UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Controller for stair ascent and descent for a humanoid robot using bio-inspired mechanisms

Dissertation presented by
Philippe VERBIST

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

Supervisor
Prof. Renaud RONSSE

Readers
Prof. Auke IJSPEERT, Prof. Paul FISETTE , Dr. Nicolas VAN DER NOOT

Academic year 2017-2018

Acknowledgments

This master thesis has been carried out both at the Université catholique de Louvain (UCL, Louvain-la-Neuve, Belgium) and at the Ecole Polytechnique Fédérale de Lausanne (EPFL, Lausanne, Switzerland). Therefore, I would like to express my sincere gratitude to the members of those two universities that helped me throughout the realization of this work.

First, I would like to express my gratitude to **Professor Renaud Ronsse**, my supervisor at the UCL, for his active and continuous support, via videoconferencing, email correspondence and numerous guidance meetings – including one in Lausanne! His advice brought new ideas and provided useful feedbacks. Thanks to him, I could get into the Biorobotics Laboratory in Lausanne, which provided me an appreciable support.

Then, I would like to thank **Professor Auke Ijspeert**, who supervised my semester project at the EPFL, for welcoming me at the Biorobotics Laboratory, for his interest in the project and for his wise advice during the first semester.

When starting this work, I had the opportunity to receive essential information from **Dr. Nicolas Van der Noot**. I want to thank him warmly for his incredible insight and expertise about the CoMan and the existing controller that helped me to understand the foundations of this work.

I also express my thanks to **Dr. Amy Wu** for her support while I was at the BioRob laboratory at the EPFL, for her time for weekly meetings, advice and supervision.

Finally, I would like to express a special thanks to my **family** and **friends** for their support, and especially to **Dr. Jean-Claude Moureau** for his careful reading and to **Esther Lecompte**, who conducted a master's thesis on a different but close topic during the same period. Her presence and her availability were for me a great source of constructive feedback and motivation.

Philippe Verbist
Louvain-la-Neuve, June 2018

Abstract

Being able to increase humanoid robots locomotor capabilities is a great challenge. Making them able to deal with some known perturbations of the ground will widen their range of possible actions in our highly unpredictable human-made environment. With a wider working environment, it might be expected that humanoid robots could help humans in a lot of new situations.

In this work, we design a bio-inspired controller for the humanoid robot CoMan, to enable it to ascent and descent stairs in simulation. The controller has an existing bio-inspired controller for flat ground as starting point and uses mechanisms such as a musculoskeletal model, reflex rules and a central pattern generator. It is developed in a two-dimensional environment and tested using the simulation software Robotran.

We enable the robot to ascent and descent stairs of a rise of up to 6 cm (14 cm for humans given the size of the robot), which corresponds almost to a standard size of rise for humans. The minimization of the metabolic energy consumption is privileged over the control of the length of the steps. We test also the robustness of the controller during ascent and descent. Moreover, the robot is also able to perform the transition between flat ground and ascending stairs using a specific controller during transition. The transition between ascending stairs and flat ground is also possible.

Furthermore, we identify the most important muscles of the model during locomotion and the most important parameters of the controller that might be used later to modify the gait of the robot. Finally, we compare the muscles activations and the kinematics of the robot with human data.

List of abbreviations

General abbreviations

CPG	Central pattern generator
DOF	Degree of freedom
GCM	Ground contact model
IDM	Inverse dynamics methodology
MBS	Multibody system
MLC	Middle-level controller
PD	Proportional-derivative
PF	Pattern formation neuron
PSO	Particle swarm optimization
RDS	Reference data set
RG	Rhythm generation neuron
SEA	Series elastic actuator
ZMP	Zero moment point

Abbreviations related to the Hill-type muscle model

MTU	Muscle tendon unit
CE	Contractile element
SE	Series elastic element
PE	Parallel elastic element
BE	Buffer elastic element

Leg sagittal muscles

BFSH	Biceps femoris short head
GAS	Gastrocnemius
GLU	Gluteus
HAM	Hamstring
HFL	Hip flexor
RF	Rectus femoris
SOL	Soleus
TA	Tibialis anterior
VAS	Vastus

Contents

Chapter 1

Introduction

1.1 Context

Humanoid robots have gained much interest over the last decade because they give the opportunity to help humans in a wide variety of situations, ranging from everyday situations (e.g. human care, cooperative work) to more advanced situations (e.g. security services, maintenance tasks of industrial plants, etc). In particular, disaster scenarios such as the Fukushima nuclear accident clearly reveal the need for robots that are capable of meeting the requirements that arise during operation in real-world, highly unstructured and unpredictable situations, where human workers cannot be deployed due to radiation, danger of collapse, or toxic contamination [SRD+17].

Moreover, synthesizing biped robots provides the opportunity to better understand the human neuro-motor circuitry and to make progress in life-science subjects, and to lead to direct applications in prosthesis and rehabilitation robotics. Such robots can be viewed as test and experimental environments for theories from neuroscience and experimental psychology, as it is often impractical to experiment directly on the human nervous system [Ron17].

One important requirement for these robots is adaptability since they must interact and evolve in a human-designed environment. For fully autonomous navigation in a three-dimensional world, a humanoid robot must be capable of stepping up and down staircases and other obstacles where a sufficient large flat surface can support the robot's feet [GFF04]. This master thesis proposes to enhance the locomotion capabilities of a humanoid robot already able to walk on a flat ground in order to help it to cope with multi-level grounds.

1.2 Project objectives

The main objective of this project is to modify a bio-inspired controller for a humanoid robot to allow it to ascend and descend stairs of constant step height. This controller is applied in simulation to the humanoid robot CoMan in a two-dimensional environment. The controller uses bio-inspired mechanisms such as a musculoskeletal model, reflex rules and a CPG. The controller knows the height of the ground that is in front of the robot, but does not control the step length.

Complementary objectives are to study the transition between flat ground and stairs, to identify the most important muscles of the model during locomotion and to determine the most important parameters of the controller that might be used later to modify the gait of the robot.

In this work we assume that the robot knows somehow the shape of the staircase - this information is provided directly to the controller -, but the recognition of the environment, using a camera for example, is not addressed.

1.3 Executive summary

This report is split up into 7 main Chapters.

First, some specific terms, concepts and definitions related to the locomotion in robotics and essential to understand the remaining of the work are presented in Chapter 2. In the same Chapter, some existing successful control strategies are reported, as well as the positioning of this project in comparison with other closely related researches.

Then, the CoMan robotic platform that is used as embodiment for the controller developed in this work is presented in Chapter 3. Since it was not possible to work directly on the real robot, this master thesis is performed in simulation. The simulation software that is used and the model of some key elements of the environment are therefore presented. In particular, we introduce a particular model of stairs, the *virtual stairs*, that allow to constrain the rise, but not the step length of the robot. The virtual stairs are used in the remaining of this master thesis. The optimization algorithm that is used to find the open parameters of the controller is also introduced.

Chapter 4 describes the existing controller used as starting point, and the modifications that we applied to it in the context of this project to allow the robot to ascend and descend stairs. In this Chapter, we begin with the general working principle. Then, the middle-level neuromuscular controller that is at the core of this research is tackled, in particular the musculoskeletal model, the Central Pattern Generator and the reflexes. An overview of the high-level controller is also provided.

There exists no precise procedure to find the modifications that have to be done in a neuromuscular controller to obtain a new kinematics of the robot. For stair ascent, the modifications were found without any additional tool. On the other hand, it was necessary to have a methodology that helped to infer qualitatively the new controller rules during stair descent. The proposed method (called inverse dynamics methodology) is presented in Chapter 5. This methodology feeds a target kinematics into an existing controller, and the output torques of the controller are compared with reference torques to infer the new rules. The methodology is presented, tested, and finally used to infer the rules for stair descent.

Chapter 6 provides five experiments based on the controller for stair ascent. First, the open parameters of the controller are optimized for different rises. Since we use virtual stairs, as explained above, we obtain the natural gait of the robot, i.e. the gait to which the robot stabilizes itself naturally (in terms of metabolic energy consumption and of step length). The initial bio-inspired controller for flat ground shows interesting robustness properties; an experiment is therefore conducted to evaluate whether the new controller for stair ascent is robust against a perturbation of the rise. Given the bio-inspired property of the controller, some human-related information is extracted in another experiment: in particular, we identify the muscles that provide the highest work over a gait cycle. Then, two techniques used to perform the transition with flat ground are introduced. Finally, the end of this Chapter is dedicated to find the key-parameters of the controller that change the most while changing the rise or the step length. This information is indeed very valuable if we want to modulate more precisely the gait of the robot in function of those two variables.

A similar general procedure is applied to the controller for stair descent, and is reported in Chapter 7. First, the open parameters of the controller are optimized for different rises, in order to find the natural gait of the robot. A robustness experiment is then accomplished. And, finally, the muscles that deliver and absorb the highest amount of energy over one gait cycle are identified. However, the transition with a flat ground and the sensitivity analyses are not performed for stair descent.

In the last main Chapter, Chapter 8, we compare the muscles activations and the kinematics obtained with the controller designed in this master thesis with human data. Indeed, as said above in Subsection 1.1, the context of this research is also to test theories from neurosciences on robots. Evaluating the controller based on the movements that we perform in our everyday life to identify the similarities and the differences, provides the best feedback for the stairs controller that is designed in this work.

Three electronic addenda are submitted with this report. The first addendum contains the Figures of Chapters 2 to 8 and the Figures of the Appendices. The second addendum contains a video related to Chapters 6 and 7. The third addendum contains the whole project's code.

Chapter 2

Locomotion in robotics

2.1 Some essential concepts and definitions

To understand this report correctly, it is necessary to start with defining some terms and underlying concepts.

2.1.1 Gait cycle

When considering legged locomotion, a common terminology is to describe the occurrence of events relatively to the **gait cycle**. The gait cycle is the time interval between two successive occurrences of one of the repetitive events of walking [Whi07]. The initial event that is usually chosen is the initial contact of the right foot. This thesis will use the same convention. The cycle continues until the right foot contacts the ground again. Within one gait cycle, each leg goes through the 7 events shown in Figure 2.1. These event can be further classified in two periods: **stance**, when the leg touches the ground, and **swing**, otherwise. The left foot goes through exactly the same series events as the right, but displaced in time by half a cycle.

In this thesis, percentages of a gait cycle correspond hence to a percentage of time of the gait cycle as defined here. The terms swing and stance as defined here will also be regularly used.

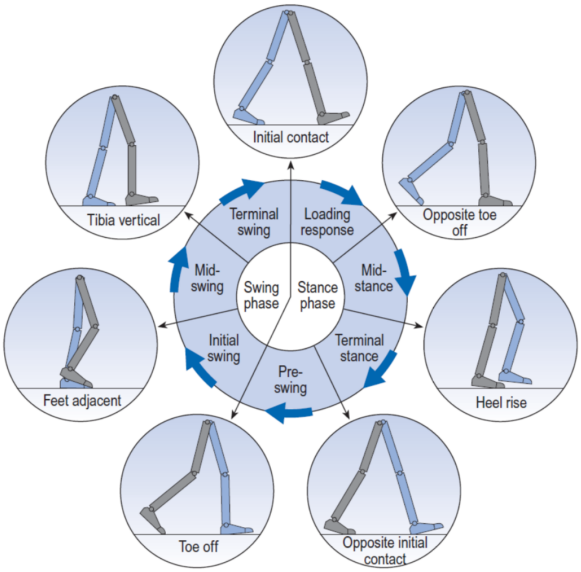


Figure 2.1: Different phases of the human gait cycle. Image courtesy from [Whi07].

2.1.2 Biomechanical planes

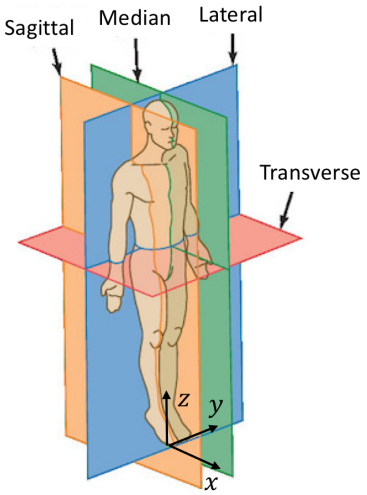


Figure 2.2: Definition of the world planes. Adapted from `unboundmedicine.com`.

To describe the rotation axis of each of the joints of a human or a robot, we will refer to the world planes depicted in Figure 2.2.

The **lateral** plane is normal to the x axis and divides the body into front and back sections. The **sagittal** plane is normal to the y axis and divides the body into left and right sections, and the **transverse** plane is normal to the z axis and divides the body into upper and lower segments.

An additional plane, the **median** plane, is parallel to the sagittal plane and divides the body into equal left and right parts.

In this work, a 2D movement concerns only the robot's DOF that act in the sagittal plane.

2.1.3 Stairs

Since this work studies the creation of a controller allowing a robot to walk on stairs, different specific terms related to the stair must be defined. We highlight the most relevant terms that will be used in the rest of this work. The definitions come from the ISO 3880/I-1977 norm that was last reviewed and confirmed in 2013; it is in force among others in Belgium and in Switzerland. The terms are illustrated in Figure 2.3.

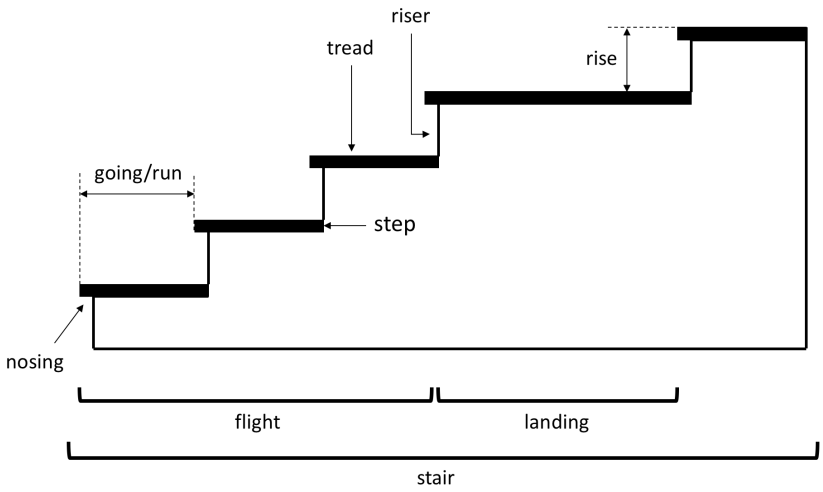


Figure 2.3: Specific terms related to a stair.

- **Stair**: a succession of horizontal stages (steps or landings) which makes it possible to pass on foot to other levels.

- **Step**: a part of a stair including a horizontal surface on which the foot is placed to go upstairs or downstairs.
- **Landing**: a platform or part of the floor structure at the ends of the flights.
- **Flight**: an unbroken series of steps between two landings.
- **Tread**: the horizontal part or upper surface of a step.
- **Riser**: the part closing the front face of the step.
- **Rise**: the vertical distance between two consecutive steps.
- **Nosing**: the front edge of a tread.
- **Going, run** (CA, US): the horizontal distance between the nosings of two consecutive steps.

In this work, we make the assumption that the stairs have no nosing. The going is therefore equal to the tread width/depth.

To make a clear difference between a *step* of the robot (the fact that the robot moves one leg forward and changes his weight repartition from one leg to the other) and a *step* of a staircase, we will always refer to the latter as a *stair step*.

A last definition that is not in the norm is added here: a **staircase** is a stair and its supporting structure. In this report, for practical purposes, we make no difference between the terms *stair* and *staircase*.

2.1.4 Gait features

The **step length** is defined as the horizontal distance between two consecutive foot strikes and is represented by the "L" in Figure 2.4. On stairs, the ideal step length is equal to the going (or tread width, if there is no nosing).

The **final step height** is the vertical distance between two consecutive foot strikes. The distance is computed when the feet are horizontal. When considering a staircase, the final step height is always equal to the rise. When walking on a flat ground, the final step height is equal to zero. The final step height is represented by the "H" in Figure 2.4.

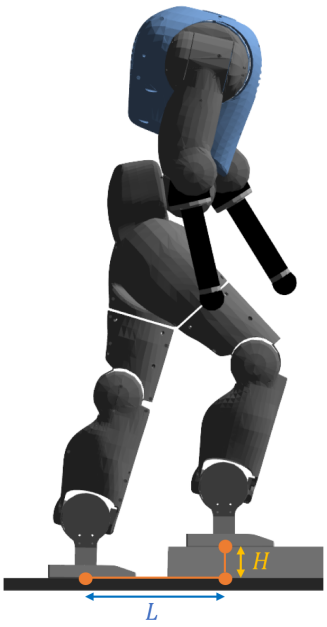


Figure 2.4: Definition of step length and final step height as used in this report.

Gait features vs controller parameters

In the context of this work, it is important to understand the difference between gait features and controller/gait parameters. Gait features will refer to the actual results obtained in terms of step length and final step height. Gait parameters on the other hand will refer to the parameters (or gains) of the controller.

2.2 Control strategies in stairs climbing

This Section emphasizes the position of this work in current researches.

2.2.1 Traditional strategies

The control of humanoid robots is a particularly difficult task due to the inherent instability of biped locomotion [FC08].

Some robots implement simple strategies to counteract this instability. For example, the electropneumatic robot EP-WAR3 ensures ground contact by suction cups. The robot movements to ascent and descent stairs are then translated into small and simple trajectories that are easily programmed on a PLC in form of gracfet flowchart [FCG03].

Other existing controllers often take advantage of the zero moment point (ZMP) criterion to predict dynamic balance and to generate walking patterns. The ZMP can be defined as the point on the ground at which the net moment of the internal forces and the gravity forces has no component along the horizontal axes [VB04]. This method requires a good trajectory generation and an effective control method.

[Shi99] proposed a 7-DOF biped robot with variable-length legs and a translatable balance weight in the body for stair climbing. The biped robot's walk relies on a careful control of the center of gravity's position whose dynamic stability is ensured by the ZMP criterion.

The ZMP algorithm is somewhat complex and computationally intensive since the full dynamics of the robot is considered. Some papers, e. g. [HCZH08] use the cart-table model (see Figure 2.5) to simplify the ZMP based control (the ZMP position can be computed very easily by $x_{ZMP} = x - \ddot{x}\frac{H}{g}$). Thanks to this simplification, they are able to propose an intuitive method for staircase walking.

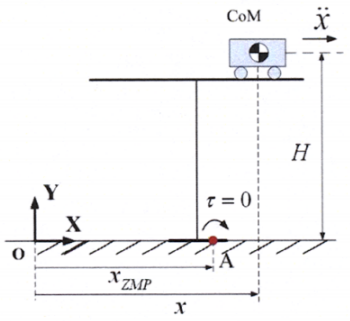


Figure 2.5: An example of cart-table model. Source: [HCZH08].

2.2.2 Bio-inspired strategies

Since humanoids anatomy is inspired from humans, it makes sense to also use control strategies inspired by nature.

In particular, humans regulate their muscles to adapt to the change of environments according to the sensory inputs [JW71]. With this consideration, [GH10] developed a muscle-reflex model of the human body that stabilize into a walking gait from its dynamic interplay with the ground. This model even predict some individual muscle activation patterns known from walking experiments. Compared to most traditional strategies, local torques are provided instead of providing fixed trajectory control. In comparison to a purely mathematical-based model, the movements of the robot are fluid, natural, energy-efficient, have limited computational costs and present a natural resistance against external perturbation. An improved model described in [SG15] shows that diverse behaviors – including stairs ascent (rise height of 10 cm and tread depth of 40 cm), but also slope ascent/descent, running and turning – are possible.

The model described in [GH10] was further improved in [VdNIR18], that added a Central Pattern Generator (CPG) on top of the muscle-reflex model. CPGs are mutually inhibiting neural circuits in the spinal cord that can produce rhythmic patterns of neural activity without rhythmic inputs [Ijs08]. In this work, the rhythmic patterns of interest are the rhythms underlying locomotion. The existence of CPGs has been demonstrated by various neurophysiological studies. CPGs exhibit several interesting properties. In particular, they produce stable rhythmic patterns, they have a few control parameters and are ideally suited to integrate sensory feedback gains. However, an adequate methodology is yet missing for designing CPGs and a solid theoretical foundation for describing their stability behavior does not exist [Ijs08].

This master thesis comes in the continuity of these bio-inspired controllers.

2.3 Related works

The bio-inspired approach described above has been used in various specific researches. The researches that are the most closely linked to this work are summarized hereafter. These contributions are all related to the design of a bio-inspired controller for the CoMan robot that we will address in the following Chapters. This work is inspired by methods developed in these related works.

Experimental validation of a reflex-based walking controller [VdNCB$^+$15]

This contribution developed a bio-inspired controller, based on a neuro-muscular model and reflex rules (inspired by [GH10]). The controller embodiment was a CoMan robot. After an optimization process, the robot was able to walk in simulation on flat ground. The controller was finally brought to the real hardware, leading to a successful walking of 50 steps.

Forward speed modulation during 2D walking gaits [VdNIR15]

As shown in [SG15], bio-inspired controllers based on reflex rules can generate diverse behavior of human locomotion. However, each specific gait needs a fine-tuning of the controller parameters, leading to difficult modulations of the gait parameters of a robot (walking speed, for example). The contribution [VdNIR15] proposes to combine reflexes (feedback signals) for leg distal muscles and a CPG (feed-forward signals) for leg proximal muscles. Large speed transitions are possible by modulating three high-level parameters as a linear function of the target speed, without further tuning of the other parameters. The resulting speeds ranged from 0.4 to 0.9 m/s.

Bio-inspired balance controller [HVdNIR16]

In this contribution, the balance of the robot in standing position is improved. The reference torques of a compliant impedance controller are translated into muscles stimulations using an inverse muscular model. These stimulations are then learned by a neural controller (cerebellar model or support vector regression algorithm). After the learning phase, the robot manages to perform a full-body compensation to reject balance perturbations in simulation.

Gait modulation of a humanoid robot, using bio-inspired mechanisms [Gre17]

Instead of modulating the speed, as in [VdNIR15], this contribution modulates in simulation two other parameters: the step length and the step height. The controller is based on reflexes and on a CPG. Eight parameters are modulated as linear or quadratic function of the target step length and step height. The resulting ranges are 20-50 cm for the step length, and 4-12 cm for the step height.

Balance improvement of a humanoid robot while walking [Lec18]

This contribution, done in parallel to this work, proposes to increase the stability of the controller developed in [HVdNIR16] by implementing a specific balance strategy during walking. A stability criterion based on the difference of position between the center of mass of the robot and the center of mass of internal models is established. When this criterion is active – meaning the robot is about to fall due to external perturbations such as pushes – an impedance controller is used to recover correct joints position.

Chapter 3

Robotic platform and tools

This Chapter presents the robot CoMan that is used as reference hardware during this work, as well as the necessary tools to design a controller on stairs for it, namely the Robotran simulation environment and the Particle Swarm Optimization algorithm.

3.1 CoMan

The robot used for this project is the CoMan (Compliant Humanoid Platform) robot (see Figure 3.1) developed by the Italian Institute of Technology (IIT) within the AMARSI European research project [AMA].

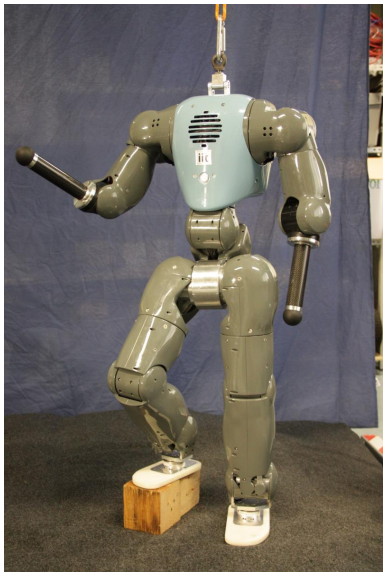


Figure 3.1: The CoMan robot. Source: [AMA].

The robot has the particularity to be modeled as a 4-year-old child and is one of the first complete humanoid robots. It is 95 cm tall, weighs 31 kg and possesses 25 degrees of freedom as depicted in Figure 3.2. Two degrees of freedom, situated in the neck, will not be used throughout this work.

Moreover, CoMan is equipped with compliant joints, using series elastic actuators (SEA) [TLSC11]. A series elastic actuator is an actuator that incorporates an elastic element as a purposeful element. Such actuators bring greater shock tolerance, accurate and stable force control and a safer interaction with the environment [PW95]. From a biological point of view, such actuators tend to model the elasticity present in legged locomotion through tendons. Thanks to all these advantages, the actuation of the robot is closer to the model of a real human joints than traditional stiff actuators.

The CoMan robot is a good candidate to test bio-inspired mechanisms, thanks to its morphological properties (human-like, actuators elasticity) that make it closer to a human than most of other robotic platforms.

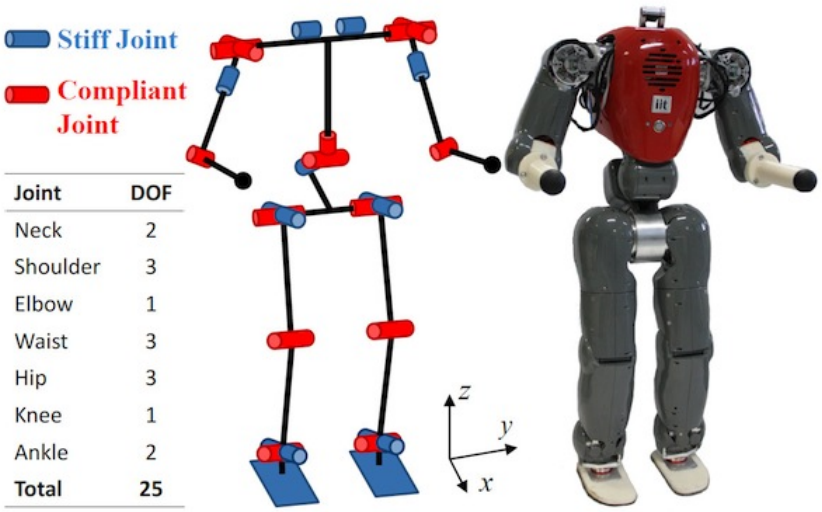


Figure 3.2: Representation of the 25 DOFs of the CoMan. Source: [Fal13]

3.2 Robotran simulation environment

The robot hardware was not available during this master thesis and it was thus needed to operate in simulation. Besides, working in simulation presents several advantages, such as the ability to test a controller without risking to damage the real robot or the ability to perform mass optimizations. We work with the Robotran simulation environment. In this environment, we need to model the CoMan, the ground reaction forces and stairs.

