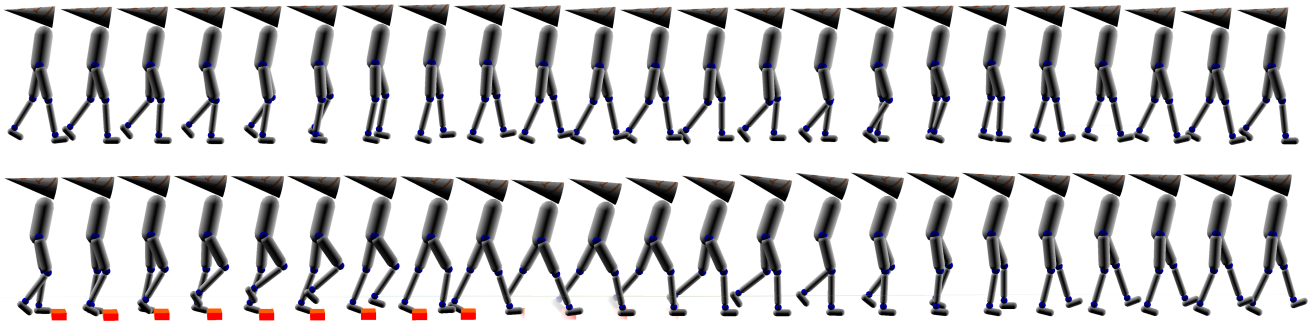
Energy consumption optimization and stumbling corrective response for bipedal walking gait

Steve Berger



Supervisors:
Jesse van den Kieboom
Prof. Renaud Ronsse
Prof. Auke Jan Ijspeert

June 22, 2011

Contents

Acknowledgments

We usually thank a lot a bit of everyone who played a role (or not) in a project. So I really want to do the same.

First, I want to thank my parents who met, mated and made me, because I am very happy to be alive. Of course, thank you very much for your constant help and for giving me the possibility to do these exciting studies. Thanks to all my ancestors as well.

Thanks to my friend Charles, who taught me how it is necessary to believe, but not sufficient. For example, he often says "Make it happen" while lying on a bed. Thank you for your teaching, at least for showing me how your philosophy brought you success in your master project!

Thanks to Bertrand who is very interested in understanding what scientists actually do in their labs. It is good to have a friend like you.

Thanks to Timothée, who seems to be definitively good in finding good jobs and a nice way to live. He always has good plans. Enjoy your trip around the world and please do not talk about that.

Thanks to Joelle-Anne.

Thanks to Regis who repeats again and again "science works at solving problems science creates itself". Thanks Regis. You're a great help.

Of course, thanks to Aurélie, who keeps me on track of the real life (the one without C++). I think I would live in a virtual world if you were not here.

Thanks to Edgar, who allow me to work with him and therefore indirectly paid my food.

Thanks to Carlos who teach me how it is possible to love the hard life even if stuck in a wheelchair forever. And I am sorry to frighten you of science...

Thanks to Toni-Toni... I'm unfortunately not allowed to write anything about you in this report.

Thanks to Walter and his nice way of life. Maybe I should follow you? Maybe you're right?

Many thanks to Kostas which opened the conference room one night and make several of my friends happy.

Thanks to John and the very interesting talks we had together. And for your fall from a table, which was really funny!

Thanks to Rolando and your everyday help in Linux! And for all the coffee you paid.

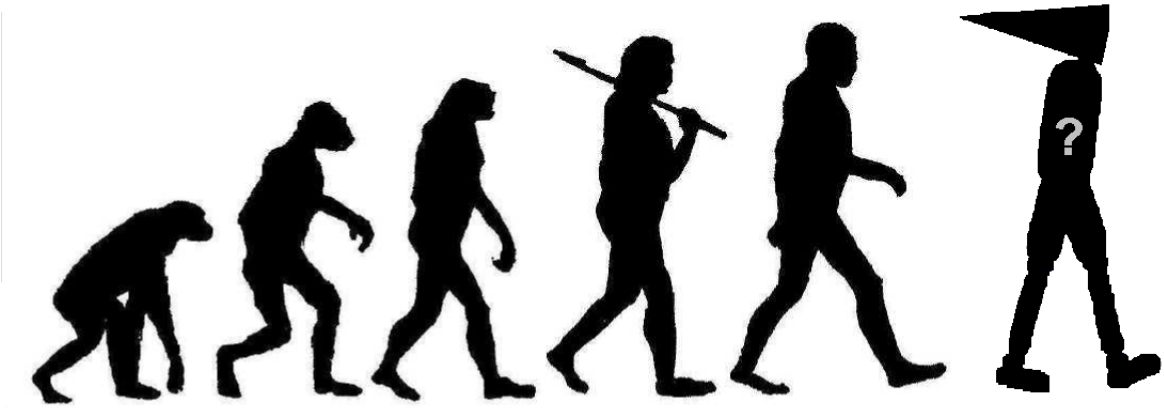
Thanks to the other students of the lab!

A special big thanks to Jesse Van Den Kieboom who supervised this project who saved my life several times, you're my hero.

Thanks to Professor Renaud Ronsse who supervised this project as well! It was a real pleasure coming to Belgium and meeting you! I hope I'll come again! By the way, you got the idea of the signal-dependent noise in muscles and for that I can say it again: thank you very much!

Thanks to Professor Auke Jan Ijspeert who gave me the possibility to work here.

That's all.



Introduction

When we look at the animal kingdom, we can agree our robotic technology is far from being able to cope with unpredictable environments as animals do. Animals manage to move in any environment with an astonishing fluidity. Decomposition of any gait reveals the complexity of such a task, as in figure 1. What are the key elements which allow a biological system to move in a given environment? The answer to this question is unfortunately far from being complete, and we will bring below only a little piece of the whole puzzle.

Early creatures which evolved limbs to cope with a terrestrial environment, rather than an aquatic one, needed a stable controller able to deal with unpredictable terrains. Such controllers for legged locomotion seem very complex indeed, since it is easier to make a wheel roll than a legged machine walk. And if natural wheels does not emerge from evolution, how can a controller for legs emerge despite its supposed high complexity?

Maybe the complexity operates at another level, like on the intrinsic properties of the legs. Could legs have intrinsic properties which help them to be stable, as the intrinsic properties of a bouncing ball help the ball to be bouncy?

Let us imagine an artificial leg purely controlled by a motor. To make the leg jump, we should control the motor to first bend the leg and then stretch it suddenly, which would use energy, computational power, etc... Now, if we have an elastic leg, we would only need to use the motor to bend the leg, the elastic properties would then perform the jumping job. So it is obvious the properties of materials used to build an artificial leg are determinant to settle its intrinsic stability/dynamic.

It has been shown by [54] that muscles possess properties which make legs more stable for a hopping gait. They call such properties *preflexes*, since the reaction of muscles against an external perturbation is faster than those of *reflexes*. It is therefore some properties of living tissues (such as muscles) which help animals to move and to be stable.

In 2003, Hartmut Geyer shows how a model of a muscle could help a bouncing mono-leg to be stable [28]. Such a muscle model has been extended in 2010 to a full bipedal walker which manages to walk in the sagittal plane and to be stable against slight slopes [31]. This model captures principles of neuromuscular feedbacks and predicts muscle activation patterns observed in leg muscles. Such a bio-inspired approach is very important to better understand how biological systems work and to formulate the basic properties of legged agent. If these properties would be better understood in the future, we could develop new technologies in the locomotion fields which deal with unpredictable terrains as animals do.

In the present master project, we will reproduce this walking model with a small modification at the ankle: a segment will be added between the ankle joint and the bottom of the foot to have an even more realistic humanoid model. The feedback parameters of the neuromuscular model has been tuned manually by H.Geyer which give very good results. We are interested to see if an optimization of feedback parameters could lead to a human-like walking gait. Since biological legged agents optimize their gaits through their life, we believe they should exist some way to measure if a gait is better than an other. Since we know a free walker will converge to a walking speed which minimize its energy consumption [19], we will investigate the measure of energy consumption as an indicator of a good gait to optimize feedback parameters.

Furthermore, we are interested to make the model more robust against signal dependent noise (SDN) in muscles, since such noise is highly present in biological motor systems [58]. Therefore, we want to see if we can optimize a walking gait in the presence of SDN and see if such process can improve the robustness of the optimized walking gait.

Finally, we want to augment the muscle model to reproduce the stumbling corrective response (SCR). The SCR is triggered every time someone hits his foot in on obstacle while walking and prevents falling.

The first chapter will introduce concepts related to the musculo-tendon complex (MTC) and show how to model such a complex. Chapter two will introduce sensory-motor feedbacks used to activate the MTC. The third chapter gives details about the humanoid model and how to use several MTCs with sensory feedbacks on a leg. In chapter four, we will see how we can use energy consumption as a criterion of walking gait optimization. Finally, chapter five will show how to design and test a set of feedback rules aimed at modeling the stumbling corrective response (SCR). We will see the humanoid can avoid falling in 95% of the cases the obstacle is hit in early- or mid-swing.

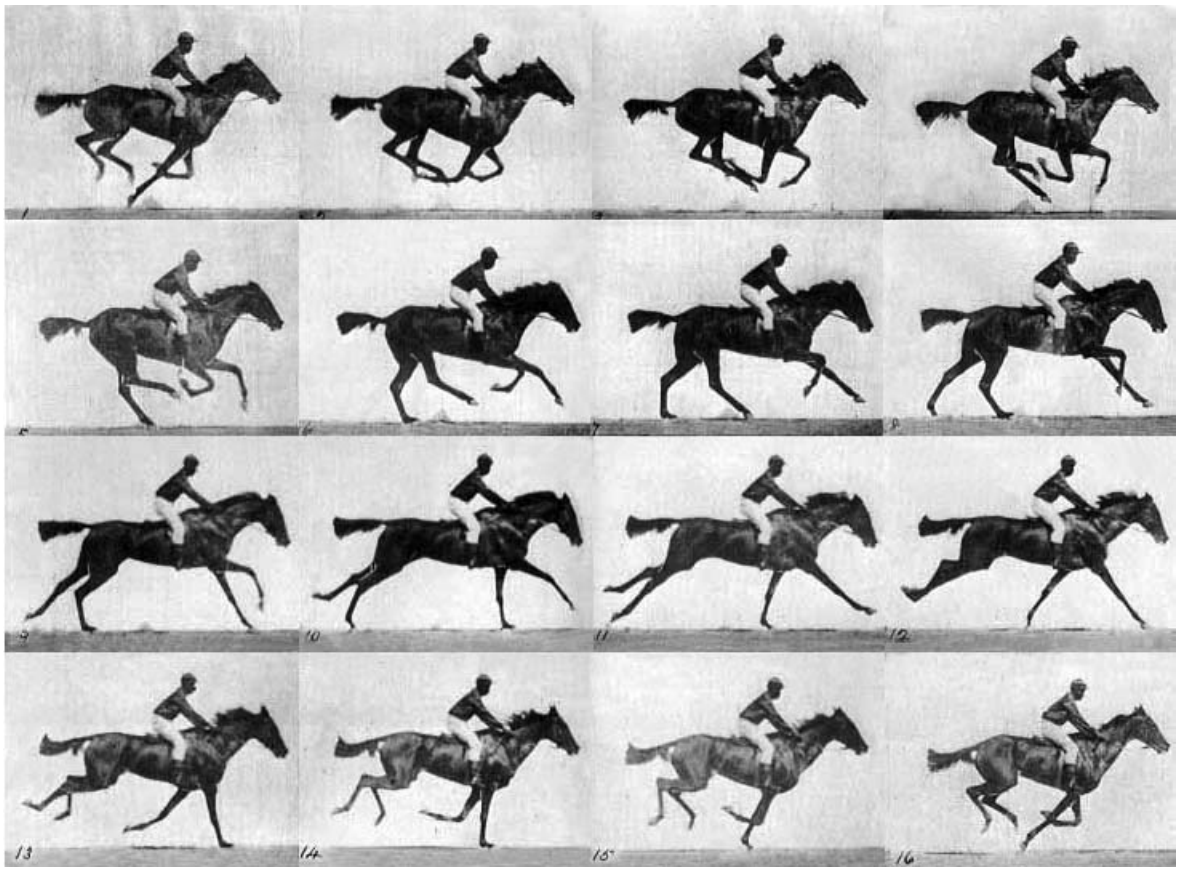


Figure 1: *Sallie Gardner at a Gallop*, a photographic experiment from Eadweard J. Muybridge, June 1878

Chapter 1

Muscle and Tendon Models

Introduction

In the following sections, we first describe the properties of biological muscle and tendon, we then give a mathematical model which captures these properties and we finally explain how to integrate this model in a simulation.

1.1 Overview of the properties of a muscle-tendon complex (MTC)

In order to move, we first need bones, otherwise we would sluggishly crawl on the floor like mollusks. Bones are specially adapted to our everyday animal life, for example, birds possess lighter bones in order to save energy while flying. The shape of bones are adapted to support our body weight and to fit with other bones. Bones are attached together with ligaments, an elastic living tissue allowing solid bones to move with respect to each others.

In this project, we will consider bones as straight rigid segments as shown on figure 1.1. Elastic properties of ligaments will not be modeled (for a model of ligaments, see [18]), however bones will freely move around virtual joints and angular soft limits will be added (see section 3.1).

Ligaments cannot control bones, they simply act passively. Muscles are required to actively move bones. However, they are not directly attached to them: tendons make the link as shown on figure 1.2. In gray, we see two bones attached together with a passive joint and forming a chain. Tendons are shown in blue. The Golgi tendon organs (GTO) - or the capsules - are force sensors [7] (in yellow). The muscles are shown in red. Show in green are the muscle spindles which are the length sensors [7] of muscles and in purple the ligaments.

In this section, we are interested in the properties of tendons as well as the properties of muscles. Tendons are passive elements which produce a force proportional to their deformation only. Muscles on contrary can generate forces if they are stimulated and thus have active properties. We will see muscles possess both active and passive properties in the next section.

It is worth to note that *stimulation* and *activation* of muscles are not the same thing (see section 1.3.2).

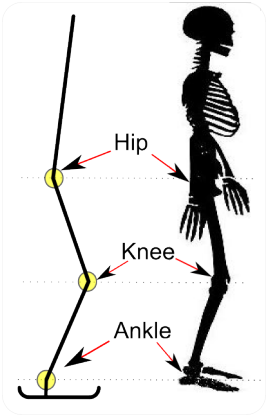


Figure 1.1: Left: artificial skeleton Right: human skeleton

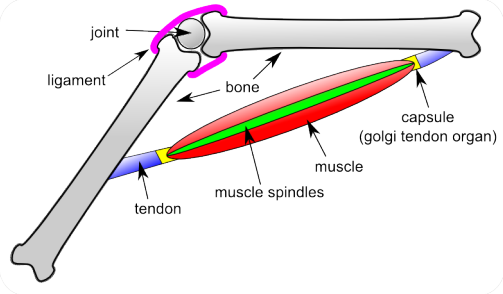


Figure 1.2: Sketch showing the position of bones, joint, tendons, muscle, muscle spindles and Golgi tendon organ.

1.1.1 Force $\Longleftrightarrow$ shortening velocity relation in muscles

In 1938, Hill develop a model of shortening muscle based on experiments with the sartorius muscle of the small *Rana temporaria* English frog [27]. At this time, the following facts were clear:

1. Active muscle, at least as ordinarily prepared and used, contains an undamped elastic element: if, during maintained isometric contraction, the muscle be instantaneously released, the tension does not fall to zero unless the amount of release be sufficient to allow the undamped element to reach its natural length.
2. Active muscle contains an apparently damped element, in series with the undamped elastic one; the former, if allowed to shorten, exerts a force which is smaller the faster the shortening, until at a certain speed it exerts no force at all: at more than that speed it cannot shorten; conversely, if made to lengthen, it exerts a force which is greater the faster the lengthening, until at a certain speed it "gives" like a wire stressed beyond the elastic limit.
3. Resting muscle contains the elastic element (1), but only to a minor degree the apparently damped element (2).
4. The active state is set up much more rapidly than the isometric tension, as is shown by applying a quick stretch at various moments after a stimulus.

Taken from [27].

Hill gives the force $\Longleftrightarrow$ velocity relation for shortening muscles:

$$(F + a)(v + b) = (F_0 + a)b$$

which represents a rectangular hyperbola with asymptotes at $F = -a$ and $v = -b$. F is the force generated by the muscle, F_0 is the isometric force of the muscle (force against which the muscle neither shortens nor lengthens), v is the shortening velocity of the muscle and a, b are two constants. The speed $v_{max} = {}^{bF_0}/_{a}$ is the shortening velocity against no load ($F = 0$). In order to compare different muscles, Hill's equation can be written in a normalized form which describe nearly all muscles (cardiac, smooth and skeletal muscles) [12]:

$$v' = \frac{1 - F'}{1 + kF'} \Leftrightarrow F' = \frac{v_{max} - v}{v_{max} + kv} \tag{1.1}$$

where $v' = {}^{v}/_{v_{max}}$, $F' = {}^{F}/_{F_0}$ and $k = {}^{F_0}/_{a} = {}^{v_{max}}/_{b}$. We will later use equation 1.1 in our muscle model.

1.1.2 Force $\Longleftrightarrow$ lengthening velocity relation in muscles

Further works have been done by B.Katz [42] to establish force $\Longleftrightarrow$ lengthening velocity relation on the sartorius muscles of the English *Rana temporaria* frog. Figure 1.3 shows the force $\Longleftrightarrow$ velocity relation for both lengthening and shortening muscle.

1.1.3 Force $\Longleftrightarrow$ length relation in muscles

Muscle force$\Longleftrightarrow$length relation have been studied among others by Aubert & al. (1951) [21].

When stretched out of its rest (or optimal) length, a muscle will passively fight against elongation like an unidirectional non-linear spring. By passively, we mean the muscle does not need to be stimulated and it is rather the intrinsic properties of its molecular structure which exhibit such spring-like behavior.

If stimulated, a muscle actively generates a force which depends on its length. By actively, we mean the muscle need to be stimulated to generate a force. A muscle is tetanised if maximally stimulated.

Both passive forces and active forces sum up to give the total muscle force (shown on figure 1.4).

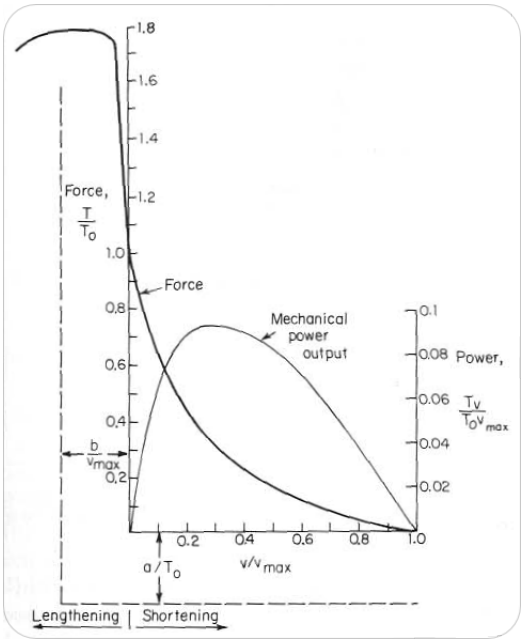


Figure 1.3: Hill's force-velocity curve. The shortening part of the curve was calculated from equ. 1.1 with $k = 0.25$. The asymptotes for Hill's hyperbola (broken lines) are parallel to the F/F_0 and v/v_{max} axes. Near zero shortening velocity, the lengthening part of the curve has a negative slope approximately six times steeper than the shortening velocity. Everything is taken from [12].

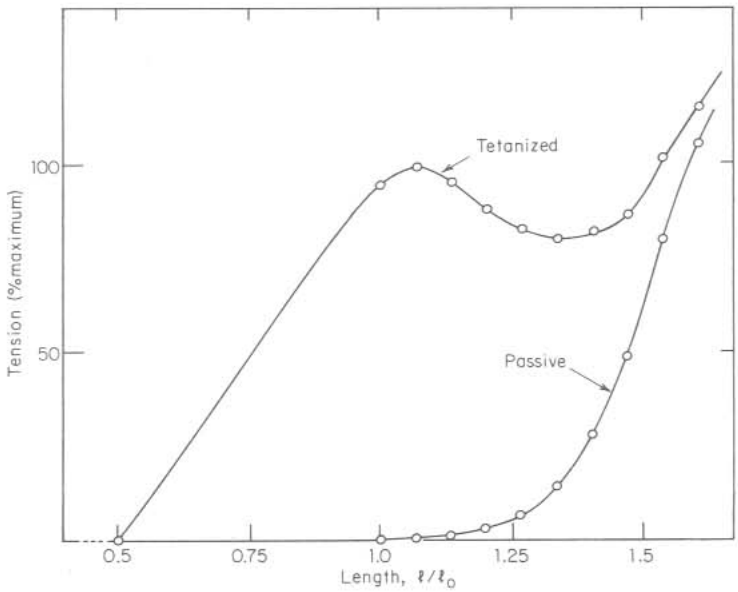


Figure 1.4: Tension-length curves for the frog sartorius muscle at 0°C. The passive curve was measured on the resting muscle at a series of different lengths. The tetanized (active) curve was measured at a series of constant lengths as the muscle was held in isometric contraction. The rest length (l_0) was the length of the muscle in the body. Everything is taken from [12].

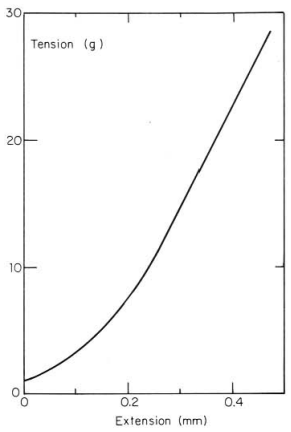


Figure 1.5: Length-tension curve for the series elastic component in a frog sartorius muscle. From [12].

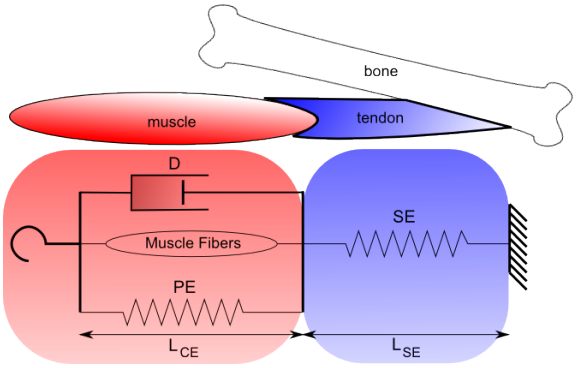


Figure 1.6: A model of a muscle-tendon complex which is a first approximation of biological muscle-tendon complex.

1.1.4 Force $\Longleftrightarrow$ length relation in tendons

The undamped elastic element in series with muscle observed by Hill has been studied by Jewell and Wilkie (1960) [41]. It behaves as an unidirectional non-linear spring (shown on figure 1.5). If a tendon is stretch out of its rest (or slack) length, it will passively fight against the elongation as a spring. So tendons possess passive elastic properties, like muscles.

1.1.5 Muscles and tendons in series

With these above discoveries, it becomes possible to model muscles and tendons linked in series as shown on figure 1.6. In this figure, a muscle is the parallel sum of:

- an active element (muscle fibers) which can generate a force depending on its stimulation
- a passive elastic (PE) element
- a passive damped (D) element

Such a muscle is connected in series with a tendon which can be seen as a serial elastic (SE). This model can be mathematically formulated but does not reflect the real above mentioned properties of biological muscle and tendon tissues.

