rhythm and pattern generation". In: *Brain research reviews* 57.1 (Jan. 2008), pp. 134–146. ISSN: 0165-0173.

[Mur+66] M. P. Murray et al. "Comparison of free and fast speed walking patterns of normal men". In: *ResearchGate* 45.1 (Mar. 1, 1966), pp. 8–23. ISSN: 0002-9491.

[NB13] Nicolas Van der Noot and Allan Barrea. *Real-time control of a humanoid robot: Dynamic walking using a bio-inspired approach*. UCL, 2013.

[NB14] N. Van der Noot and A. Barrea. "Zero-Moment Point on a bipedal robot under bio-inspired walking control". In: MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference. Apr. 2014, pp. 85–90.

[NRI15] Nicolas Van der Noot, Renaud Ronsse, and Auke J. Ijspeert. "Biped gait controller for large speed variations, combining reflexes and a central pattern generator in a neuromuscular model". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 6267–6274.

[SB04] P. Sardain and G. Bessonnet. "Forces acting on a biped robot. Center of pressure-zero moment point". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 34.5 (Sept. 2004), pp. 630–637. ISSN: 1083-4427.

[SG12] Seungmoon Song and Hartmut Geyer. "Regulating speed and generating large speed transitions in a neuromuscular human walking model". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 511–516.

[SG15a] Seungmoon Song and Hartmut Geyer. "A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion: A spinal feedback circuitry generating human locomotion behaviors". In: *The Journal of Physiology* 593.16 (Aug. 15, 2015), pp. 3493–3511. ISSN: 00223751.

[SG15b] Seungmoon Song and Hartmut Geyer. "Regulating speed in a neuromuscular human running model". In: *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 217–222.

[STG12] Alexander Schepelmann, Michael D. Taylor, and Hartmut Geyer. "Development of a Testbed for Robotic Neuromuscular Controllers". In: *ResearchGate*. Robot. Sci. Syst. VIII - Online Proc. Jan. 1, 2012.

[Sak+02] Y. Sakagami et al. "The intelligent ASIMO: system overview and integration". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 3. 2002, 2478–2483 vol.3.

[Sap+16] A. A. Saputra et al. "Biologically Inspired Control System for 3-D Locomotion of a Humanoid Biped Robot". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46.7 (July 2016), pp. 898–911. ISSN: 2168-2216.

[Scr69] D. R. Scrutton. "Footprint Sequences of Normal Children under Five Years Old". In: *Developmental Medicine & Child Neurology* 11.1 (Feb. 1, 1969), pp. 44–53. ISSN: 1469-8749.

[Sof] *Software Documentation | BioRob*. URL: http://biorob.epfl.ch/page-36418-en.html (visited on 01/07/2017).

[Tag94] Gentaro Taga. "Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment". In: *Physica D: Nonlinear Phenomena* 75.1 (1994), pp. 190–208.

[VB04] Miomir Vukobratović and Branislav Borovac. "Zero-moment point — thirty five years of its life". In: *International Journal of Humanoid Robotics* 01.1 (Mar. 1, 2004), pp. 157–173. ISSN: 0219-8436.

[Wag+15] A. Wagoner et al. "Humanoid robots rescuing humans and extinguishing fires for Cooperative Fire Security System using HARMS". In: *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. 2015 6th International Conference on Automation, Robotics and Applications (ICARA). Feb. 2015, pp. 411–415.

[Wan+12] Jack M. Wang et al. "Optimizing Locomotion Controllers Using Biologically-Based Actuators and Objectives". In: *ACM transactions on graphics* 31.4 (July 2012). ISSN: 0730-0301.

[Whi07] Michael Whittle. *Gait Analysis: an introduction*. 4th ed. Vol. 1. Butterworth-Heinemann Elsevier, 2007. ISBN: 9 780 7506 8883 3.

[ZHR15] Petr Zaytsev, S. Javad Hasaneini, and Andy Ruina. "Two steps is enough: No need to plan far ahead for walking balance". In: IEEE, May 2015, pp. 6295–6300. ISBN: 978-1-4799-6923-4.

[bKb93] T. Öberg, Alek Karsznia, and K. Öberg. "Basic gait parameters: reference data for normal subjects, 10-79 years of age". In: *Journal of rehabilitation research and development* 30 (1993), pp. 210–210.

[AMA16] AMARSi Project. *COmpliant HuMANoid Platform (COMAN)*. 2016. URL: https://www.amarsi-project.eu/coman.

[CER16] CEREM. *Robotran - Homepage*. 2016. URL: http://www.robotran.be/.

[IIT16] IIT. *Humanoids and Human Centered Mechatronics*. www.iit.it. 2016. URL: https://www.iit.it/research/lines/humanoids-human-centered-mechatronics.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**