3.2.1 Overview of Robotran

Robotran is a general purpose multibody simulation environment developed by the Multibody research group of the Mechatronic, electrical energy, and dynamic systems (MEED) division of the institute of Mechanical, Material and Civil engineering (iMMC) of the Université catholique de Louvain (UCL)[CER].

A multibody system (MBS) is a mechanical system made of several rigid or flexible bodies interconnected by joints [SF03]. These bodies can undergo internal or external forces and torques. In our case, those forces can result from electro-mechanical components in the robot itself, such as the series elastic actuators (SEA) or from interactions with the external world such as gravity or the contact with the ground. In Robotran, specific model features such as sensors are made available to the user. Robotran uses the symbolic approach to generate the multibody motion equations (kinematics, dynamics, inverse-dynamics).

3.2.2 Modelization of the CoMan

This thesis uses the Robotran model of the CoMan used by Nicolas Van der Noot and Allan Barrea [BVdN13]. In particular, this model simulates the CoMan using only inputs and outputs available on the real robot, making it easy to transfer the code developed in simulation on the real robot. For this latter reason, some non-idealities of the real robot are also included in the model, such as the actuators dynamics [ZHVdN$^+$17] and the sensors noise.

3.2.3 Modelization of the ground reaction forces

Each foot of CoMan in simulation is modeled using a mesh of 200 points. At each point, a contact force with the ground is computed as follows:

- the tangantial component is computed using a linear spring-damper model for static friction or the Coulomb's law for dynamic friction;
- the vertical force is computed using a non linear spring-damper model.

The high density of the mesh allows to have a precise repartition of the ground reaction forces on the feet. All elementary forces are then added together to find the resulting force on the feet.

3.2.4 Modelization of stairs

In the model, the ground height h under one particular point of the foot can be described using a simple customizable function $h = h(x, y)$. Since we work in a 2D environment, the function depends only on one coordinate: $h(x, y) = h(x)$. This Subsection explains how we design the function h for a stair.

A natural way to describe a staircase on a map is to fix statically its coordinates, as shown on Figure 3.3.

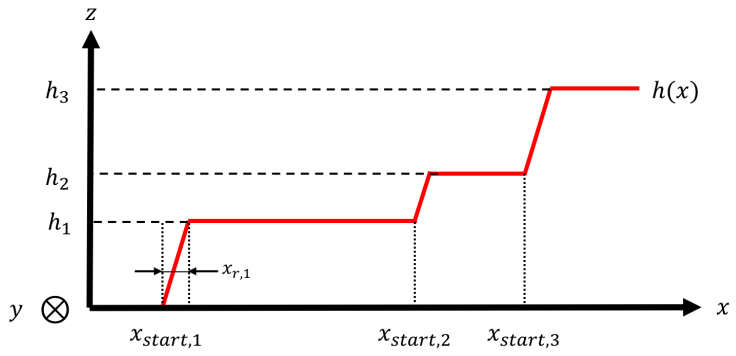


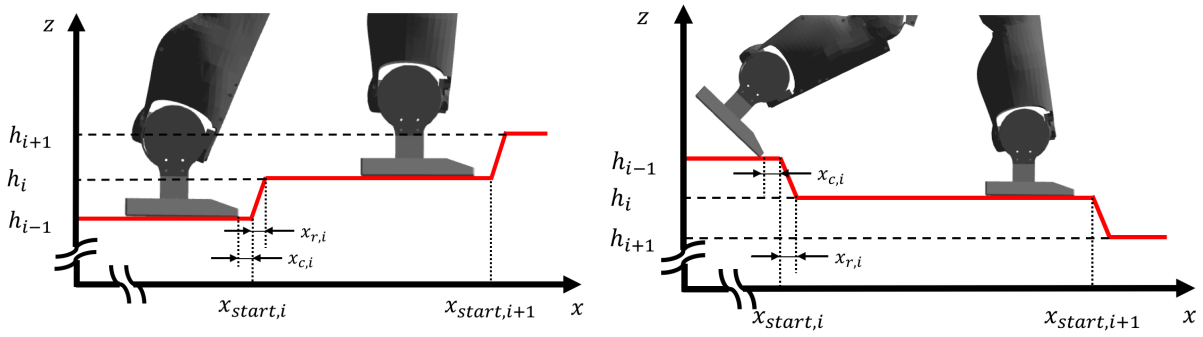
Figure 3.3: Natural description of the parameters of a stair. The robot walks towards the $+x$-direction. The i^{th} step starts at $x_{start,i}$ and has an absolute height h_i. The riser has a horizontal length of $x_{r,i}$ in the $+x$-direction and is set to a non-zero value to avoid an infinite slope. Axis x and z have unit of distance.

According to the notations of this Figure, the function representing the stair is therefore:

$$h(x) = \begin{cases} h_{i-1} + (h_i - h_{i-1}) \cdot \left(\frac{x - x_{start,i}}{x_{r,i}}\right) & \text{for } \{x, i\} \text{ s.t. } x_{start,i} \leq x < x_{start,i} + x_{r,i} \\ h_i & \text{for } \{x, i\} \text{ s.t. } x_{start,i} + x_{r,i} \leq x < x_{start,i+1} \end{cases}$$

We do not expect the robot to perform always the same step length, because the bio-inspired controller that will be introduced in the next Chapter does not guarantee the exact strikes positions, and also because some non-idealities are modeled (see Subsection 3.2.2). After a few steps, even a small error in the foot placement increases in importance and can lead the robot to miss a step, and therefore to fall off the staircase.

To avoid this problem, we introduce **virtual stairs**: their particularity is to set the starting position $x_{start,i}$ of each step a distance $x_{c,i}$ in front of the toes of the foot performing the i^{th}



(a) Ascending

(b) Descending

Figure 3.4: Virtual stair model. The starting position of each stair step is equal to the position of the toe of the last foot strike plus an additional clearing distance $x_{c,i}$.

strike. It is important to emphasize that the parameters $x_{start,i}$ are constructed as the robot walks along the stair and are therefore automatically adapted to the step length of the robot.

Virtual stairs are shown for ascending and descending in Figures 3.4a and 3.4b.

In this work, the horizontal length of the riser $x_{r,i}$ and the clearance length $x_{c,i}$ are both set arbitrarily to 1 cm.

3.3 Particle Swarm Optimization

In Chapter 4 we will introduce a controller with a lot of open parameters that need to be found. An efficient optimization framework is therefore needed that will be able to identify the optimal parameters given an objective function (or fitness function) to maximize. As in [VdNIR18], this work will use an heuristic optimization technique called Particle Swarm Optimization (PSO). PSO is a population-based algorithm for optimization that is based on social-psychological principles [Zeu11].

A PSO algorithm starts by generating a set of N random solutions, where each parameter is taken at random (inside a user-defined range). The fitness of each particle is then computed and a new set of N particles is generated. Each solution is attracted both by its own best fitness position and by other solutions with higher fitnesses. We have no guarantee that this heuristic optimization will find the global optimum, since it might get stuck in a local minimum, and multiple optimizations for a single evaluation are thus needed. The details of the implementations of the PSO algorithm can be found in [vdK14].

During this work, we had access to the hardware cluster and optimization framework (including PSO) of the BioRob laboratory of the EPFL, Lausanne.

Chapter 4

Bio-inspired controller

As already stated, the controller used to compute the reference torques at the articulations of the robot is bio-inspired.

An important point is that we make the working assumption that the robot moves in a 2D environment: the waist of the robot is constrained to stay in the sagittal plane, removing the need to perform a lateral balance control.

The controller general architecture is taken from [VdNIR15]. In [Gre17], a slight modification was made to the controller to use two more muscles to control more precisely the position of the leg. This work uses the same modification. The controller is further modified to be able to cope with stairs.

4.1 General working principle

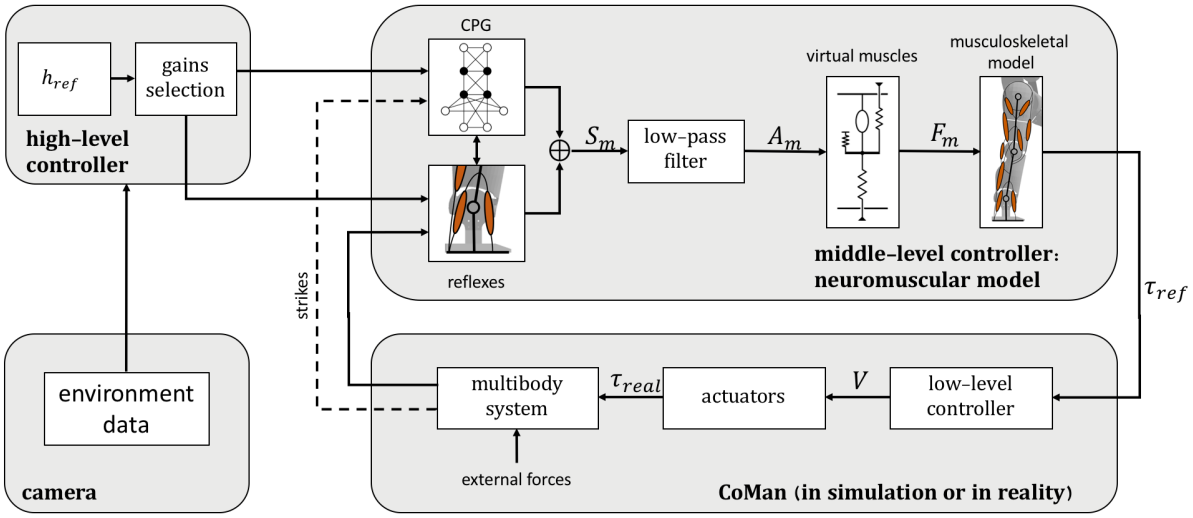


Figure 4.1: Bio-inspired controller, adapted from [VdNIR18].

The controller pictured in Figure 4.1 aims at providing appropriate voltages to the robot motors, in order to achieve locomotion. The voltages are either sent to the real robot motors or to their model in simulation. The controller has three main parts: a high-level controller, a middle-level neuromuscular controller and a low-level impedance controller.

Since the low-level impedance controller is not the main concern of this work, it is not studied in detail here. More details can be found in [DMMC+13]. On the other hand, the functioning of the middle- and the high-level controllers are important for the present work and are detailed hereafter.

4.2 Middle-level neuromuscular controller

The body of the robot is composed of three main parts that will be actuated differently: the sagittal DOFs of the legs, the non-sagittal DOFs of the legs, and the DOFs of the torso and the arms.

The non-sagittal DOFs of the legs are out of interest because we work in a 2D environment: those joints are given a constant zero position reference. Besides, it was not useful to change the controller for the DOFs of the torso and the arms because it gives already satisfying results.

In this Section, we are particularly interested in the actuation of the sagittal DOFs of the legs, and we will thus focus on those 6 DOFs.

4.2.1 Musculoskeletal model

The key principle of the controller is to model muscles in the sagittal plane of each leg of the robot. By choosing appropriate stimulation for these muscles, we cause them to react, resulting in a movement of the legs. These muscles are of course virtual, and their behavior will be simulated and replicated on the robot.

First, a virtual skeletal structure is needed to attach the muscles. Each leg is therefore divided into 3 parts: the thigh, the low leg and the foot. The joints that we study are the hip, the knee and the ankle.

Then, nine muscle groups from the human anatomy are attached on this structure, as shown in Figure 4.2. The muscles that are considered in this work are: hip flexors (HFL), gluteus (GLU), hamstring (HAM), vastus (VAS), gastrocnemius (GAS), soleus (SOL), tibialis anterior (TA), biceps femoris short head (BFSH) and rectus femoris (RF). The two last muscles were added to the original model because they allow to modulate more precisely the control of the knee bending. Indeed, adding a muscle on a joint gives more possibilities to act on this joint and increases the maximum torque that can be applied to it. We expect that a more precise control of the knee will be useful during a walk on a stair.

When stimulated, a muscle produces a given force and generates a torque, which can be computed using the attachment point on the skeletal structure. The reference torque of each joint is thus computed as the sum of the torques generated by each muscle.

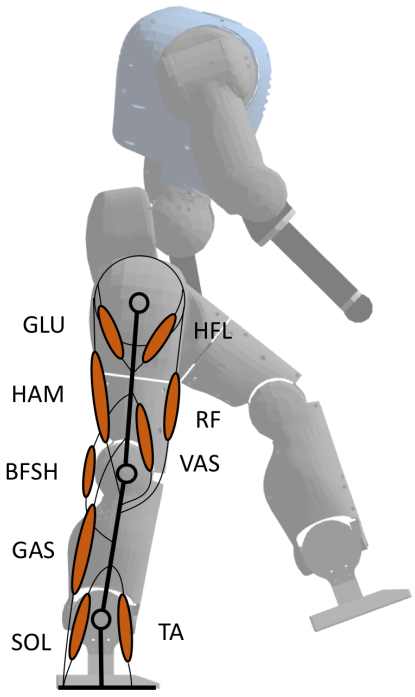


Figure 4.2: The 9 muscle groups acting on leg sagittal DOFs.

4.2.2 Muscle Tendon Unit

Each muscle group is modeled as a Hill-type Muscle Tendon Unit (MTU), as suggested in [SG15] and represented in Figure 4.3.

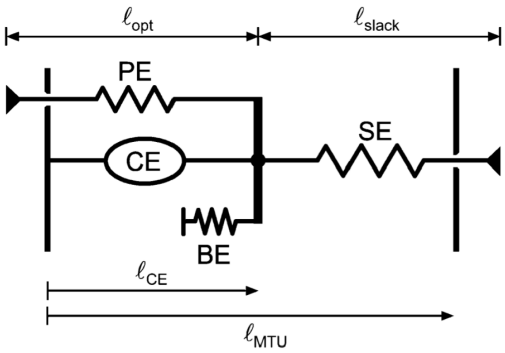


Figure 4.3: Muscle-tendon unit from the Hill-model. Source: [GH10].

Model Each MTU is made of two main elements: a contractile element (CE) and a series elastic element (SE). A parallel elastic element (PE) engages when the CE stretches beyond its optimum length and a buffer elastic element (BE) prevents the CE from collapsing when CE is under its minimal length.

The length of the contractile element l_{ce} is the only state variable of the muscle. The speed of this contractile element v_{ce} is a function of the length of the MTU l_{mtu} and a function of the muscle activation A_m. The lengths of the MTUs are computed from the angular configurations of the robot.

Joint soft limits The joints of the CoMan have range limits, like human joints. As the Hill-model does not take this into account, an additional torque is applied on each joint that stretches beyond its limit.

Dynamic scaling The muscle parameters were gathered on an adult-like model of 80 kg and transformed to fit the model of a 4-year-old child. The transformation was done using dynamic scaling, that uses fundamental physical variables (mass, length, time) to obtain system quantities at different scales.

4.2.3 Muscle stimulations

The muscles have two sources of stimulations. The proximal[1] muscles are stimulated by a Central Pattern Generator (CPG), while the distal muscles are stimulated by reflex rules.

The stimulations are then filtered by a low-pass filter, resulting in the activations of the muscles.

[1]The proximal muscles are the muscles that are the most close to the torso, while the distal muscles are the most distant from the torso.

4.2.4 Central Patern Generator

A Central Pattern Generator (CPG) is a neural circuit capable of producing rhythmic patterns without receiving rhythmic inputs. The CPG used is represented in Figure 4.4 and uses 10 neurons to generate its outputs. The explanations given here are adapted from [VdNIR15]. The four numbered neurons N_1, N_2, N_3 and N_4 are called Rhythm Generation (RG) neurons and are the central elements of the whole CPG network. The six other neurons, denoted with letters, are called Pattern Formation (PF) neurons and form the stimulation patterns for the muscles.

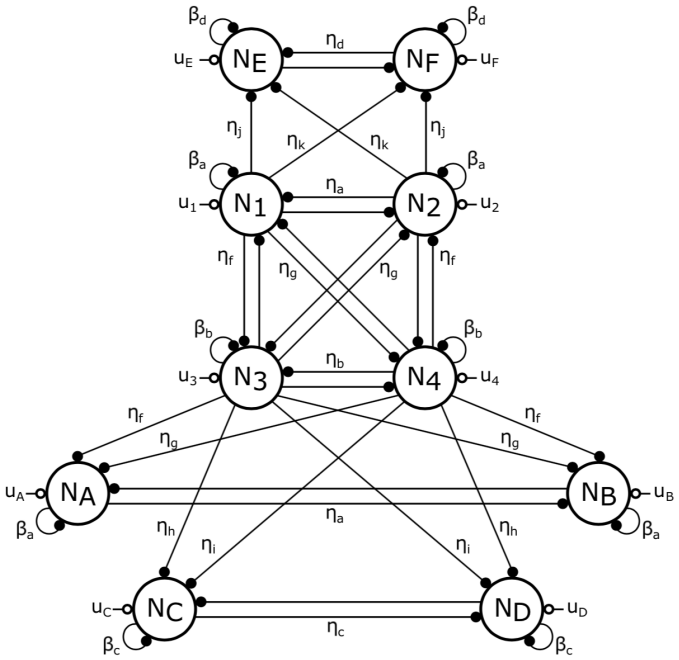


Figure 4.4: Central Pattern Generator. Adapted from [VdNIR15].

Structure 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 (see later). 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. Each neuron is characterized by two state variables: its firing rate x and its self-inhibition v.

Firing rates The main state of each neuron is captured by its firing rate x_i, the evolution of which is described by the well known equations [Tag94]:

$$\dot{x}_i = \frac{1}{\tau}\left(-x_i - \beta_i v_i - \sum_{\forall j \neq i} \eta_j [x_j]^+ + u_i\right). \tag{4.1}$$

In this equation, τ is the time constant of the CPG, v_i is the self-inhibition, modulated by a constant β_i to be optimized[2], η_j is the connection strength to be optimized and u_i is an external excitation associated to each neuron. The function $[\bullet]^+$ takes the positive part of its argument,

[2]The optimization process will be described in Chapter 6.

and saturates to zero if the argument is negative (i.e. $[x]^+ = max(0, x)$), capturing the fact that the activation of a given neuron decreases when another is active. 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 full equations are given in Appendix A.1

Self-inhibitions The self-inhibitions v_i are computed using Equation 4.2, where γ_i are constants to be optimized. The full equations are available in Appendix A.2.

$$\dot{\nu}_i = \frac{1}{\gamma_j \tau} \left(-\nu_i + [x_i]^+ \right) \tag{4.2}$$

Synchronization To keep the CPG in synchronization with the robot behavior, 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 related, depending on when the foot strikes happen and on the current state 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 4.3 and 4.4. 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**.

$$\begin{aligned} u_1 &= u - [x_1]^+_{SL} + [x_1]^-_{Str,R} & u_C &= u - [x_C]^+_{SL} \\ u_2 &= u - [x_2]^+_{SR} + [x_2]^-_{Str,L} & u_D &= u - [x_D]^+_{SR} \\ u_3 &= u - [x_3]^+_{SL} - [x_3]^+_{Str,L} & u_E &= u - [x_E]^+_{SL} \\ u_4 &= u - [x_4]^+_{SR} - [x_4]^+_{Str,R} & u_F &= u - [x_F]^+_{SR} \end{aligned} \tag{4.3}$$

$$\begin{aligned} u_A &= u - [x_A]^+_{SL} + [x_A]^-_{Str,R} - [x_A]^+ [\theta_{ref} - \theta_t]^+_{1/0} \\ u_B &= u - [x_B]^+_{SR} + [x_B]^-_{Str,L} - [x_B]^+ [\theta_{ref} - \theta_t]^+_{1/0} \end{aligned} \tag{4.4}$$

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 $[\bullet]^+_{1/0}$ returns 1 if its argument is positive, 0 otherwise (similarly for $[\bullet]^-_{1/0}$ with negative arguments). The variable θ is the torso sagittal lean angle. This allows to reduce the excitations u_A and u_B to help track these reference angles more accurately.

If the CPG is **ahead of the walking gait**, 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.

Outputs Finally, to use the firing rates as stimulations for the muscles, 6 different outputs are computed, as defined in Equations 4.5.

$$\begin{aligned} y_1 &= \left[[x_A]^+ - [x_3]^+\right]^+ & y_4 &= \left[[x_D]^+ - [x_4]^+\right]^+ \\ y_2 &= \left[[x_B]^+ - [x_4]^+\right]^+ & y_5 &= \left[[x_E]^+ - [x_1]^+\right]^+ \\ y_3 &= \left[[x_C]^+ - [x_3]^+\right]^+ & y_6 &= \left[[x_F]^+ - [x_2]^+\right]^+ \end{aligned} \tag{4.5}$$

Muscles stimulations The outputs of the CPG are designed to feed the appropriate stimulations to the leg proximal muscles at different times.

$$\begin{aligned} S_{BFSH,R} &= k_{BFSH,1} \cdot y_4 + k_{BFSH,2} \cdot y_6 + k_{BFSH,3} \cdot y_1 + k_{BFSH,4} \cdot y_5 \\ S_{GLU,R} &= k_{GLU,1} \cdot y_1 + k_{GLU,2} \cdot y_6 \\ S_{HAM,R} &= k_{HAM,1} \cdot y_1 + k_{HAM,2} \cdot y_2 + k_{HAM,3} \cdot y_6 + k_{HAM,4} \cdot y_5 \\ S_{HFL,R} &= k_{HFL,1} \cdot y_4 + k_{HFL,2} \cdot y_6 \\ S_{RF,R} &= k_{RF,1} \cdot y_4 + k_{RF,2} \cdot y_6 + k_{RF,3} \cdot y_5 \end{aligned} \tag{4.6}$$

$$\begin{aligned} S_{BFSH,L} &= k_{BFSH,1} \cdot y_3 + k_{BFSH,2} \cdot y_5 + k_{BFSH,3} \cdot y_2 + k_{BFSH,4} \cdot y_6 \\ S_{GLU,L} &= k_{GLU,1} \cdot y_2 + k_{GLU,2} \cdot y_5 \\ S_{HAM,L} &= k_{HAM,1} \cdot y_2 + k_{HAM,2} \cdot y_1 + k_{HAM,3} \cdot y_5 + k_{HAM,4} \cdot y_6 \\ S_{HFL,L} &= k_{HFL,1} \cdot y_3 + k_{HFL,2} \cdot y_5 \\ S_{RF,L} &= k_{RF,1} \cdot y_3 + k_{RF,2} \cdot y_5 + k_{RF,3} \cdot y_6 \end{aligned} \tag{4.7}$$

The gains $k_\bullet$ are constants to be optimized. The constant $k_{HFL,2}$ has a non-zero value only for descending stairs. In addition to these rules, the stimulations of the HFL and the GLU are adapted to track more easily the torso sagittal angle. First, a proportional-derivative (PD) controller on the torso sagittal lean angle is computed as follows:

$$\Delta_{\theta,\{R,L\}} = \left(k_{p,\theta}(\theta_{ref} - \theta_t) - k_{d,\theta}\dot{\theta}_t\right) \tilde{F}_{gd,\{R,L\}}, \tag{4.8}$$

where θ_t is the torso sagittal lean angle, and $\tilde{F}_{gd,\{R,L\}}$ is the vertical force below the right/left foot normalized to the walker weight. $k_{p,\theta}$, $k_{d,\theta}$ and θ_{ref} are parameters to be optimized.

Stimulations are then added to the muscles:

$$\begin{aligned} S_{HFL} &\stackrel{+}{=} \left[\Delta_{\theta,\{R,L\}}\right]^+ \\ S_{GLU} &\stackrel{+}{=} \left[\Delta_{\theta,\{R,L\}}\right]^- \end{aligned} \quad \text{if} \begin{cases} y_1 = 0 \text{ and } y_4 = 0 \text{ (for right leg)} \\ y_2 = 0 \text{ and } y_3 = 0 \text{ (for left leg)} \end{cases} \tag{4.9}$$

4.2.5 Reflexes

The reflexes are mainly applied on distal muscles. They are based on some internal states of the muscles, such as their force or their length. The reflexes are different depending on whether we consider, on one hand, a flat ground or stair ascent, or on the other hand, stair descent.

Flat ground or ascending stairs When the robot walks on flat ground or on ascending stairs, it uses the stimulations presented in the following table, where $\tilde{F}_m$ is the force produced by the muscle m normalized to its maximal force, $\tilde{l}_{ce,m}$ is the length of the CE element of the muscle m normalized to its optimal length. The gains $G_\bullet$ and the lengths $l_{TA,st}$ and $l_{TA,sw}$ are parameters to be optimized.