First, the springs should be unidirectional. For both the tendon and muscle, spring effects (SE for serial elastic and HPE or PE for high parallel elastic) take place only if they are stretch out of their rest length (called optimal length for muscles and slack length for tendons).

Second, to prevent the model of muscle fibers to reach a null length (which is not realistic at all), we need to add an artificial unidirectional spring (LPE for low parallel elastic).

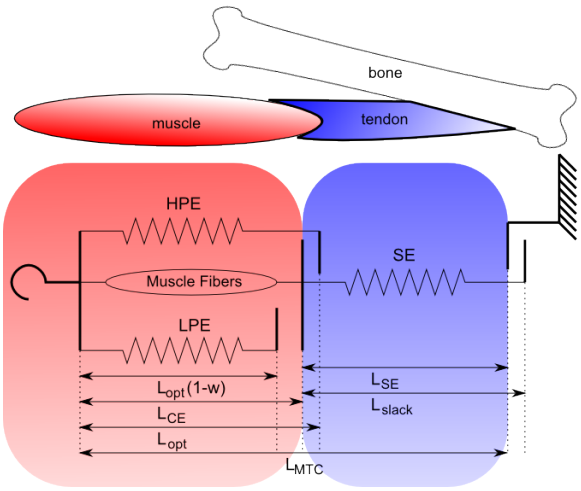


Figure 1.7: A model of a muscle-tendon complex which is close to biological observation

Third, the damping properties of a muscle are linked to its stimulation, which means damped-like behavior is an active property embedded within the muscle fibers [32]. In other words, if there is no stimulation, there is no damped-like effect.

Finally, as seen before, the force generated by muscle fibers under stimulation depend on their length as well.

Figure 1.7 illustrates a muscle-tendon model which is closer to biological observations.

In the next section, we will see in more details how biological muscles work.

1.2 Biological muscles

For the sake of completeness, we will explain some basic features and mechanisms of biological muscles. This whole section is adapted from [10, 11, 4].

There are three main classes of muscles:

- Smooth muscles (found in guts, blood vessels, uterus, bladder, etc...)
- Heart muscles
- Skeletal and facial muscles

The two first classes are involuntary muscles (we do not consciously control them, they are rather controlled by the sympathetic and parasympathetic divisions of the nervous system). In this project, we are interested by the third class (which are voluntary muscles) and more precisely by skeletal muscles. Each time we talk about muscles, we actually refer to skeletal muscles.

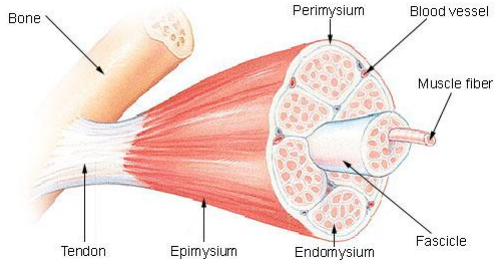
1.2.1 Overview of molecular events in skeletal muscles

A sarcomere is the contractile unit of muscle fibers

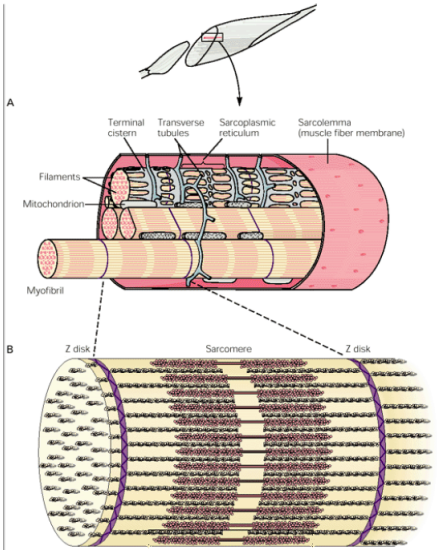
Figure 1.8a shows the very general macroscopic structure of skeletal muscles. Basically, skeletal muscles are made of muscle fibers in parallel and in series. The length of a lower limb muscle fiber can range from 11% up to 90% of the whole muscle length (according to data taken from [55]).

A muscle fiber is a muscle cell and is made of several *myofibrils* in parallel (fig. 1.8b) . Each myofibril (1 to 2 $[\mu m]$ of diameter) is made of several *sarcomeres* (each about 1.5 to 3.2 $[\mu m]$ long) in series. Therefore, a muscle fiber would be made of about 20'000 sarcomeres in series.

Figure 1.9 sketches the structure of a sarcomere. In a few words, a sarcomere is a cylindrical shaped structure limited by two discs (the *Z-discs*) anchoring *actin* filaments (or thin filament). Actin is a cytoskeleton protein (for more details, see [11]) which keep a constant length in sarcomeres. *Myosin II* (or thick filament) is a protein made



(a) Structure of skeletal muscles, taken from [13].



(b) **A**: A muscle fiber is made of myofibrils in parallel and series. Each myofibril is made of several sarcomere in series. **B**: Each sarcomere is limited by the Z-disc and filled with actin and myosin II. Taken from [6].

Figure 1.8

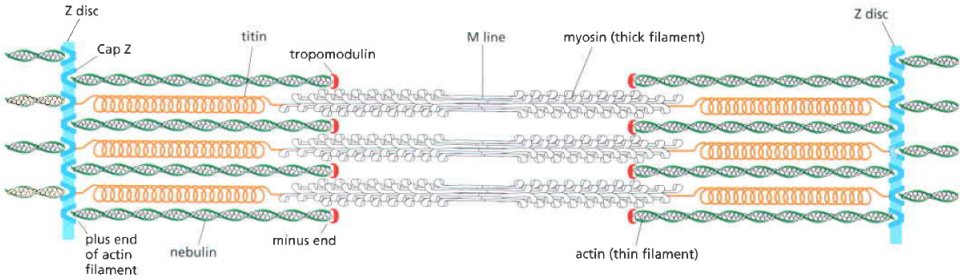


Figure 1.9: A lateral sketch of a sarcomere (taken from [11]).

of a head and a tail. The tail can agglomerate with other tails of myosin II molecules to create a bipolar polymer (fig. 1.10) while the head can bind with actin. To prevent such binding, *nebulin* coats the actin filament. The bipolar polymers of myosin II are held in the center of each sarcomere by a spring-like molecule, the *titin* (also known as *connectin*).

Muscle contraction process

Muscles contract if they are stimulated by neurons. Figure 1.11 shows a motoneuron (which is a special kind of neuron aimed at stimulating muscles) linked to a muscle fiber through a neuromuscular junction. At this junction, the axon of the motoneuron splits in several synaptic knobs. Each synaptic knob releases a chemical compound (the acetylcholine or ACh) if an electric spike comes from the motoneuron. ACh travel in the synaptic cleft and binds to ACh-receptors (or equivalently called ACh-gated channel on fig. 1.11).

An ACh-receptor is an ion channel which opens if bound by ACh to let sodium ions flow inside a muscle fiber. Increase in concentration of Na^+ opens another ion channel which lets Ca^{2+} flow inside the sarcomeres. Ca^{2+} will then link to nebulin and change its conformation to uncoat the actin and let myosin II bind freely with actin.

The myosin head can be detached from the actin filament only if an ATP[1] molecule is around. Hydrolysis of

[1]ATP (or Adenosine triphosphate) is a molecule which releases energy if hydrolyzed in ADP (Adenosine diphosphate). ATP is the energy unit of living systems, it is as a microscopic battery which can be used only once. Therefore, living organisms have to continuously produce ATP in the whole body.

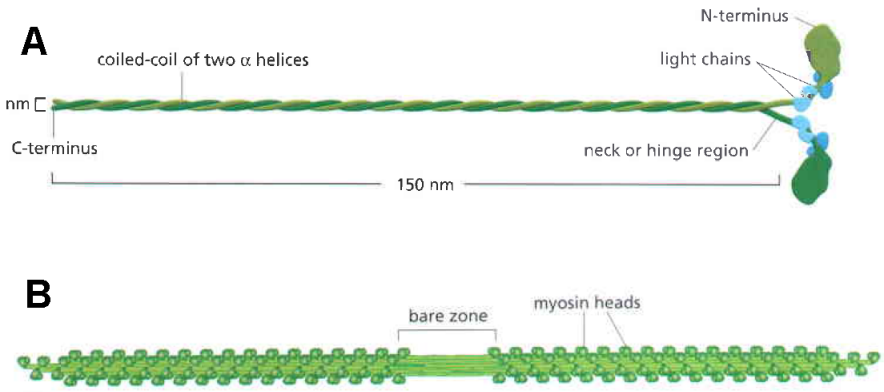


Figure 1.10: **A**: Myosin II bimere molecule is composed of two heavy chains and four light chains. **B**: About 250 bimere of myosin II create a bipolar thick filament. Taken from [11].

ATP to ADP gives enough energy to change the conformation of the myosin head which will bind a bit further on the actin filament. Removing ADP allows the myosin head to pull on the actin (fig. 1.12) and generates a force (the displacement is about $0.06[\mu m]$). If no ATP is available, myosin will stick indefinitely on actine (which explains why cadavers are stiff).

This whole chemical reactions and therefore the muscle activation takes time. We will see later how it is possible to simulate the activation/deactivation of a muscle fiber given the stimulation from a motoneuron.

To summarize, stimulation from a motoneuron triggers a chemical reaction at the membrane of a muscle fiber which lets Ca^{2+} ions flow in the sarcomeres. Such events allow myosin and actin to interact and shrink sarcomeres. If several sarcomeres are shrunk together, they provoke the contraction of the whole muscle.

1.2.2 From microscopic molecules to macroscopic properties

We have seen how molecules interact to produce a contracting force in muscle fibers. These microscopic structures determine the macroscopic force $\Longleftrightarrow$ length and force $\Longleftrightarrow$ velocity properties of muscle fibers.

Force $\Longleftrightarrow$ length relation Figure 1.13 illustrates how the passive and active force of a muscle fiber changes with respect to its length.

If a muscle fiber is at its optimal length, the surface of interaction between actin and myosin II will be maximum, allowing the muscle to generate a maximal force. On contrary, if the muscle is elongated, the region of overlap decreases until no more myosin II can pull on actin, therefore the capability of the muscle to actively produce force is reduced to zero. Finally, if the muscle is fully contracted the myosin II filament will reach the end of the actin filament (and touch the Z-disc) which prevents further contraction of the sarcomere (the muscle is not capable of producing force).

The titin (connectin) proteins act as springs and generate a passive force if elongated. The total passive force of muscles is explained by the titin structure and other tissues surrounding muscle fibers.

Force$\Longleftrightarrow$ velocity relation Figure 1.14 illustrates how the active force of a sarcomere depends on its contractile velocity.

If a sarcomere is lengthening, the direction of the sliding motion between actin and myosin allow them to have a better interaction. On the contrary, if a sarcomere unit is shortening, the myosin head needs to detach and reattach at a higher frequency to cope with the already shortening fibers. The active force generated by sarcomeres will therefore decreas if the muscle is shortening. At the limit, the shortening velocity can be higher than the capacity of myosin to actuate its pulling behavior. In such case, the muscle can not produce active force anymore. We call this limit velocity v_{max}.

To summarize, the active forces of a muscle are determined by the cross bridges between actin and myosin II and by the contractile velocity of sarcomeres. Passive forces are determined by the elasticity of the titin (connectin) and other tissues surrounding the muscle fibers.

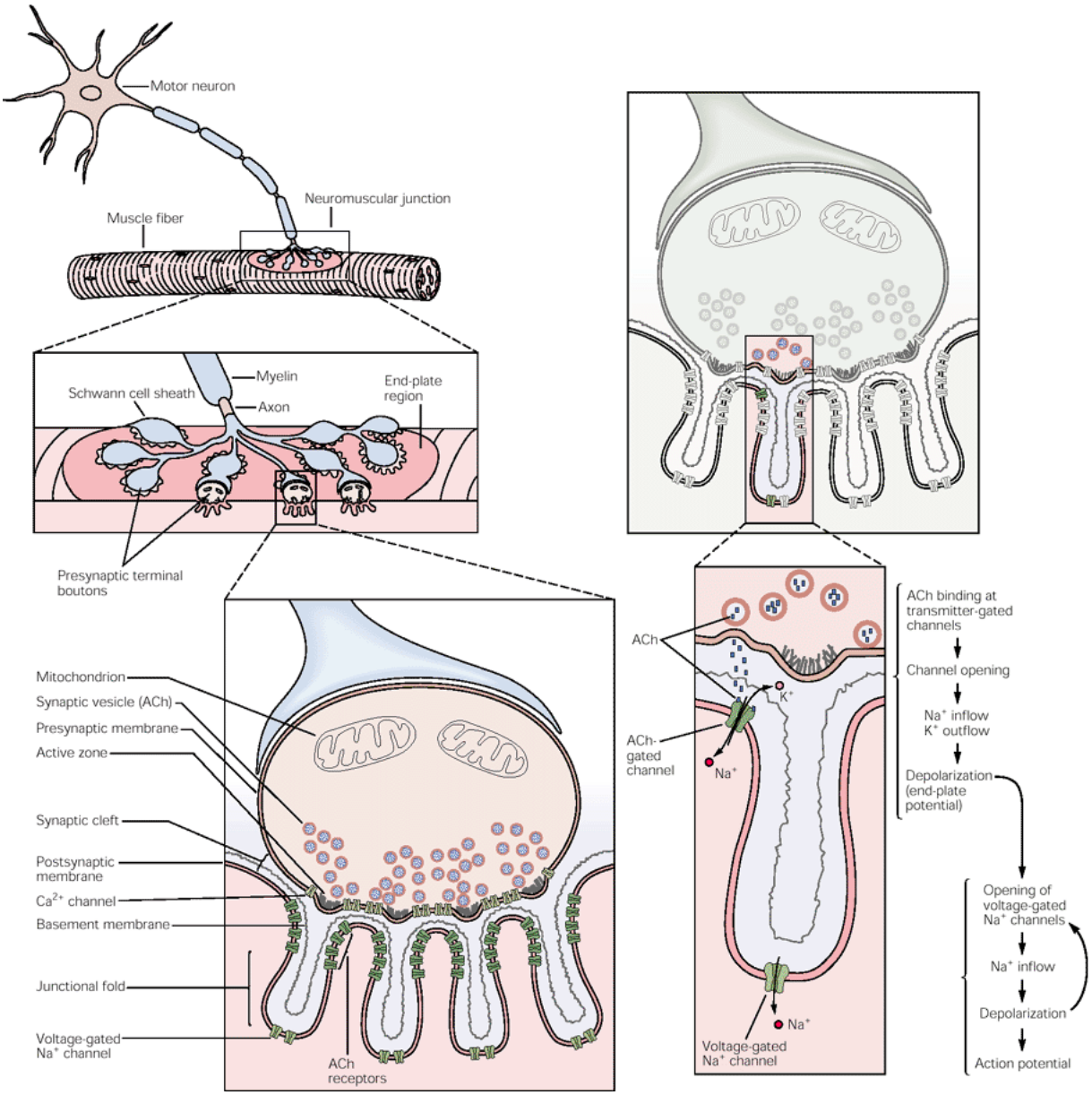


Figure 1.11: Schematic view of a neuromuscular junction. Acetylcholine is released from the synaptic knob if the motoneuron spikes and binds to ACh-receptors on the muscle side which allow ions to flow in and out of the muscle fiber (taken from [4])

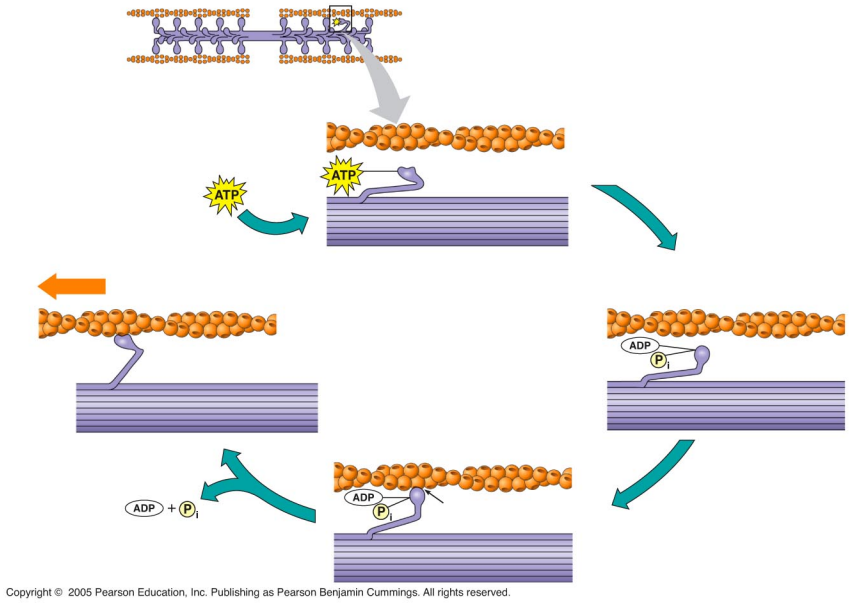


Figure 1.12: Actin-myosin interaction, taken from [14].

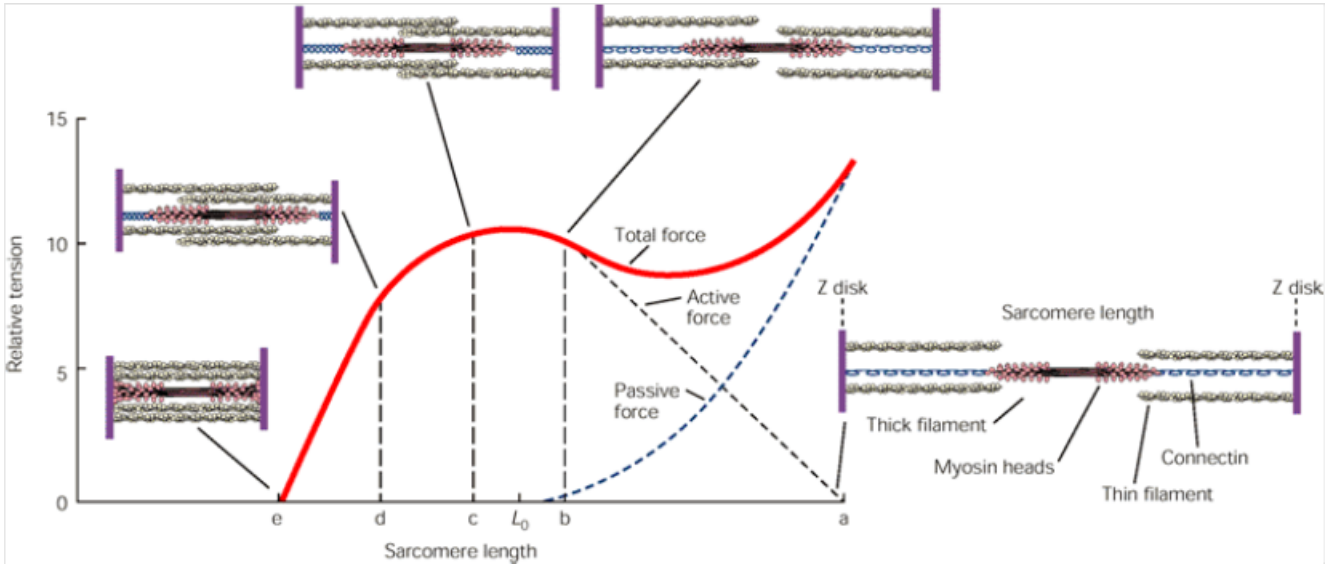


Figure 1.13: Molecular structures explain muscle force $\Longleftrightarrow$ length relation. Taken from [6]

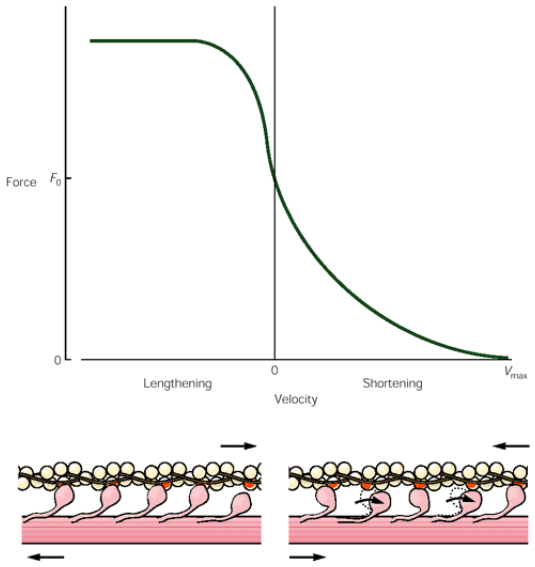


Figure 1.14: Muscle force$\Longleftrightarrow$velocity relation from a molecular point of view. Taken from [6]

1.2.3 Muscle fibers are pennated

The total force of a muscle depends on how many muscle fibers work in parallel. A way to increase the amount of parallel muscle fibers in a muscle of fixed size is to pennate them. The pennation angle is the angle θ (fig. 1.15) between the muscle's attaching axis and the orientation of the muscle fibers within the muscle.

If muscle fibers are pennated, they will generate a force which is not aligned with the muscle axis. Therefore, if the pennation angle is too big (if close to $\theta = \pi/2$) the force pulling on the bones is reduced to zero.

In other words, the pennation angle increases the number of muscle fibers that can work in parallel (and therefore the total force generated with an angle θ) but decreases the total force pulling on bones (since muscle fibers are not aligned anymore with the muscle axis). Typically, a highly pennated muscle will produce a higher force but on a smaller range of motion.

It is worth to note the pennation angle increases when a muscle shortens (fig. 1.15C) and can vary within the same muscle. For the sake of simplicity, we will consider in our model a muscle as a cylindrical shape with a constant pennation angle.

We call the *physiological cross-sectional area* (PCSA) the area of a muscle cut orthogonally to muscle fibers.

There exists a relation between the PCSA, the pennation angle θ, the muscle volume V and the muscle fibers length l_{opt} [55] :

$$PSCA = \frac{m \cdot \cos(\theta)}{\rho \cdot l_{opt}} \tag{1.2}$$

where m is the mass of the muscle and $\rho = 1.056\, g/cm^3$ its density (taken from [55]).

1.3 Mathematical model of MTC

In the following sections we will see in detail how to mathematically formulate the passive and active properties of a muscle-tendon complex (MTC). This musculo-tendon model is taken from [28], and the human soleus muscle (fig. 1.16) has been taken as an example to illustrate the model with the parameters given in table 1.1.

1.3.1 Modeling the properties of tendons

As we see above, tendons behave like unidirectional non-linear springs:

$$F_{SE}(\varepsilon) = \begin{cases} F_{max} \cdot \left(\frac{\varepsilon}{\varepsilon_{ref}}\right)^2, & if\ \varepsilon > 0 \\ 0, & else \end{cases} \tag{1.3}$$

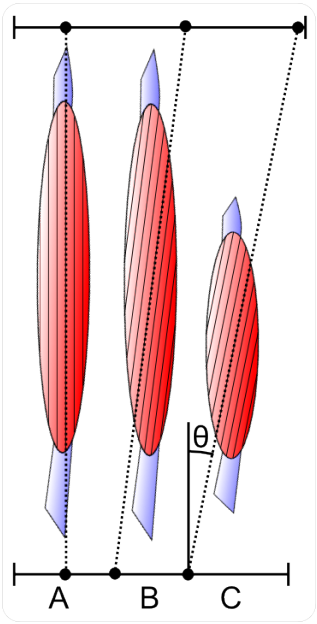


Figure 1.15: **A**: a non-pennated muscle. **B**: a pennated muscle. **C**: the angle of pennation θ increases when the muscle shortens.