Stance		Swing	
$S_{VAS} =$	$S_{MIN} + G_{VAS}\tilde{F}_{VAS}$	$S_{VAS} =$	S_{MIN}
$S_{GAS} =$	$S_{MIN} + G_{GAS}\tilde{F}_{GAS}$	$S_{GAS} =$	S_{MIN}
$S_{TA} =$	$S_{MIN} + G_{TA,st}(\tilde{l}_{ce,TA} - l_{TA,st})$	$S_{TA} =$	S_{MIN}
	$-G_{SOL,TA}\tilde{F}_{SOL}$		$+G_{TA,sw}\left(\tilde{l}_{ce,TA} - l_{TA,sw}\right)$
$S_{SOL} =$	$S_{MIN} + G_{SOL}\tilde{F}_{SOL}$	$S_{SOL} =$	S_{MIN}

Moreover, in stance, $S_{VAS} = S_{MIN}$ during double support phase or when the knee is close to over-extension: $\phi_k < \phi_{k,th}$ and $\dot{\phi}_k < 0$, where ϕ_k is the angle of the knee and $\phi_{k,th}$ is a parameter to be optimized.

Descending stairs When the robot descends stairs, the main difference concerns the VAS muscle. In particular, we force it to bend during stance thanks to a PD controller:

$$\Delta_{\phi,k,\{R,L\}} = k_{c,\phi_k} + \left(k_{p,\phi_k}(\phi_{k,ref} - \phi_{k,\{R,L\}}) - k_{d,\phi_k}\dot{\phi}_k\right)\tilde{F}_{gd,\{R,L\}},$$

where $\phi_{k,\{R,L\}}$ is knee angle of the left and the right foot, $\tilde{F}_{gd,\{R,L\}}$ is the vertical force below the right/left foot normalized to the walker weight, and $k_{c,\phi_k}, k_{p,\phi_k}, k_{d,\phi_k}$ and $\phi_{k,ref}$ are parameters to be optimized.

The parameters are summarized in the following table, where the gains $G_\bullet$, $K_{VAS,sw}$ and the lengths $l_{TA,st}$ and $l_{TA,sw}$ are parameters to be optimized.

Stance		Swing	
$S_{VAS} =$	$S_{MIN} + \left[\Delta_{\phi,k,\{R,L\}}\right]^+$	$S_{VAS} =$	$S_{MIN} + K_{VAS,sw}$
			$-G_{VAS,HAM}\tilde{F}_{HAM}$
$S_{GAS} =$	$S_{MIN} + G_{GAS}\tilde{F}_{GAS}(t_l)$	$S_{GAS} =$	S_{MIN}
$S_{TA} =$	$S_{MIN} + G_{TA,st}(\tilde{l}_{ce,TA} - l_{TA,st})$	$S_{TA} =$	S_{MIN}
	$-G_{SOL,TA}\tilde{F}_{SOL}$		$+G_{TA,sw}\left(\tilde{l}_{ce,TA} - l_{TA,sw}\right)$
$S_{SOL} =$	$S_{MIN} + G_{SOL}\tilde{F}_{SOL}$	$S_{SOL} =$	S_{MIN}

Moreover, in stance, $S_{VAS} = S_{MIN}$ when the knee is close to over-extension ($\phi_k < \phi_{k,th}$ and $\dot{\phi}_k < 0$) or during double support phase. In swing, a similar rule applies to prevent hip over-extension: $S_{HFL} = S_{RF} = S_{MIN}$ when $\phi_h < \phi_{h,th}$ and $\dot{\phi}_h < 0$, where ϕ_h is the hip angle and $\phi_{h,th}$ is a parameter to be optimized.

4.3 High-level controller

As seen in the previous Section, the middle-level controller is made of a high number of gains that have to be optimized. In the context of stairs, these gains are different depending on the desired gait features, and in particular the desired final step height, that depends on the rise of the following stair step.

The high-level controller :

- analyzes the ground in front of the robot,
- chooses the next reference final step height h_{ref},
- and selects the according set of gains of the middle-level controller.

For now, we suppose that the robot has a perfect knowledge of the ground that lies in the median plane in front of him and knows therefore the rise of the following stair step. In the real CoMan, this information could be brought by a vision system.

Chapter 5

Inverse dynamics methodology

The previous Chapter described the controller we developed for walking on stairs, based on a previous controller for flat ground.

To design this new version of the controller, it was necessary to:

- change some stimulation rules of the muscles,
- find ranges for various gains of the model that will allow to find a functional solution (i.e. a solution walking on stairs) during an optimization process.

This is true in particular for stair descent, where the rules are more complicated.

For a bio-inspired controller, there does not exist yet a rigorous and detailed procedure to find these informations. However, this Chapter proposes a methodology – called the inverse dynamics methodology (IDM) – that provides qualitative informations related to those two aspects.

The name inverse dynamics methodology comes from the fact that this methodology helps to find the muscles stimulations (and therefore the corresponding torques at the joints) based on a reference kinematics.

5.1 Methodology

5.1.1 General presentation

Normally, the middle-level controller (MLC) is used in closed-loop with the robot (in reality or in simulation), as depicted in Figure 4.1. The input data of the MLC contains mainly kinematics data, as well as data from the interaction with the environment (forces below the feet). The outputs are the torques $\boldsymbol{\tau}_{ref}$ that must be applied to the actuators of the robot.

In the following, we are studying in the case where the MLC does not produce the correct torque patterns. In this case it is useless to try to use the MLC in closed-loop since the robot falls immediately. The solution proposed here is to isolate the MLC and to impose a known and correct kinematics as input. The outputs of the MLC are then observed, and adjustments can be made based on the observations. This procedure is repeated until the MLC produces correct torques.

More precisely, the proposed implementation (that is called inverse dynamics methodology) is presented in Figure 5.1 and proceeds as follows:

- first, isolate the MLC from the robot (in reality or in simulation) to study its behavior;
- find a reference data set (RDS) that contains the needed inputs for the MLC (kinematics, contact forces with the ground, etc) and the expected torques at the joints level (see Subsection 5.1.3 for more details);
- feed the MLC with input data D_{in}^{RDS} from the RDS, and run a simulation;
- compare:
  - the output torques τ_{ref} of the MLC with the torques τ_{real}^{RDS} of the RDS;
  - the muscles stimulations S_m computed in the MLC with the stimulations S_m^{RDS} of the RDS. If there is no reference muscles at our disposal, it is possible to use the inverse muscular model developed in [HVdNIR16] to find the stimulations needed to render a given torque profile;
- use the differences highlighted during the last point to infer:
  - new stimulation rules of the muscles,
  - new ranges for the gains of the model.

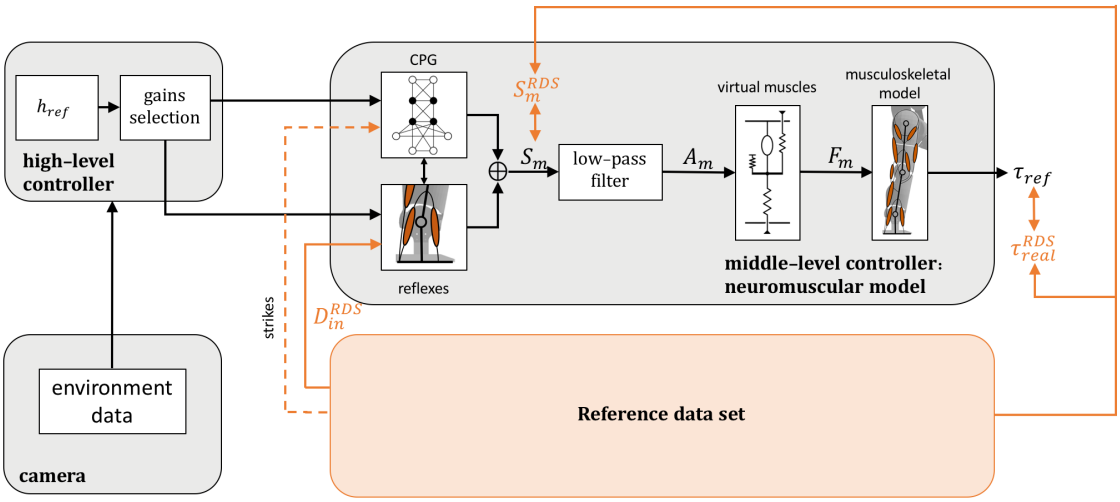


Figure 5.1: Implementation of the inverse dynamics methodology.

In this methodology, we have two DOFs on which we can act:

- changing the MLC,
- changing the reference data set for a different kinematics (level walking, stair ascent, stair descents, ...).

5.1.2 Assumptions

The IDM described above is based on three assumptions.

First, as already stated, we work in a 2D environment. Therefore, we study only the movements and the muscles simulations in the sagittal plane.

A second assumption is that the reference torques delivered by the MLC are equal to the torques that would produce the actuators and therefore, the body of the robot: $\boldsymbol{\tau}_{ref} \approx \boldsymbol{\tau}_{real}$ (see Figure 4.1). In other words, the low-level controller has an infinite bandwidth. This assumption is important to have a meaningful comparison between $\boldsymbol{\tau}_{ref}$ (from the MLC) and $\boldsymbol{\tau}_{real}^{RDS}$ (from the RDS). A simple experiment is made to validate this assumption: the two signals are compared for the three sagittal joints of each leg. Figure 5.2 shows the results of the comparison for the left leg[1]. The two signals are almost always equal, as expected.

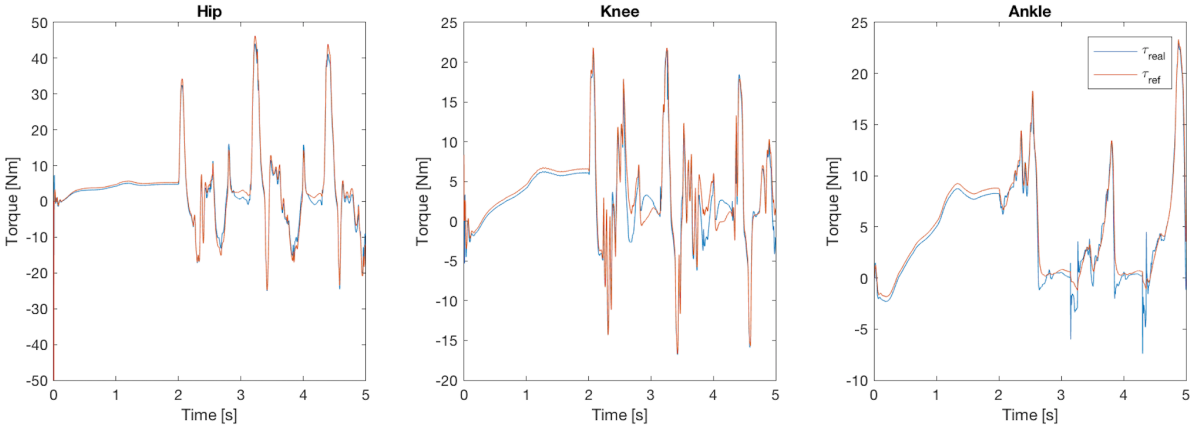


Figure 5.2: Comparison between the reference torques $\boldsymbol{\tau}_{ref}$ computed by the MLC and the torques at the joints $\boldsymbol{\tau}_{real}$. When the MLC is used in closed-loop, these two torques are very similar, validating the assumption $\boldsymbol{\tau}_{ref} \approx \boldsymbol{\tau}_{real}$. Here, the results are displayed when the robot walks on a flat ground.

A last assumption is that forces below the feet can be approximated following a simple rule:

- when the data clearly indicates that one leg is in stance and the other one is in swing, the ground reaction force acting on the stance leg is equal to w_{CoMan}, the weight of the CoMan (278.2 N). The ground reaction force acting on the other leg is equal to zero,
- at the estimated heel strike, the weight is progressively moved linearly from one foot to the other during a period of time corresponding to 10% of the gait cycle. This value corresponds to the duration of the double support phase in a typical gait cycle [Whi07].

This assumption is strong and it was initially planned to remove it. However, the results were exploitable enough and the assumption was finally kept.

5.1.3 Reference data set

In this Subsection, we give more details about the reference data set.

The input data $\boldsymbol{D}_{in}$ of the MLC must be composed of:

- the joints positions $\mathbf{q}$,
- the joints speeds $\dot{\mathbf{q}}$,
- the forces below the feet $\mathbf{F}_f$,
- the absolute orientation $\mathbf{R}_{IMU}$ of the waist,
- the angular speed $\boldsymbol{\omega}_{IMU}$ of the waist.

[1] In the rest of this Chapter, the results will always be displayed for the left leg. Similar results are systematically obtained for the right leg.

The positive angles for the joints described above are defined as follows: flexion for the trunk, extension for the hip, flexion for the knee and plantarflexion for the ankle, as depicted in Figure 5.3. This convention is used throughout this Chapter and the following.

The reference data $\mathbf{D}_{in}^{RDS}$ and $\boldsymbol{\tau}_{real}^{RDS}$ are taken from human data from two sources:

- Riener & Al. [RRF02] for the angle positions $\mathbf{q}$ and angle torques $\boldsymbol{\tau}_{real}$. Angle speeds $\dot{\mathbf{q}}$ are found by computing the first derivative of the angle positions.
- Krebs & Al. [KWJ+92] for the absolute orientation $\mathbf{R}_{IMU}$ of the waist. The angular speed $\boldsymbol{\omega}_{IMU}$ is found by derivation.

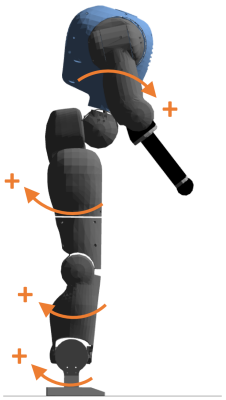


Figure 5.3: Direction of positive angles.

As explained above, the forces below the feet $\mathbf{F}_f$ are estimated using linear functions. The reference muscle stimulations $\mathbf{S}_m^{RDS}$ are found by using the inverse muscular model developed in [HVdNIR16]. Since all data found in the literature is normalized over the cycle time, we define a general cycle duration reference of 1.1 s, the mean time needed to perform two steps (one cycle). The data is reproduced during four cycles, and only the fourth cycle is analyzed.

The reference data from [RRF02] and [KWJ+92] is reproduced in Appendix B, Figure B.1 and Figure B.2. The data are available for level walking, stair ascent and stair descent.

5.2 Sanity test

A sanity test is made in order to check the proper implementation of the IDM. We proceed as follows:

1. a simulation of the CoMan walking on flat ground is run as usual (MLC in closed-loop: Figure 4.1);
2. the input and output data $\mathbf{D}_{in}$, $\boldsymbol{\tau}_{ref}$ and $\mathbf{S}_m$ are saved;
3. the initialization part is removed;
4. these data are then used as reference data $\mathbf{D}_{in}^{RDS}$, $\boldsymbol{\tau}_{real}^{RDS}$ and $\mathbf{S}_m^{RDS}$ while using the IDM (Figure 5.1);
5. the output torques and muscles stimulations are compared.

Since the data was produced by the middle-level controller during the original run (step 1 above), the output of this controller should be the same during the IDM (step 4).

A comparison between $\boldsymbol{\tau}_{ref}$ and $\boldsymbol{\tau}_{real}^{RDS}$ is shown in Figure 5.4. Figure 5.5 shows a comparison between $\boldsymbol{S}_m$ and $\boldsymbol{S}_m^{RDS}$. As expected, there is a good matching between the two signals. However, there are small differences that can be explained by the fact that the initialization phase of the real simulation was removed. The muscle states are therefore not the same at the initialization, resulting in differences at the beginning that vanish over time.

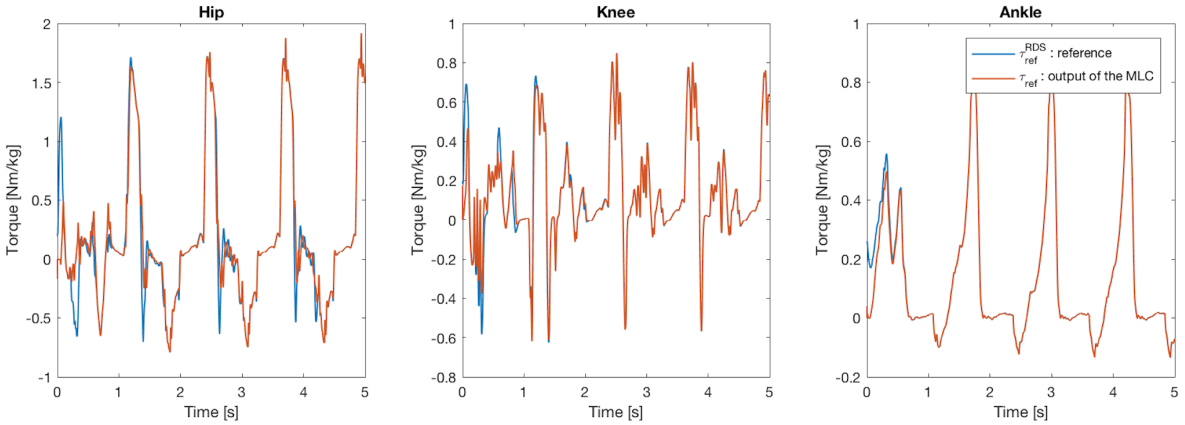


Figure 5.4: Sanity test. The output torques $\boldsymbol{\tau}_{ref}$ of the IDM are compared to the reference torques $\boldsymbol{\tau}_{real}^{RDS}$.

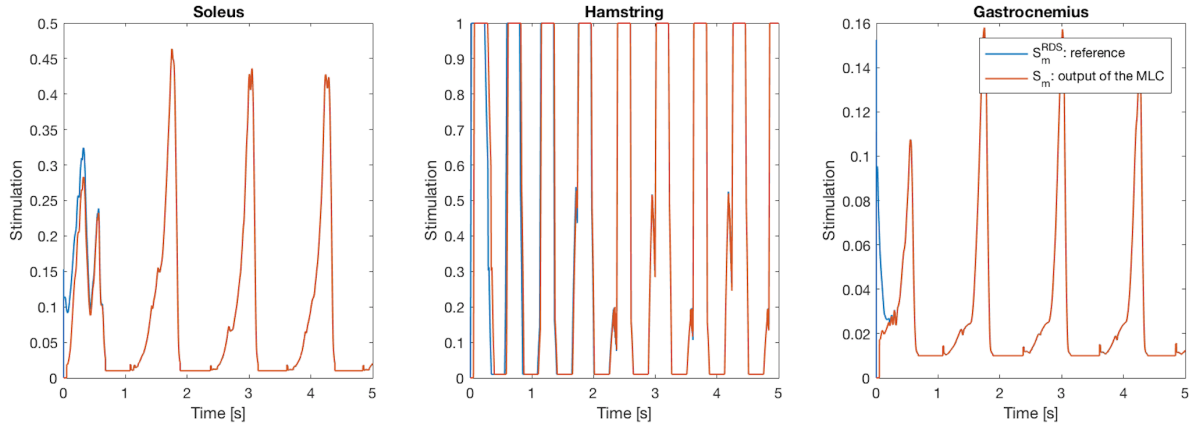


Figure 5.5: Sanity test. The output muscles stimulations $\boldsymbol{S}_m$ of the IDM are compared to the reference stimulations $\boldsymbol{S}_m^{RDS}$. Three of the nine muscles of the left leg are shown. Similar results for the other muscles.

5.3 Results

Generally one can apply this methodology to any desired kinematics. In this work, the IDM was used in particular to find new rules for stair descent when we had only the controller for flat ground at our disposal. We will show this particular application here.

We perform the IDM explained above with

- the initial controller for flat ground,
- the reference kinematics of stair descent.

A comparison between the torques is provided in Figure 5.6 and between the muscles stimulations in Figure 5.7.

Concerning the hip, it can be seen that just after foot strike (0-40 % of gait cycle) the torque produced by the current controller is too high and is directed in the wrong direction. Indeed, a negative torque is needed to dissipate energy and to slow down the robot. The stimulations of GLU and HAM should therefore be reduced to almost zero. After foot off (50-70 % of gait cycle), the current controller tries to extend quickly the hip, which is not required for stair descent: as can be seen in Appendix B, Figure B.1, the amplitude of the oscillations of the hip angle is about 40% smaller during stairs descent compared to level walking. The HFL should therefore not receive a peak of stimulation at that moment but rather a more regular stimulation throughout the gait cycle.

Concerning the knee articulation, the extension torque before toe off (50-70 % of gait cycle) is too important, and should be reduced by modulating more precisely the VAS.

The torque at the level of the ankle is too low during the stance phase (0-50 % of gait cycle). This could be done by increasing the gains for GAS and SOL.

The observations detailed here were used to design some rules of the controller presented in Chapter 4. The result is a robot able to walk on descending stairs in Chapter 7.

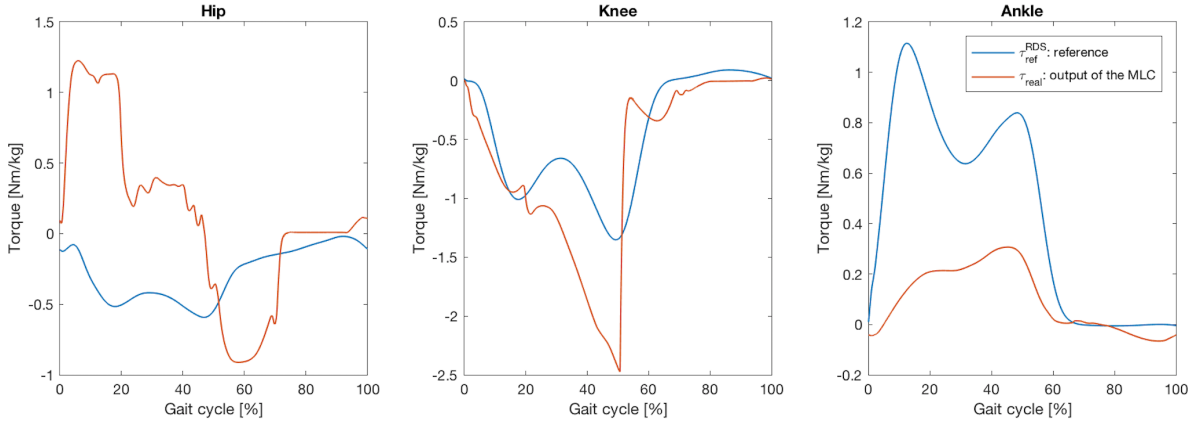


Figure 5.6: IDM with reference kinematics of stair descent and the initial controller for flat ground. Comparison between the torques at the three sagittal joints of the leg.

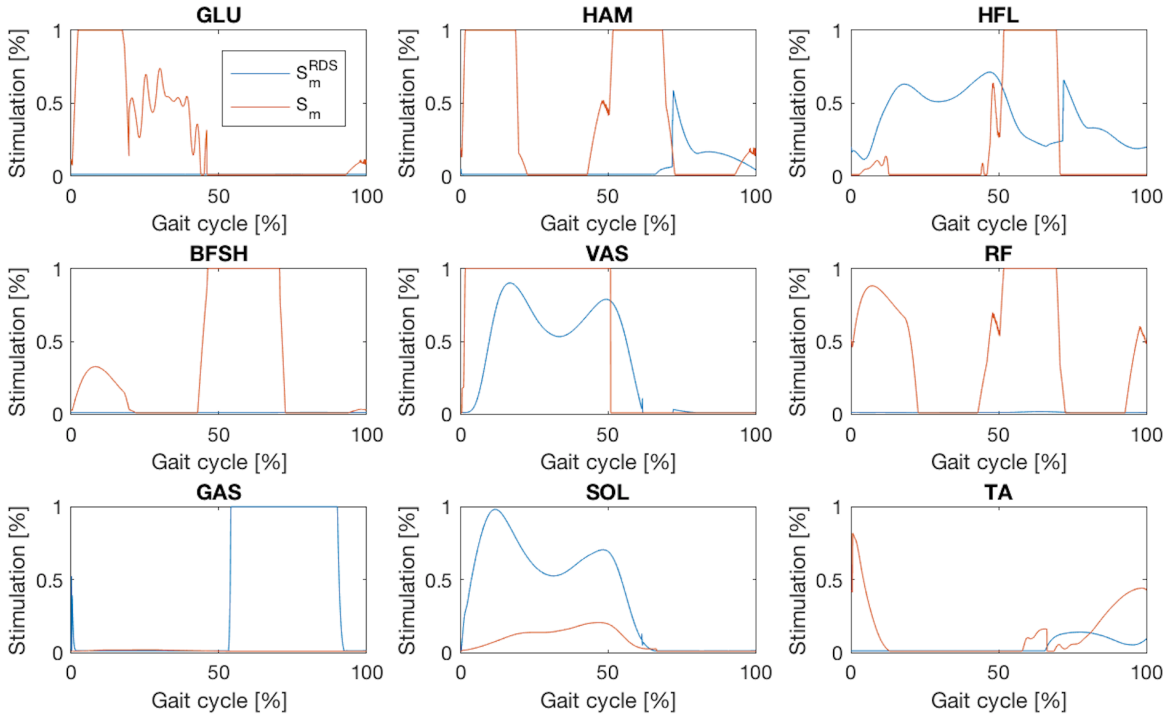


Figure 5.7: IDM with reference kinematics of stair descent and the initial controller for flat ground. Comparison between the stimulations of the nine sagittal muscles of the leg.

Chapter 6

Stair ascent

Chapter 4 introduced a controller for ascending stairs. This Chapter provides some experiments based on this controller.

In the first experiment, we find the sets of parameters such that the robot walks at its own peace on stairs. To do so, we constrain only the final step height, by using the virtual stairs introduced in Chapter 3.

Then, we evaluate the robustness of the controller against a perturbation of the rise.

In the third experiment, the muscles that deliver and absorb the most mechanical energy over one gait cycle are identified.

Then, we study the ability of the robot to perform a transition between a flat ground and an ascending stair. To make the transition possible, we use a cognitive module.

Finally, the controller having a lot of open parameters, it is interesting to find the parameters that are the most related to the value of the rise, or to the desired step length. A sensitivity analysis for each of those two aspects is thus performed.

The performances obtained in this Chapter are also presented in a video in the electronic addendum that is submitted with this report.

6.1 Natural gait

The first experiment that is carried out makes the robot able to walk on virtual stairs of four different rises (3, 4, 5 and 6 cm).

These values of rise chosen for CoMan correspond to the values of rise for an adult indicated in Table 6.1. These correspondences are found based on the general ratio between all the lengths of the CoMan and all the lengths of an adult-like model of 80 kg. This ratio is approximately 0.427 [Gre17].

The rise of 6 cm (14 cm for a human) is almost a standard size for the rise[1]. The other values of rise (3, 4, and 5 cm) where chosen to have other samples. The convergence for higher values of rise (> 6 cm) is difficult to obtain with the current controller and are therefore not reported here.

Since we choose the rise of the stair, we constrain indirectly the final step height of the robot.

[1]The rise has usually a value between 17 and 21 cm for humans.

Size for CoMan	Size for an adult-like model of 80 kg
3 cm	7.0 cm
4 cm	9.4 cm
5 cm	11.7 cm
6 cm	14.1 cm

Table 6.1: Correspondence between sizes for CoMan and sizes for a human-like model.

We decide also to constrain the metabolic energy consumption of the robot, by minimizing it.

However, we decide not to constrain the step length because in this work, we are interested by the step length to which the robot will naturally stabilizes itself for different rises. As will be seen in this Section, the step length to which the robot stabilizes naturally (without constraint) is usually higher than the going of a real staircase[2]. However, as we experimented it, constraining the step length such that it corresponds to a normal step length leads either to a very high energetic consumption (twice as expected) and/or to an unstable result. In this work, we report therefore not such results.

6.1.1 Optimization

Chapter 4 showed that a lot of parameters of the controller need to be found. The algorithm that is used to find the best parameters is the Particle Swarm Optimization, as explained in Section 3.3.

The optimization process has two big issues.

The first issue is that there is a large number of open parameters and multiple combinations of parameters can make the robot walk. The optimizer risks to get stuck in a local minimum area of the search space, leading to a non-optimal result.

The second issue is that the optimization is not deterministic because of the randomness present in the PSO algorithm. Each run of the optimization can potentially provide a different set of parameters with a different optimum.

Because of these two issues, it is required to run multiple optimizations to have a meaningful gait to analyze. However, each optimization requires a lot of time and a huge computational power. Our option was to run only a limited amount of optimizations. Most of the time, we run 5 optimizations.

During an optimization process, the design of the fitness function is fundamental. The general methodology is explained hereafter, following the methodology presented in [VdNIR18].

Fitness The project uses a fitness function that works by phases as shown in Figure 6.1, where each phase is unlocked only if the necessary performances are obtained in the previous phases. Each phase computes a fitness that ranges from 0 to 100. The fitness phases described here are used to find the parameters to allow the biped to walk directly on a stair.

The first phase is the **minimum distance phase**. It requests the robot to walk an arbitrary

[2]The going has usually a value between 21 and 27 cm for humans, corresponding to a value between 9 and 11.5 cm for the CoMan.

Figure 6.1: Phases of the fitness function. Each phase produces a value between 0 and 100. Phases represented on the same level are computed in parallel. Phases are unlocked from left to right.

minimal distance $d_{min} = 15\,\text{m}$. A reward is provided proportionally to the traveled distance as

$$Fitness = \min\left(100, 100 \cdot \frac{d}{d_{min}}\right) \tag{6.1}$$

where d is the actual distance walked. This phase makes sure the robot actually moves and does not stay in its initial upright position. The next phases are unlocked only if this fitness is equal to 100.

The second phase is the **walking time phase** that requires the robot to walk during a simulation time $T_s = 60\,\text{s}$. The fitness is proportional to the walking time as

$$Fitness = 100 \cdot \frac{t}{T_s} \tag{6.2}$$

where t is the time in seconds during which the robot walks without falling. The next phases are unlocked only if this fitness is equal to 100.

The third phase is the minimization of the **metabolic energy consumption**. This objective aims at emulating energy saving mechanisms that are likely to take place in real human walking. The metabolic energy consumption model used computes the total rate of energy consumption $\dot{E}$ as the sum of the rate at which heat is liberated $\dot{H}$ plus the rate at which positive work is done $\dot{W}$ [BPA04]

$$\dot{E} = \dot{H} + \dot{W} = (\dot{A} + \dot{M} + \dot{S} + \dot{B}) + \dot{W},$$

where the heat rate was further decomposed into muscle activation heat rate $\dot{A}$, muscle maintenance heat rate $\dot{M}$, muscle shortening heat rate $\dot{S}$ and basal metabolic rate $\dot{B}$. The energy E is found by numerical integration of the energy rate $\dot{E}$. Note that in $\dot{W}$ we compute the work done by the contractile element (CE) of the muscle, and not the total work done by the muscle

$$\dot{W} = \begin{cases} F_{CE} \cdot v_{CE} & \text{if } v_{CE} < 0 \\ 0 & \text{if } v_{CE} \geq 0 \end{cases} \tag{6.3}$$

where F_{CE} is the force and v_{CE} is the speed of the CE. The fitness function uses the bell-shaped equation described in Equation 6.4 where x is the parameter to be constrained, x^* is the reference and α is a weight. In this particular case, x is the metabolic energy E normalized per step and per unit mass of the robot (J/(step·kg)), x^* is set to 2[3], and $\alpha = 0.05$.

$$Fitness = 100 \cdot e^{-\alpha(x-x^*)^2} \tag{6.4}$$

The last phase is the minimization of the **CPG phase error**. It is computed at the same time as the third phase. We want to encourage the emergence of solutions that do not need the

[3]The value of x^* is ideally zero but we choose a value of 2 because we expect the walker to consume a minimum amount of energy.

synchronization mechanisms of the CPG. It uses also Equation 6.4 where x is the mean phase error time expressed in seconds of the CPG, x^* is zero since we want to minimize it and $\alpha = 250$.

6.1.2 Results

Given the fitness function explained above, we run 5 optimizations for 4 different stair rises: 3 cm, 4 cm, 5 cm and 6 cm. The optimizations are performed with the bounds given in Appendix C.1. We use 400 particles and 50 generations.

As a reminder, we use the virtual stairs introduced in Subsection 3.2.4: we impose the final step height of each step, but we do not constraint the step length. The gaits obtained after optimization are therefore the gaits at which the robot stabilizes itself naturally on a stair, by choosing appropriately its step length. It is the reason why we speak of natural gait.

The optimized parameters sets are summarized in Figure 6.2. In this Figure, each set is described by two characteristics of the resulting gait of the controller: the step length and the metabolic energy.

We can make the following observations.

- The optimizations for one given stair rise are concentrated in one specific cluster when the rise is small (3, 4 cm), and are more dispersed for the bigger rises (5, 6 cm).
- For one particular rise: the lower the step length, the lower the metabolic energy consumption. This relation can be captured by a linear approximation.
- The optimizer found a set of parameters for the maximal rise (6 cm) that has a lower consumption per step than any of the other parameters sets (with a lower rise).

From these observations we can conclude that the metabolic energy consumption per step is more related to the step length than to the final step height. Indeed, the potential energy (normalized with respect to the mass) needed to climb up a step of 1 cm is:

$$\Delta E = \Delta h \cdot g = 0.01 \cdot 9.81 \approx 10^{-1}[J/kg], \tag{6.5}$$

which is about 2 order of magnitude smaller than the energy of a step (even on flat ground). In the above equation, g is the gravitational term.

For each rise, we choose one parameters set (indicated by the full circles in the Figure) that will be considered as the reference in the remaining of this Chapter. We choose the parameters sets that have the lowest metabolic energy consumption, because they are more likely to generate a human-like gait, which is the objective of the use of bio-inspired mechanisms.

For illustration, snapshots of the CoMan climbing stairs of 3 cm and of 6 cm (the two extremes) are shown in Figure 6.4 and Figure 6.5. For concision, the snapshots when the CoMan climbs stairs of 4 cm and 5 cm are not illustrated here.

Moreover, the robot is also able to walk on a stair of increasing rise, as shown in Figure 6.3. We change at each foot strike the parameters set such that it corresponds successively to the parameters set of a rise of 3, 4, 5 and 6 cm.

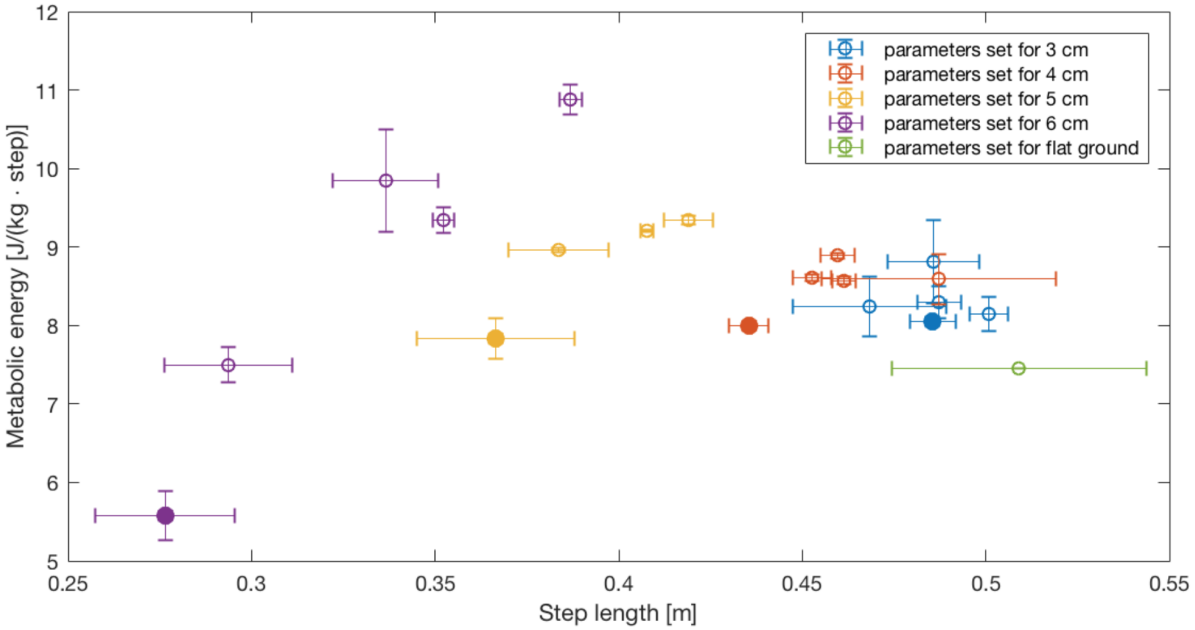


Figure 6.2: Metabolic energy consumed by the muscles of the CoMan per step and normalized by the mass of the CoMan in function of the step length for different optimized parameter sets. 5 parameters sets (from 5 different optimizations) are shown for 4 different rises (3 cm - 4 cm - 5 cm - 6 cm), represented by the different colors. We use virtual stairs, which explains that the step length is not constrained and may vary. The horizontal and vertical bars represent one standard deviation, obtained by running 5 times the same controller. For each rise, we choose 1 set of parameters indicated by the full circles.

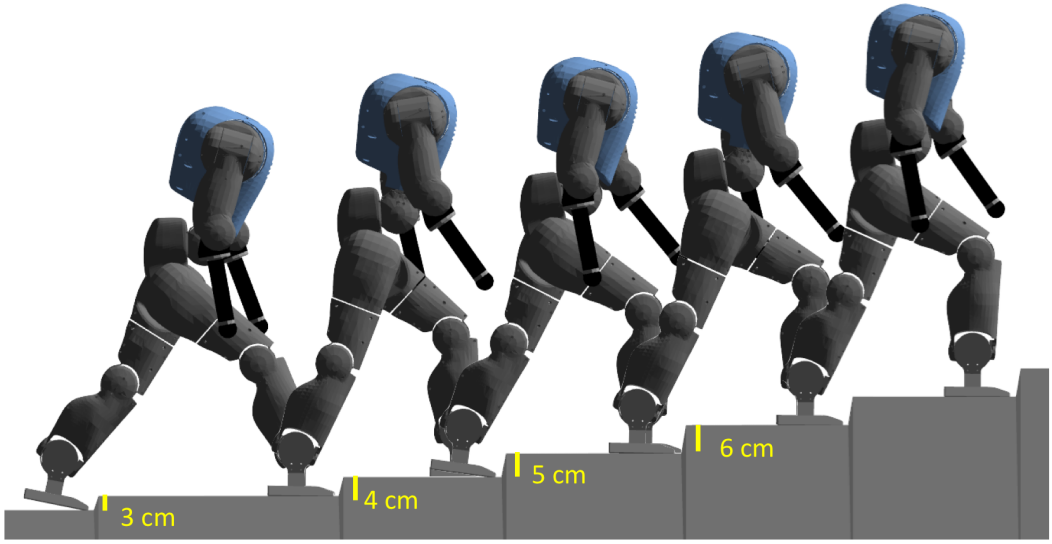
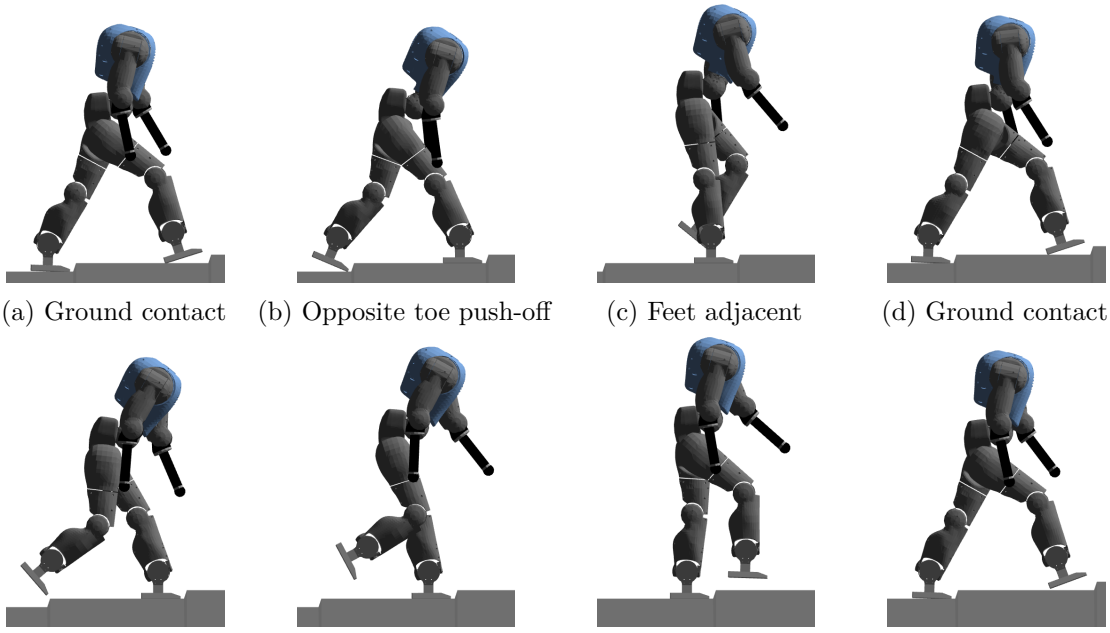


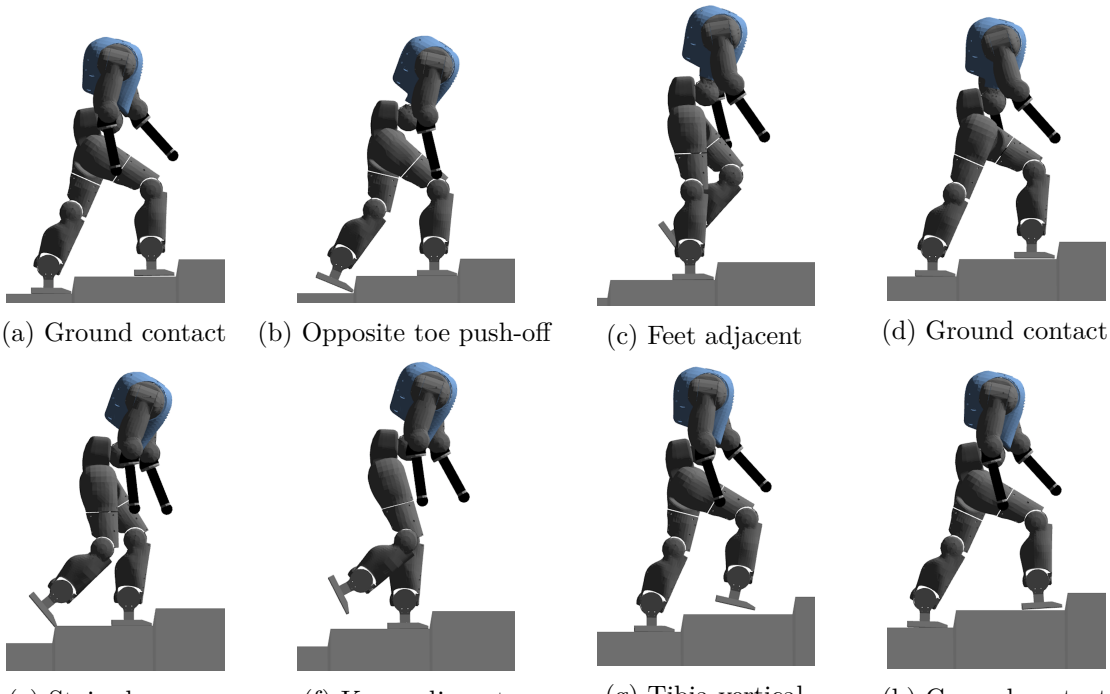
Figure 6.3: The robot is able to walk on stairs of increasing rise: 3, 4, 5 and 6 cm.



(a) Ground contact (b) Opposite toe push-off (c) Feet adjacent (d) Ground contact

(e) Stair clearance (f) Knee adjacent (g) Tibia vertical (h) Ground contact

Figure 6.4: Snapshots of different gait events occurring during one gait cycle when the CoMan climbs up stairs of 3 cm.



(a) Ground contact (b) Opposite toe push-off (c) Feet adjacent (d) Ground contact

(e) Stair clearance (f) Knee adjacent (g) Tibia vertical (h) Ground contact

Figure 6.5: Snapshots of different gait events occurring during one gait cycle when the CoMan climbs up stairs of 6 cm.

6.2 Robustness on the rise

We want to know for each parameters set if it can resist against a perturbation of the rise.

In that purpose, we change the rise when the robot is in a steady state and we evaluate if the robot is able to walk without falling on the modified stair during 4 steps, before walking again on stairs steps with the initial rise.

Figure 6.6 shows that the parameters set for an initial rise of 3 cm resists against perturbations ranging from -22.5 % to 25%, the parameters set for an initial rise of 4 cm against perturbations ranging from -10 % to 22.5%, the parameters set for 5 cm against perturbations ranging from -12.5% and 5% and the parameters set for 6 cm against perturbations ranging from -5% to 12.5%.

If we transform the expression of the perturbation ranges into absolute value of the rise, we find that the parameters set for 3 cm is robust against perturbations ranging from -0.675 cm to 0.75 cm, the parameters set for an initial rise of 4 cm against perturbations ranging from -0.4 cm to 0.9 cm, the parameters set for 5 cm against perturbations ranging from -0.625 cm to 0.25 cm and the parameters set for 6 cm against perturbations ranging from -0.3 cm to 0.75 cm.

If we do an average of these numbers, we find that the controller – any initial rise combined – is robust in average to perturbations ranging from -0.5 cm to 0.66 cm.

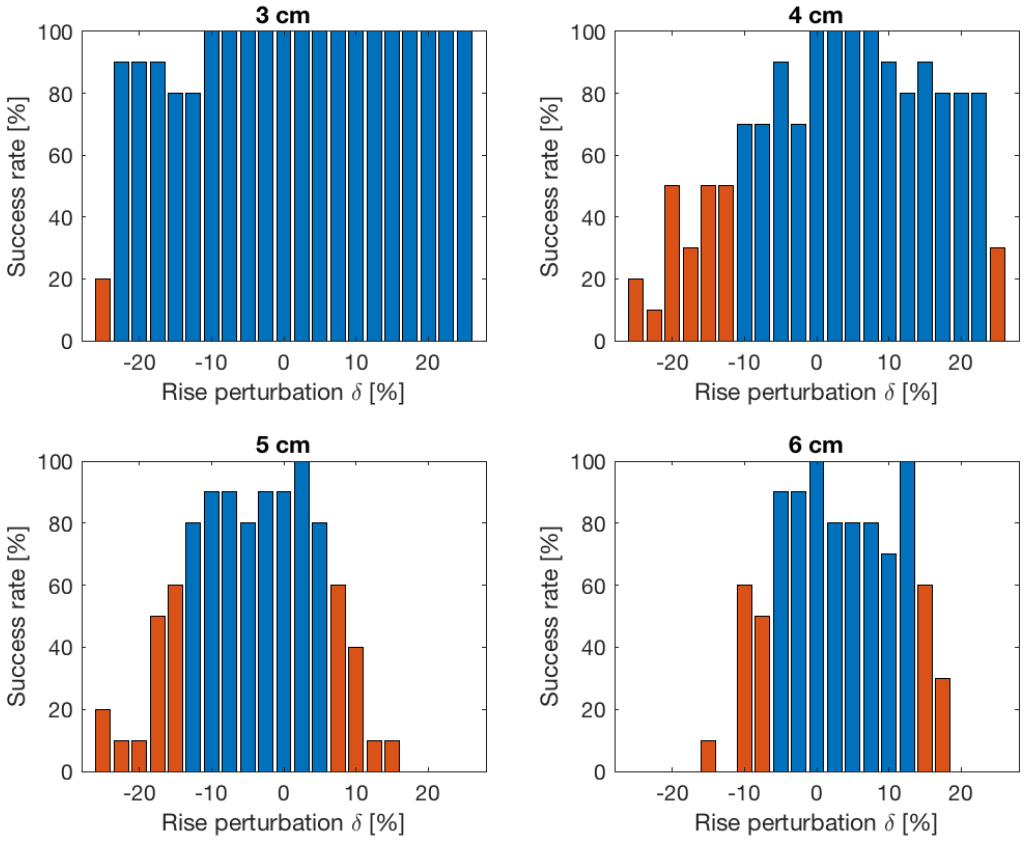


Figure 6.6: Success rates of the simulations when the rise is perturbed. The robot performs at first 8 steps on a staircase with reference rise $h = h_{ref}$ (h_{ref} corresponds to the rise for which the parameters set was optimized, namely 3, 4, 5 or 6 cm). The rise of the next 4 steps are modified: $h = (1 + \delta)h_{ref}$, where δ has a value between -25% and 25%, with a discretization step of 2.5%. The robot walks then until the end of the simulation time (30 seconds) on stairs with reference rise h_{ref}. The test is run 10 times for each reference rise h_{ref} and for each variation δ. If the robot walks until the end of the simulation without falling, then the simulation is considered as successful. If more than 7 simulations have succeeded for the same perturbation (success rate greater or equal to 70%), we consider that the controller is able to face the perturbation and the bar is colored in blue. Otherwise, we consider that the controller is not able to face the perturbation and the bar is colored in red.

6.3 Muscles work

The power delivered or absorbed by each muscle of the model during a gait cycle is represented in Figure 6.7. Table 6.2 reports the energy delivered and absorbed by each muscle during the swing and the stance phases.

We emphasize the fact that we compute the power developed or absorbed by the whole MTU (and not only the power developed by its contractile element, as it is done for the computation of $\dot{W}$ in the metabolic energy, see Subsection 6.1.1) as follows:

$$P = -F_{MTU} \cdot v_{MTU} \tag{6.6}$$

where F_{MTU} and v_{MTU} are the force and the velocity of the whole Muscle-Tendon Unit.

Concerning the power delivered (Figure 6.7 (left)), we see that the muscle VAS delivers the highest amount of energy during stance, while the HFL delivers the highest amount of energy during the swing phase.

Concerning the power absorbed (Figure 6.7 (right)), we notice that the HAM is mainly responsible for absorbing energy during stance. During swing, we notice that the HAM absorbs also a large amount of energy.

During a full gait cycle, the total mechanical energy produced by the muscles of one leg is 647.4 [mJ/(kg · gait cycle)].

Muscle	Energy delivered during stance	Energy delivered during swing	Energy absorbed during stance	Energy absorbed during swing
BFSH	11.9	41.4	-69.2	-21.2
GAS	10.7	3.5	-24.4	0
GLU	198.8	1.1	-13.1	-10.3
HAM	172.9	0.5	**-84.6**	**-163.9**
HFL	56.3	**270.4**	-57	-0.1
RF	7.1	47.7	-40	-76.1
SOL	11.7	11.3	-38.1	-2.5
TA	10.3	3.5	-5.6	-6.3
VAS	**487.5**	6.9	-75.7	-18
Totals	967.2	386.3	-407.7	-298.4
	1353.5		-706.1	
	647.4			

Table 6.2: Value of the energy absorbed and delivered by each muscle of one leg during the stance and during the swing phase when the robot ascends stairs with rise of 6 cm. The values are expressed in [mJ/(kg · gait cycle)]. For each column, the highest energies (in absolute value) are showed in bold.