Figure 1.16: The soleus muscle

ε_{ref}	0.04 [ ]
l_{slack}	0.26 $[m]$
F_{max}	4000 $[N]$
l_{opt}	0.04 $[m]$
c	ln(0.05) [ ]
w	0.56 [ ]
v_{max}	6.0 $[l_{opt}/s]$
K	5.0 [ ]
N	1.5 [ ]

Table 1.1: Parameters used for the human soleus muscle-tendon model

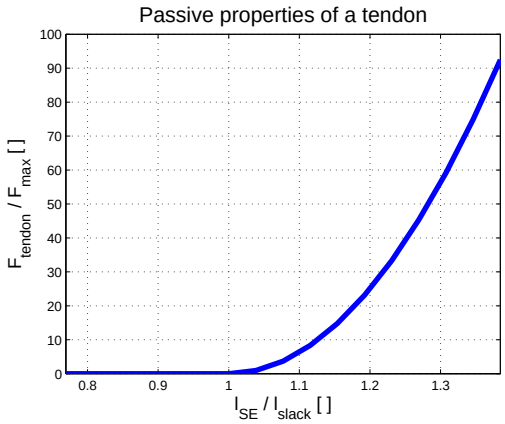


Figure 1.17: Dimensionless force generated by the SE in function of its dimensionless length

with

$$\varepsilon(l_{SE}) = \frac{l_{SE} - l_{slack}}{l_{slack}}$$

where F_{SE} is the force generated by the tendon (or equivalently called the serial elastic), F_{max} is a maximum force of this specific muscle, ε is the tendon strain, ε_{ref} is the tendon reference strain for which $F_{SE}(\varepsilon_{ref}) = F_{max}$, l_{SE} is the length of the serial elastic and l_{slack} is its slack length.

Figure 1.17 shows the dimensionless passive force of the Achilles tendon in function of its dimensionless length.

1.3.2 Modeling the properties of muscles

Passive elements

The passive elements of muscles are two unidirectional non-linear springs:

$$F_{HPE} = \begin{cases} F_{max} \cdot \left(\frac{l_{CE} - l_{opt}}{l_{opt} \cdot w}\right)^2, & if\ l_{CE} > l_{opt} \\ 0, & else \end{cases} \tag{1.4}$$

and

$$F_{LPE} = \begin{cases} -F_{max} \cdot \left(\frac{l_{CE} - l_{opt} \cdot (1-w)}{l_{opt} \cdot \sqrt{w/2}}\right)^2, & if\ l_{CE} \leq l_{opt} \cdot (1-w) \\ 0, & else \end{cases} \tag{1.5}$$

where F_{HPE} is the force generated by the high parallel elastic, F_{LPE} is the force generated by the low parallel elastic, F_{max} is a maximum force which is muscle specific,$(1-w)$ defines the dimensionless length of the LPE, as well as the reference strain of the high parallel elastic, l_{CE} is the muscle fibers length (CE stands for Contractile Element) and l_{opt} is the muscle fiber's length at rest.

Figure 1.19 shows the dimensionless force generated by the HPE and LPE springs in function of the muscle fibers dimensionless length.

Active element

The active elements of a muscle are the muscle fibers which generate a force depending on their activation A, on their length l_{CE} and on their velocity v_{CE} :

$$F_{MF} = F_{max} \cdot A \cdot f_l(l_{CE}) \cdot f_v(v_{CE}) \tag{1.6}$$

where F_{MF} is the force generated by the muscle fibers, F_{max} is a maximum force which is muscle specific, $A \epsilon [0;1]$ stands for the muscle activation, $f_l(l_{CE}) \epsilon [0;1]$ is a coefficient which depends on the length of the muscle fibers and $f_v(v_{CE}) \epsilon [0;1.5]$ is a coefficient which depends on the velocity of the muscle fibers.

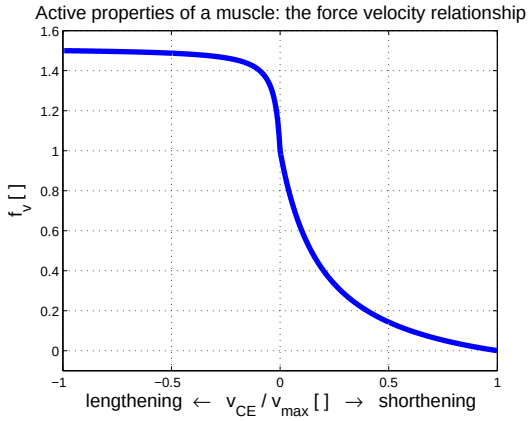


Figure 1.18: Velocity coefficient f_v in function of the velocity v_{CE} of the muscle fibers.

Velocity coefficient

The *velocity coefficient* $f_v(v_{CE})$ is computed as:

$$f_v(v_{CE}) = \begin{cases} \frac{v_{max}-v_{CE}}{v_{max}+K\cdot v_{CE}}, & \text{if } v_{CE}>0 \text{ (shortening)} \\ N+(N-1)\cdot\frac{v_{max}+v_{CE}}{7.56\cdot K\cdot v_{CE}-v_{max}}, & \text{else (lengthening)} \end{cases} \tag{1.7}$$

where v_{max} is the maximum velocity of muscle fibers, v_{CE} is the velocity of muscle fibers, K is a curvature parameter and N is the dimensionless amount of force $^{F_{MTC}}/_{F_{max}}$ reached at a lengthening velocity $v_{CE} = -v_{max}$ (F_{MTC} stands for the force generated by the Muscle Tendon Complex). The equation used for a shortening muscle is the same as the Hill's force-shortening velocity equation 1.1. Figure 1.18 shows the velocity coefficient f_v in function of the muscle fiber velocity v_{CE}.

Length coefficient

The *length coefficient* $f_l(l_{CE})$ is computed as:

$$f_l(l_{CE}) = \exp\left(c\cdot\left|\frac{l_{CE}-l_{opt}}{l_{opt}\cdot w}\right|^3\right) \tag{1.8}$$

where w is the width of the bell-shaped curve and c is a constant.
Figure 1.19 shows the passive and active forces generated by the muscle depending on its length.

Activation

As we already noted, the activation of muscle fibers is not the same as their stimulation. The stimulation $S\epsilon[0;1]$ is the signal sent by motoneurons from the spinal cord to the muscle fiber. The activation $A\epsilon[0;1]$ is the consequence of the stimulation in the sarcomeres. There are different ways to model the activation (see for example [34], which uses a second order differential equation and takes into account the fact that activation is faster than deactivation in biological systems) but for the sake of simplicity, we choose to use here a first order differential equation, taken from [31, 28]:

$$\frac{dA}{dt}(t+1) = \frac{S(t)-A(t)}{\tau} \tag{1.9}$$

with $\tau = 0.01$ is a time constant for both activation and deactivation.

Passive and active elements work together

The total force F_{tot} of a muscle is the sum of the passive and active forces:

$$F_{tot} = \overbrace{F_{HPE}+F_{LPE}}^{passive} + \overbrace{F_{MF}}^{active}$$

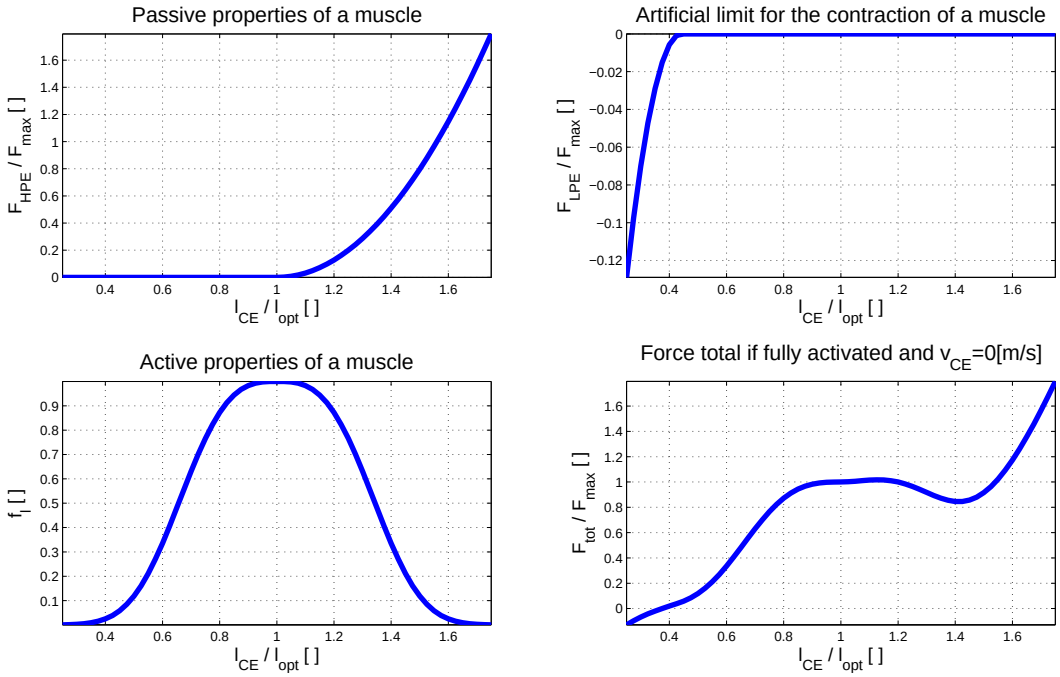


Figure 1.19: UP: dimensionless forces generated by the two passive spring LPE & HPE in function of their dimensionless length. BOTTOM LEFT: length coefficient f_l of the muscle fibers in function of its dimensionless length. BOTTOM RIGHT: total dimensionless force of a muscle if fully activated and held in an isometric condition ($v_{CE} = 0$) in function of its dimensionless length. We can see the similarity with figure 1.13.

If the muscle is fully activated ($A = 1.0$) the total dimensionless force of the muscle fibers in function of their dimensionless velocity and length is shown of figure 1.20 (note a positive force shrinks the muscle).

1.3.3 The intrinsic properties of muscles make them more stable against perturbation

In [54] it is shown that a periodic movement of a simplified one-dimensional hill-type muscle hopping model is more stable if the muscle model possesses the force$\Longleftrightarrow$velocity relation we see above.

To explain such behavior, lets imagine the situation shown on fig 1.21. In this sketch, an inverted pendulum is held in an upward position by two muscles. Both muscles have exactly the same properties, they are both fully activated ($A = 1$) and have the same length if the pole is vertical. If we push the mass on the right, then φ decreases, muscle 1 shortens and muscle 2 elongates. Since $v_{CE}^{muscle1} > v_{CE}^{muscle2}$ we have $f_v^{muscle1} < f_v^{muscle2}$ and then $F_{MF}^{muscle1} < F_{MF}^{muscle2}$. In that situation, the force$\Longleftrightarrow$velocity relation acts against the perturbation and makes the pole stable.

Furthermore, if the fiber's lengths are slightly shorter than their optimal lengths when the pole is vertical, an elongated muscle allows a larger active force production than a shortening one. This can be explained by the active force$\Longleftrightarrow$length relation: after the perturbation $l_{CE}^{muscle1} < l_{CE}^{muscle2}$, we have $f_l^{muscle1} < f_l^{muscle2}$ and therefore $F_{MF}^{muscle1} < F_{MF}^{muscle2}$. Against, the active force$\Longleftrightarrow$length relation in muscles acts against the perturbation and makes the pole more stable.

To summarize, the intrinsic properties of activated muscles allow them to be more stable against perturbations. In [54], such an event is called a *preflex*, since the intrinsic reactions of muscles against perturbations are even faster than reflexes.

1.4 Simulation of an MTC

Now that we can independently compute the force generated by tendons and muscle fibers, we will see how to mathematically compute the total force F_{MTC} of the MTC, the muscle fibers velocity v_{CE}, the muscle fibers length l_{CE} and the tendon length l_{SE}.

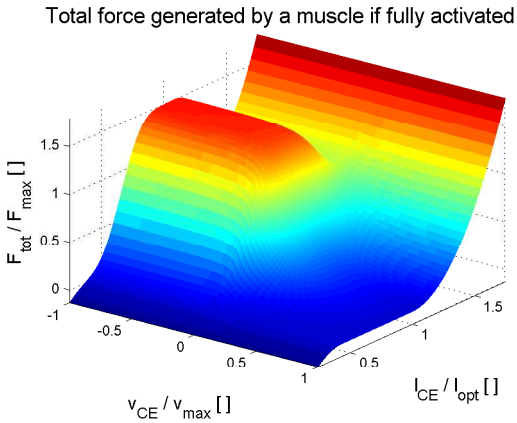


Figure 1.20: Total dimensionless force generated by the muscle in function of its dimensionless length and velocity.

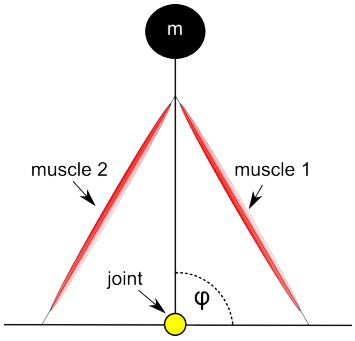


Figure 1.21: An inverted pendulum held by two activated muscles.

First, because the muscle fibers and the tendon are in series, we can write:

$$F_{tot} = F_{SE} \quad \Longrightarrow \quad F_{LPE} + F_{HPE} + F_{MF} = F_{SE} \tag{1.10}$$

This equilibrium can be understood by looking at the point between the serial elastic and the muscle fibers. Such a point does not move at equilibrium, which means $\vec{F_{tot}} + \vec{F_{SE}} = \vec{0}$ (the amount of force pulling this point to the left is equal to the amount of force pulling the point to the right).

Second, the following equality always holds:

$$l_{MTC} = l_{CE} + l_{SE} \tag{1.11}$$

1.4.1 Initialization of an MTC

Given an initial MTC length l_{MTC} and an initial muscle activation of zero, we want to compute l_{CE} and l_{SE} at equilibrium (so $v_{CE} = 0$).

- If $l_{MTC} \leq l_{slack} + l_{opt}$, the MTC is slack and we have thus:

$$\begin{cases} l_{CE} = l_{opt} \\ l_{SE} = l_{MTC} - l_{CE} \end{cases}$$

- If $l_{MTC} > l_{slack} + l_{opt}$, the MTC is stressed. Since the activation is zero and the low parallel elastic is not contracted, we have $F_{MF} = F_{LPE} = 0$. Because the muscle fibers are stretched and the tendon is not slack, we have:

$$\begin{gathered} F_{HPE} = F_{SE} \\ F_{max} \cdot \left(\frac{l_{CE} - l_{opt}}{l_{opt} \cdot w} \right)^2 = F_{max} \cdot \left(\frac{l_{SE} - l_{slack}}{l_{slack} \cdot \varepsilon_{ref}} \right)^2 \\ \frac{l_{CE} - l_{opt}}{l_{opt} \cdot w} = \frac{l_{SE} - l_{slack}}{l_{slack} \cdot \varepsilon_{ref}} \\ \frac{l_{MTC} - l_{SE} - l_{opt}}{l_{opt} \cdot w} = \frac{l_{SE} - l_{slack}}{l_{slack} \cdot \varepsilon_{ref}} \end{gathered}$$

Finally:

$$\begin{cases} l_{SE} = l_{slack} \cdot \frac{l_{opt} \cdot w + \varepsilon_{ref} \cdot (l_{MTC} - l_{opt})}{l_{opt} \cdot w + \varepsilon_{ref} \cdot l_{slack}} \\ l_{CE} = l_{MTC} - l_{SE} \end{cases}$$

1.4.2 Integration steps to compute $l_{SE}(t)$ and $l_{CE}(t)$

In order to compute the new length $l_{SE}(t)$, $l_{CE}(t)$ of the MTC at each time step given the velocity $v_{CE}(t)$ and the activation $A(t)$ of the muscle fibers, we proceed as follow:

- Compute $F_{SE}(t)$ with equation 1.3.
- Compute $F_{HPE}(t)$ with equation 1.4.
- Compute $F_{LPE}(t)$ with equation 1.5.
- Compute $f_l(t)$ with equation 1.8. For reasons of numerical stability, f_l should not be smaller than 0.001.
- Compute $F_{MF}(t)$ from equation 1.10.
- Compute the activation $A(t)$ from equation 1.9. We have to limit the activation in the range $A \epsilon [0.01; 1.0]$ for reasons of stability.
- Compute $f_v(t)$ from equation 1.6. Note that f_v should always be in the range $f_v \epsilon [0.0; 1.5]$, as shown in figure 1.18.
- Compute $v_{CE}(t)$ from equation 1.7:

$$v_{CE}(t) = \begin{cases} v_{max} \cdot \frac{1 - f_v(t)}{1 + K \cdot f_v(t)}, & if\ f_v(t) < 1.0 \\ v_{max} \cdot \frac{f_v(t) - 1}{7.56 \cdot K \cdot (f_v(t) - N) + 1 - N}, & else \end{cases}$$

- Compute $l_{CE}(t) = l_{CE}(t-1) + v_{CE}(t) \cdot dt$.
- Compute $l_{SE}(t) = l_{MTC}(t) - l_{CE}(t)$ from equation 1.11.

In this last calculation, we need to know l_{MTC}. The following section 1.4.3 explains how we can deduce l_{MTC} from the position of the bones.

1.4.3 An example of a joint actuated by an MTC

Now we have a numerical model of MTCs, we can use and place them on the leg of a robot. But we first need to understand how such a complex can actuate a joint.

As an example, lets look at figure 1.22 where an MTC is attached to a wall and to an arm with the following assumptions:

- muscle fibers are not stimulated, so only passive forces occur in the MTC: $F_{SE} = F_{LPE} + F_{HPE}$.
- the arm (blue) is massless.

In such a situation, the only torque acting on the arm is the one from the MTC which acts passively. If initially elongated, the MTC shrinks until $l_{SE} = l_{slack}$ and $l_{CE} = l_{opt}$, which defines the rest length of the MTC: $l^0_{MTC} = l_{opt} + l_{slack}$. At this rest length, $F_{SE} = F_{HPE} = F_{LPE} = 0$.

Now, with φ being the angle of the joint, lets define two constant angles:

1. φ_{max} is the φ angle required to have an orthogonal angle between the arm and the MTC ($\beta = \pi/2$).
2. φ_{ref} is the φ angle required to have an MTC length of l^0_{MTC}.

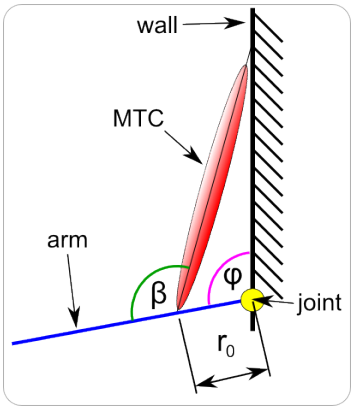


Figure 1.22: A muscle attached to a wall.

Computing the torque τ at the joint

To update the angle φ of the arm, we need to know the torque τ the MTC applies to the arm:

$$\tau = r_0 \cdot F^{\perp}_{MTC}$$

where r_0 is the distance between the joint and the point of attached of the MTC on the arm, and $F^{\perp}_{MTC}$ is the orthogonal force the MTC applies to the arm. This orthogonal force is maximum for $\varphi = \varphi_{max}$ and zero if $\beta = 0$ or $\beta = \pi$. In other words, the torque will depend on r_0, φ and F_{MTC}, so we can write:

$$\tau = r(\varphi) \cdot F_{MTC} \tag{1.12}$$

with

$$r(\varphi) = r_0 \cdot \cos(\varphi - \varphi_{max}) \tag{1.13}$$

.

The force $F_{MTC} = F_{SE}$ can be computed easily if we know l_{SE} (see equation 1.3). Because $l_{SE} = l_{MTC} - l_{CE}$ we need to compute the MTC length l_{MTC}.

Computing l_{MTC}

Any small change of the angle φ changes l_{MTC}. More exactly:

$$dl_{MTC} = r(\varphi) \cdot d\varphi \tag{1.14}$$

To understand why we use $r(\varphi)$ in this calculation, imagine $\beta = \pi/2$ (which defines the angle φ_{max}). For any change $d\varphi$ we would then have $dl_{MTC} = r_0 \cdot d\varphi$. On the contrary, if $\beta = 0$ or $\beta = \pi$, we would have $dl_{MTC} = 0$. Using equation 1.13 is a good way to simulate the change of MTC length dl_{MTC} given an angle φ.

Now, we can integrate equation 1.14:

$$\int_{l^0_{MTC}}^{l} dl_{MTC} = r_0 \cdot \int_{\varphi_{ref}}^{\varphi} \cos(\varphi - \varphi_{max}) \cdot d\varphi$$

$$\triangle l_{MTC} = l_{MTC} - l^0_{MTC} = r_0 \cdot [\sin(\varphi - \varphi_{max}) - \sin(\varphi_{ref} - \varphi_{max})]$$

This last equation shows how we can compute the length l_{MTC} given the angle φ of a joint. Since muscle fibers are pennated, the length change they experience is reduced. To account for this effect, we add a pennation factor $\rho \epsilon [0; 1]$:

$$\triangle l_{MTC} = \rho \cdot r_0 \cdot [\sin(\varphi - \varphi_{max}) - \sin(\varphi_{ref} - \varphi_{max})] \tag{1.15}$$

It is worth to note the MTC length l_{MTC} does not reflect the total length of the whole muscle plus the whole tendon, but only the length of one tendon in series with a one muscle fiber, which is enough to compute the torque τ.

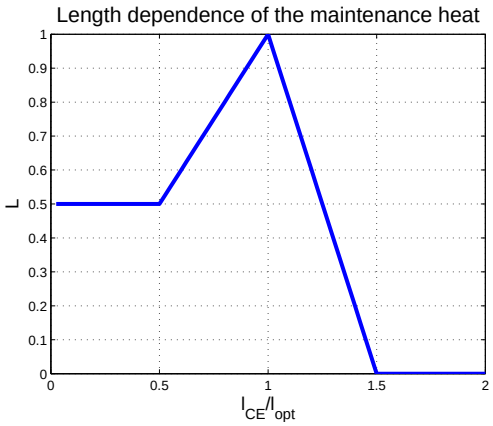


Figure 1.23: The piecewise function $L\left(l_{CE}^{normalised}(t)\right)$

Integration steps

To update the position of a bone at each time step, we do the following:

- Given a joint angle φ, we compute the MTC length $l_{MTC}(t) = l_{MTC}^0 + \triangle l_{MTC}$ with the help of equation 1.15.
- We then compute all the forces, velocities and lengths of tendons and muscle fibers (see section 1.4.2) which allow us to know $F_{MTC}(t) = F_{SE}(t) = F_{tot}(t)$.
- We compute the torque $\tau(t)$ acting on the joint with equation 1.12.
- We update the angle φ of the joint according to the torque $\tau(t)$.

1.5 A model of energy consumption in muscle

In this section, we will see how to compute the energy consumption in a muscle, which will be useful later in this project.

[20] gives a model of such energy consumption made of five terms, and we choose to use only the two most relevant:

- The *maintenance heat*, which is the heat liberated during the recycling of actin-myosin cross-bridges. The recycling depends on the activation $A(t)$. The cross-bridges within the whole muscle depend on the length of the muscle fibers (see fig. 1.13) times the number of muscle fibers. The number of muscle fibers can be estimated with the mass of the muscle. At any time step, we can compute the maintenance heat for a given muscle as:

$$dM = L\left(l_{CE}^{normalised}(t)\right) \cdot m \cdot C_M \cdot A(t)$$

where $m\,[kg]$ is the mass of the muscle, $C_M = 100\,[J/Kg \cdot s]$ is a constant we evaluate from [20], $A(t)$ is the activation of the muscle and $L\left(l_{CE}^{normalised}(t)\right)$ is a piecewise function shown on figure 1.23.

- The *work*, which is the energy consumed for any mechanical displacement, can be computed for each muscle as:

$$dW = \begin{cases} F_{MF} \cdot dl_{CE} & \text{if the muscle is shortening} \\ 0 & \text{else} \end{cases}$$

Finally, the total energy consumption is given by:

$$E = \int_{t_0}^{t} \left(\sum_{x \in all\ muscles} dW_x + dM_x \right) dt$$

where x is one of the seven MTC.

Chapter 2

Sensory - Motor Feedback

2.1 Some aspects of the walking gait

We generally call a *step* the interval from the left heel strike to the right heel strike. A *stride* is the interval between two successive heel strike of the same leg, therefore one stride is two steps or one *gait cycle* (fig.2.1).

One stride can be divided in two phases: the *stance* phase is the part of the gait cycle where the foot touch the ground and the *swing* phase is the part of the gait cycle where the foot is kept in the air.

However, it is possible to further subdivide a walking gait cycle, as illustrated on figure 2.2. Each phase of the gait can be associated with specific activation of muscle groups as shown on figure 2.3.

In this project, a gait cycle is divided in three phases: stance, double support finishing stance and swing. We will see later these three subdivisions of the gait cycle are sufficient to allow a humanoid to walk.

From the seventeenth century, walking and running have been seen as two different mechanical behaviors. Walking is generally seen as an inverted pendulum and running as a spring-mass model. However, if the mass-sping model model efficiently match with running human data, the inverted pendulum model does not fit with reality (see fig. 2.4 for an illustration).

[29] shows both gaits can be explained with the spring-mass model. Proper tuning of the spring stiffness and angle of attack of the leg allows a spring-mass model to walk and fit nicely with human walking ground reaction force data.

It is therefore the spring-like property of the legs which design the two typical bumps of the ground reaction force observed in walking gait. The first bump occurs in early stance and is mainly due to the activation of the vastus muscle group to absorb and decelerate the forward movement. The second bump is mainly created by the activation of soleus and gastrocnemius muscles in late swing to push the body forward [46]. Therefore, there exist a division of tasks between the vastus muscle group and the soleus-gastrocnemius muscle groups which is illustrated in figure 2.3.

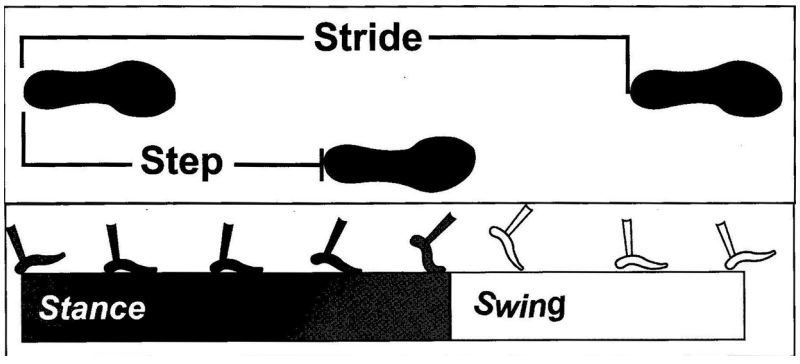


Figure 2.1: Top: Step and stride Bottom: division of a stride in two phases, the stance and swing phase. Taken from [3].

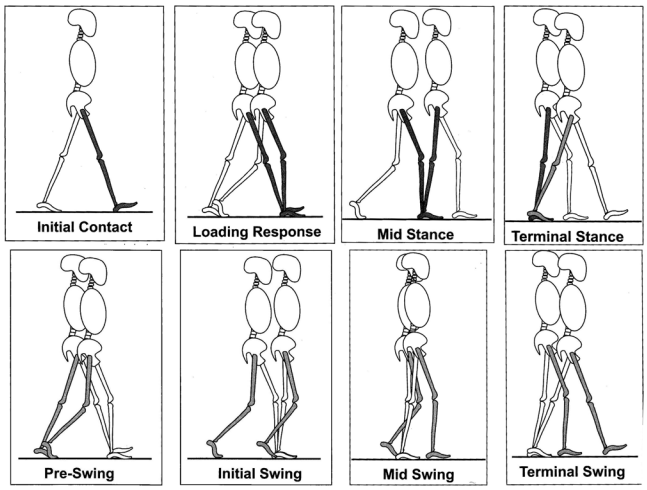


Figure 2.2: Subdivision of the gait cycle as proposed in [3].

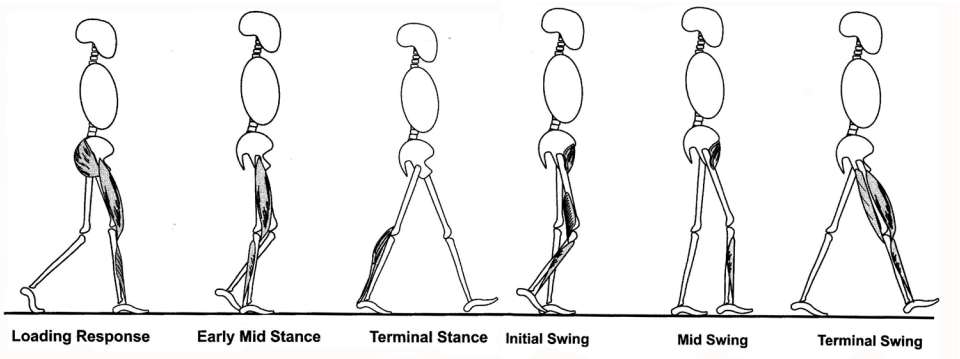


Figure 2.3: Activation of muscle groups in function of the phase of the gait cycle. Note we do not simulate the short head biceps femoris in initial swing in our model, however every other muscle is simulated. Taken from [3].

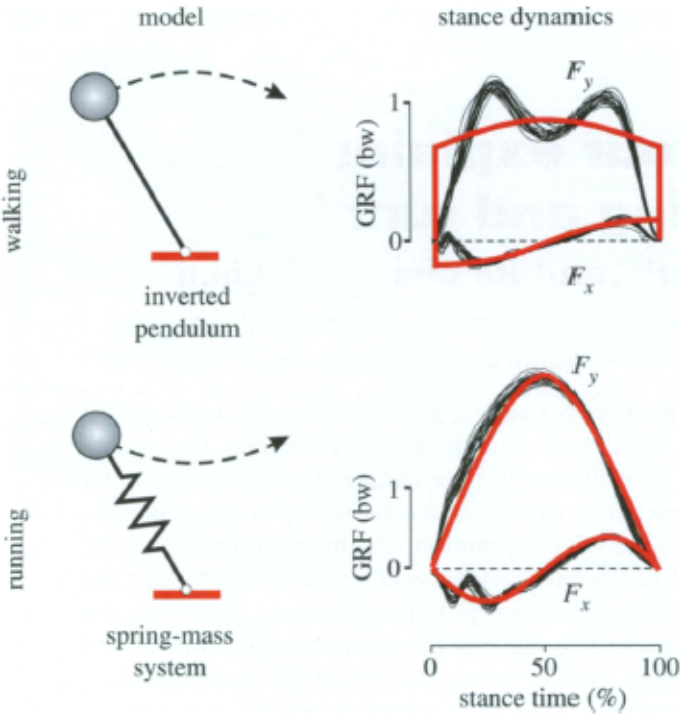


Figure 2.4: The two mechanical paradigms of walking (an inverted pendulum) and running (a spring-mass model). The thick red trace shows the GRF of the model and the thin black are experimental human data. We clearly see how the spring-mass model effectively match reality but how the inverted pendulum model does not fit with experimental data. Taken from [29].

2.2 Introduction to feedback

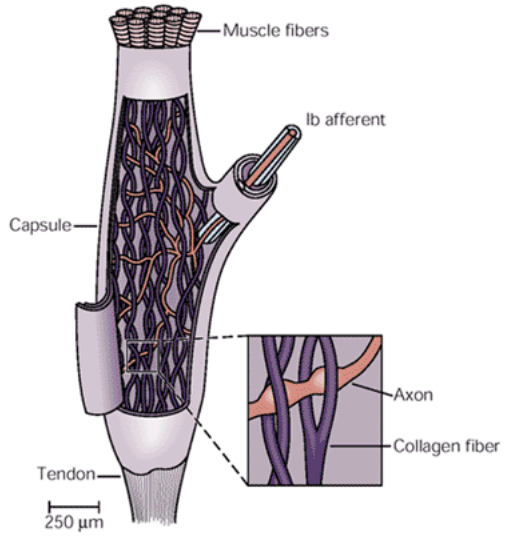
As stated earlier, to make the robot walk we need to stimulate the muscle fibers in a proper manner. What actually stimulates the muscle fibers? And on what information does it rely to know how much and when a muscle fiber should be stimulated? These are the two main questions we answer in this section.

Let's imagine a person who is walking and touching the ground with his right heel. Some muscles of the right leg have to be activated in order to bear the weight of this man. As an example, let's think about the vastus muscles group which is stimulated at the heel-ground contact. The sensory information from the heel is transmitted by a neuron (cutaneous Ib afferent fiber) toward the dorsal part of the spinal cord [7]. There, several connections can be made with interneurons (*plurisynaptic* connections) or directly to α-motoneurons (*monosynaptic* pathway). Motoneurons exit the spinal cord through its ventral side and run in the body to stimulate the vastus muscle fibers.

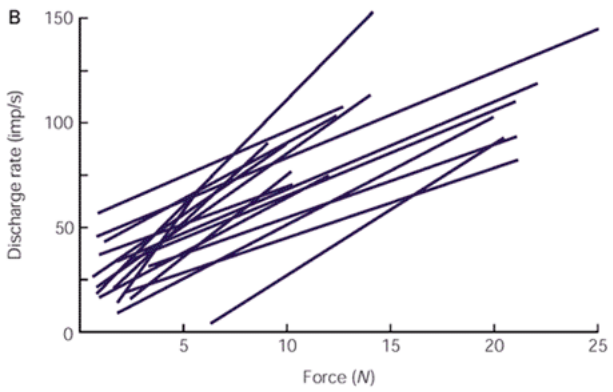
We say a connexion is *homonymous* if sensory information comes from a particular MTC and is sent back to the same MTC. On the contrary, if information from a particular MTC is fed back to another MTC, we say the connexion is *heteronymous*.

To summarize, sensory neurons exist which relay information from sensory organs to the spinal cord. There, some neural networks process these inputs and send an appropriate command to muscle fibers through α-motoneurons. This whole loop is so fast that the muscles are activated before the information about the heel-ground contact has time to reach our brain.

Some reflexes are innate, which means we have them by default at birth, they are genetically encoded. Some other reflexes are highly trained during our life, for example, people practicing fencing will develop some new reflexes to dodge attacks.



(a) An illustration of the Golgi tendon organ, taken from [7]



(b) The discharge rate of a population of Golgi tendon organs signals the force in a MTC. Taken from [7]

Figure 2.5: Biological systems

2.3 Sensory input and motor output

2.3.1 The Golgi tendon organ feels the force of the MTC

As we already see on figure 1.2, a Golgi tendon organ (GTO), or equivalently a capsule, feels the force acting on the tendon. Since tendon and muscle fibers are connected in series, the Golgi tendon organs feel the overall force of the MTC.

In biological systems, GTOs are made of collagen fibers (figure 2.5a). When a force stretches them, collagen fibers compress intermix axons, which in turn fire through the Ib afferent neuronal sensory pathway. It has been shown that the firing rate is linearly coupled with the force the capsule experiences [7]. Figure 2.5b illustrates such linear regression on the soleus muscle of a cat.

We can therefore model in a first approximation the sensory-motor force feedback as:

$$S_y(t) = G \cdot F_{MTC,x}(t)$$

where $S_y(t)$ is the stimulation of the MTC y which depends on the force of the MTC x and G is a gain constant.

We consider here a linear relationship between stimulation and force feedback, however Ib afferent axon can make plurisynaptic connexion with α-motoneuron in the spinal cord, so it is very likely the relation between $S(t)$ and $F_{MTC}(t)$ is non-linear, as usually in biological systems.

2.3.2 The muscle spindles feel the length of the muscle fibers

We remember how muscle spindles are embedded in the muscle mass (figure 1.2). Since muscle spindles are connected in parallel with muscle fibers, they are always stretched together. As illustrated on figure 2.6, Ia afferent axons are rolled around muscle spindles and increase their firing rate if elongated. In other words, Ia afferent pathway feels the length of the muscle spindles which in turn reflect the length l_{CE} of the muscle fibers.

In order to be always under tension, muscle spindles are connected in series with contractile regions. These contractile regions are controlled by efferent axons, the γ-motoneurons. Figure 2.6B shows different type of muscle spindles as well as different type of γ-motoneurons and afferent pathways. There are required to separately detect *change in length* and *static length*. So muscle spindles can also provide information about the velocity v_{CE} of the muscle fibers (more details in [7]).

In this project, we will model the muscle fibers' length feedback as:

$$S_y(t) = G \cdot \left(l_x^{CE}(t) - l^{off}\right)$$

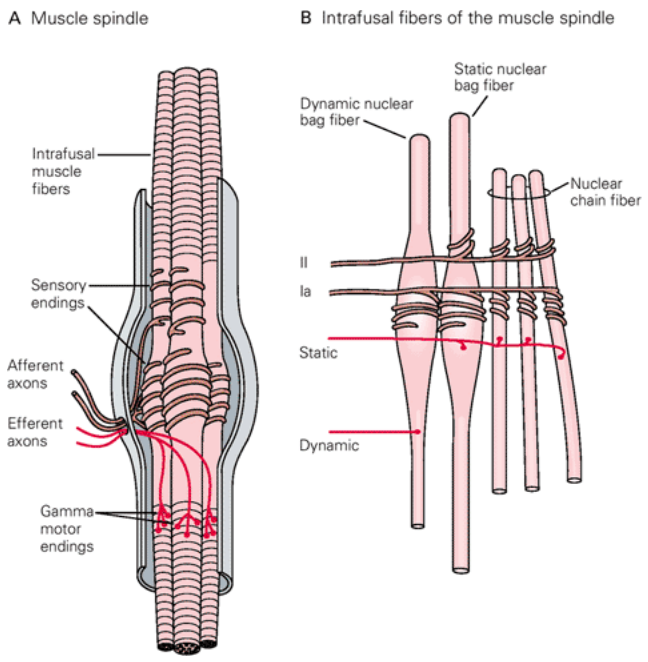


Figure 2.6: Illustration of muscle spindles. Taken from [7]

where $S_y(t)$ is the stimulation of the MTC y which depends on the length $l^{CE}(t)$ of the MTC x. l^{off} is a reference length and G a gain constant.

2.3.3 Other sensory organs

We see two sensory organs (GTOs and muscle spindles) out of many others. The body of an animal possesses a lot of sensors which can feel the environment and themselves. Think about the skin, which is a surface of high definition sensors. Eyes, ears, nose, mouth and the vestibular organs are examples of the richness of sensory capacity animals possess. Others sensory organs feel the inside of our body and relay information to proprioceptive neurons. We do not have consciousness of the major flux of them, since they are mainly used to tune our heart-beat, to breath, to digest, etc... Plasticity within the neural networks allow our nervous systems to integrate and process properly this rich flow of inputs.

It is clear sensors from the feet are highly required for stabilization while walking. Unfortunately, actual robots lack such high definition sensory skins. But even if they would possess some, could we know how to use such high sensory flow?

In this project, we will see how a walking gait can rely only on force and length feedback from MTCs and on two force sensors (one at the heel and one at the ball) under each foot.

2.4 Walking with feedback information

2.4.1 Feedback rules

A whole gait cycle is achieved if one leg (the right leg for example) starts to walk at a given position (lets say when the heel stroke the ground) and comes back at this position. We can divide a cycle in two phases:

- the *swing* phase: when the foot is in the air.
- the *stance* phase: when the foot touch the ground.

The feedback rules (taken from [31]) are divided in two sets: one for the stance phase and one for the swing phase (see table 2.1). Note we do not implemented feedback delay for the sake of simplicity.

Stance

Support of the body weight In stance, the extensor muscles have to be stimulated to bear the body weight. A positive force feedback (observed in cats [47] and modeled in [28]) of the extensor muscles can generate a compliant leg behavior. Therefore, the soleus, gastrocnemius and vastus muscle groups are self-stimulated by their own force feedback ($G_x \cdot \frac{F_x}{F_x^{max}}$). This positive force feedback of the soleus and gastrocnemius helps the final push off at the end of the stance phase.

The tibialis muscle group is inhibited by the force feedback of the soleus ($-G_{SOL-TA} \cdot \frac{F_{SOL}}{F_{SOL}^{max}}$) to avoid a useless co-contraction of both soleus and tibialis.

Hyper extension of the knee We prevent the hyper extension of the knee by inhibiting the vastus muscle group if the knee angle exceeds some threshold ($-k_\varphi \cdot \left(\varphi_{knee} - \varphi_{knee}^{off}\right)$, if $\varphi_k > \varphi_k^{off}$ and if $\dot{\varphi}_k > 0$, with $\varphi_k^{off} = 170°$).

Trunk balance The trunk balance is controlled with a proportional and differential strategy, where θ is the angle between the trunk and the gravitation and $\theta^{ref} = 6°$ is a reference angle: $k_p \cdot \left(\theta - \theta^{ref}\right) + k_d\dot{\theta}$, where k_p and k_d are the proportional and differential gain constants. Since the trunk can be controlled only if the leg is bearing weight, we have: $\left[k_p \cdot \left(\theta - \theta^{ref}\right) + k_d\dot{\theta}\right] \cdot k_{bw} \cdot \frac{\left\|F_{leg}^{ipsi}\right\|}{F_g}$ where k_{bw} is a gain constant, $\left\|F_{leg}^{ipsi}\right\|$ is the amount of weight the leg is bearing and F_g is the total weight of the humanoid.

Double support

When the two feet are both touching the ground, we need to transfer the weight from one leg to the next. To this end, we modulate the stimulation of the leg which is finishing stance to initiate the swing phase.

First, we inhibit the vastus muscle with $-k_{bw} \cdot \frac{\left\|F_{leg}^{contra}\right\|}{F_g}$.

Second, we inhibit the gluteus muscle by a fixed amount $-\triangle S$.

Finally, we excite the hip flexor muscle group with the same fixed amount $\triangle S$.

Swing phase

Protraction of the leg The hip flexor muscle group has to be stimulated to protracte the leg at the begining of the swing phase: $G_{HF} \cdot \left(l_{HF}^{CE} - l_{HF}^{off} \cdot l_{HF}^{opt}\right)$. Since the required protraction speed depends on the trunk's forward lean, we need to bias such stimulation by the trunk angle at take off (θ^{TO}) : $k_{lean} \cdot \left(\theta^{TO} - \theta^{ref}\right)$ where k_{lean} is a gain constant.

Retraction of the leg The retraction of the leg improves the mechanical self-stability against disturbances [33]. Three reflexes are required.

- Inhibition of the hip flexor muscle group by the stretch reflex of the hamstring: $-G_{HAM-HF} \cdot \left(l_{HAM}^{CE} - l_{HAM-HF}^{off} \cdot l_{HAM}^{opt}\right)$.
- Gluteus and hamstring are both self stimulated by a force feedback: $G_x \cdot \frac{F_x}{F_x^{max}}$

Foot clearance In swing, we need to clear the foot from the ground by exciting the tibialis muscle with a length feedback: $G_{TA} \cdot \left(l_{TA}^{CE} - l_{TA}^{off} \cdot l_{TA}^{opt}\right)$, where $l_{TA}^{off} \cdot l_{TA}^{opt}$ is the length the muscle fibers should reach. Such a reflex allows the foot to initiate the next stance phase with the heel.

2.4.2 Rescaling the parameters

We see on table 2.1 that we need to tune 25 parameters. Finding these parameters will be discussed later in chapter 3.

Some parameters have small values (ranging for example from 0.01 to 0.1) and others have larger values (ranging from 0.0 to 20.0 for example). Therefore, applying a small modification to a large parameter would have negligible effect. On the contrary, the same modification applied to a small parameter could lead to a large change in the

Stance

SOL $$S_{SOL} = S^0_{SOL} + G_{SOL} \cdot \frac{F_{SOL}}{F^{max}_{SOL}}$$

TA $$S_{TA} = S^0_{TA} + G_{TA} \cdot \left(l^{CE}_{TA} - l^{off}_{TA} \cdot l^{opt}_{TA}\right) - G_{SOL-TA} \cdot \frac{F_{SOL}}{F^{max}_{SOL}}$$

GAS $$S_{GAS} = S^0_{GAS} + G_{GAS} \cdot \frac{F_{GAS}}{F^{max}_{GAS}}$$

VAS $$S_{VAS} = S^0_{VAS} + G_{VAS} \cdot \frac{F_{VAS}}{F^{max}_{VAS}} - k_{\varphi} \cdot \overbrace{\left(\varphi_{knee} - \varphi^{off}_{knee}\right)}^{if \begin{cases} \varphi_k > \varphi^{off}_k \quad \text{and} \\ \dot{\varphi}_k > 0 \end{cases}} - \overbrace{k_{bw} \cdot \frac{\left\| F^{contra}_{leg} \right\|}{F_g}}^{if \begin{cases} \text{double support and} \\ \text{finishing stance} \end{cases}}$$

HAM $$S_{HAM} = S^0_{HAM} + \left[k_p \cdot \left(\theta - \theta^{ref}\right) + k_d \dot{\theta}\right] \cdot k_{bw} \cdot \frac{\left\| F^{ipsi}_{leg} \right\|}{F_g}$$

GLU $$S_{GLU} = S^0_{GLU} + \left[0.68 \cdot k_p \cdot \left(\theta - \theta^{ref}\right) + k_d \dot{\theta}\right] \cdot k_{bw} \cdot \frac{\left\| F^{ipsi}_{leg} \right\|}{F_g} \overbrace{-\triangle S}^{if \begin{cases} \text{double support and} \\ \text{finishing stance} \end{cases}}$$

HF $$S_{HF} = S^0_{HF} - \left[k_p \cdot \left(\theta - \theta^{ref}\right) + k_d \dot{\theta}\right] \cdot k_{bw} \cdot \frac{\left\| F^{ipsi}_{leg} \right\|}{F_g} \overbrace{+\triangle S}^{if \begin{cases} \text{double support and} \\ \text{finishing stance} \end{cases}}$$

Swing

SOL $$S_{SOL} = S^0_{SOL}$$

TA $$S_{TA} = S^0_{TA} + G_{TA} \cdot \left(l^{CE}_{TA} - l^{off}_{TA} \cdot l^{opt}_{TA}\right)$$

GAS $$S_{GAS} = S^0_{GAS}$$

VAS $$S_{VAS} = S^0_{VAS}$$

HAM $$S_{HAM} = S^0_{HAM} + G_{HAM} \cdot \frac{F_{HAM}}{F^{max}_{HAM}}$$

GLU $$S_{GLU} = S^0_{GLU} + G_{GLU} \cdot \frac{F_{GLU}}{F^{max}_{GLU}}$$

HF $$S_{HF} = S^0_{HF} + G_{HF} \cdot \left(l^{CE}_{HF} - l^{off}_{HF} \cdot l^{opt}_{HF}\right) - G_{HAM-HF} \cdot \left(l^{CE}_{HAM} - l^{off}_{HAM-HF} \cdot l^{opt}_{HAM}\right) + k_{lean} \cdot \left(\theta^{TO} - \theta^{ref}\right)$$

Table 2.1: Feedback rules

N°	Name of the parameter	Min	Max	N°	Name of the parameter	Min	Max
1	G_{SOL}	0.8	1.6	14	k_{bw}	0.8	1.4
2	G_{TA}	1.0	10.0	15	k_p	1.5	3.0
3	G_{SOL-TA}	0.1	0.8	16	k_d	0.15	0.4
4	G_{GAS}	0.8	1.6	17	$\triangle S$	0.1	1.0
5	G_{VAS}	0.95	1.6	18	S^0_{SOL}	0.01	0.03
6	G_{HAM}	0.3	1.0	19	S^0_{TA}	0.01	0.03
7	G_{GLU}	0.2	0.85	20	S^0_{GAS}	0.01	0.03
8	G_{HF}	0.2	1.0	21	S^0_{VAS}	0.01	0.1
9	G_{HAM-HF}	15.0	30.0	22	S^0_{HAM}	0.01	0.1
10	l^{off}_{TA}	0.0	0.71	23	S^0_{GLU}	0.01	0.1
11	l^{off}_{HF}	0.2	0.8	24	S^0_{HF}	0.01	0.1
12	l^{off}_{HAM-HF}	0.75	1.0	25	k_{lean}	0.5	2.0
13	k^{knee}_{φ}	1.5	2.5				

Table 2.2: Range of each parameters

N°	Name of the parameter	From [31]	Rescaled	N°	Name of the parameter	From [31]	Rescaled
1	G_{SOL}	1.2	0.50	14	k_{bw}	1.2	0.67
2	G_{TA}	1.1	0.01	15	k_p	1.91	0.27
3	G_{SOL-TA}	0.3	0.29	16	k_d	0.25	0.40
4	G_{GAS}	1.1	0.37	17	$\triangle S$	0.25	0.17
5	G_{VAS}	1.15	0.31	18	S^0_{SOL}	0.01	0.00
6	G_{HAM}	0.65	0.50	19	S^0_{TA}	0.01	0.00
7	G_{GLU}	0.4	0.31	20	S^0_{GAS}	0.01	0.00
8	G_{HF}	0.35	0.19	21	S^0_{VAS}	0.09	0.89
9	G_{HAM-HF}	4.0	-0.73	22	S^0_{HAM}	0.05	0.44
10	l^{off}_{TA}	0.71	1.00	23	S^0_{GLU}	0.05	0.44
11	l^{off}_{HF}	0.6	0.67	24	S^0_{HF}	0.05	0.44
12	l^{off}_{HAM-HF}	0.85	0.40	25	k_{lean}	1.15	0.43
13	k^{knee}_{φ}	1.5	0.50				

Table 2.3: Parameters and rescaled parameters used by H.Geyer in [31]

reflex feedback. It is therefore preferable to rescale every parameter from their respective $[min; max]$ to $[0; 1]$ with the following rescaling rule:

$$x^{rescaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Table 2.2 summarizes these $[min; max]$ ranges we manually tuned.