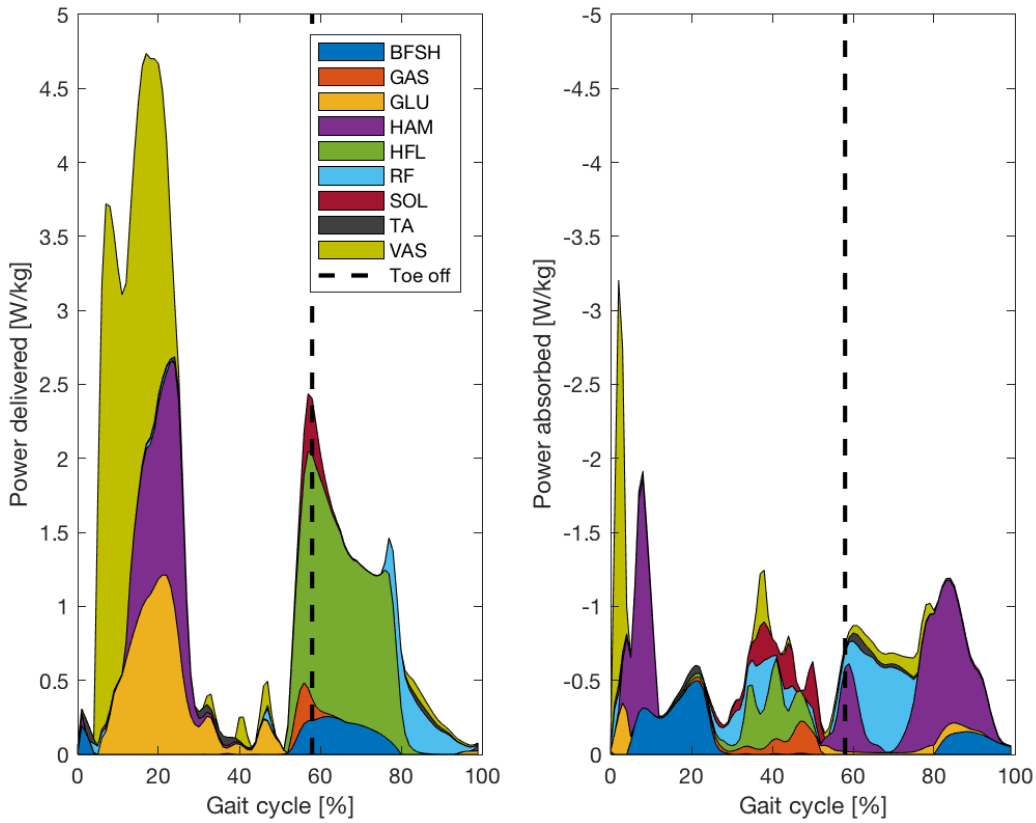


Figure 6.7: Power delivered (left) and power absorbed (right) by the different muscles over a gait cycle when the robot ascends stairs of 6 cm with the corresponding optimized parameters set. The values are stacked in top of each other, showing the relative contribution of each muscle. The values are averaged over 114 gait cycles. The vertical dotted line indicates the transition from stance to swing.

6.4 Transition with flat ground

Until now, we have a controller able to control walk in a steady-state on a flat ground (developed by [VdNIR15] and [Gre17] and called "Flat A" in this Section) and a controller able to control walk in a steady state on an ascending stair (developed in this work and called "Stairs up A" in this Section). We want the robot to be able to perform two types of transition: from flat ground to ascending stairs and from ascending stairs to flat ground.

Concerning the first transition, changing from one controller (able to walk on flat ground) to the other (able to walk on an ascending stair) at the last step of the robot on flat ground is not enough for the robot to perform the transition between the two types of ground without falling.

Two solutions are considered here. The first solution consists in constraining the optimization of the two controllers, such that the transition is possible. The second solution is to add a cognitive module only during the transition. We study these two methods hereafter.

Concerning the other type of transition, changing from one controller (able to walk on ascending stairs) to the other (able to walk on flat ground) at the last step of the robot on stairs is enough for the robot to perform the transition. No specific controller is therefore added in this case.

6.4.1 Optimization constraint

We perform an optimization as shown in Figure 6.8. During this optimization, the optimizer must find two parameters sets: one for flat ground (called parameters set "Flat B") and another one for stair ascent (called parameters set "Stairs up B"). The bounds of the parameters sets are given in Appendix C.1. Each evaluation of these parameters sets is divided into three parts:

- first, we evaluate the parameters set "Flat B": a first simulation is run (on flat ground), during which the fitness function with the usual phases is evaluated, namely: walking distance, walking time, metabolic energy and CPG phase error.
- the second part evaluates the second parameters set "Stairs up B": a second simulation is run (on ascending stairs with a rise of 5 cm), during which the same fitness function is evaluated.
- Finally, if the "walking time" phase has succeeded for the two first simulations described above (flat ground and ascending stairs), then the last part of the optimization is unlocked. 5 simulations are run during which the robot must walk 8 steps on a flat ground and the rest on an ascending stair with a rise of 5 cm until the end of the simulation. The parameters set "Flat B" is used during the 8 first steps. At the 8^{th} foot contact with the ground, we change sharply the parameters to use the parameters set "Stairs up B". For the 5 optimizations, the fitness function has only one phase: the "Walking time" phase with a simulation time of $T_s = 20$ s. Indeed, less than 10 s is generally required for the robot to perform the 8 first steps. If it walks during 20 s, it means that it succeeded the transition from the flat ground to the ascending stairs.

As a reminder, each fitness stage has a value between 0 and 100 if all necessary performances are obtained in the previous phases. The value is equal to 0 otherwise. The total fitness function is the sum of the value of all phases.

This optimization allows to find parameters sets for flat ground and for stair ascent that are able to perform the transition. However, we note that these two parameters sets lead to an over-consumption of the metabolic energy, as shown in Figure 6.9.

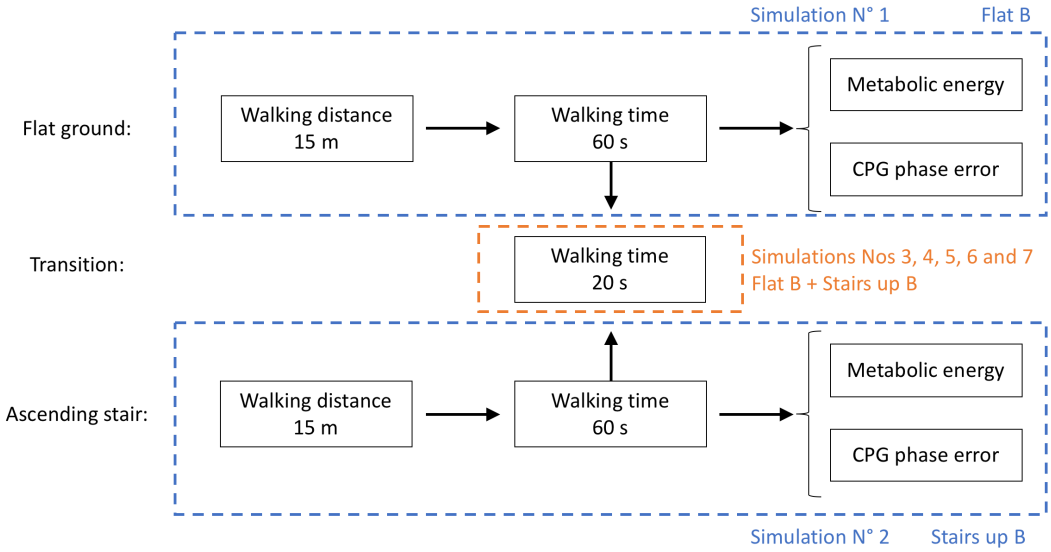


Figure 6.8: Organization of the optimization used to find parameters sets for flat ground and for stair ascent, such that they are able to perform the transition.

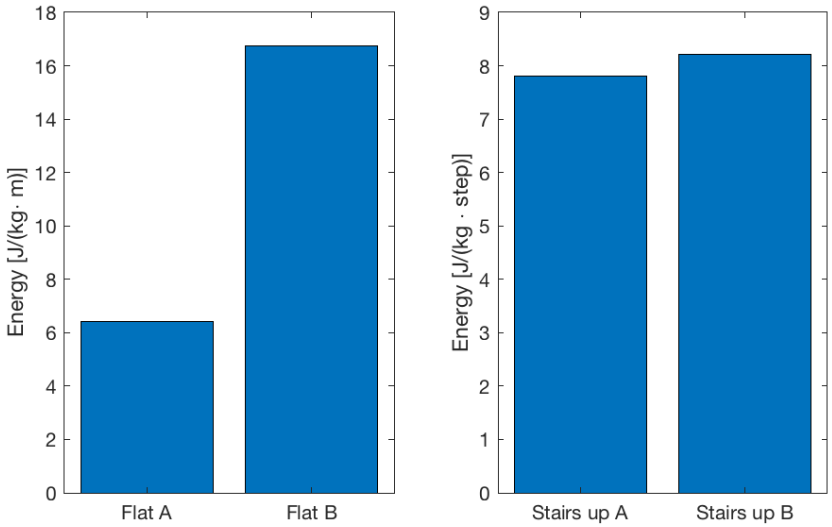


Figure 6.9: Comparison of the metabolic energy consumed by different controllers. Controller "Flat A" is optimized on flat ground only. Controller "Flat B" is optimized on flat ground, with the constraint that is must be able to perform the transition. Controller "Stairs up A" is optimized only on ascending stairs. Controller "Stairs up B" is optimized on ascending stairs, with the constraint that is must be able to perform the transition.

6.4.2 Cognitive module

To avoid the over-consumption of metabolic energy, the ideal is to use the controller optimized for flat ground only (controller "Flat A") and the controller optimized on ascending stairs only (controller "Stairs up A"), and to operate at the transition between both.

We developed therefore a module that delivers muscle stimulations that adds up to the stimulations computed by the controller on ascending stairs during the 2 first steps on the staircase. This module aims at adjusting the angular positions of the joints, allowing the robot to perform a transition without falling.

We call this module "cognitive module". Indeed, we use rhythmic patterns when walking on flat ground or on an ascending stair in a steady-state. However, during a transition, the movements are more "conscious" to position correctly the feet on stairs.

This module consists in 6 proportional controllers (one for each joint of each leg) that track reference angular positions:

$$\begin{aligned}
\Delta_{\phi_h,\{R,L\}} &= k_{p,h,tr}(\phi_{h,ref,\{R,L\}} - \phi_{h,\{R,L\}}) \\
\Delta_{\phi_k,\{R,L\}} &= k_{p,k,tr}(\phi_{k,ref,\{R,L\}} - \phi_{k,\{R,L\}}) \\
\Delta_{\phi_a,\{R,L\}} &= k_{p,a,tr}(\phi_{a,ref,\{R,L\}} - \phi_{a,\{R,L\}})
\end{aligned}$$

7 muscles allowing to correct the errors are then stimulated:

$$\begin{array}{ccc}
S_{HFL,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_h,\{R,L\}}\right]^- & S_{VAS,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_k,\{R,L\}}\right]^- & S_{TA,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_a,\{R,L\}}\right]^- \\
S_{GLU,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_h,\{R,L\}}\right]^+ & S_{HAM,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_k,\{R,L\}}\right]^+ & S_{SOL,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_a,\{R,L\}}\right]^+ \\
 & S_{GAS,\{R,L\}} \stackrel{+}{=} \left[\Delta_{\phi_k,\{R,L\}}\right]^+ &
\end{array}$$

The function $[\bullet]^+$ takes the positive part of its argument and saturates to zero if the argument is negative. The function $[\bullet]^-$ takes the absolute value of its argument if it is negative and saturates to zero otherwise.

The values of $k_{p,h,tr}$, $k_{p,k,tr}$ and $k_{p,a,tr}$ are found by optimization.

The values of the reference angles $\phi_{h,ref}$, $\phi_{k,ref}$ and $\phi_{a,ref}$ are obtained by extracting the value of these angles obtained by the previous method (Subsection 6.4.1) during the transition.

6.4.3 Results

Figures 6.10 and 6.11 show the robot performing the transition from flat ground to ascending stairs and the transition from ascending stairs to flat ground. For the first transition, we used the cognitive module. For the latter, we did not used any particular supplementary controller.

The robot is able to perform several times the transitions.

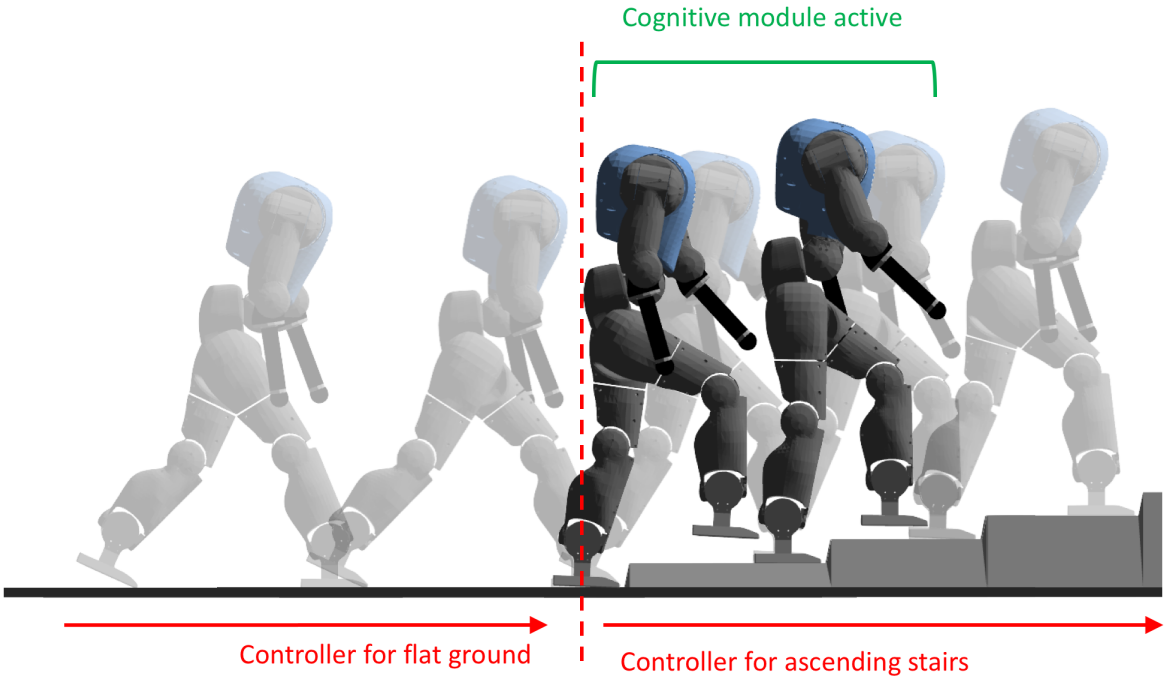


Figure 6.10: Snapshots of the CoMan during a transition from flat ground to ascending stairs. The transparent snapshots are taken at toe off. The opaque snapshots illustrate at which step the cognitive module is active.

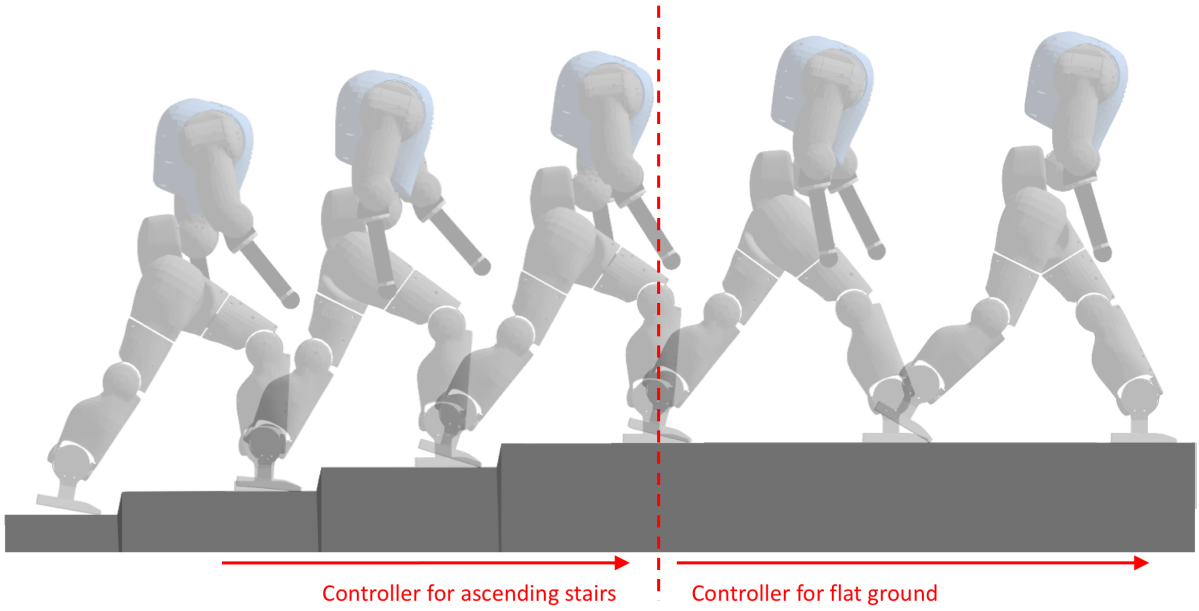


Figure 6.11: Snapshots of the CoMan during a transition from ascending stairs to flat ground. The transparent snapshots are taken at toe off.

6.5 Sensitivity analyses

In this Section, we identify the parameters of the controller that change the most while varying the rise and the step length.

These results can be used in a further work to modulate some parameters of the controller to modify the gait of the controller when the robot is on a staircase.

6.5.1 Rise

The controller presented in Chapter 4 has a total of 46 open parameters. Among these parameters, we select the parameters of the CPG that are not related to the interactions between neurons, namely the 14 parameters of modulation of the outputs $k_\bullet$ and the time constant τ. We choose also the torso lean angle θ_{ref}.

We analyze the influence of the rise on these 16 parameters as follows. An optimization is performed for a rise of 6 cm (we use the result from Section 6.1). Then, all optimized parameters are frozen, except the 16 parameters indicated above. Then, we run optimizations with a step rise varying between 5 cm and 7 cm, with a discretization step of 0.5 cm.

For each rise, we run 5 optimizations. We compute then the mean of each parameter for the 5 optimizations. Finally, we approximate the relative evolution of each parameter by the best linear fit, in the least-squares sense. The evolution of the means and the linear approximations are represented in Appendix D.1, Figures D.1 and D.2.

The slopes of the linear fits are then extracted and compared to each other in Figure 6.12. We conclude that the parameters that vary the most (i.e. the parameters for which the slope of the linear fit is greater than 10% of the slope of the parameter that vary the most) are, in order of importance, $k_{HAM,3}$, $k_{BFSH,4}$, $k_{HAM,4}$, $k_{HAM,2}$, $k_{BFSH,2}$, $k_{RF,1}$, $k_{BFSH,3}$ and $k_{GLU,2}$.

This indication shows the parameters that must be modulated in priority if we wish to modulate these parameters according to the rise value.

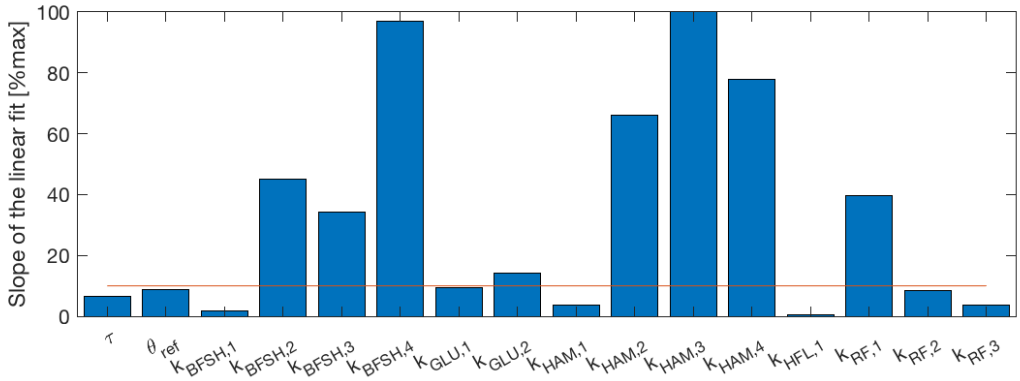


Figure 6.12: Comparison between the absolute value of the slope of the linear fits for 16 parameters of the controller. The slopes are normalized with respect to the maximal value. The parameters that have the higher slope are the parameters that are the most sensible to a change of the rise. The horizontal line corresponds to a slope of 10%.

6.5.2 Step length

A procedure similar to the one used during the sensitivity analysis on the rise is used to identify the parameters among the 16 parameters identified above that influence the most the step length.

We proceed as follows. An optimization is performed for a rise of 6 cm and for a step length of 27.5 cm. To constrain the step length, we use the fitness function depicted in Figure 6.13. This fitness function has one more phase than the original fitness function (Figure 6.1): the step length phase. This phase uses the Equation 6.4 where x is the mean step length during the simulation, x^* is the reference step length and $\alpha = 200$. This phase is considered as succeeded when the error on the step length is smaller than 3 cm in absolute value.

All optimized parameters of the controller are frozen, except the 16 parameters indicated above. Then, we run optimizations using the fitness function depicted in Figure 6.13 with a reference step length varying between 22.5 cm and 35 cm, with a discretization step of 2.5 cm.

For each step length, we run 5 optimizations. We compute then the mean of each parameter for the 5 optimizations. Finally, we approximate the relative evolution of each parameter by the best linear fit, in the least-squares sense. The evolution of the means and the linear approximations are represented in Appendix D.1, Figures D.3 and D.4.

The slopes of the linear fits are then extracted and compared to each other in Figure 6.14. We conclude that the parameters that vary the most (i.e. the parameters for which the slope of the linear fit is greater than 10% of the slope of the parameter that vary the most) are, in order of importance, $k_{HAM,3}$, $k_{GLU,2}$, $k_{HAM,2}$, $k_{HFL,1}$, $k_{HAM,4}$, $k_{RF,3}$, $k_{BFSH,3}$ and $k_{RF,2}$.

This indication allows to know the most important parameters that must be modulated in priority if we wish to modulate these parameters according to the desired step length.

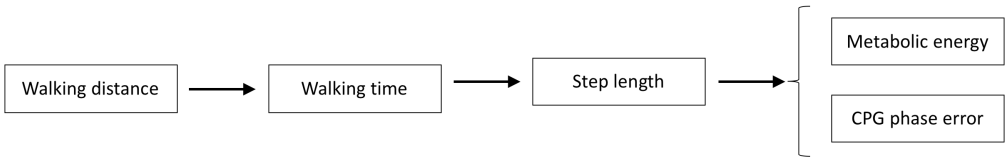


Figure 6.13: Fitness used for the optimizations constraining the step length.

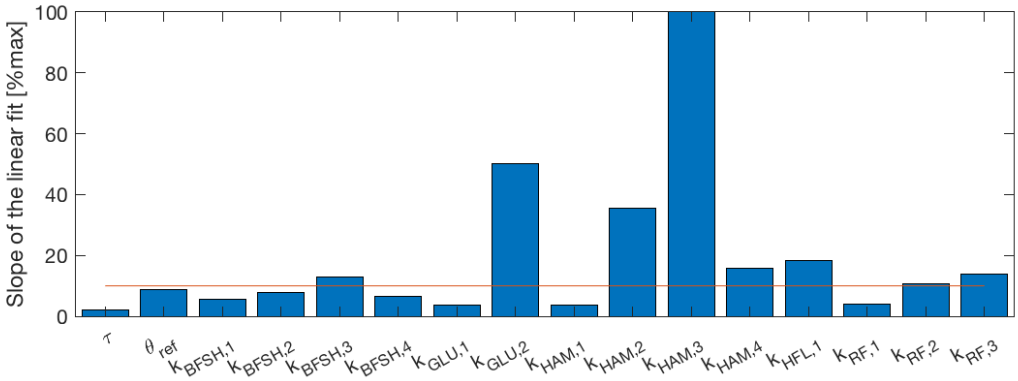


Figure 6.14: Comparison between the absolute value of the slope of the linear fits for 16 parameters of the controller. The slopes are normalized with respect to the maximal value. The parameters that have the higher slope are the parameters that are the most sensible to a change of step length. The horizontal line corresponds to a slope of 10%.

Chapter 7

Stair descent

The last Chapter studied the behavior of the controller on ascending stairs. In this Chapter, we are interested in the behavior of the controller for descending stairs, that was introduced in Chapter 4. We use the same first experiments as in Chapter 6.

First, we find the sets of parameters such that the robot walks at its own peace on descending stairs for different rises. Then, we evaluate the robustness against a change of rise. Finally, walking on stairs requires to absorb energy, we identify therefore the muscles that are responsible of this absorption.

7.1 Natural gait

The first experiment that is carried out makes the robot able to walk on virtual stairs of four different rises (-3, -4, -5 and -6 cm).

The choice of these values of rise is explained in Section 6.1. In the same Section, we explain why we choose to constrain the final step height, the metabolic energy, and not the step length.

We use the optimization process described in Subsection 6.1.1. We run 5 optimizations for 4 different rises: -3, -4, -5 and -6 cm. Note that we use the virtual stairs presented in Subsection 3.2.4. The optimizations are performed with the bounds given in Appendix C.2. We use 400 particles and 100 generations. Compared to the stair ascent, we had to perform more generations (100 instead of 50) because the convergence to an optimum is slower.

For each rise, we select among the 5 optimized parameters sets the one that consumes the minimum amount of metabolic energy. The selected parameters sets are summarized in Figure 7.1. In this Figure, each set is described by two characteristics of the resulting gait: the step length and the metabolic energy per step. For completeness, Figure D.5 in Appendix D shows all the parameters sets that were found. In particular, we see that all parameters sets are grouped almost in the same cluster.

For illustration, snapshots of the CoMan descending stairs of 3 cm and of 6 cm (the two extremes) are shown in Figure 7.3 and Figure 7.4. For concision, the snapshots when the CoMan descends stairs of 4 cm and 5 cm are not illustrated here.

Moreover, the robot is also able to walk on a stair of decreasing rise, as shown in Figure 7.2. We change at each foot strike the parameters set such that it corresponds successively to the parameters set of a rise of -4, -5 and -6 cm.

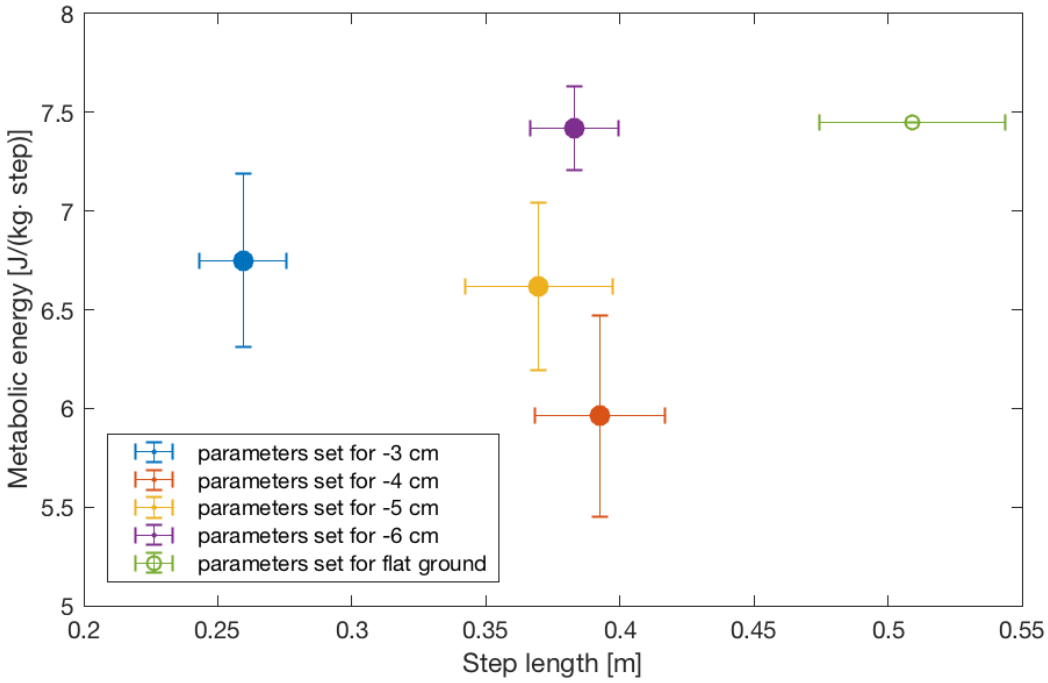


Figure 7.1: Metabolic energy consumed by the muscles of the CoMan per step and normalized by the mass of the CoMan in function of the step length for different optimized parameters sets. 5 optimizations are run for 4 different rises (-3 cm, -4 cm, -5 cm and -6 cm). For each rise, the result that minimizes the metabolic energy consumption is chosen and shown here. We use virtual stairs, which explains that the step length is not constrained and may vary. The horizontal and vertical bars represent one standard deviation, obtained by running 5 times the same controller.

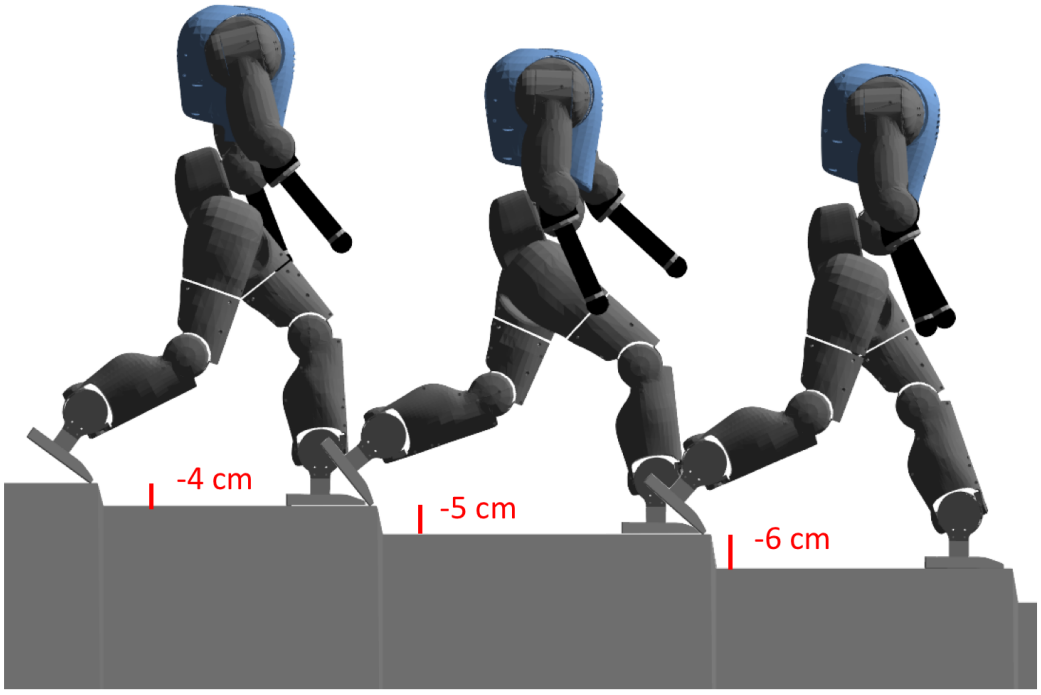
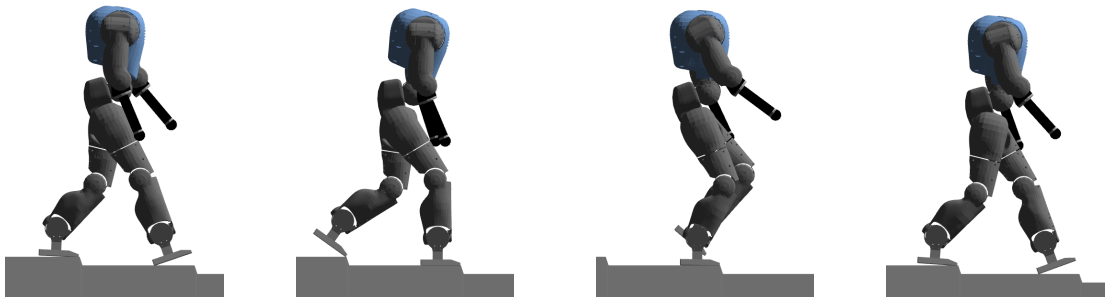
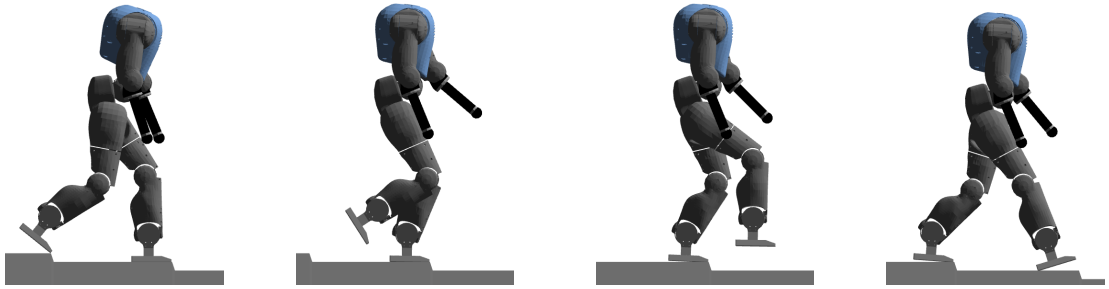


Figure 7.2: The robot is able to walk on stairs of decreasing rise: -4, -5 and -6 cm.

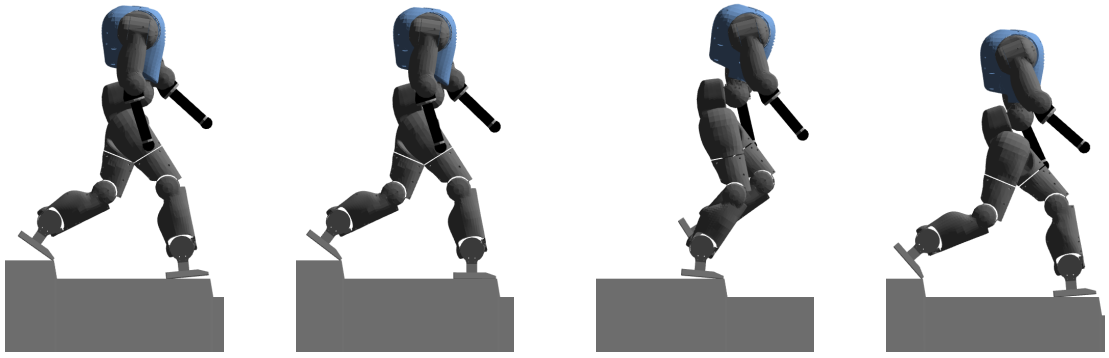


(a) Ground contact (b) Opposite toe push-off (c) Feet adjacent (d) Ground contact

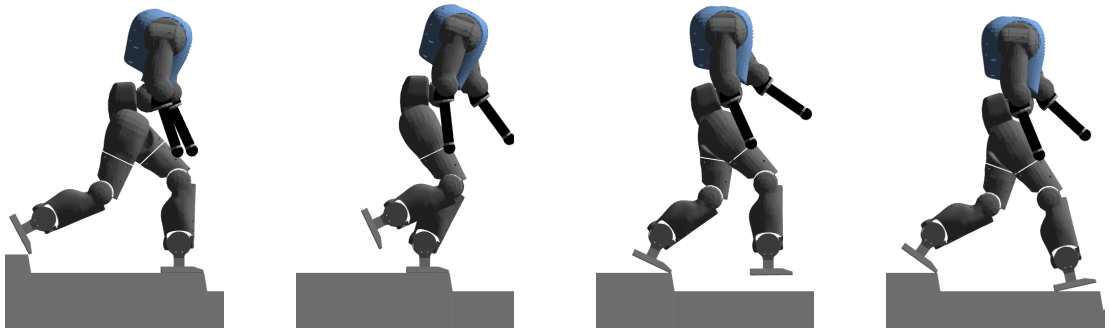


(e) Stair clearance (f) Knee adjacent (g) Tibia vertical (h) Ground contact

Figure 7.3: Snapshots of different gait events occurring during one gait cycle when the CoMan descends stairs of 3 cm.



(a) Ground contact (b) Opposite toe push-off (c) Feet adjacent (d) Ground contact



(e) Stair clearance (f) Knee adjacent (g) Tibia vertical (h) Ground contact

Figure 7.4: Snapshots of different gait events occurring during one gait cycle when the CoMan descends stairs of 6 cm.

7.2 Robustness on the rise

We want to know for each parameters set if it can resist against a perturbation of the rise.

We proceed as in the previous Chapter: we wait for the robot to be in a steady-state on the stair and we change then the rise of the following 4 steps. If the robot continues to walk during more than 7 simulations out of 10, we consider that it can handle the perturbation. The exact procedure and the results are summarized in Figure 7.6 .

This Figure shows that the parameters set for an initial rise of -3 cm resists against perturbations ranging from -12.5% to 27.5%, the parameters set for an initial rise of -4 cm against perturbations ranging from -35% to 52.5%, the parameters set for -5 cm against perturbations ranging from -47.5% to 50% and the parameters set for -6 cm against perturbations ranging from -45% to 32.5%.

If we transform the expression of the perturbation ranges into absolute value of the rise, we find that the parameters set for -3 cm is robust against perturbations ranging from -0.375 cm to 0.825 cm, the parameters set for an initial rise of -4 cm against perturbations ranging from -1.4 cm to 2.1 cm, the parameters set for -5 cm against perturbations ranging from -2.375 cm to 2.5 cm and the parameters set for -6 cm against perturbations ranging from -2.7 cm to 1.95 cm.

We conclude that the parameters sets for -3 cm has the lowest robustness. On the other hand, the parameters sets for -4, -5 and -6 show a similar robustness to perturbation, with a perturbation that can range in average from -42.5% to 45%. The fact that the parameters sets for -4, -5 and -6 cm can handle multiple centimeters of change explains probably why the parameters sets for those 3 rises can be found in a similar cluster in Figure 7.1.

Theses numbers are high compared to the robustness of the controller for stair ascent. The differences come probably from the fact that it is easier to reposition correctly a leg when descending compared to ascending, as can be seen in Figure 7.5. The hypothesis of virtual stairs plays also a role in the robustness of the controller.

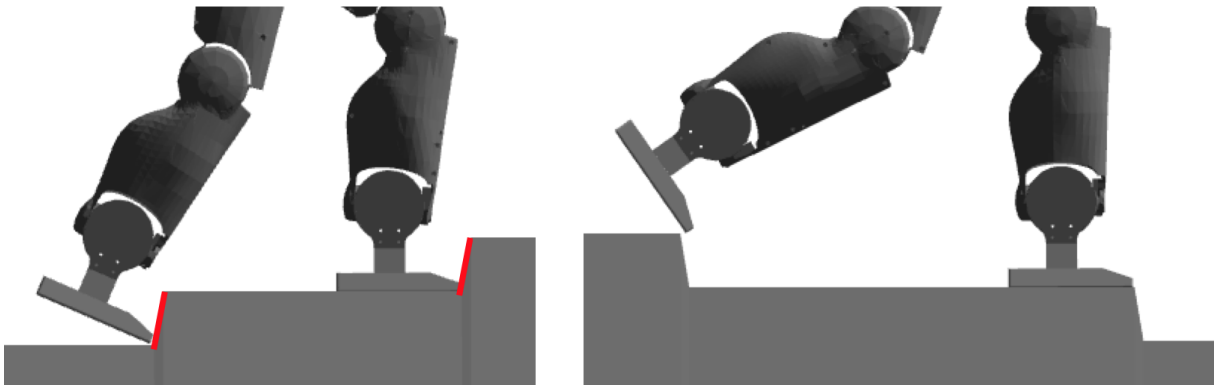


Figure 7.5: During ascent, in the case of a perturbation, the leg can touch the risers indicated in red and that are just in front of the feet, which could result in a fall. During descent, the intermediate step is the main obstacle, which can be easily avoided by a flexion of the knee.

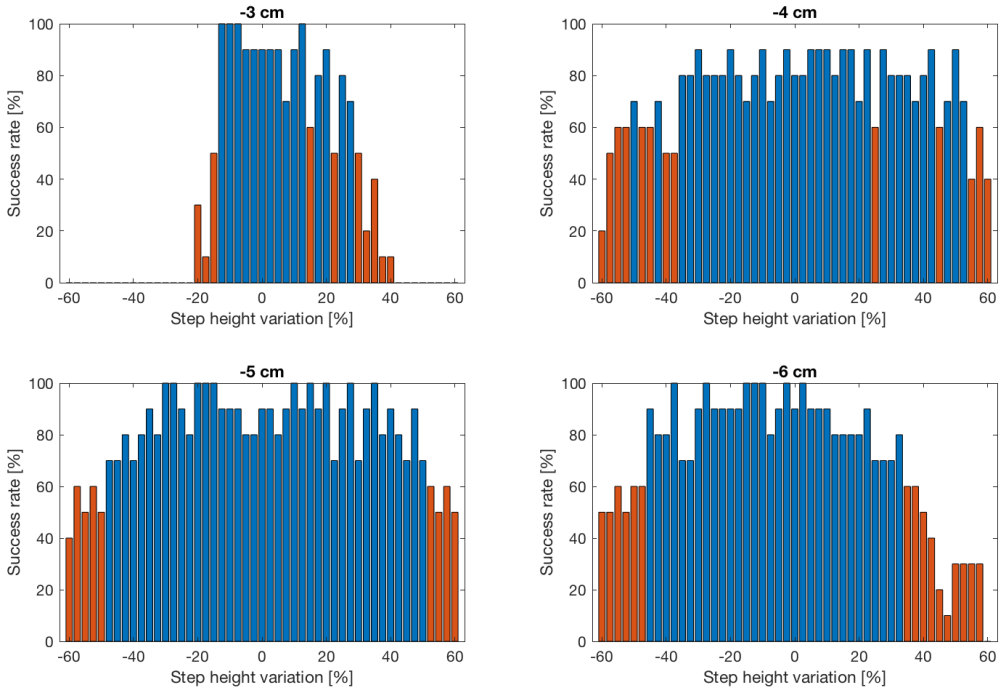


Figure 7.6: Success rates of the simulations when the rise is perturbed. The robot performs at first 8 steps on a staircase with reference rise $h = h_{ref}$ (h_{ref} corresponds to the rise for which the parameters set was optimized, namely -3, -4, -5 or -6 cm). The rise of the next 4 steps are modified: $h = (1 + \delta)h_{ref}$, where δ has a value between -60% and 60%, with a discretization step of 2.5%. The robot walks then until the end of the simulation time (30 seconds) on stairs with reference rise h_{ref}. The test is run 10 times for each reference rise h_{ref} and for each variation δ. If the robot walks until the end of the simulation without falling, then the simulation is considered as successful. If more than 7 simulations have succeeded for the same perturbation (success rate greater or equal to 70%), we consider that the controller is able to face the perturbation and the bar is colored in blue. Otherwise, we consider that the controller is not able to face the perturbation and the bar is colored in red.

7.3 Muscles work

The power delivered or absorbed by each muscle of the model during a gait cycle is represented in Figure 7.7. Table 7.1 reports the energy delivered and absorbed by each muscle during the swing and the stance phases.

As explained in Section 6.3, we emphasize the fact that we compute the power developed or absorbed by the whole MTU (and not only the power developed by its contractile element, as it is done for the computation of $\dot{W}$ in the metabolic energy).

Concerning the power delivered (Figure 7.7 (left)), we see that the muscle HAM delivers the highest amount of energy during stance, while the HFL delivers the highest amount of energy during the swing phase.

Concerning the power absorbed (Figure 7.7 (right)), we notice that the VAS is mainly responsible for absorbing energy during stance. Indeed, the knee bends slowly to allow the opposite foot to touch the next stair step. We notice that the HFL and GLU have also an important impact.

During a full gait cycle, the total mechanical energy produced by the muscles of one leg is 144.5 [mJ/(kg · gait cycle)]. As expected, this value is lower than the energy produced during stair ascent (647.4 [mJ/(kg · gait cycle)], see Subsection 6.3). A similar computation for walking on flat ground reveals that the mechanical energy developed by the muscles of one leg is equal to 348 [mJ/(kg · gait cycle)]. Again, as expected, this value is between the mechanical energy of stair descent and of stair ascent.

Interestingly, during stance, the VAS is the muscle that delivers the highest amount of energy during ascent (see Section 6.3) and the muscle that absorbs the highest amount of energy during descent. This justifies the fact that the main modification in the controller for descending stairs (compared to the original controller for flat ground) concerns this muscle (see Chapter 4).

Muscle	Energy delivered during stance	Energy delivered during swing	Energy absorbed during stance	Energy absorbed during swing
BFSH	15.8	23.2	-15.8	-1.9
GAS	1	0.1	-2.7	-0.3
GLU	81.4	44.3	-74.2	-9.4
HAM	**225.9**	0	-41.5	**-97.8**
HFL	56.3	**242.6**	-84.9	-50
RF	2.1	24.9	-17.6	-0.6
SOL	19.4	1.5	-39	-0.4
TA	10.2	0.2	-5.7	-0.8
VAS	68.5	38.5	**-240.9**	-27.9
Totals	480.6	375.3	−522.3	−189.1
	855.9		−711.4	
	144.5			

Table 7.1: Value of the energy absorbed and delivered by each muscle of one leg during the stance and during the swing phase when the robot descends stairs with rise of -6 cm. The values are expressed in [mJ/(kg · gait cycle)]. For each column, the highest energies (in absolute value) are showed in bold.

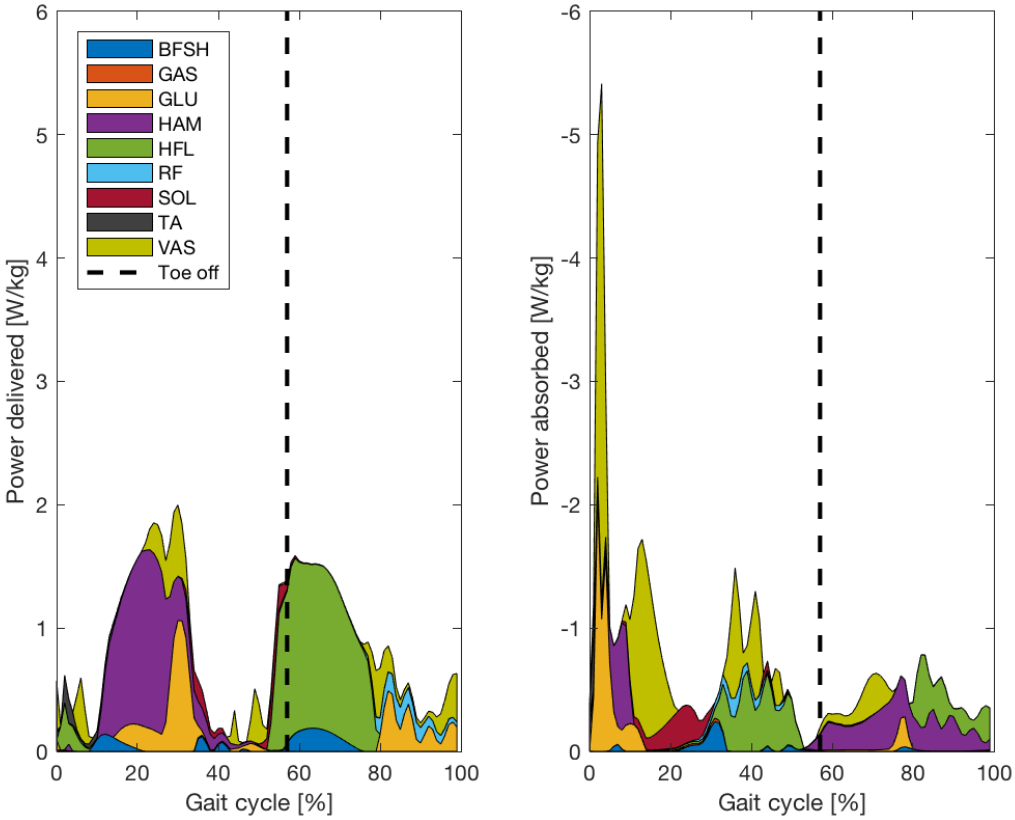


Figure 7.7: Power delivered (left) and power absorbed (right) by the different muscles over a gait cycle when the robot descends stairs of -6 cm with the corresponding optimized parameters set. The values are stacked in top of each other, showing the relative contribution of each muscle. The values are averaged over 89 gait cycles. The vertical dotted line indicates the transition from stance to swing.

Chapter 8

Comparison with human data

In this Chapter, we compare the gait obtained with the controllers for stair ascent and stair descent with human data. We consider the parameters sets of 6 cm and of -6cm, because they provide the highest final step height and the smaller step length, resulting in the highest inclination.

Even with these parameters that maximize the inclination of the stair, the resulting inclinations are 12° during ascent (rise of 6 cm and going of 28 cm) and 9° during descent (rise of -6 cm and going of 38 cm). These inclinations are small compared to the inclination of human stairs (usually between 24° and 42° [RRF02]). This is due to the fact that we did not constrain the step length, as already justified in Section 6.1. However, similar experimentations using bio-inspired mechanisms reported also a small value of inclination. For example, in [SG15], the reported stair inclination is 14°.

We will compare the muscles activations and the kinematics.

8.1 Muscles activations

The activations of the muscles during **ascent** are compared with human data in Figure 8.1.

The activations of the muscles of the ankle GAS, SOL and TA have the best fit. Interestingly, these muscles are driven by reflex rules.

The VAS and the GLU have also a rather good matching pattern, but are over-activated just after foot strike and also just before toe push off.

The muscles that have the worst matching pattern are the BFSH and the RF. We note in particular their over-activation during a major part of the gait cycle. These are the muscles that were added to the neuromuscular model at the level of the knee. The maximal force of these muscles is small compared to the other muscles. Their over-activation does therefore not change a lot the behavior of the robot.

Finally, human data is not provided for the HAM and HFL muscles and it is therefore not possible to perform a comparison.

Even if some deviations with human data are visible, the neuromuscular model predicts in general rather correctly the muscle activation profiles.

The activations of the muscles during **descent** are compared with human data in Figure 8.2.

The differences between the controller and the human data are in this case more visible. Interestingly, the GAS of the robot is not stimulated at all. This muscle is indeed mainly useful at toe off to propel the body. This function is not necessary during stair descent because the problem is precisely to dissipate energy.

Concerning **ascent and descent**, the differences are probably due to the following reasons:

- as explained above, the step length of the robot is not constrained. As a consequence, the inclination of the stairs is smaller than for humans, and a lack of fit can result from the fact the kinematics is not exactly the same;
- the robot walks in a 2D environment, and it possible that some valuable informations are missed in the orthogonal planes;
- the rules that are used in the controller to stimulate the muscles might be different from the mechanisms that take place in the human body. Also, we extended the controller for flat ground to stair ascent and descent. Some rules that come from level waking might be incorrect in the case of stairs ascent or descent.