Since this work is mainly inspired from H. Geyer, we copy in table 2.3 the parameters and rescaled parameters taken from [31].

Chapter 3

Modeling a humanoid and its experimental framework

3.1 A humanoid model

In this project, we only rely on simulation, no work has been done on a real robot. We choose to work with the physics based simulator software Webots [1] and to create the same bipedal model as H.Geyer in [31] (with a different look however), except for the foot where we add a more realistic ankle (L_{ankle}, see fig. 3.2b) and a curved part at the heel and ball (we think that could help the agent to rotate on the tip of its feet since no toes have been implemented). Figure 3.1 shows our bipedal humanoid. Note the capsule-like trunk and the cone-like head are only shapes without any physical significance. The weight of the trunk is distributed in a 80 cm height and 20 cm wide and depth parallelepiped.

Each leg is made of three segments (a thigh, a shin and a foot) and of three joints (hip, knee and ankle). The name of the segments, their positions and their joints angles are shown on figure 3.2b.

A leg can now be controlled with MTCs if we define for each of them l_{opt}, l_{slack}, v_{max}, F_{max}, ρ and if we define for each joint-MTC interaction r_0, φ_{ref} and φ_{max}. Figure 3.2a illustrates where we place seven different MTCs on one leg of our bipedal humanoid. In biological systems, one MTC is made of several muscles, therefore an MTC simulates a group of muscles rather than a single one.

The names of the MTCs are the following:

Extensor	Flexor	Biarticular
SOL Soleus muscle group	TA Tibialis muscle group	GAS Gastrocnemius muscle group
VAS Vasti muscle group	HF Hip flexor muscle group	HAM Biarticular hamstring muscle group
GLU Gluteus muscle group		

We provide each foot of the humanoid with two touch sensors (one at the heel and the other at the ball, both are sensitive only on the curved part of the foot).

Segments parameters

The length of the segments are given in table 3.1 and the mass, center of mass (CoM) and inertia in table 3.2. The positions of the CoM have been measured from the hip for the trunk, from the knee for the thigh, from the ankle for the shin and from the hind foot (without the curved part) for the foot.

L_{trunk}	L_{thigh}	L_{shin}	L_{ankle}	$L_{hind\,foot}$	$L_{fore\,foot}$	R_{curve}
0.80	0.50	0.50	0.10	0.02	0.14	0.025

Table 3.1: Lengths of the segments of the humanoid $[m]$

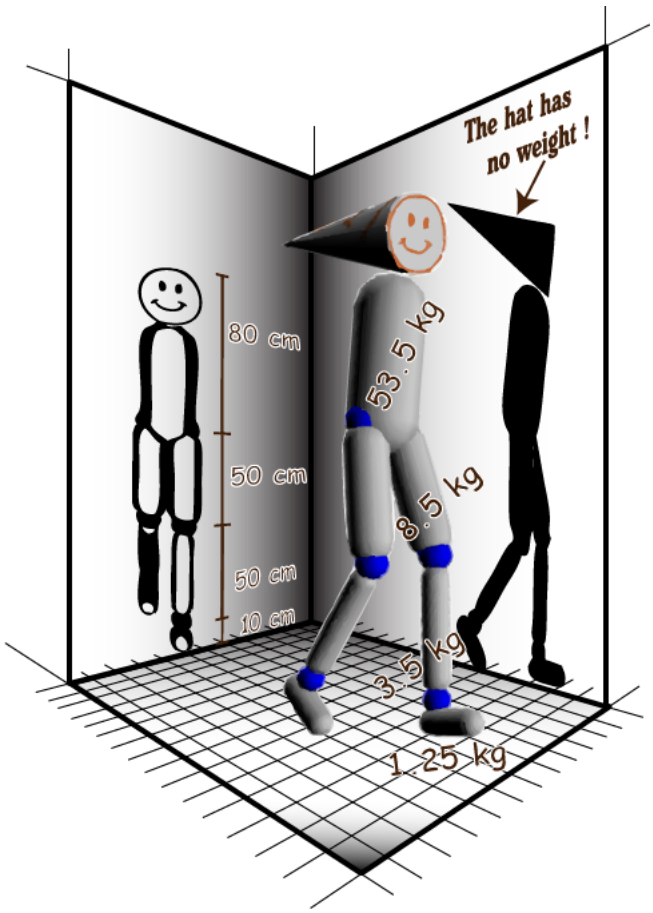
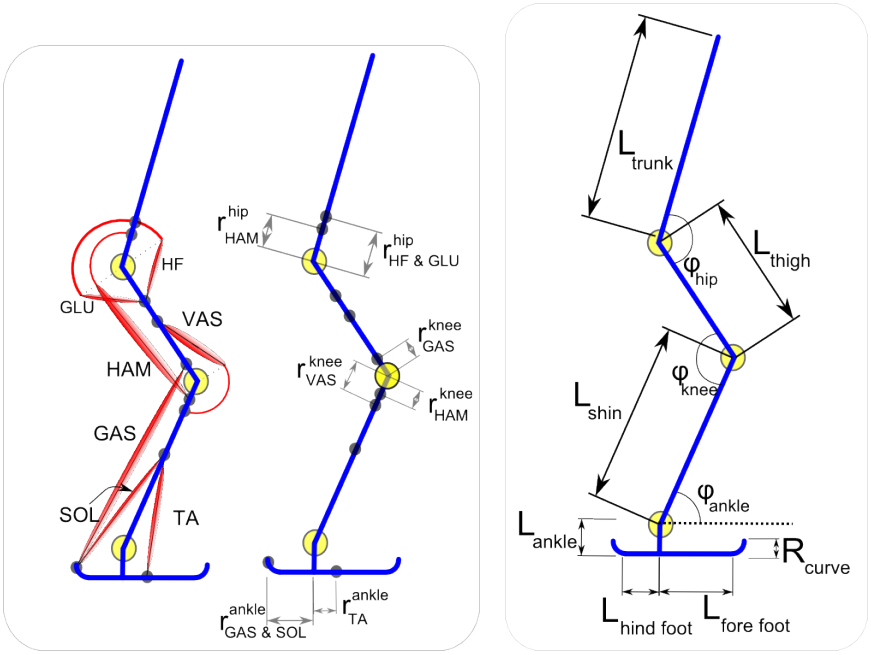


Figure 3.1: An illustration of our simulated humanoid



(a) Names and positions of MTCs

(b) Lengths of the different segments

Figure 3.2: A bipedal humanoid model

	Trunk	Thigh	Shin	Foot
Mass: $m\,[kg]$	53.500	8.500	3.500	1.250
CoM: $x\,[m]$	0.350	0.300	0.300	0.020
Inertia: $I\,[kg \cdot m^2]$	3.000	0.150	0.050	0.0050

Table 3.2: Mass, center of mass (CoM) and inertia of the segments

$\varepsilon_{ref}\,[]$	$c\,[]$	$w\,[]$	$K\,[]$	$N\,[]$	$\tau\,[]$	$\rho\,[^{kg}/_{m^3}]$	$P\,[^{N}/_{m^2}]$
0.04	$\log(0.05)$	0.56	5.0	1.5	0.01	1060	$2.5{\cdot}10^5$

Table 3.4: Common parameters for every MTC

MTCs parameters

The parameters specific for each MTC are given in table 3.3. Given the maximum force F_{max}, we can compute the PCSA for each muscle ([31] assumes a force of 25 $[^{N}/_{cm^2}]$) and deduce their mass from equation 1.2. The mass of each muscle will be used to estimate their energy consumption.

Table 3.4 summarizes the common parameters for every MTC.

	SOL	TA	GAS	VAS	HAM	GLU	HF
Maximal force: $F_{max}\,[N]$	4000	800	1500	6000	3000	1500	2000
Maximal speed: $v_{max}\,[^{l_{opt}}/_{s}]$	6	12	12	12	12	12	12
Optimal muscle fibers length: $l_{opt}\,[m]$	0.04	0.06	0.05	0.08	0.10	0.11	0.11
Optimal tendon length: $l_{slack}\,[m]$	0.26	0.24	0.40	0.23	0.31	0.13	0.10
Pennation factor: $\rho\,[\,]$	0.5	0.7	0.7	0.7	0.7	0.5	0.5
Mass:$m\,[kg]$	1.36	0.29	0.45	2.91	1.82	1.40	1.87

Table 3.3: Parameters specific to a given MTC

Joints-MTCs parameters

The parameters specific for each joint are given in table 3.5. We can see no φ_{max} are required for the hip actuation, which can be understood if one looks at figure 3.2a: for any hip angle, the MTCs HAM, GLU and HF will always be orthogonal to the trunk bone. In other words, the torque $\tau(t)$ applied on the trunk as well as the change of length $\triangle l_{MTC}$ for these hip MTCs do not depend on the hip angle φ_{hip}. Equation 1.13 becomes: $r(\varphi) = r_0$.

	Ankle			Knee			Hip		
	SOL	TA	GAS	GAS	VAS	HAM	HAM	GLU	HF
$r_0\,[N]$	0.05	0.04	0.05	0.05	0.06	0.05	0.08	0.10	0.10
$\varphi_{max}\,[deg]$	110	80	110	140	165	180	-	-	-
$\varphi_{ref}\,[deg]$	80	110	80	165	125	180	155	150	180

Table 3.5: Parameters specific to the interaction between MTCs and joints

Soft limits of the joints

We already said we do not model ligaments (since ligaments have an exponential shaped spring behavior [18]), however we add soft limits to each joint:

$$\tau_{soft\,limit} = \begin{cases} K \cdot \triangle\varphi \cdot (1 + \triangle\dot{\varphi}) & \text{if } \triangle\varphi > 0 \text{ and if } \triangle\dot{\varphi} > 1\,[^{deg}/_{s}] \\ 0 & \text{else} \end{cases}$$

where $\tau_{soft\,limit}\,[Nm]$ is a torque applied on a joint with $K = 0.3\,[^{Nm}/_{deg}]$ and

	ankle	knee	hip
φ_{min}	70°	45°	20°
φ_{max}	130°	175°	230°

3.2 Experimental framework

In this section, we will briefly review the framework of the experiment.

Initialization of the walking gait The simulated robot lives in an infinite flat world.

Initially, the humanoid is hung in the air slightly above the ground. We force the angle of the trunk at $\theta_{trunk} = 15°$, the right hip at $\varphi_{hip}^{right} = 120°$ and the left hip at $\varphi_{hip}^{left} = 160°$ such that the right leg is before the left with an angle of attack which helps to start the walking gait. All the other joints are free to move according to the torque generated by MTCs.

From [31], we know the walking gait is stable if the humanoid possess the right balance between kinetic and potential energy. Therefore, starting with an initial speed of zero (with no energy) is not adapted to our walking reflexes (the robot would fall). To give an initial amount of energy to the humanoid, we give it a forward speed of $1.0\,[m/s]$. After a random time ranging from $4.0\,[s]$ to $5.0\,[s]$, we let the robot go. This random time window ensures the humanoid will always start the walking gait with a different initial condition since the knee and ankle joints are free to move. Therefore, each simulation is different from another.

For the sake of clarity, we will call this whole walking gait initialization the *launching phase*.

Since we are interested in forward walking gait, and not in balance problem, we will constrain the robot in the sagittal plan. Therefore, the robot can not move at all to the right or to the left.

If the speed of the robot reaches zero while walking, we consider it as a fall and any fall is followed by a reinitialisation of the walking gait (launching phase).

All constraints (hanging, initial forward velocity, sagittal plan) have been implemented in a physics plugin with the Open Dynamic Engine library used by Webots.

Chapter 4

Optimizations

4.1 Introduction

We need to find the 25 feedback parameters (shown in table 2.2) to allow the humanoid to walk properly. To this end, we could manually find such set, which means a lot of manual trials and errors. This approach, even if used with high success by H.Geyer in [31], can be long and exhausting.

In this project, we are interested in optimization. The nice side of optimization is the possibility to adapt the parameters to the uniqueness of a humanoid body and its environment. If we change the size, the mass or if we add some constraints to the humanoid or to the environment, we should be able to find in a systematic manner a good set of parameters.

In this chapter, we will first present how random generations of parameters allow the robot to walk without falling. In section 4.3, we will introduce a home made optimization algorithm aimed at optimizing the energy consumption of the walking gait. We will then show in section 4.4 how the well known Particle Swarm Optimization can efficiently do the same.

4.2 Random generation of parameter sets

4.2.1 Method

The laziest way to find a good initial set of parameters is to generate them randomly until we find a good one. "A good one" means the robot should be able to walk without falling for 3 times 100 steps, each time with different initial conditions due to the launching phase. We choose to repeat the experiment three times to be certain the robot is stable against different initial conditions.

4.2.2 Experiment 1: Results

Parameter sets

To have a statistical estimation of the chance to obtain such good initial set, we generate 1000 good solutions and record for each of them the rescaled values of the parameters as well as the number of random generation needed to find this good solution.

On figure 4.1, we see we need about 375 trials to obtain a good initial set with a probability of 95%. Such probability are highly dependent on how we choose the boundaties of the parameters: if the range $[min; max]$ of each parameters in table 2.2 are too large, the number of trials required to get a good set of parameters would increase.

Figure 4.2 shows an histogram for each parameter. We can note there is a huge variety of possible parameter configurations for all the muscles which still result in some stable gait. This is an indication of the suitability of the intrinsic properties of the muscle modeling (and reflexes) in human legs that is specifically "designed" for bipedal locomotion. With the help of this visual tool, we can see if the range we tune in table 2.2 are too wide or too narrow. We repeat two times this experiment (first result not shown here) in order to better range the parameters. The red dots on each subplot show the parameter used by H. Geyer in [31].

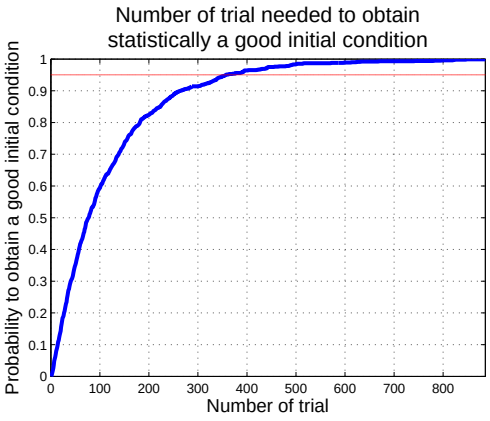


Figure 4.1: Experiment 1: Probability to find a good initial condition given a number of random generation of sets of parameters

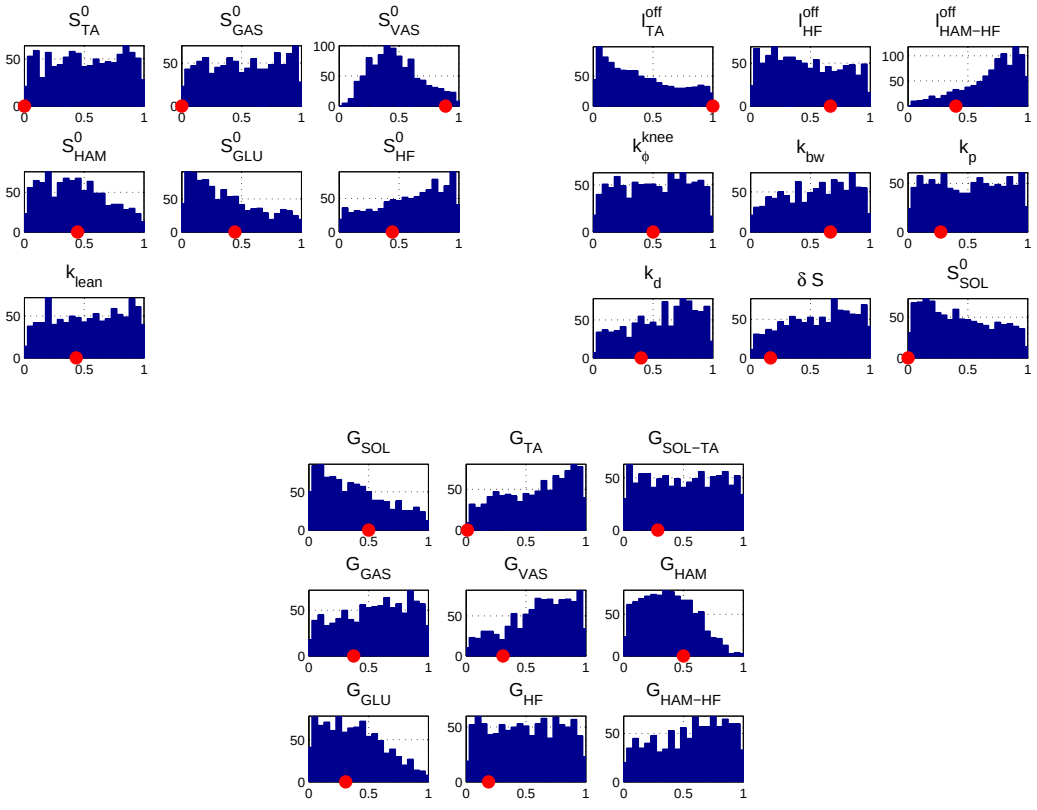


Figure 4.2: Experiment 1: Histogram for each parameters, showing the number of times we obtained a given value for a given parameter. The red dot shows the parameter of Hartmut Geyer in [31]

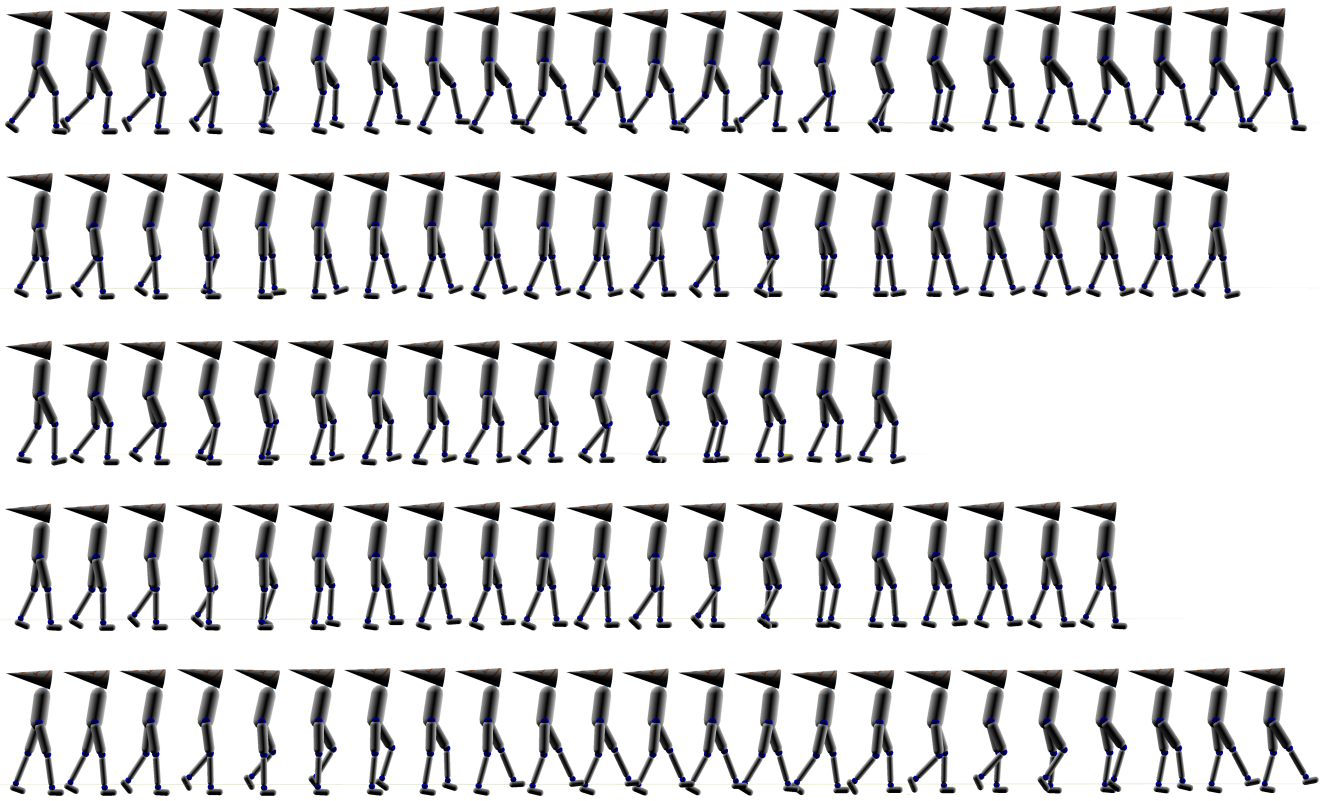


Figure 4.3: Experiment 1: Five examples of gait. From top to bottom: parameter sets 1, 2, 7, 8 and 12. We sample the pictures each 50[ms]. The robot walks mainly on the toes with parameter set 1. Parameter sets 2 and 7 have a small step length. Parameter set 12 has a kicking like gait. Finally, the parameter set 8 seems to be the more natural of all.