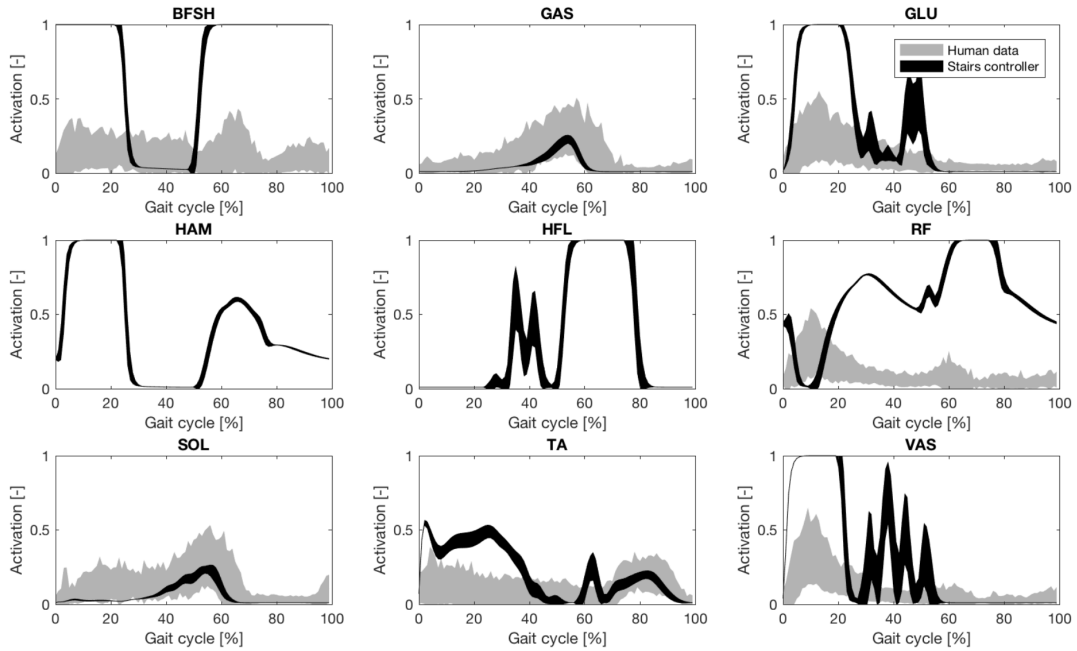


Figure 8.1: Stair ascent - Comparison between the activation profiles for the 9 considered muscles of the neuromuscular model of the controller and human data (when available). The human EMG data comes from [BRMF11] (adult, stairs ascent). The graphs represent the mean values augmented by one standard deviation for all data. For the stairs controller, the means and the standard deviations are computed over 113 gait cycles.

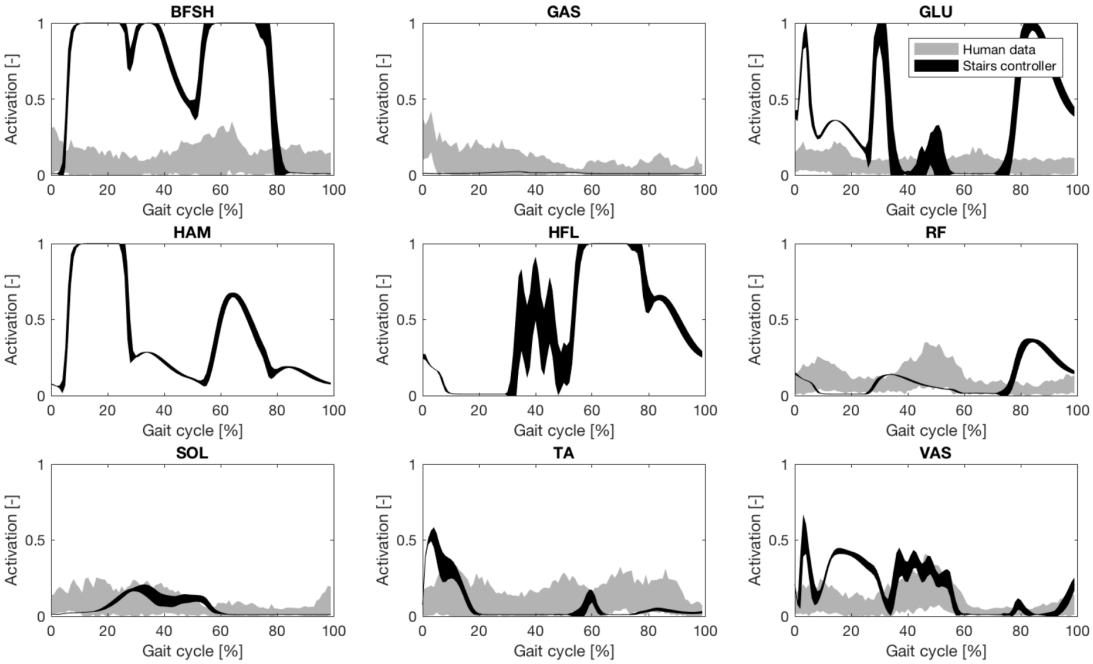


Figure 8.2: Stair descent - Comparison between the activation profiles for the 9 considered muscles of the neuromuscular model of the controller and human data (when available). The human EMG data comes from [BRMF11] (adult, stairs descent). The graphs represent the mean values augmented by one standard deviation for all data. For the stairs controller, the means and the standard deviations are computed over 89 gait cycles.

8.2 Kinematics

The comparison between the human and the robot kinematics during **ascent** is shown in Figure 8.3. To have a closer look at the results, we will analyze successively the joint angles, the torque and the power.

Concerning the joint angles, we observe a good fit for the hip and for the knee. Small differences remain, probably caused by the fact that the inclination of the staircase used during the human data acquisition (24°) is different from the inclination of the staircase on which climbs the robot(12°). Concerning the ankle, the occurrence of the maximal value of its angle in the gait cycle is delayed by approximately 10 %. Besides, the angle of this joint at ground contact is almost equal to zero in the case of the robot (see also the snapshots 6.5a, 6.5d and 6.5h: at ground contact, the tibia is vertical and the foot is horizontal). In human data, on the contrary, the ankle is in plantarflexion, causing the toes to touch the ground at first. When the rise decreases (snapshots 6.4a, 6.4d and 6.4h), the characteristics of a walk on a flat ground (in particular, the heel strikes) reappear.

Concerning the torques, at the beginning of the gait cycle (0-25%), the hip of the robot provides more torque than expected, while the knee produces less torque. We conclude that the robot uses mainly its hip to propel itself forward and to climb a step, unlike what happen in the human gait. In this latter, we see in the graphs that it is the knee that provides the necessary energy to climb the stair. For the ankle, we see a good general fit.

Finally, concerning the power of the hip and the knee we can raise the same remark: most of the work is done by the hip instead of the knee. For the ankle, except some oscillations, the general shape looks correct.

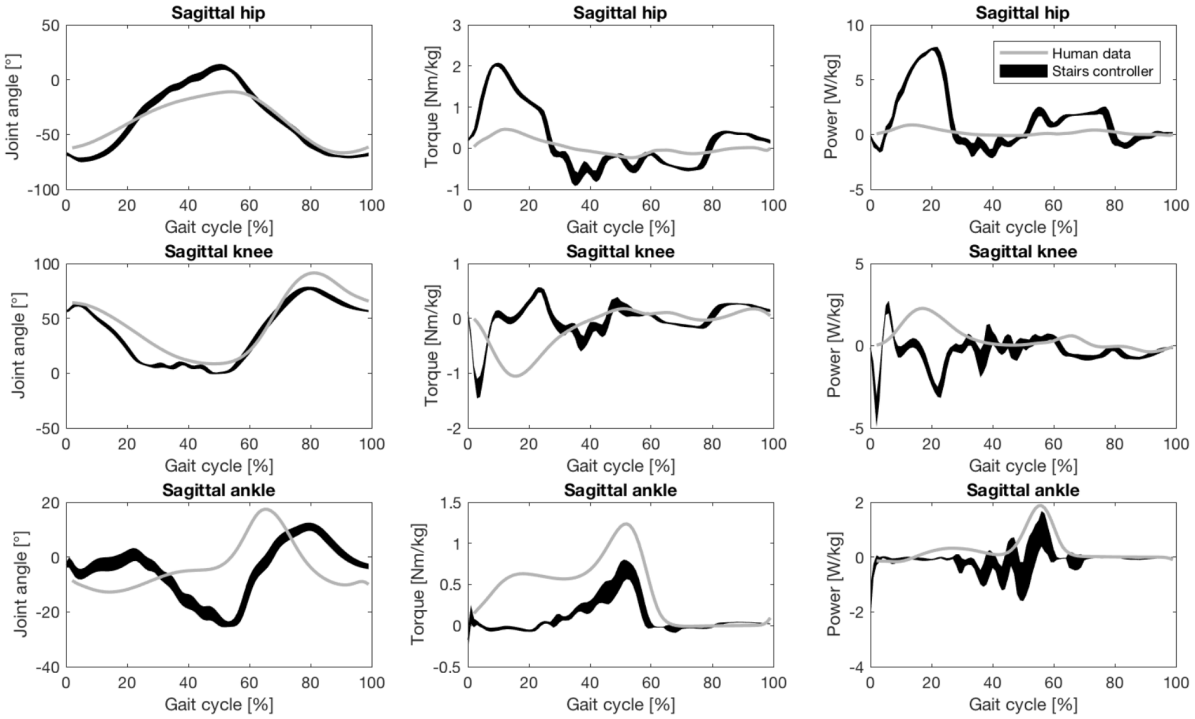


Figure 8.3: Stair ascent - Comparison of the angular position, torque and power of the three sagittal leg joints during one gait cycle. The shaded area for the Stairs controller represents one standard deviation over 113 gait cycles. The line for human data represent the mean value. Human data from [RRF02].

The comparison of the kinematics during **descent** is shown in Figure 8.4. We see good correlations between the signals but some differences exist and are detailed hereafter.

Concerning the angle of the hip, we see large variations for the robot, where there should be only small movements. Those variations explain the fact that the robot performs a large step length. We observe a good general fit for the knee and the ankle, but small differences are nevertheless visible, such as the delay corresponding to 10% of the gait cycle for the knee. The ankle should undergo a more important plantarflexion at the end of the swing phase. This plantarflexion corresponds to the fact that the toes touch the ground at first, helping to reduce energy.

Concerning the torques, we see that the torque of the hip is to high during stance. Instead of slowing the robot down and absorbing energy, it propels the robot forward, resulting in a large variation of that joint. However, during swing, we see a correct behavior of the hip. The knee produces a negative torque to absorb energy as expected. A small positive torque peak at 10% of the gait cycle is however present, before becoming negative again. For the ankle, the torque peak arises when the foot is already flat on the ground, which absorbs less energy.

Finally, for the power, we notice as already stated the high displacement of the hip due to a high power. The knee absorbs energy in particular at the beginning of the stance phase, while this behavior should be spread over this all period, and particularly at the end of the stance phase. The ankle, as already said, does not absorb as much energy as it should do just after foot contact.

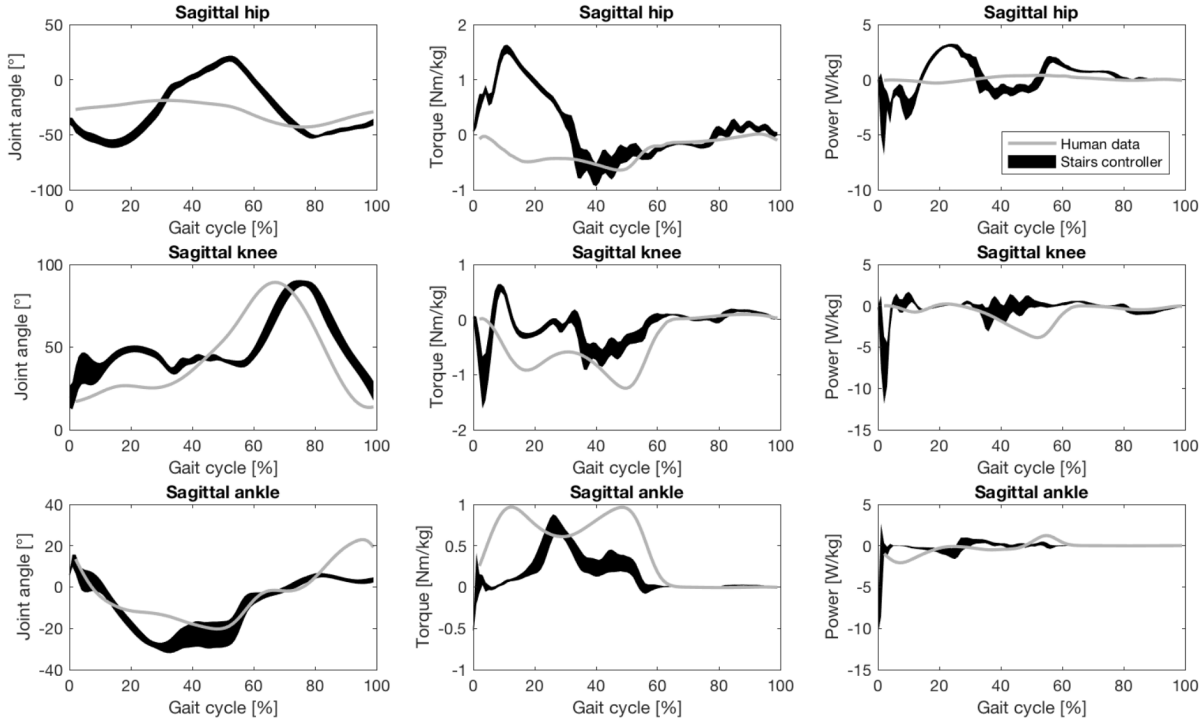


Figure 8.4: Stair descent - Comparison of the angular position, torque and power of the three sagittal leg joints during one gait cycle. The shaded area for the Stairs controller represents one standard deviation over 89 gait cycles. The line for human data represent the mean value. Human data from [RRF02].

Both during **ascent and descent**, we see encouraging results: this comparison shows that the behavior of the robot is close to the general behavior of a human. This conclusion is comforted by the observation of the snapshots of the CoMan shown in Figure 6.5 and 6.4. Moreover, the observations made above coincide with the observations made for the work done by the muscles, in Sections 6.3 and 7.3.

Chapter 9

Conclusion

The main objective of this master thesis is achieved with the design of a controller for the humanoid robot CoMan that allows it to ascend and descend stairs in simulation and in a two-dimensional environment. In this conclusion, we underline the achieved results as well as the complementary objectives that were also obtained. Finally, we propose future possible extensions emanating from the results of this thesis.

Concerning **stair ascent**, our controller enables the robot, after an optimization process, to walk on stairs with constant rise of 3, 4, 5 or 6 cm. When scaled to the size of the robot, the rise of 6 cm for the robot is almost a standard size for human rise. During the optimization process, we mainly minimized the metabolic energy consumption of the robot to emulate energy saving mechanisms that are likely to take place in real human walking. We therefore obtained the natural gait of the robot on stair, that is different for each of the four considered rises. However, these natural gaits lead to step lengths (28-48 cm for the CoMan) higher than normal step lengths (10 cm, for the CoMan). At this stage of the development of the controller, it is therefore not possible to use the controller on real stairs, with constrained step length.

The controller, as expected, shows some robustness against a perturbation of the rise. Indeed, the rise of 4 consecutive steps can be changed between -0.5 and 0.66 cm in average, without leading to a fall of the robot.

We were also able to perform the transition between a flat ground and an ascending stair, using a cognitive module. This contribution acts only during the transition, on top of the energy-optimized steady-state controllers. This module corrects the position of joints angles to track the position of reference angles. Another proposed mechanism for the transition, consisting in constraining the optimization process, leads also to successful results, but with the drawback that the gait obtained during a steady-state were less energy-efficient. Besides, the transition between an ascending stair and a flat ground is possible without any further control mechanism.

Finally, we identified thanks to two sensitivity analyses the parameters of the controller that change the most while varying the rise and the step length of the robot on stairs.

Concerning **stair descent**, it was also possible to enable the robot to walk on stairs with constant rise of -3, -4, -5 and -6 cm. To find the correct muscles stimulations for stair descent, we used in particular the inverse dynamics methodology. This tool, developed during this work, provided qualitative information to adjust the initial controller (only working for flat ground) to a target

kinematics (here, stair descent).

The controllers show a good robustness against a perturbation of the rise of 4 consecutive steps, that range between -2.7 cm and 1.95 cm in the case of the controller for rise of -6 cm.

Both for **ascent and descent**, we identified the muscles that generate and that absorb the highest amount of energy. Interestingly, we showed that, in stance, one of the muscles of the neuromuscular model that act on the knee (the VAS) delivers the highest amount of energy during stair ascent while it absorbs the highest amount of energy during stair descent, as it occurs in human walking.

When comparing the controllers to human data, we showed that they predict rather correctly the muscle activation profiles and the kinematics, comforting the bio-inspired foundation of the controllers. The controllers show nevertheless some deviations, mainly due to the high step lengths already stated. Replicating more exactly the human behavior is a complex task that will require further researches.

In this work, we decided at first not to control the step length of the robot, because doing so leads to a high energetic consumption and/or to an unstable result and also because at first we did not know which of the numerous parameters of the model needed to be modulated. Therefore, a first **perspective** would be to control the step length. The sensitivity analysis provided in this work could help to target the modulation of the most useful parameters. If the step length can be accurately controlled, then it would be possible to remove the hypothesis of virtual stairs used during this work. The robot would therefore climb real stairs (i.e. without simplifying assumption).

Another perspective is to perform the same experiments in a 3-dimensional environment. A challenge here is to make sure that the robot does not fall laterally.

A last perspective is to test the controller on the real robot, to validate the results experimentally, with the condition that the robot must first be able to modulate its step length.

Being able to increase the humanoid robots locomotor capabilities is a great challenge. Making them able to deal with some known perturbations of the ground will widen their range of possible actions, specifically in our highly unpredictable human environment. With a wider working environment, it might be expected that humanoid robots could help humans in a lot of new situations.

Appendix A

Equations of the CPG

A.1 Neuron firing rates

$$
\begin{aligned}
\dot{x}_1 &= \frac{1}{\tau}\left(-x_1 - \beta_a v_1 - \eta_a[x_2]^+ - \eta_f[x_3]^+ - \eta_g[x_4]^+ + [u_1]^+\right)\\
\dot{x}_2 &= \frac{1}{\tau}\left(-x_2 - \beta_a v_2 - \eta_a[x_1]^+ - \eta_g[x_3]^+ - \eta_f[x_4]^+ + [u_2]^+\right)\\
\dot{x}_3 &= \frac{1}{\tau}\left(-x_3 - \beta_b v_3 - \eta_f[x_1]^+ - \eta_g[x_2]^+ - \eta_b[x_4]^+ + [u_3]^+\right)\\
\dot{x}_4 &= \frac{1}{\tau}\left(-x_4 - \beta_b v_4 - \eta_g[x_1]^+ - \eta_f[x_2]^+ - \eta_b[x_3]^+ + [u_4]^+\right)\\
\dot{x}_A &= \frac{1}{\tau}\left(-x_A - \beta_a v_A - \eta_f[x_3]^+ - \eta_g[x_4]^+ - \eta_a[x_B]^+ + [u_A]^+\right)\\
\dot{x}_B &= \frac{1}{\tau}\left(-x_B - \beta_a v_B - \eta_g[x_3]^+ - \eta_f[x_4]^+ - \eta_a[x_A]^+ + [u_B]^+\right)\\
\dot{x}_C &= \frac{1}{\tau}\left(-x_C - \beta_c v_C - \eta_h[x_3]^+ - \eta_i[x_4]^+ - \eta_c[x_D]^+ + [u_C]^+\right)\\
\dot{x}_D &= \frac{1}{\tau}\left(-x_D - \beta_c v_D - \eta_i[x_3]^+ - \eta_h[x_4]^+ - \eta_c[x_C]^+ + [u_D]^+\right)\\
\dot{x}_E &= \frac{1}{\tau}\left(-x_E - \beta_d v_E - \eta_j[x_1]^+ - \eta_k[x_2]^+ - \eta_d[x_F]^+ + [u_E]^+\right)\\
\dot{x}_F &= \frac{1}{\tau}\left(-x_F - \beta_d v_F - \eta_k[x_1]^+ - \eta_j[x_2]^+ - \eta_d[x_E]^+ + [u_F]^+\right)
\end{aligned}
$$

A.2 Neurons self-inhibition

$$\dot{v}_1 = \frac{1}{\gamma_a \tau}\left(-v_1 + [x_1]^+\right) \qquad \dot{v}_B = \frac{1}{\gamma_a \tau}\left(-v_B + [x_B]^+\right)$$
$$\dot{v}_2 = \frac{1}{\gamma_a \tau}\left(-v_2 + [x_2]^+\right) \qquad \dot{v}_C = \frac{1}{\gamma_c \tau}\left(-v_C + [x_C]^+\right)$$
$$\dot{v}_3 = \frac{1}{\gamma_b \tau}\left(-v_3 + [x_3]^+\right) \qquad \dot{v}_D = \frac{1}{\gamma_c \tau}\left(-v_D + [x_D]^+\right)$$
$$\dot{v}_4 = \frac{1}{\gamma_b \tau}\left(-v_4 + [x_4]^+\right) \qquad \dot{v}_E = \frac{1}{\gamma_d \tau}\left(-v_E + [x_E]^+\right)$$
$$\dot{v}_A = \frac{1}{\gamma_a \tau}\left(-v_A + [x_A]^+\right) \qquad \dot{v}_F = \frac{1}{\gamma_d \tau}\left(-v_F + [x_F]^+\right)$$

Appendix B

Complements for the inverse dynamics methodology

The most useful data from the literature for the inverse dynamics methodology is reproduced here.

B.1 Reference leg joints angles

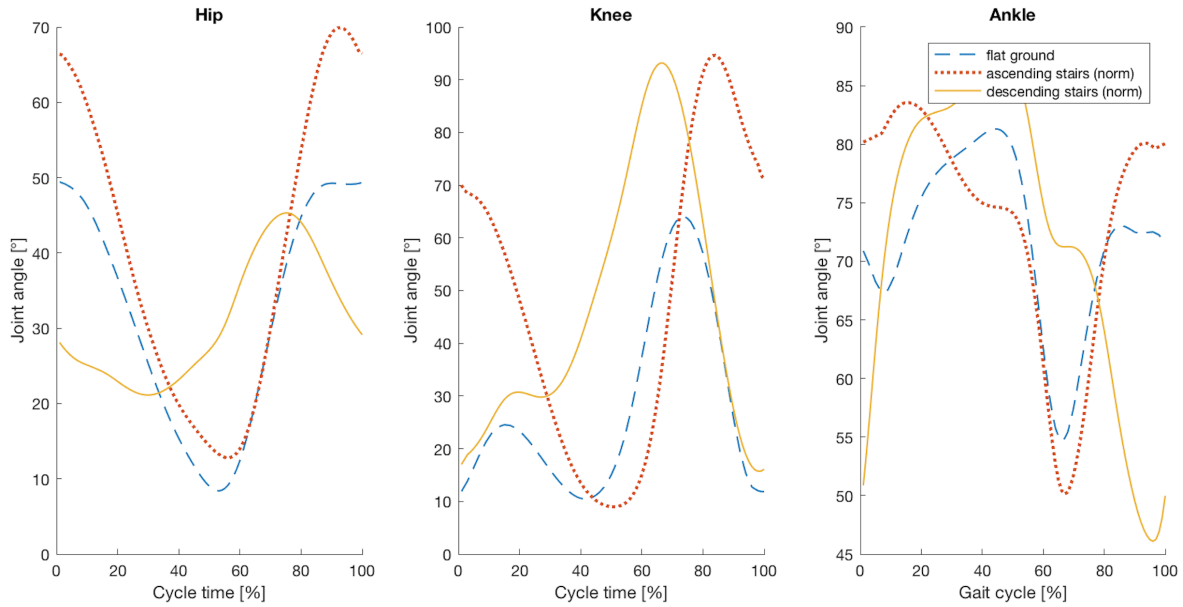


Figure B.1: Angular position for the sagittal joints of the leg over a cycle. The cycle starts with foot contact. Data from [RRF02]. In this reference, the data for ascending and descending stairs is given for stairs with a minimum (24°), normal (30°) and maximum (42°) inclination. Only the normal inclination is used here.

B.2 Reference trunk angle

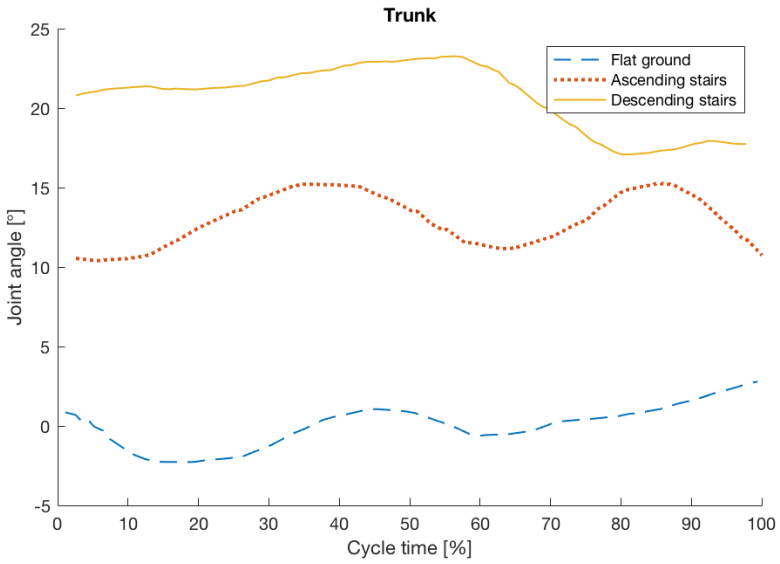


Figure B.2: Trunk angle expressed in the room coordinates. The cycle starts with foot contact. Data from [KWJ+92].