Walking gait analysis

We analyze the first twelve good set of parameters obtained (see table 7.1). We obtain very different kinds of gaits: some are slightly crouched, some have a small stride length, some walk mainly on the toes and some seem to kick in a ball at each step. None of them look really natural. Figure 4.3 shows five gait out of twelve (from top to bottom: parameter sets 1, 2, 7, 8 and 12). The robot walks mainly on the toes with parameter set 1. Parameter sets 2 and 7 have a small step length (0.55[m] against 0.7[m] in general). Parameter set 12 has a kicking like gait. Finally, the parameter set 8 seems to be the more natural of all. Movies are available on [56].

For each parameter set, we record the muscle activation of the right leg as well as its ground reaction force and the angles of the joints over one stride. Since we want to compare our results with biological data, we plot each graph against data taken from [3]. We use the muscle activation of the soleus (SOL), tibialis anterior (TA), gastrocnemius (GAS), vastus lateralis (VAS), semimembranosis (HAM), upper gluteus maximum (GLU) and adductor longus (HF). We use the maximum cross-correlation to quantify the shift Δ between human data curves and our simulation curves and the Pearson's correlation to determine the similarity R between the two curves.

The parameter set 7 gives the best match with biological data (shown on figure 4.4). However, parameter set 8 is visually more natural (as one can see on figure 4.3). Therefore, it is not possible the use the maximum cross-correlation and the Pearson's correlation to determine whether or not the gait look natural. To better quantify the natural visual aspect of a gait, we would need to take in account more information, like the angle of the trunk with respect to gravitation and the angular velocity of the joints.

From the feedback rules (table 2.1) we can directly guess the model can not predict accurately the late swing activation of the vastus muscle in biological data. In the model, this muscle can only be activated in stance. The same early activation is observed for the hamstring and gluteus muscle in end of swing (see human data on figure 4.4a). Activation of HAM and GLU muscles in late swing allow the walker to retract the swing leg (increasing therefore the stability of the gait [33]). Activation of VAS muscle increases the stiffness of the knee before heel strike. In the model, the retraction of the leg is achieved by the excitatory force feedback at the HAM and GLU

muscles and by the inhibitory stretch of HAM on the HF muscle. These feedback rules cannot predict accurately the activation of human data since they are active during the whole swing phase and not only in late swing. Therefore, comparing human data and simulations will never give a good match for these muscles. We can see however on figure 4.4a a nicer prediction can be made for SOL, TA and GAS muscles.

We measure the ground reaction force in the direction of the gravitation (fig. 4.4b). For each result, we first observe an early peak of GRF in the first 3 % of the stance phase followed by a drop in intensity. A second wider peak at 10 % of the stride correlates with human data but is always of higher intensity. We explain these results by the fact the VAS muscle is not activated in late swing in the model. Therefore, the leg lands with a knee lacking stiffness. The first peak of GRF is due to the heel strike on the ground. Since the knee is slack, the foot bounce on the floor which explain the drop of GRF intensity around 4% of the stride. The wider pike around 10% of the stride correlate with the activation of the VAS muscle. Since the stance is initiate with a slack knee, the force feedback of the VAS muscle should be high enough to avoid the walker to collapse, leading to a high force production which explain the exaggerated second GRF pike. Starting the stance with such GRF pattern will corrupt the rest of the stance phase. Therefore, even if we can see some similarity between human GRF and simulation, we will not observe the nice shock absorption of human data which produce a GRF starting to zero, increasing smoothly with two nice peak at early and late stance.

Finally, figure 4.4c shows the joint angle for one stride. The solution 7 walk with the knee slightly flexed which explain why simulation have lower knee angles, ankle angles and hip angles.

Robustness

We want to quantify the stability against the initial condition, the robustness (stability against noise) and to estimate the width of the basin of attraction of the limit cycle for a given parameter set.

For each parameter set, we launch the robot 100 times (therefore the initial condition of the walking gait is different at each trial, see section 3.2) and let it walk for 100 steps. We compute the probability to fall which is shown on the upper graph in figure 4.5a. We can see all solutions are quite stable, except solutions 5, 8 and 9. These results are coherent with the parameter sets selection criterion which validate a good set only if the robot is able to walk 3 times 100 steps.

We then want to measure the robustness against noise in the muscle. To this end, we add a signal dependent white noise ε in the activation of muscles:

$$\varepsilon = \frac{r - \mu}{\sqrt{\sigma}} \cdot A^2 \cdot \eta \tag{4.1}$$

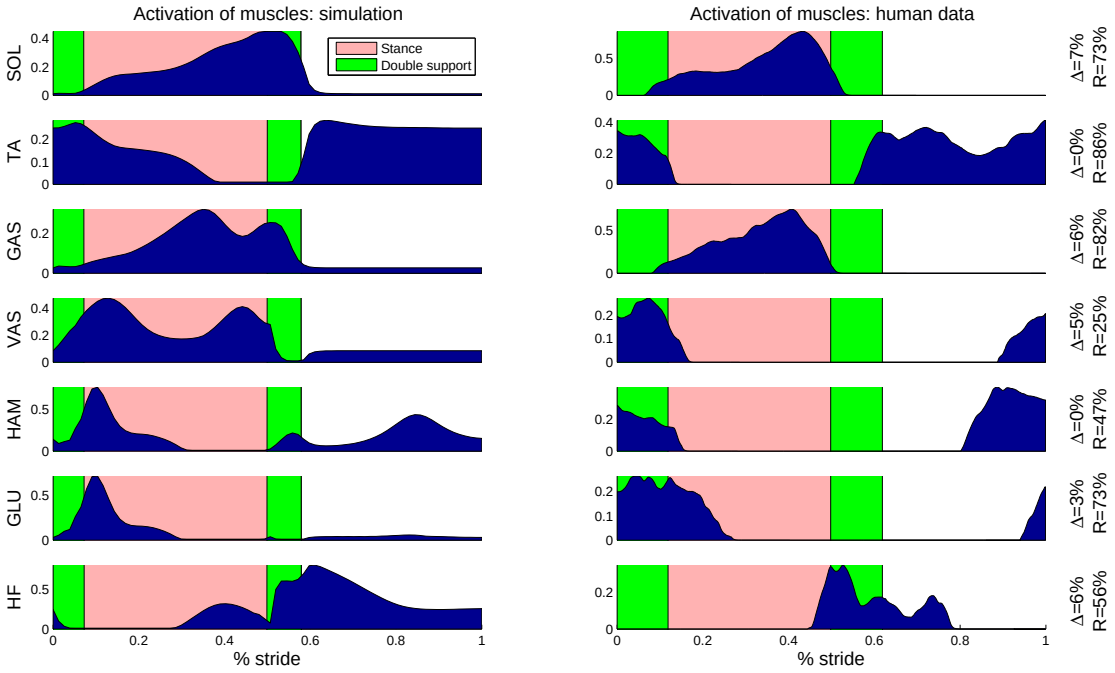
where r is a uniform random number between 0 and 1 which has $\mu = 0.5$ mean and $\sigma = 0.0833$ variance, A is the activation of muscles and η is a constant. $\eta = 0$ when the robot is launched, and increase of 0.01 at each 10 steps. We record the final value of η when the robot falls and repeat the experiment 100 times. The lower graph on figure 4.5a shows for each parameter set the maximum average value of η the walker can stand without falling.

We see the solutions which are the most stable against different initial condition (set 1 for example) are not necessarily the ones which can stand a high amount of noise. On the contrary, parameter set 5 can support more noise than parameter set 1.

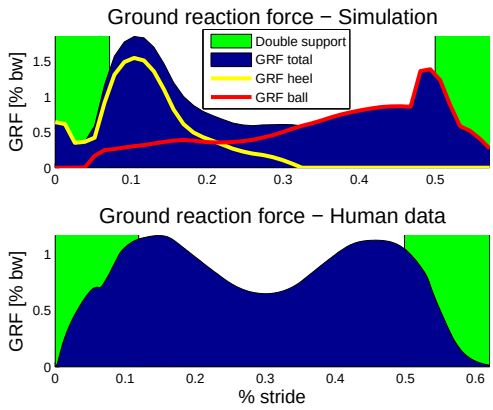
Finally, we are interested to see how fast the walker converges to a limit cycle during the first steps and how close to this limit cycle the walker will stay while walking. To this end, we record at each steps the state of the robot: linear velocity, angle and angular velocity of each body as well as activation, contractile element length, contractile element velocity and tendon length of each muscle. We let the robot walk 50 steps and compute the difference between two steps by summing the individual difference of each state indicators. We repeat the experiment 20 times and average the difference for each step. Figure 4.5b shows the results. We can see parameter set 3 converges within 2 steps to its limit cycle and tends to be very regular from one step to the next. On contrary, parameter set 9 need 7 steps to reach its limit cycle and will be less regular from one step to the next. In general, a parameter set which is not stable will need more steps to converge to its limit cycle. However, the regularity of the gait from one step to the next does not impact the stability of the walker nor the robustness of the gait.

Discussion

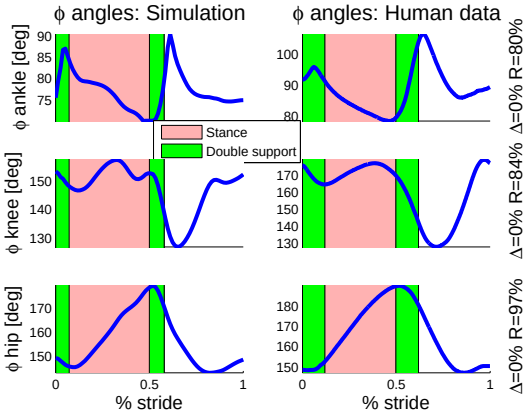
In this first experiment, we generate random parameter sets and observed we need about 375 trials to obtain a good set with probability of 95%. Each solution gives a different gait, but none of them look very natural and smooth.



(a) Activation of muscles over one stride. The left plots shows our results and the right plots shows human data taken from [3].

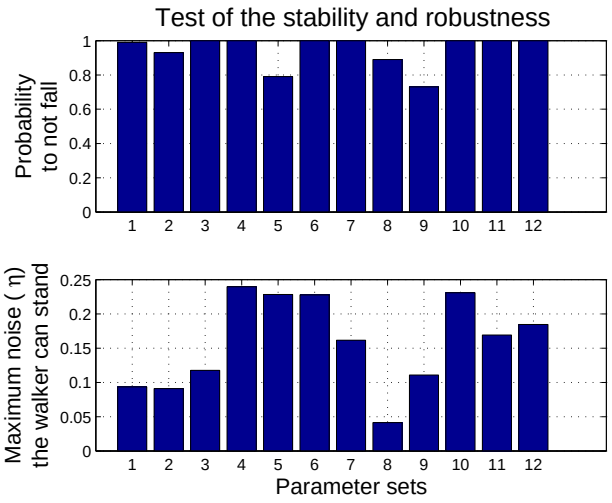


(b) The ground reaction force in the direction of the gravitation, measured from the foot sensors.

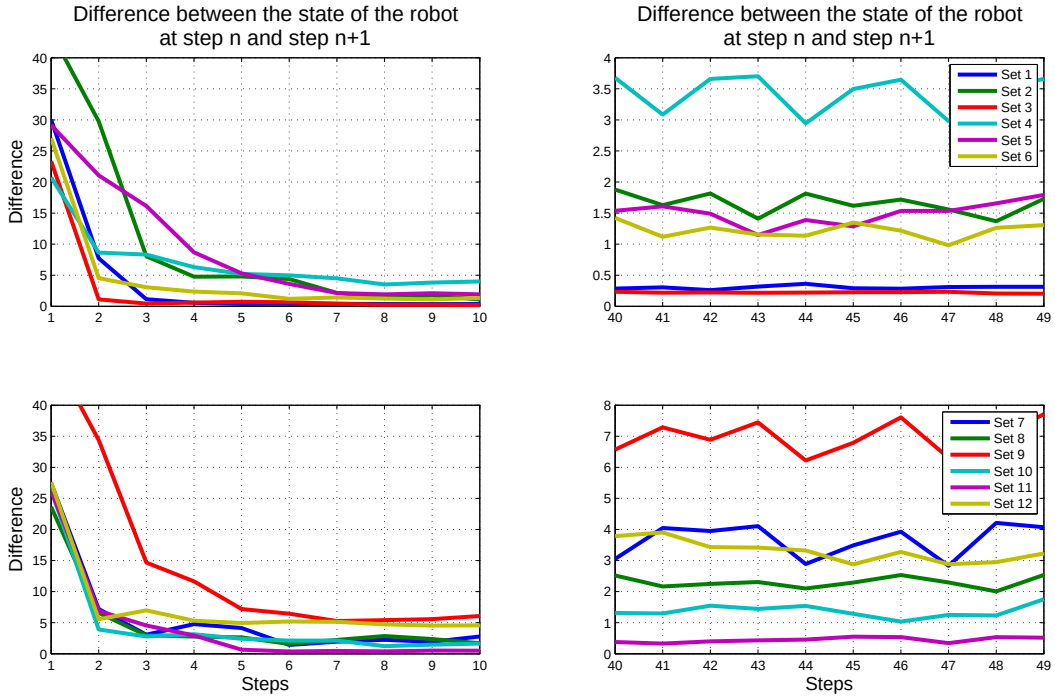


(c) Angle of the right hip, knee and ankle versus human data.

Figure 4.4: Experiment 1, Solution 7: Gait analysis over one stride for the right leg. The humanoid is walking with a forward speed of 1.45$[m/s]$.



(a) Up: Stability against the initial condition Down: Robustness against noise in muscles



(b) Left: Convergence to the limit cycle when the robot is launched Right: regularity of the steps.

Figure 4.5: Experiment 1: Stability and robustness of 12 tested parameter sets.

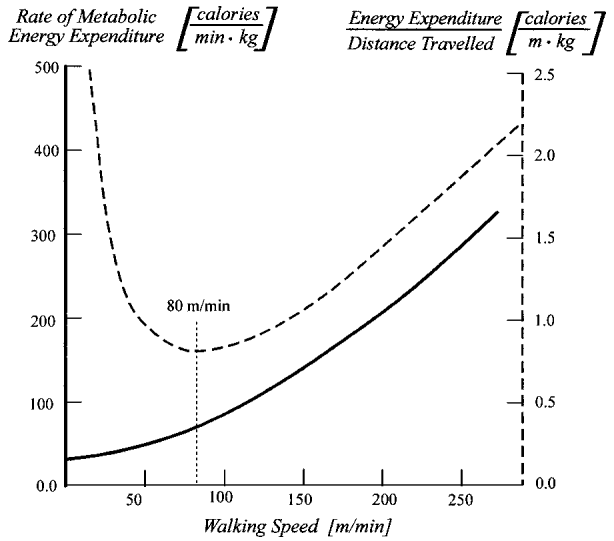


Figure 4.6: Metabolic energy expenditure plotted as a function of walking speed. The rate of metabolic energy expenditure increases parabolically as walking speed increases (solid line). When the rate of metabolic energy consumption is normalized by the distance traveled, an optimal walking speed is predicted at 80 [m/min] (dashed line). Taken from [19]

The comparison with human data can be made, but we have to keep in mind the underlying feedback rules do not reflect the exact biological muscle activations. Furthermore, we did not take into account the joint angular velocity and trunk angle which make the quantification of the difference between human and simulated data difficult.

Therefore, the solutions which give the highest match with biological data are not the ones which look the most natural.

We observe that some solutions are more stable than others against different initial condition. However, these stable solutions are not necessarily robust against noise in muscles. Therefore, some solutions are well adapted to the way we launch the robot from the air but are not robust against noise. On contrary, some solutions are robust against noise in muscles, but are not so well adapted to the launching phase.

Finally, a walker which is less stable needs generally more steps to converge to its limit cycle. But neither the stability nor the robustness against noise in muscles seems to impact the regularity of the steps while walking.

4.3 Optimization

4.3.1 Introduction

In the previous section, we show it is possible to generate stable and quite robust walking gait with random generation of parameter sets. However, looking at the gait teach us they lack smoothness and they do not seem natural. How can we further tune the parameter sets such as to obtain a smoother gait?

Minimum energy consumption as a fitness function

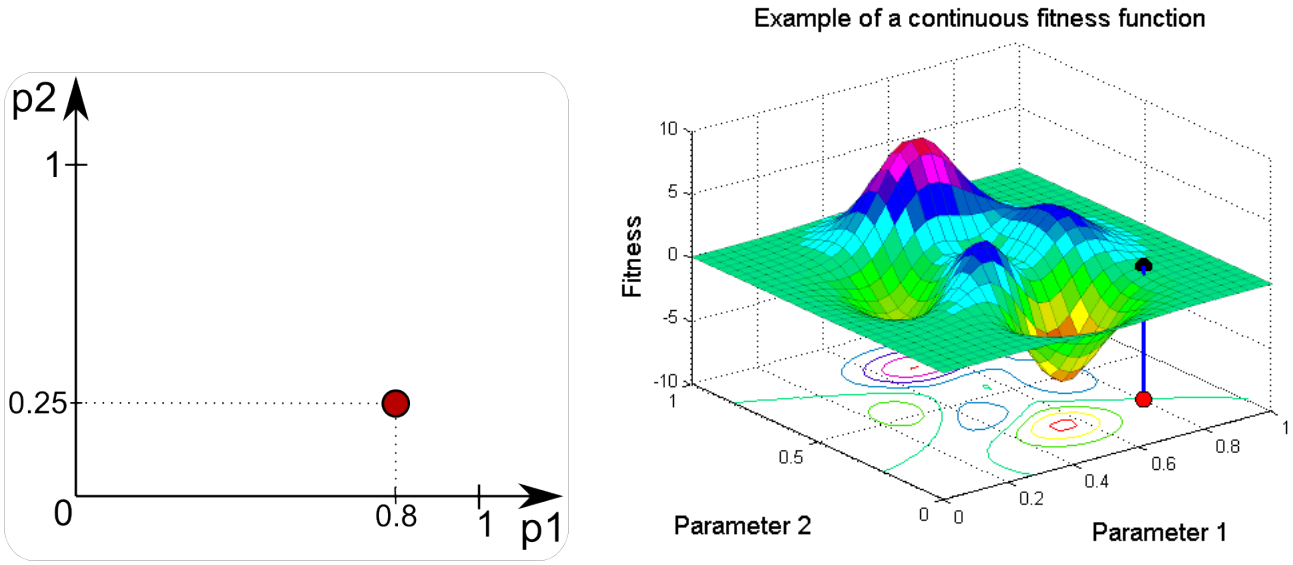
If one asks an adult to walk at an undefined speed, the walker will spontaneously walk at a speed which minimize its energy consumption [53, 19]. By minimizing the energy consumption, we mean minimizing the energy used per unit distance traveled. Figure 4.6 shows such energy expenditure as a function of the distance traveled with a minimum energy per unit distance achieved at a speed of 80[m/min]=1.33[m/s].

Furthermore, animals generally tend to use the minimum energy to reach a given goal. Measuring the energy per distance traveled seems to be a good way to measure how well the robot is walking. Therefore, we believe we can improve the gait of the humanoid by optimizing its energy consumption.

To this end, we will use two optimization algorithms: a home made optimization algorithm (see below) and a PSO algorithm (see section 4.4) and we will compare their performances.

The general method of adaptation

Every optimization algorithm uses the following principles:



(a) An example of a particle in a parameter space

(b) An example of a continuous fitness function

Figure 4.7

- A given set of parameter values is evaluated, which means we measure the performance of the system given these parameter values. This measure is generally called the *fitness* function.
- After we evaluate several sets of parameter values, we *select* the best ones.
- Finally, we add little *variation* to the selected sets.

Parameter space

Imagine a simplified system which is define by only two parameters p_1 and p_2. A set of parameters (for example $p_1 = 0.8$ and $p_2 = 0.25$, as illustrated on figure 4.7a) is called a *particle* in the parameter space. The parameter space is all the possible sets of parameters that could exist, i.e. the area where the red particle can be. Now we can evaluate the system for this red particle with a fitness function. Evaluating systematically every point of the parameter space leads to a fitness surface, as illustrated on figure 4.7b.

The dimension N of the parameter space is the number of parameters ($N = 25$ in our walking problem) and the fitness surface extend in the $(N+1)^{th}$ dimension. As one can see on figure 4.7b, the fitness function can be continuous and smooth, but in some other cases the fitness function can be discontinuous.

If the fitness function is discontinuous, we would prefer to use a random search of parameters (GA for example). On contrary, if the fitness function is continuous, we can take advantage of the gradient of the fitness surface to know how to change the parameters. PSO[1] [48, 49], SPSA [52], Nelder-Mead [45] or any Gradient Descent optimization algorithms are example of such smooth fitness surface exploitation.

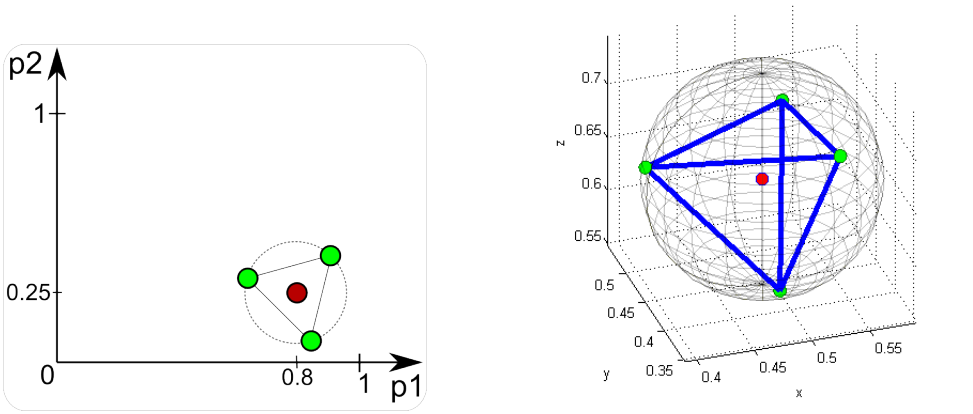
4.3.2 Method

At the beginning of each trial we provide the humanoid with a credit of energy (5000[J]). While walking, the muscles of the robot will consume energy (see section 1.5) and the trial stops when the credit is exhausted. We then measure the distance traveled. Since minimizing the energy consumption per unit distance traveled is the same as walking as far as possible with an initial credit of energy, the measured distance can be used as a fitness function.

Optimization process

Now we have a way to measure how good is the walking gait of the robot, we are able to discriminate one set of parameters from another.

[1] Note that PSO does not use nor does it estimate the gradient.



(a) A simplex around the CoM

(b) The result of a simplex formation in 3D

Figure 4.8: Simplex formation around a CoM

Our strategy will be to generate several solutions around a reference one, to evaluate them all, to select the best and to move a bit our actual set of parameters toward the selected one. We found here the general idea of adaptation processes: variation and selection.

We will start our optimization from the initial parameter sets obtained in section 4.2. Let call this initial particle the *center of mass* (CoM).

To be able to know where to move our CoM in the parameter space, we need to generate and evaluate several particles around the CoM. If these particles are too close from the CoM, their fitness could be similar to the fitness of the CoM and we would not be able to discriminate a better solution. On the contrary, if these particles are too far away from the CoM we risk to generate solutions which do not work (not able to walk). Therefore, we choose to generate particles randomly on a hypersphere radius R with the CoM at the center.

Now, if two particles are too close to each other, we risk to evaluate twice the same solution. On the contrary, if they are too far apart, we risk to miss a good solution in between. Therefore, we need to place optimally the particle on the cycle with our simplex formation algorithm (see below).

Finally, if we generate too many particles, the simulation would take too much. On the contrary, with only a few particles, we could loose some good solutions. Therefore, we choose to generate $N+1$ particles, where N is the dimension of the parameter space.

Figure 4.8a illustrates a CoM in red, and three particles in green in a 2 dimensional parameter space. We can see how the green particles are placed on a cycle with an equal distance from each other, forming therefore an equilateral triangle. In higher dimensional space, an equilateral triangle is called a *simplex*.

The simplex formation algorithm In order to position the particles optimally on the surface of an hypersphere, we use the following algorithm:

1. Initialize randomly $N+1$ particles on an hypersphere, where N is the dimension of the parameter space.
2. For each of them, move them a bit away from every other particle until the algorithm converges.

Figure 4.8b illustrates such simplex formation in a 3D parameter space.

The optimization algorithm This home made optimization algorithm is as follow:

1. Generate a random initial solution (see section 4.2) and call it the CoM.
2. Generate $N+1$ particles at a distance R from the CoM and apply the simplex formation algorithm.
3. Evaluate the CoM and each particles (with the minimum energy consumption as the fitness function).
4. Move the CoM in the direction of the best particle through a distance $\triangle d$.
5. Go to (2) until the limit number of epoch t_{max}.

N	R	$\triangle d$	credit of energy	t_{max}
25	0.1	0.03	5000 [J]	1000

Table 4.1: Parameters used in the home made optimization algorithm

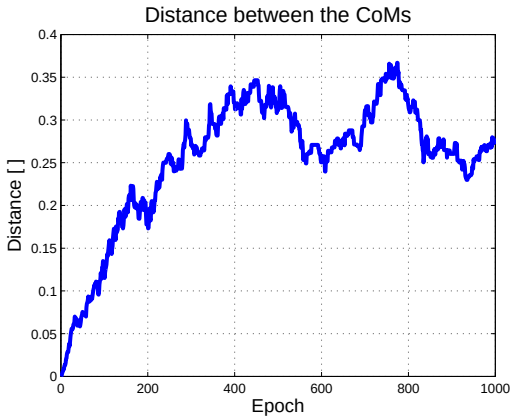


Figure 4.9: Experiment 2: Distance between the 10 CoMs in the parameter space. We can see the distance increase through optimization, therefore each optimization leads a different final result.

Table 4.1 summarizes the parameters used in our optimization.

We call a *trial* the evaluation of one particle (which is launching the humanoid with 5000[J] and let it walk until depletion of energy). We call an *epoch* the evaluation of every particle. We call a *run* the whole optimization.

Since we use 26 particles and we evaluate each of them over 1000 epoch, we have a total of 26'000 trials per run.

4.3.3 Results

4.3.3.1 Experiment 2: running the optimization from a single initial parameter set

In this experiment, we run 10 times the optimization algorithm with the same initial set of parameters. The main goal of this experiment is to see if this algorithm always converge to the same result at each run. We choose the set number 1 in table 7.1 without any criterion of selection.

Results of the optimization

Figure 4.9 shows the sum of the distance between each pair of CoM in the parameter space, which is growing through the epochs. Therefore, even if we start from the same initial condition, each optimization gives a different final result, which indicates that there exist several local maxima on the fitness function. Table 7.2 summarizes the best fitness and the best parameters for each of the 10 trials.

We are now interested to visualize the differences between the H.Geyer's parameters, our initial parameters and our final parameters. Figure 4.11 shows the single H.Geyer's parameter set with black dots, our single initial parameter set with blue dots and our 10 final parameter sets with boxplot.

A boxplot displays one red line at the median of the data (in our case the value of a parameter), one blue line at the lower and upper quartile and notches indicate where the 95% of the data are located (see fig. 4.10). We call the inter quartile distance (IQD) the distance between upper and lower quartile. Any data which is 1.5 times the IQD above the upper quartile or which is 1.5 times the IQD below the lower quartile are called outliers and is marked with a red cross. Black whiskers shows the maximum data (which are not outliers) above the upper quartile and below the lower quartile.

On figure 4.11, we see our final parameters does not converge to the H.Geyer's parameter set. It seems we obtain a different pattern of parameters. We explain this difference by the fact we add a foot of 10 [cm] height (distance from the foot plant to the ankle joint, see fig. 3.1), which is not present in [31]. However, it is difficult to evaluate how this change in the humanoid model impacts the parameters of the walking gait and we simply observe the difference.

Furthermore, some parameters change a lot during optimizations (for example parameters 9, 21, 24, etc...) while others stay close to their initial value (1, 3, ...). Some final parameters can be spread and others stay confined within

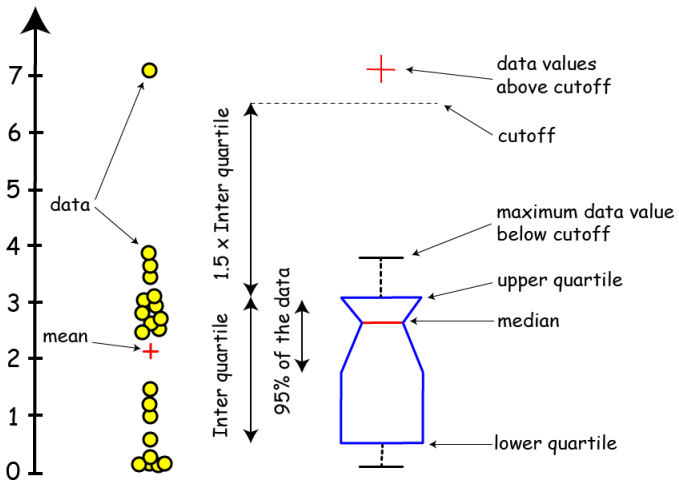


Figure 4.10: Boxplot explanation.

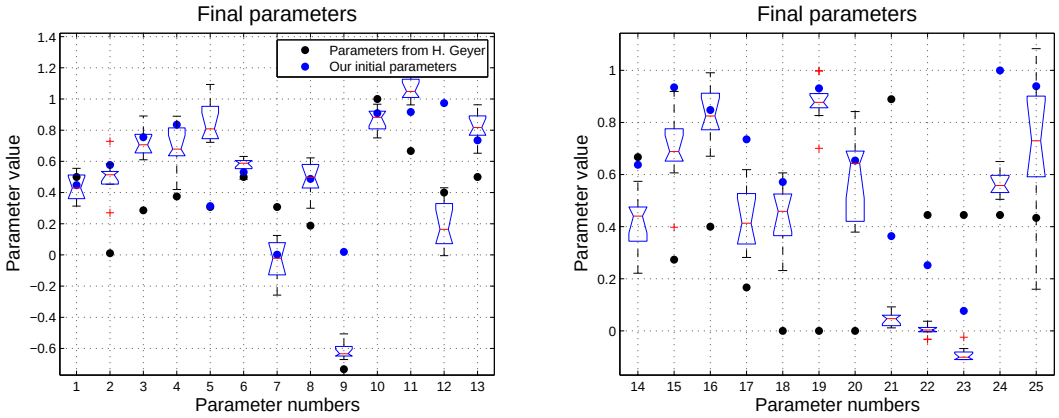


Figure 4.11: Experiment 2: Boxplot showing the spread of our final parameters.

a small value range (parameter 9 for example). Therefore, it seems very likely several local maximum exist.

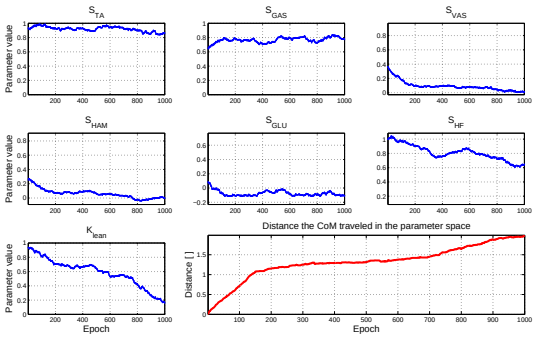
Figure 4.12 shows the optimization for the trial number 8, which gives the best optimization of energy consumption. Figure 4.12a,4.12b and 4.12c show the optimization of each parameter (blue curve). Note the rescaled parameters are not limited in the range $[0; 1]$, they can become bigger or smaller through optimization. However, the parameters (not rescaled) can never be smaller than 0. We can see some parameters almost do not change while others change a lot during the first 200 epochs (G_{HAM-HF} for example, which is parameter 9). The red curve on figure 4.12a shows the distance traveled by the CoM in the parameter space which is computed as the Euclidean distance from its initial position to its position at epoch n. The distance walked by the humanoid as well as its energy consumption per unit of distance traveled for each epoch are shown on figure 4.12d. We can see a constant optimization of energy consumption, and therefore of distance walked, is realized. Initially, the humanoid is barely able to walk 20 meters with 5000 joules. After optimization, the robot can walk 30 meters with the same amount of energy.

Every run shows a rapid optimization within the first 200 epochs followed by a slow optimization rate. We stop every run after 1000 epochs, however they all seems to be further optimizable.

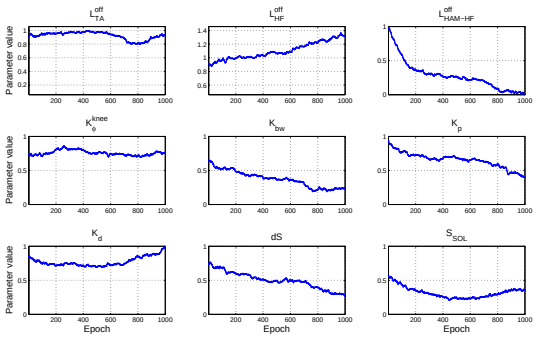
Walking gait analysis

The final walking gait of optimization run number 8 is shown in figure 4.13. The optimization leads to a smoother gait and an increase in step length.

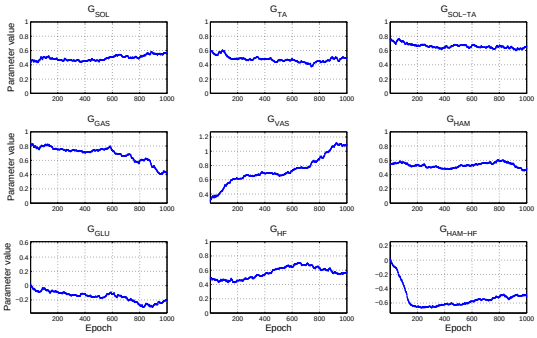
The final walking gait has been analyzed over one stride (fig. 4.14). Figure 4.14a shows the activation of each muscle. We can see a double bump in the SOL, GAS and VAS muscle which can be explained by the spring-like behavior of the leg. In humans, the vastus muscle is responsible of the first bump which occurs during the initial



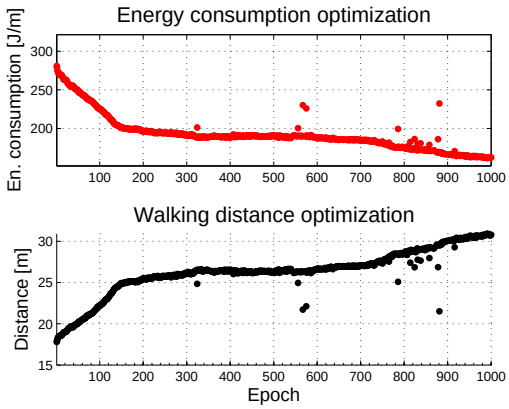
(a) Optimization of parameters and distance traveled by the CoM in the parameter space.



(b) Optimization of parameters



(c) Optimization of parameters



(d) Optimization of energy consumption and the distance walked with 5000[J].

Figure 4.12: Experiment 2: Optimization run number 8.

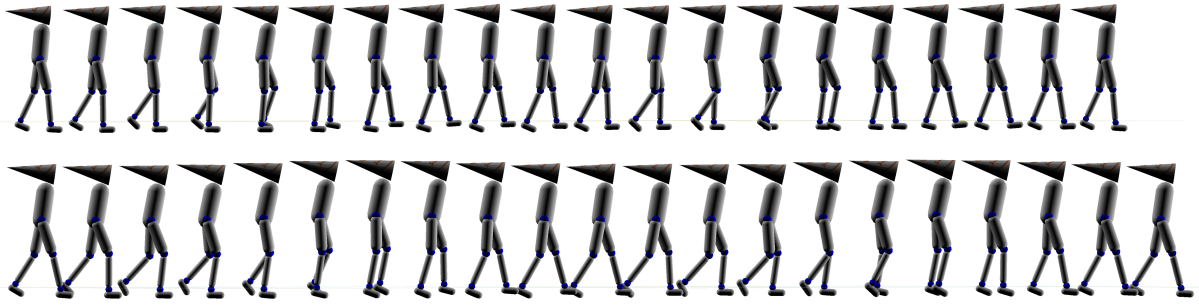


Figure 4.13: Experiment 2: Walking gait obtained with the final result of optimization run number 8. We sample the pictures each 50[ms]. Up: before optimization (taken from experiment 1) Down: after optimization. Movies are available on [56].

swing phase. Soleus and gastrocnemius muscles are responsible of the second bump in late swing [46]. Here we do not observe such separation of the task by different muscle groups. In our results, SOL, GAS and VAS are all responsible for the two bumps. The little bump of TA activation in stance phase is a consequence of the low activation of the SOL (see feedback rules in table 2.1).

Figure 4.14b shows the ground reaction force. We see a nicer double wave than in experiment 1 (fig. 4.4b), however the heel strikes the ground twice: the first time at heel strike and a second time at 10% of the stride. Again, we explain this behavior by the absence of shock absorption at heel strike.

Figure 4.14c shows the angle of each joint. The rebound of the leg is clearly visible at the ankle, which is correlated with the undesired second heel strike.

Robustness analysis

Figure 4.15a show the stability against initial conditions. Before optimization, we had a stability of 100% against different initial condition. After optimization, the stability is lower, ranging from 10% to 70%. We explain this result by the fact we did not add pressure of selection against the initial condition. In other words, we did not test several times each solutions, as we did in experiment 1 (each solution was tested 3 times before validation).

Robustness against noise in muscles are lower after optimization (from $\eta = 0.03$ to $\eta = 0.07$) than before ($\eta = 0.09$), except for run number 9 which show a higher robustness against noise ($\eta = 1.12$). Since we did not include noise in the optimization, it sounds logical this aspect has not been improved.

Figure 4.15b shows the number of steps required to converge from the initial condition to the limit cycle as well as the regularity of the gait from one step to the next. This time, there is a clear evidence that a stable gait will converge faster to the limit cycle and will be more regular, as shown by parameter sets 2,3,8,9 and 10.

Discussion

This home made optimization algorithm effectively optimize the energy consumption of the robot. Running 10 times the optimization algorithm from the same initial condition leads to 10 different final solutions. We think there should exist several local maximum in the fitness function. However every optimization shows a constant but slow optimization rate at the end of each run. Therefore we can not say each optimization would never converge to the same final solution, since we never try to reach it, but it seems very unlikely (and very computationally expensive).

The final parameters do not match with the parameter set of H.Geyer. Since our humanoid model has a 10 centimeter ankle which is not present in [31], it could explain these differences.

The final walking gait is smoother, energetically more efficient and looks more natural. However, the leg does not absob the shock at heel strikes and this can be seen on the ground reaction force plot. Furthermore, we do not observe the single vastus activation at early swing and common soleus-gastrocnemius activation in late swing. These three muscle groups rather work together to create the spring-like property of the legs.

The stability against initial condition and muscle noise robustness are poorer than before optimization because we did not take them in account while optimizing. Finally, we observe that a a stable gait will converge faster to the limit cycle and will be more regular.

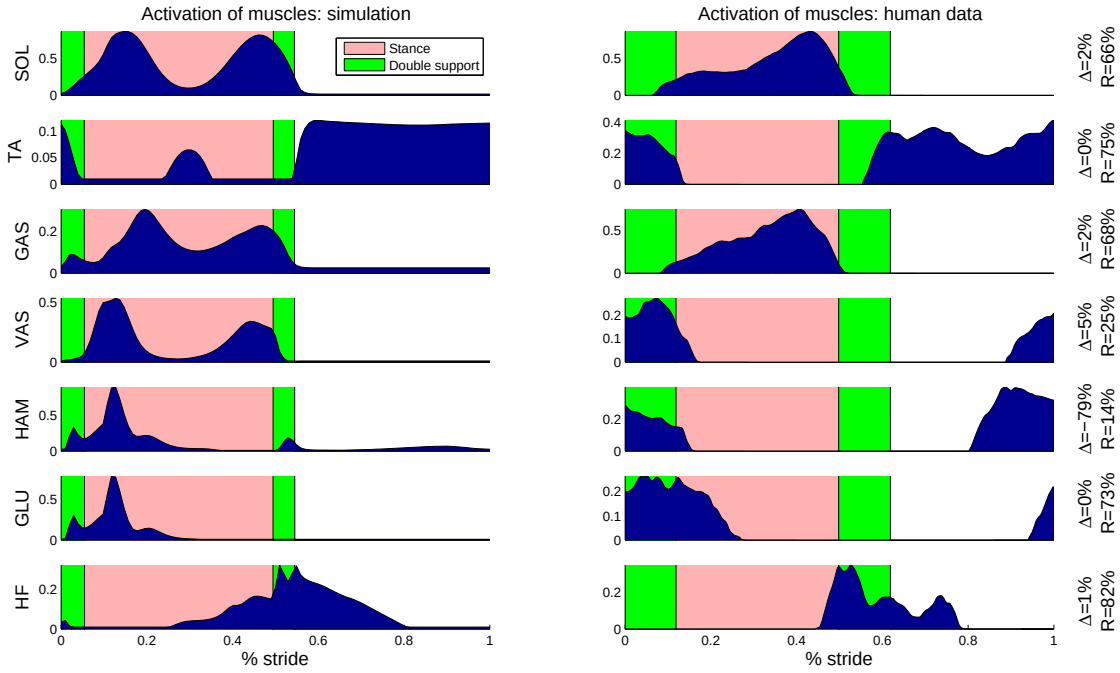
4.3.3.2 Experiment 3: running the optimization from 12 different initial parameter sets

In this experiment, we run the optimization on the 12 sets presented on table 7.1. The best fitness and the best parameters for each of the 12 runs are shown on table 7.3.

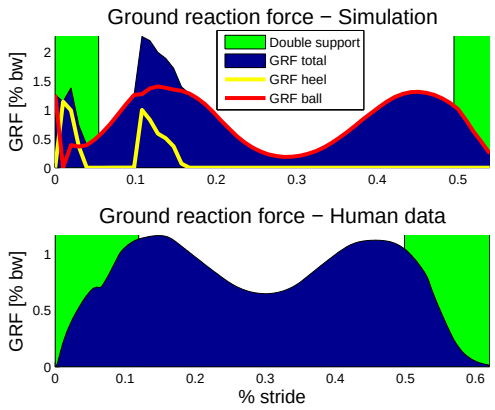
Results of the optimization

The boxplot on figure 4.16 shows the distribution of our final parameters. Again, we did not converge to the same parameter set as H.Geyer but we are statistically closer in this experiment than in the previous one. However, we can not draw conclusions on this because we run only 12 optimizations and we would need much more to have a statistical solid fact.

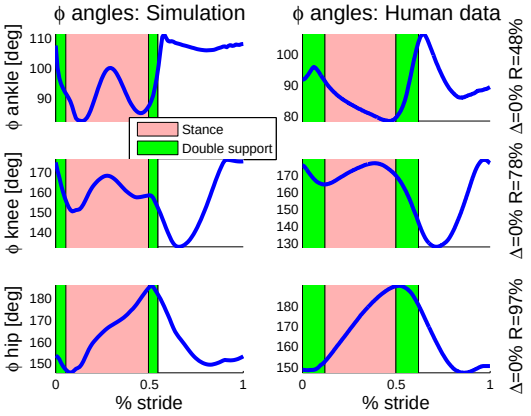
The best optimization has been realized with initial parameter set 6. On figure 4.17a, 4.17b and 4.17c we can see the optimization of each parameter individually. Figure 4.17d shows the energy (and distance walked) optimization. With 5000[J] the robot was able to walk 20 meters before optimization and 36 meters after.



(a) Activation of muscles over one step.

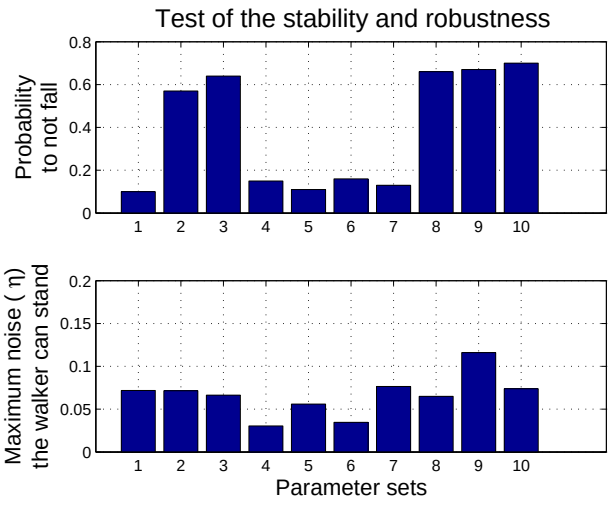


(b) The ground reaction force in the direction of the gravitation, measured from the foot sensors.

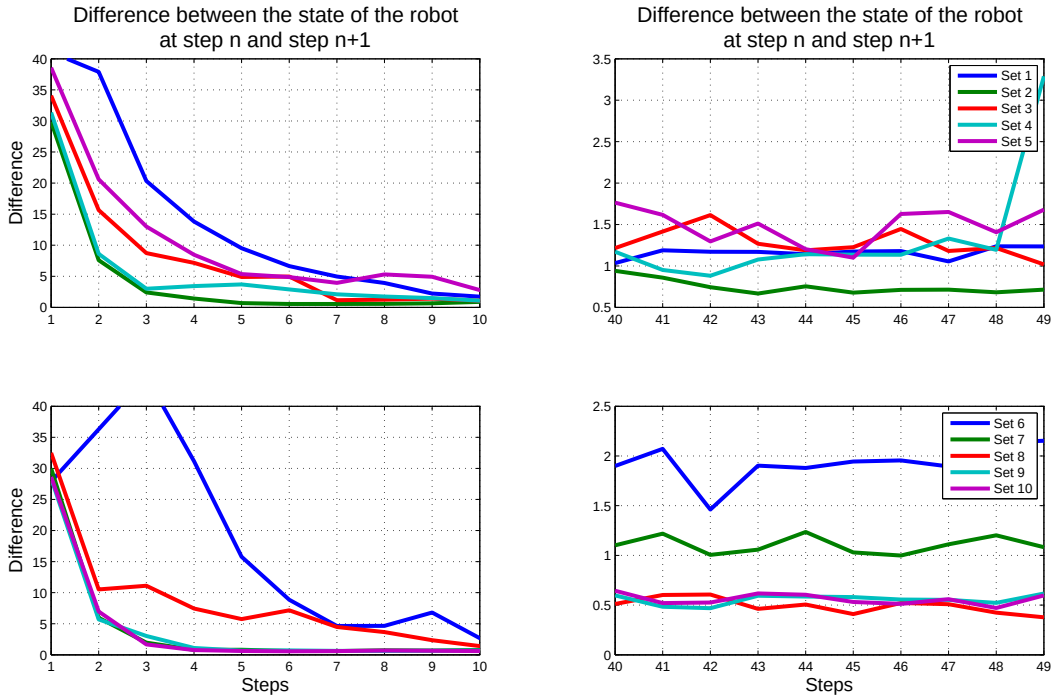


(c) Angle in degree of the right hip, knee and ankle.