Appendix C

Bounds for the optimizations

C.1 Bounds for flat ground and for ascending stairs

Reflex rules	min	max	CPG modulation	min	max
G_{GAS}	0.4	0.8	$k_{BFSH,1}$	3.0	12.0
G_{SOL}	0.6	0.9	$k_{BFSH,2}$	0.0	0.5
$G_{SOL,TA}$	0.3	1.0	$k_{BFSH,3}$	0.0	12.0
$G_{TA,st}$	1.5	5.5	$k_{BFSH,4}$	0.0	12.0
$G_{TA,sw}$	1.5	6.0	$k_{GLU,1}$	1.3	5.1
G_{VAS}	30.0	45.0	$k_{GLU,2}$	0.0	3.3
$l_{TA,st}$	0.6	0.75	$k_{HAM,1}$	3.0	8.0
$l_{TA,sw}$	0.8	1.0	$k_{HAM,2}$	1.2	4.2
$\phi_{k,th}$	0.0	0.35	$k_{HAM,3}$	0.0	5.0
$k_{p,\theta}$	8.0	14.0	$k_{HAM,4}$	0.0	5.0
$k_{d,\theta}$	0.2	0.8	$k_{HFL,1}$	4.2	8.0
			$k_{RF,1}$	0.8	2.8
			$k_{RF,2}$	2.5	12.0
			$k_{RF,3}$	1.7	2.7
			θ_{ref}	0.17	0.21

CPG parameters	min	max	CPG parameters	min	max
β_a	4.0	8.0	η_a	4.5	6.0
β_b	2.0	4.5	η_b	4.5	7.0
β_c	2.0	5.0	η_c	2.5	4.5
β_d	3.0	4.5	η_d	4.0	6.5
γ_a	4.0	6.0	η_e	2.0	3.0
γ_b	2.0	3.5	η_f	3.0	4.0
γ_c	1.0	5.5	η_g	4.0	5.0
γ_d	2.5	3.5	η_h	3.7	4.8
τ	0.08	0.15	η_i	2.5	3.5
			η_j	1.0	4.0
			η_k	0.0	4.0

C.2 Bounds for descending stairs

Reflex rules	min	max	CPG modulation	min	max
G_{GAS}	0.4	0.8	$k_{BFSH,1}$	3.0	12.0
G_{SOL}	0.4	0.9	$k_{BFSH,2}$	0.0	0.5
$G_{SOL,TA}$	0.4	1.1	$k_{BFSH,3}$	0.0	12.0
$G_{TA,st}$	1.5	5.5	$k_{BFSH,4}$	0.0	12.0
$G_{TA,sw}$	1.5	6.0	$k_{GLU,1}$	0.0	1.0
$l_{TA,st}$	0.6	0.75	$k_{GLU,2}$	0.0	3.3
$l_{TA,sw}$	0.8	0.9	$k_{HAM,1}$	3.0	8.0
$\phi_{k,ref}$	0.7	1.0	$k_{HAM,2}$	1.2	4.2
$k_{p,\theta}$	10.0	16.0	$k_{HAM,3}$	0.0	3.0
$k_{d,\theta}$	0.2	0.8	$k_{HAM,4}$	0.0	5.0
$G_{VAS,HAM}$	1.0	5.0	$k_{HFL,1}$	0.0	5.0
$K_{VAS,sw}$	0.4	0.8	$k_{HFL,2}$	0.0	3.0
k_{c,ϕ_k}	0.1	0.4	$k_{RF,1}$	0.0	1.0
k_{p,ϕ_k}	0.5	1.5	$k_{RF,2}$	0.0	1.0
k_{d,ϕ_k}	0.0	0.2	$k_{RF,3}$	0.0	1.0
$\phi_{h,th}$	-1.3	-0.9	θ_{ref}	0.03	0.15

CPG parameters	min	max	CPG parameters	min	max
β_a	4.0	8.0	η_a	4.5	6.0
β_b	2.0	4.5	η_b	4.5	7.0
β_c	2.0	5.0	η_c	2.5	4.5
β_d	3.0	4.5	η_d	4.0	6.5
γ_a	4.0	6.0	η_e	2.0	3.0
γ_b	2.0	3.5	η_f	3.0	4.0
γ_c	1.0	5.5	η_g	4.0	5.0
γ_d	2.5	3.5	η_h	3.7	4.8
τ	0.08	0.15	η_i	2.5	3.5
			η_j	1.0	4.0
			η_k	0.0	4.0

Appendix D

Complements

D.1 Sensitivity on rise

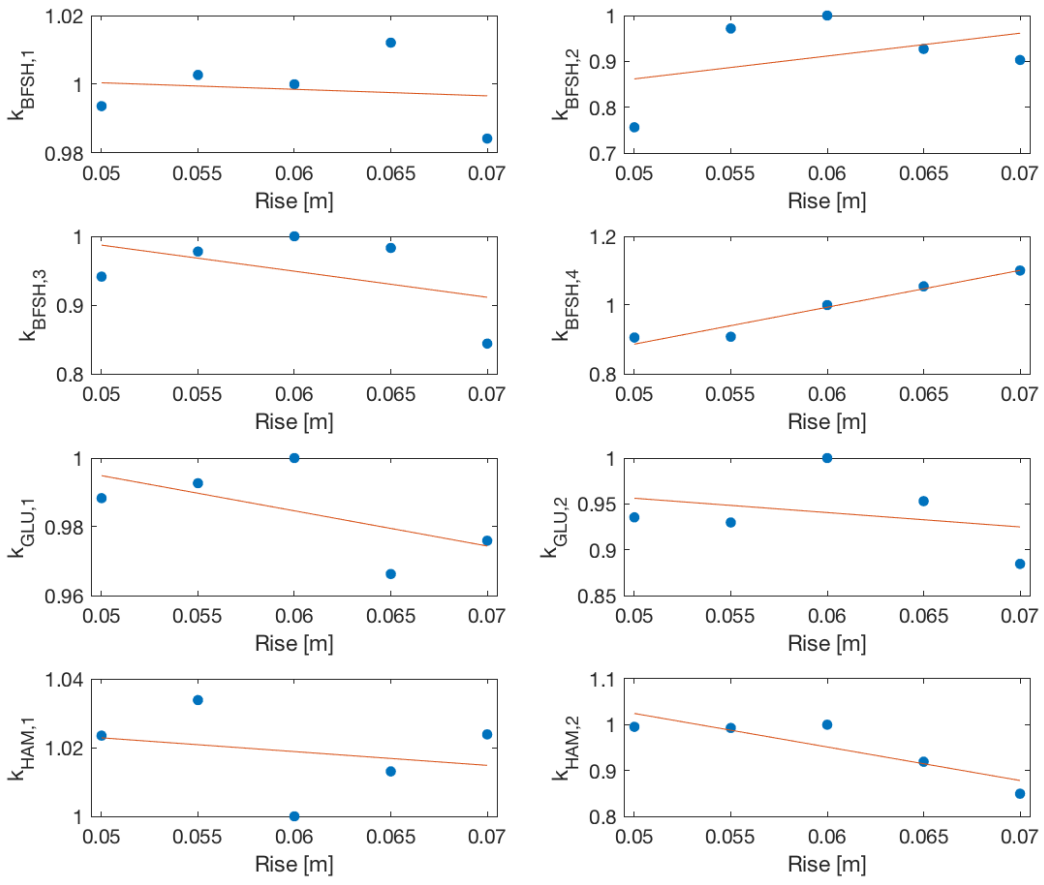


Figure D.1: Mean values of the parameters $k_{BFSH,1}$, $k_{BFSH,2}$, $k_{BFSH,3}$, $k_{BFSH,4}$, $k_{GLU,1}$, $k_{GLU,2}$, $k_{HAM,1}$ and $k_{HAM,2}$ based on 5 optimizations for 5 different rises. The mean values are expressed in relative variation compared to the rise of 6 cm. These values are then approximated by the best linear fit, in the least-squares sense.

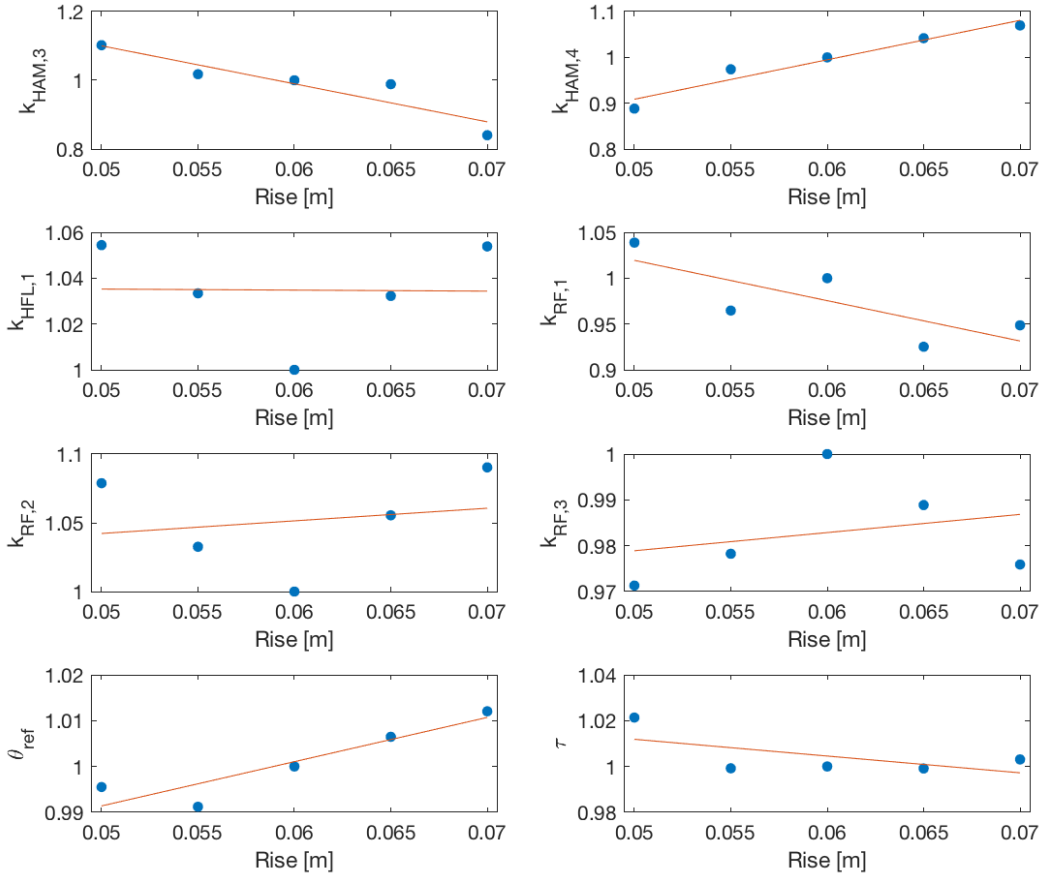


Figure D.2: Mean values of the parameters $k_{HAM,3}$, $k_{HAM,4}$, $k_{HFL,1}$, $k_{RF,1}$, $k_{RF,2}$, $k_{RF,3}$, θ_{ref} and τ based on 5 optimizations for 5 different rises. The mean values are expressed in relative variation compared to the rise of 6 cm. These values are then approximated by the best linear fit, in the least-squares sense.

D.2 Sensitivity on step length

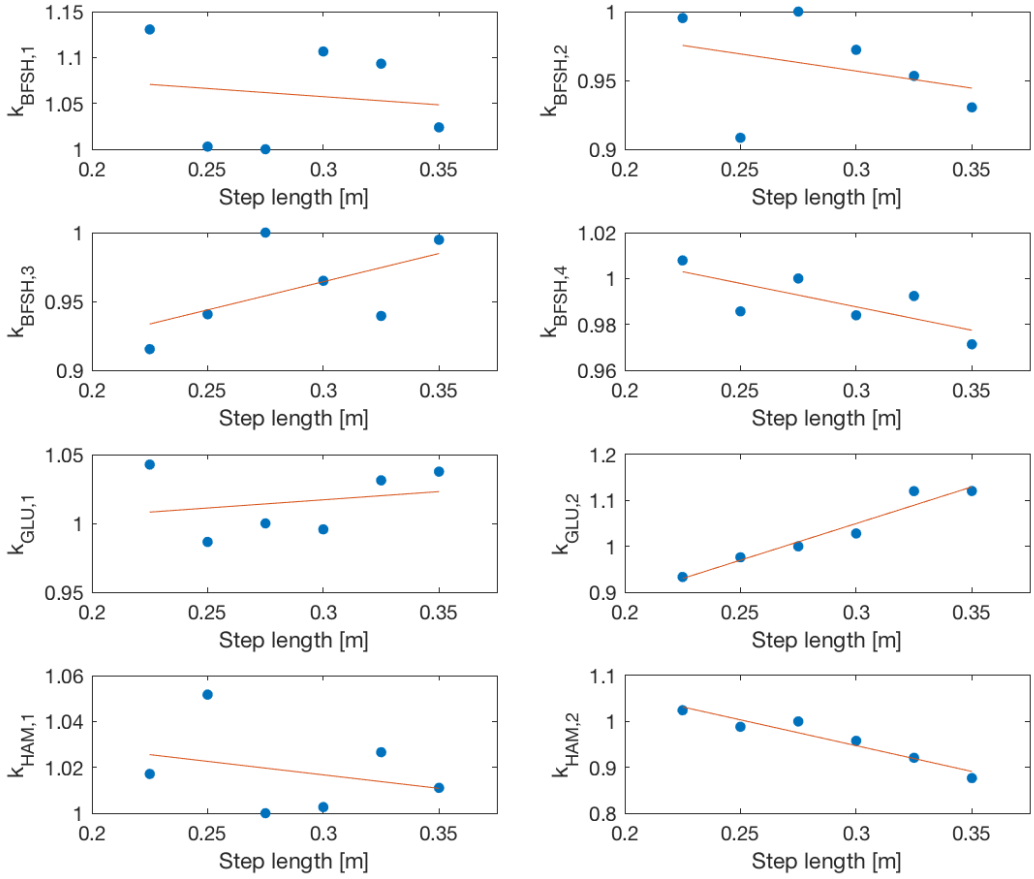


Figure D.3: Mean values of the parameters $k_{BFSH,1}$, $k_{BFSH,2}$, $k_{BFSH,3}$, $k_{BFSH,4}$, $k_{GLU,1}$, $k_{GLU,2}$, $k_{HAM,1}$ and $k_{HAM,2}$ based on 5 optimizations for 6 different step lengths. The mean values are expressed in relative variation compared to the step length of 27.5 cm. These values are then approximated by the best linear fit, in the least-squares sense.

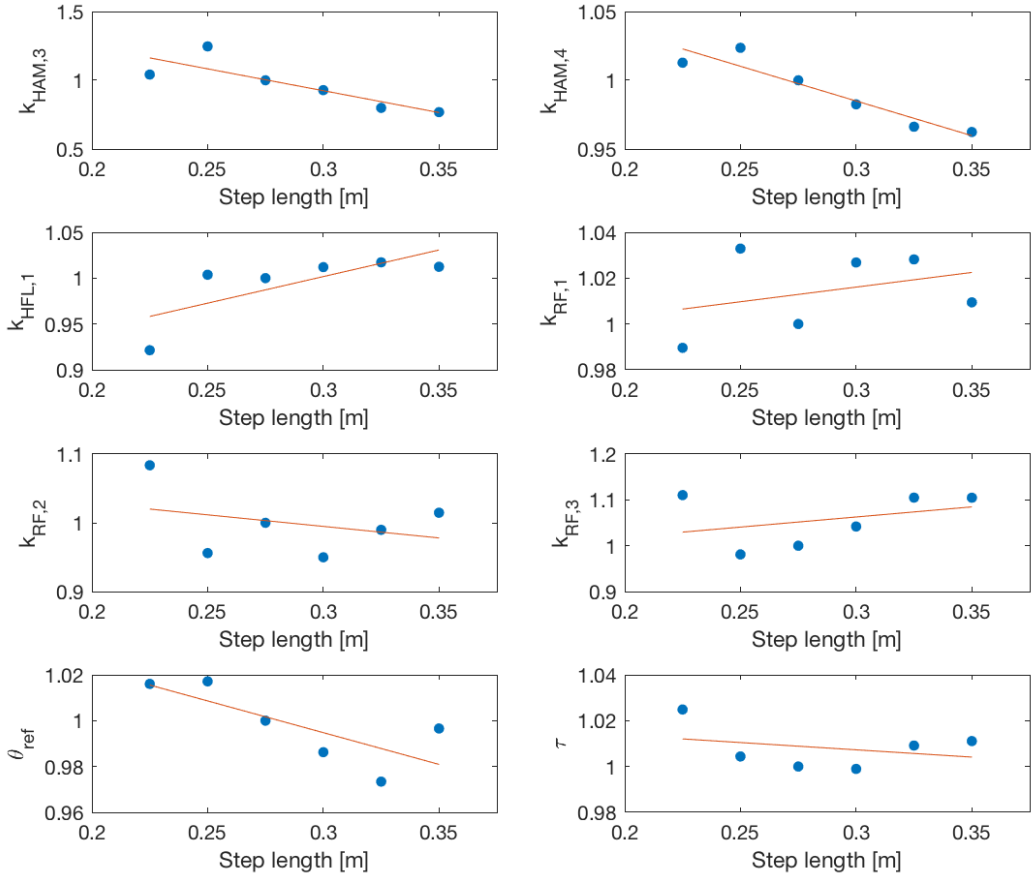


Figure D.4: Mean values of the parameters $k_{HAM,3}$, $k_{HAM,4}$, $k_{HFL,1}$, $k_{RF,1}$, $k_{RF,2}$, $k_{RF,3}$, θ_{ref} and τ based on 5 optimizations for 6 different step lengths. The mean values are expressed in relative variation compared to the step length of 27.5 cm. These values are then approximated by the best linear fit, in the least-squares sense.

D.3 Complement for descending

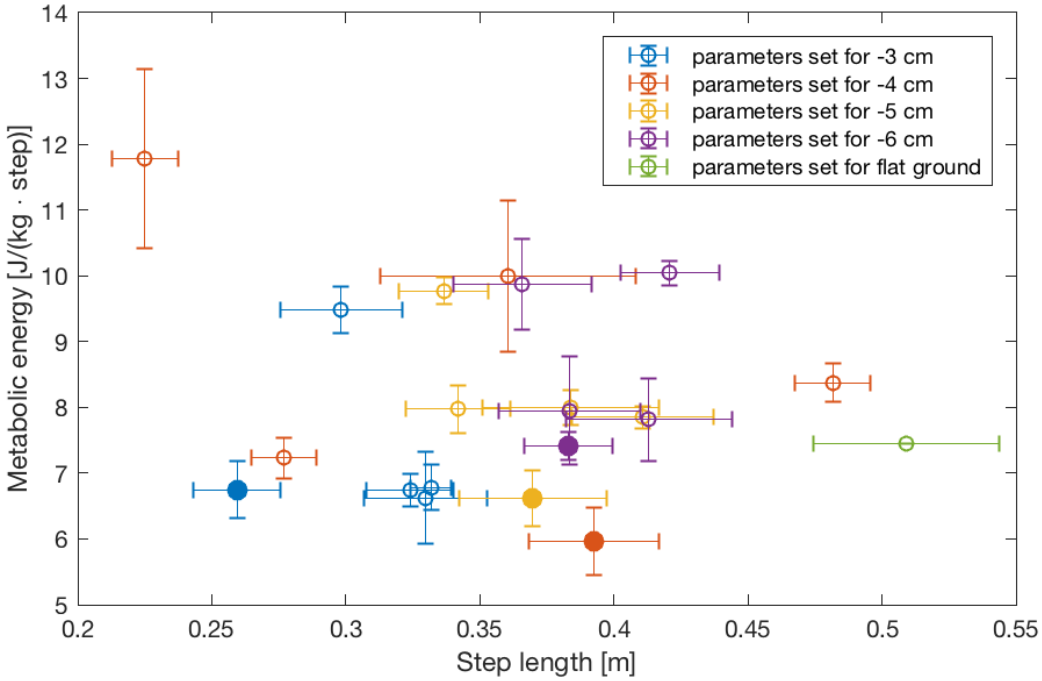


Figure D.5: Metabolic energy consumed by the muscles of the CoMan per step and normalized by the mass of the CoMan in function of the step length for different optimized parameter sets. 5 parameters sets (from 5 different optimizations) are shown for 4 different rises (-3 cm, -4 cm, -5 cm, -6 cm), represented by the different colors. We use virtual stairs, which explains that the step length is not constrained and may vary. The horizontal and vertical bars represent one standard deviation, obtained by running 5 times the same controller. For each rise, we choose 1 set of parameters indicated by the full circles.

Bibliography

[AMA] AMARSI. COmpliant HuMANoid Platform (COMAN). `https://www.amarsi-project.eu/coman`.

[BPA04] L. J. Bhargava, M. G. Pandy, and F. C. Anderson. A phenomenological model for estimating metabolic energy consumption in muscle contraction. *Journal of Biomechanics*, 37(1):81–88, 2004.

[BRMF11] G. Bovi, M. Rabuffetti, P. Mazzoleni, and M. Ferrarin. A multiple-task gait analysis approach: Kinematic, kinetic and EMG reference data for healthy young and adult subjects. *Gait & Posture*, 33(1):6–13, 2011.

[BVdN13] A. Barrea and N. Van der Noot. *Real-time control of a humanoid robot: Dynamic walking using a bio-inspired approach.* Université Catholique de Louvain, Louvain-la-Neuve, June 2013.

[CER] CEREM. Robotran website. `https://www.robotran.be`.

[DMMC+13] H. Dallali, M. Mosadeghzad, G. A. Medrano-Cerda, N. Docquier, P. Kormushev, N. Tsagarakis, Z. Li, and D. Caldwell. Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach. In *2013 IEEE International Conference on Mechatronics (ICM)*, pages 598–603, 2013.

[Fal13] J. Falconer. This Humanoid Robot Gets Pushed Around But Stays on Its Feet. `https://spectrum.ieee.org/automaton/robotics/humanoids/iit-coman-humanoid-robot`, 2013.

[FC08] C. Fu and K. Chen. Gait Synthesis and Sensory Control of Stair Climbing for a Humanoid Robot. *IEEE Transactions on Industrial Electronics*, 55(5):2111–2120, May 2008.

[FCG03] G. Figliolini, M. Ceccarelli, and M. Di Gioia. Descending stairs with EP-WAR3 biped robot. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 2, pages 747–752 vol.2, 2003.

[GFF04] J. S. Gutmann, M. Fukuchi, and M. Fujita. Stair climbing for humanoid robots using stereo vision. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 2, pages 1407–1413 vol.2, September 2004.

[GH10] H. Geyer and H. Herr. A Muscle-Reflex Model That Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(3):263–273, June 2010.

[Gre17] Ph. Greiner. *Gait modulation of a humanoid robot, using bio-inspired mechanisms.* Université Catholique de Louvain, 2017.

[HCZH08] W. Huang, C. Chew, Y. Zheng, and G. Hong. Pattern generation for bipedal walking on slopes and stairs. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 205–210, 2008.

[HVdNIR16] F. Heremans, N. Van der Noot, A. J. Ijspeert, and R. Ronsse. Bio-inspired balance controller for a humanoid robot. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 441–448, 2016.

[Ijs08] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, 2008.

[JW71] G. M. Jones and D. G. D. Watt. Observations on the control of stepping and hopping movements in man. *The Journal of Physiology*, 219(3):709–727, 1971.

[KWJ+92] D. E. Krebs, D. Wong, D. Jevsevar, P. O. Riley, and W. A. Hodge. Trunk kinematics during locomotor activities. *Physical Therapy*, 72(7):505–514, 1992.

[Lec18] E. Lecompte. *Balance improvement of a humanoid robot while walking.* Université Catholique de Louvain, 2018.

[PW95] G. A. Pratt and M. M. Williamson. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 399–406 vol.1, August 1995.

[Ron17] R. Ronsse. *LMECA2732 - Robotics.* Université Catholique de Louvain, 2017.

[RRF02] R. Riener, M. Rabuffetti, and C. Frigo. Stair ascent and descent at different inclinations. *Gait & Posture*, 15(1):32–44, February 2002.

[SF03] J.-C. Samin and P. Fisette. *Symbolic Modelling of Multibody Systems.* Kluwer Academic Publisher, 2003.

[SG15] S. Song and H. Geyer. A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion. *The Journal of Physiology*, 593(Pt 16):3493–3511, August 2015.

[Shi99] C. Shih. Ascending and descending stairs for a biped robot. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 29(3):255–268, May 1999.

[SRD+17] M. Schwarz, T. Rodehutskors, D. Droeschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke. NimbRo Rescue: Solving Disaster-response Tasks with the Mobile Manipulation Robot Momaro. *Journal of Field Robotics*, 34(2):400–425, March 2017.

[Tag94] G. Taga. Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment. *Physica D: Nonlinear Phenomena*, 75(1):190–208, 1994.

[TLSC11] N. G. Tsagarakis, Zhibin Li, J. Saglia, and D. G. Caldwell. The design of the lower body of the compliant humanoid robot iCub. In *2011 IEEE International Conference on Robotics and Automation*, pages 2035–2040, May 2011.

[VB04] M. Vukobratović and B. Borovac. Zero-moment point - thirty five years of its life. *International journal of humanoid robotics*, 1(1):157–173, 2004.

[vdK14] J. van den Kieboom. Optimization Framework Conceptual Overview, February 2014.

[VdNCB+15] N. Van der Noot, L. Colasanto, A. Barrea, J. van den Kieboom, R. Ronsse, and A. J. Ijspeert. Experimental validation of a bio-inspired controller for dynamic walking with a humanoid robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 393–400, September 2015.

[VdNIR15] N. Van der Noot, A. J. Ijspeert, and R. Ronsse. 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)*, pages 6267–6274, May 2015.

[VdNIR18] N. Van der Noot, A. Ijspeert, and R. Ronsse. Bio-inspired controller achieving forward speed modulation with a 3d bipedal walker. *International Journal of Robotics Research (IJRR)*, 2018.

[Whi07] M. Whittle. *Gait Analysis: an introduction*, volume 1. Butterworth-Heinemann Elsevier, 4th edition, 2007.

[Zeu11] T. Zeugmann. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 745–753. Springer, 2011.

[ZHVdN+17] A. A. Zobova, T. Habra, N. Van der Noot, H. Dallali, N. G. Tsagarakis, P. Fisette, and R. Ronsse. Multi-physics modelling of a compliant humanoid robot. *Multibody System Dynamics*, 39(1-2):95–114, January 2017.

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