Figure 4.14: Experiment 2, Solution 8: Gait analysis over one step for the right leg. The humanoid is walking with a forward speed of $1.5[m/s]$.



(a) UP: Stability against the initial condition DOWN: Robustness against noise in muscles



(b) LEFT: Convergence to the limit cycle when the robot is launched RIGHT: regularity of the steps.

Figure 4.15: Experiment 2: Stability and robustness of 10 optimizations run all from the same initial parameter set.

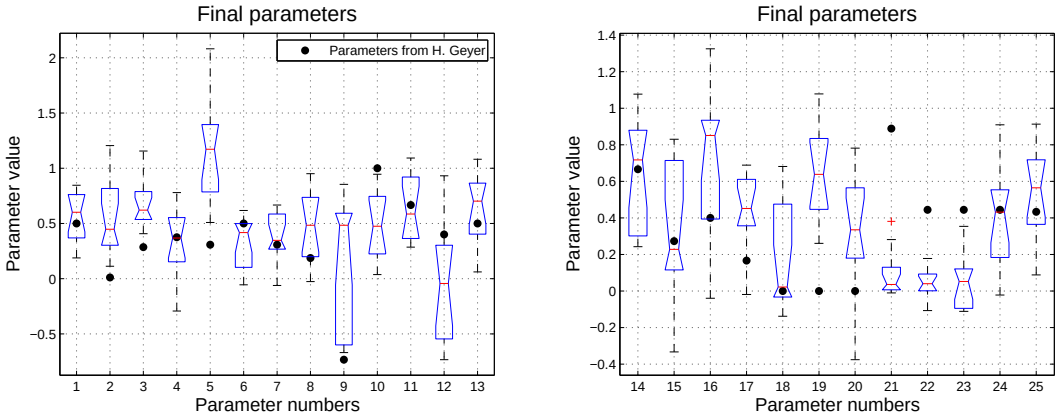
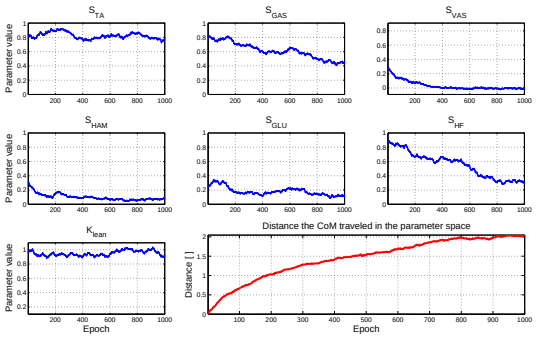
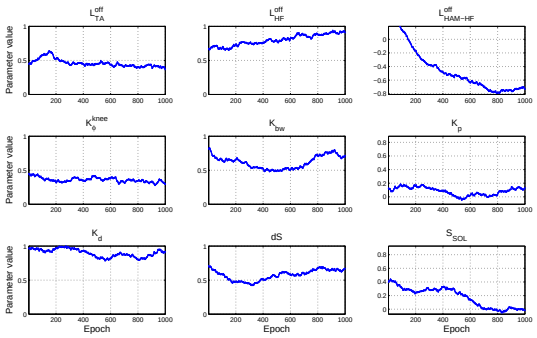


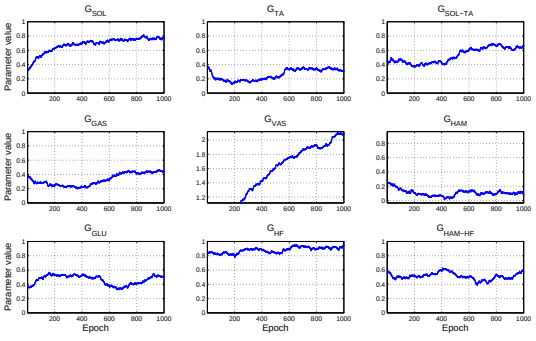
Figure 4.16: Experiment3: Boxplot showing the distribution of 12 final parameter sets after optimization from 12 different initial parameter sets.



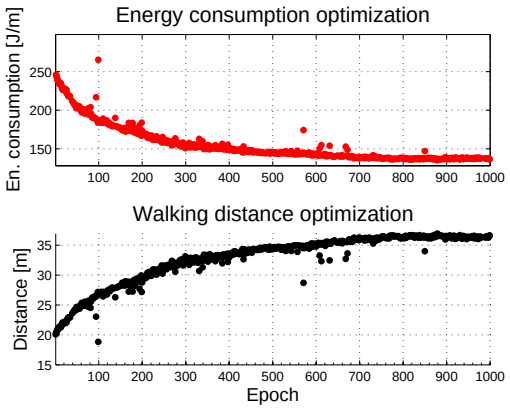
(a) Optimization of parameters and the distance traveled by the CoM in the parameter space.



(b) Optimization of parameters.



(c) Optimization of parameters.



(d) Energy consumption optimization and the distance walked with 5000[J].

Figure 4.17: Experiment 3: Optimization of the parameter set number 6.

Figure 4.18: Experiment 3: Parameter set 6. We sample the pictures every 50[ms]. UP: Before optimization DOWN: after optimization

Walking gait analysis

Figure 4.18 shows the walking gait before and after optimization, each picture have been sampled every 50[ms]. After optimization, we clearly see the stance leg is straighter, the whole body is positioned in the direction of the walk and the swing leg is not lifted as much as before optimization.

Figure 4.19a shows the activation of muscles. This time, we observe the vastus muscle is only active in the early swing. Gastrocnemius is mainly active in mid and late swing, contributing to the push off. Soleus muscle is both active in early swing and late swing. Again, these activations define the ground reaction force feature.

On figure 4.19b we clearly see the absence of heel strike absorption in the first 5% of the stride, the influence of the vastus activation between 10% and 15%, the influence of the soleus activation between 15% to 25% and the action of both soleus and gastrocnemius after 25%.

Figure 4.19c shows the joints angles. We see how the knee reaches its limit of extension and how it stays as straight as possible. This gait feature can be understood by the fact a straight leg in stance needs a smaller muscle activation to bear the body weight and therefore uses less energy. Having a straight leg in stance is energetically efficient. Furthermore, no muscle activation (and no energy consumption) is required to stop the hyper extension of the knee in swing because the knee just reaches its soft limit.

Humans do not walk like this solution because it would be too painful to stop the extension of knee in swing phase with the simple mechanical limit of bones and ligaments. We can see here it is not sufficient to optimize energy consumption to have a human like gait, we need to simulate the "pain" by penalizing any joint angle which exceed some threshold. Nothing like this has been done in this project.

Robustness analysis

Here again, nothing has been done to optimize the gaits against noise in muscles and against different initial condition. Therefore, the stability and the robustness after optimization (fig. 4.20) is lower than before optimization (fig. 4.5).

We do not observe a correlation between stability and/or robustness and convergence to the limit cycle and/or regularity.

Discussion

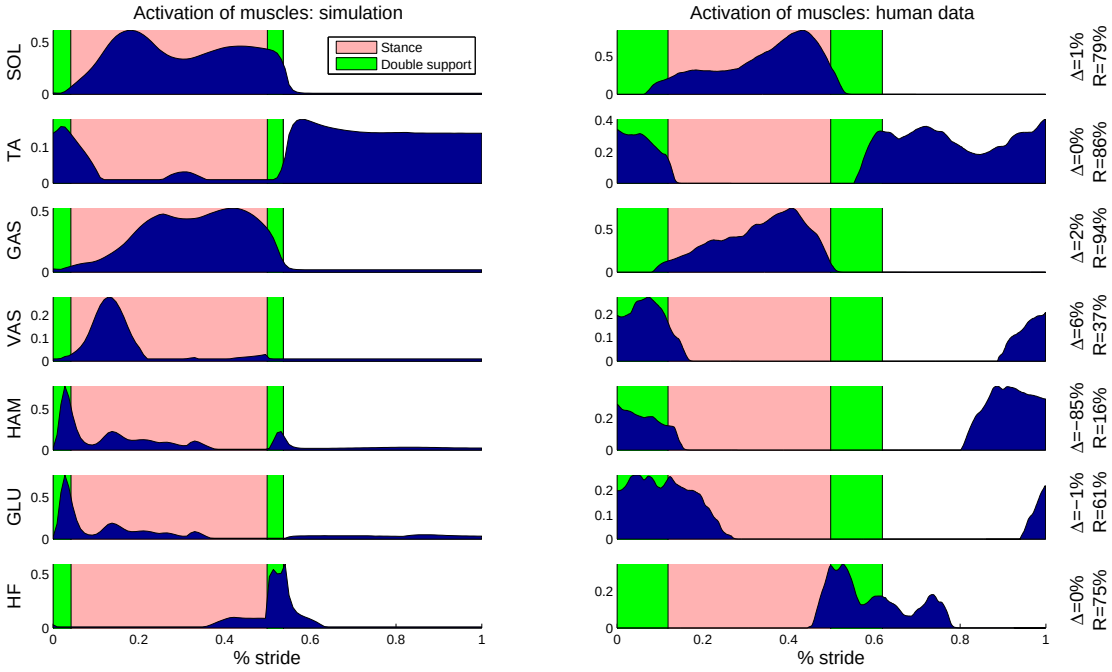
The final parameters do not closely match with the parameter set of H.Geyer. However, we obtained different parameter sets which allow the humanoid to minimize its energy consumption while walking.

We observed how the optimization takes advantage of the soft limit of the knee joint to achieved the walking gait while minimizing energy consumption. Such behavior is not realistic because it would be too painful for our knee to walk like this.

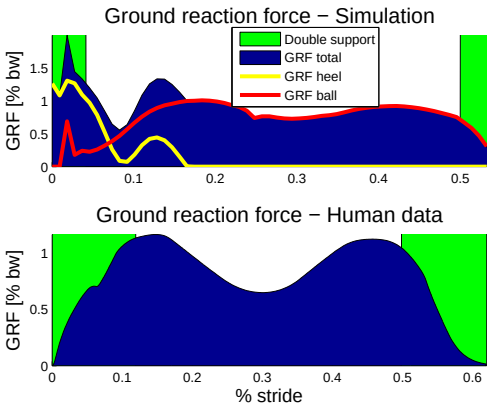
Finally, since we did not include muscle noise and stability against initial condition during the optimization, our final parameters are less stable and less robust than the initial ones.

4.4 Particle Swarm Optimization

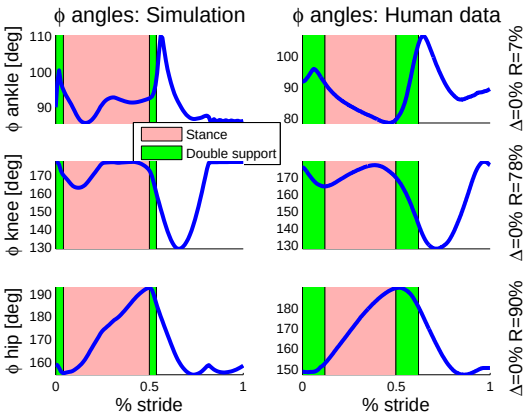
In this section, we will use the well known particle swarm optimization algorithm.



(a) Activation of muscles over one step.

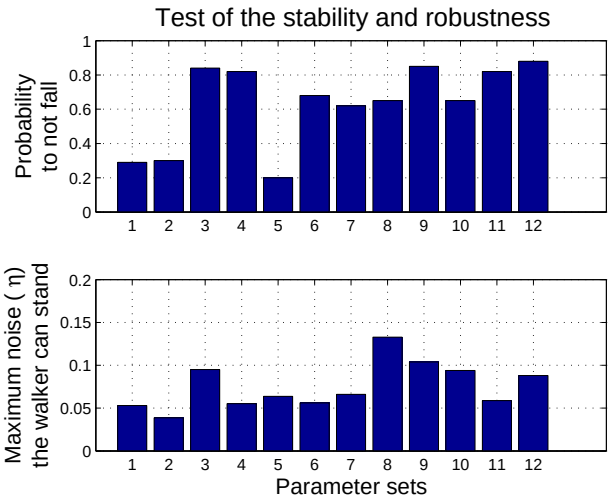


(b) The ground reaction force in the direction of the gravitation, measured from the foot sensors.

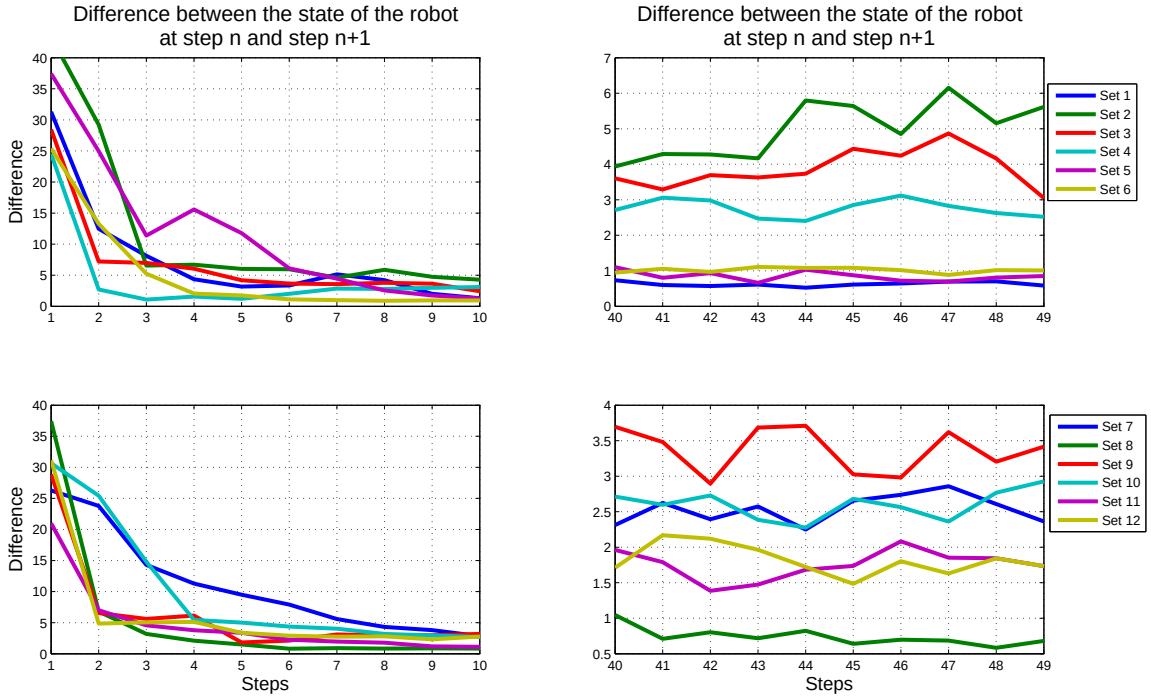


(c) Angle in degree of the right hip, knee and ankle.

Figure 4.19: Experiment 3: analyze of the walking gait over one stride of parameter set 6.



(a) UP: Stability against the initial condition DOWN: Robustness against noise in muscles



(b) Convergence to the limit cycle

Figure 4.20: Experiment 3: Stability and robustness

n	t_{max}	v_{max}	w_1	w_2	k
60	400	0.03	2.05	2.05	1.0

Table 4.2: Parameters of the PSO algorithm

4.4.1 Introduction to Particle Swarm Optimization algorithm

The concept of the particle swarm optimization algorithm comes from social behavior studies. In 1995, James Kennedy and Russell Eberhart set up a quite simple algorithm able to mimic the bird flocking or fish schooling behavior [48].

We remember the concept of fitness function introduced in 4.7b: for each point in a parameter space we can measure its corresponding fitness. The goal of any optimization is to maximize this fitness.

The PSO algorithm is initialized with n particles dispatched randomly in a parameter space (or in a subspace of the whole parameter space). Each particle p is defined by it position $\overrightarrow{x_p}$ and its velocity $\overrightarrow{v_p}$ in the parameter space.

At each epoch $t\epsilon\,[0; t_{max}]$, we:

1. Evaluate the fitness of each particle.
2. Update the speed of each particle: $\overrightarrow{v_p}(t) = \overrightarrow{v_p}(t-1) + w_1 \cdot r_1 \cdot \left(\overrightarrow{x_p^{best}} - \overrightarrow{x_p}(t-1)\right) + w_2 \cdot r_2 \cdot (\overrightarrow{x_{best}} - \overrightarrow{x_p}(t-1))$
3. Update the position of each particle: $\overrightarrow{x_p}(t) = \overrightarrow{x_p}(t-1) + \overrightarrow{v_p}(t)$
4. Back to 1. until the algorithm converges or the maximum number of epoch t_{max} is reached.

where r_1 and r_2 are two random numbers in the range $[0; 1]$, w_1 and w_2 are two constants to tune, $\overrightarrow{x_p^{best}}$ is the coordinate where particle p experienced its best fitness and $\overrightarrow{x_{best}}$ is the coordinate of the best fitness ever found by every particles and the norm of the speed v_p can not exceed a speed threshold v_{max}.

Any particle will then be statistically attracted by its own best position in the parameter space and by the best position of the whole population of particle. With time, the flock will converge to the best solution found so far. The best solution can be a local maxima, it is therefore not guaranty to find the optimum solution.

To avoid the premature collapse of the flock in a solution, [49] gives a variant of the update of the particle speed:

$$\overrightarrow{v_p}(t) = \chi \left(\overrightarrow{v_p}(t-1) + w_1 \cdot r_1 \cdot \left(\overrightarrow{x_p^{best}} - \overrightarrow{x_p}(t-1)\right) + w_2 \cdot r_2 \cdot (\overrightarrow{x_{best}} - \overrightarrow{x_p}(t-1)) \right)$$

with

$$\chi = \frac{2k}{|2 - w - \sqrt{w^2 - 4w}|}$$

and

$$w = w_1 + w_2 > 4$$

χ is called the constriction coefficient and is responsible for the decaying norm of the speed $\overrightarrow{v_p}(t)$ through time, where $k\epsilon\,[0; 1]$ define the speed of convergence of the algorithm. If $k = 1$ then we are in presence of a slow convergence.

Table 4.2 summarizes the parameters used in our simulation.

4.4.2 Method

With the PSO algorithm we do not need to start from an initial condition as with the previous home made optimization algorithm. Every particle are randomly initialized in our rescaled parameter space and the optimization is run from this initial condition. Therefore, each run is different from an other.

Since we have 60 particles each of them being evaluated over 400 epochs, we have a total of 24'000 trials, which is 2'000 trials less than with our previous optimization algorithm.

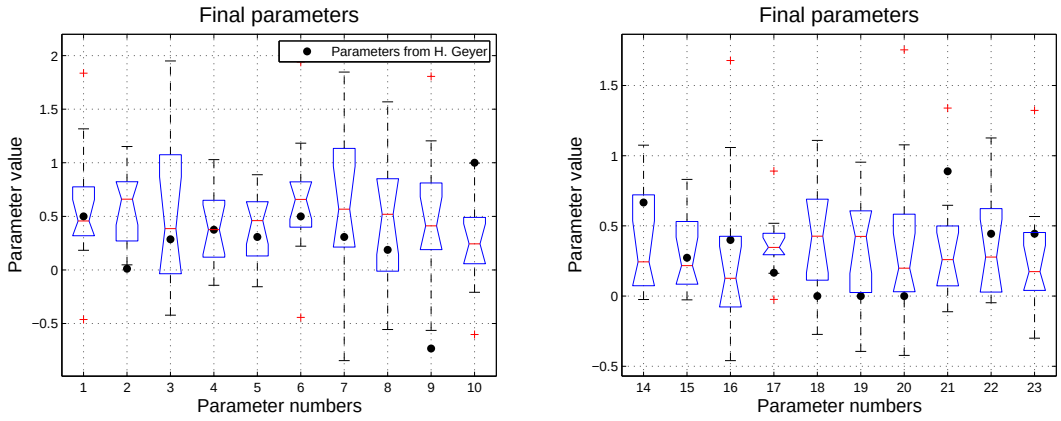


Figure 4.21: Experiment 4: Boxplot showing our final parameters and the H.Geyer parameter set.

4.4.3 Results

4.4.3.1 Experiment 4: running the PSO algorithm

Results of the optimization

We run 10 times the PSO algorithm. Table 7.4 summarizes our ten final parameter sets and their fitness. The best result is achieved with parameter set number 3 which allow the robot to walk 39 meters with 5000[J].

Figure 4.21 shows the H.Geyer parameter set and boxplots showing our best parameters obtained with the PSO.

Figure 4.22a, 4.22b and 4.22c shows the value of each parameter at each epoch taken from the best particle of the population. We can see the parameters converge within 200 epochs to their final value. Figure 4.22d shows the optimization of energy which reach a threshold after 200 epochs.

Walking gait analysis

Figure 4.23 shows the walking gait of parameter set 3 after optimization. As a results of energy consumption optimization, the whole body is very static: the trunk do almost not move and the feet are just above the ground in swing phase.

Figure 4.24a shows the activation of muscles. The results are very similar to the ones found in experiment 3 with the home made optimization algorithm (fig. 4.19a).

Figure 4.24b shows the ground reaction force which shows again an absence of shock absorption at heel strike.

Finally, figure 4.24c shows the joint angles of the right leg which are very similar to the one found in experiment 3 (fig. 4.19c).

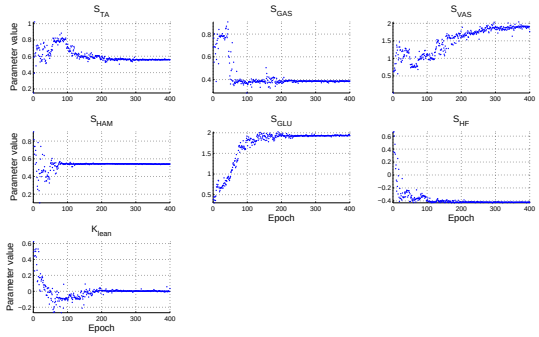
Robustness analysis

Figure 4.25a shows the stability against different initial conditions and the robustness against muscle noise. We can clearly see how the solutions are not stable at all. How can we explain the optimization algorithm found good solutions which are not stable (7 out of 10 runs have even 0% of stability)?

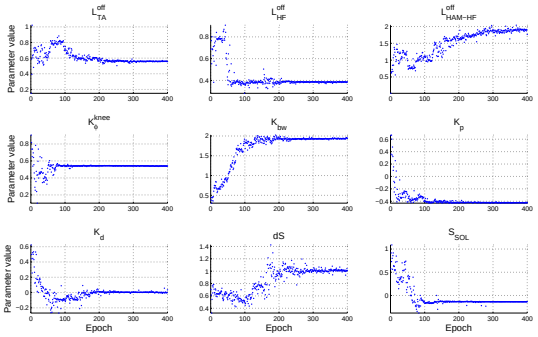
We remember the particles are attracted by their own best position and by the position of the best solution found so far. Therefore, if a parameter set has a very low probability of success and if by chance the humanoid does not fall, it can still achieve a very efficient trial in terms of energy consumption. Such a parameter set will then act as an attractor for the particles and they will converge toward the best trials ever experienced.

We remember we run 100 trials of 100 steps with each final parameter sets to compute the stability (the probability to not fall). For parameter set 3, none of these 100 trials has been successful, leading to a stability of 0%. However, the real probability to not fall is slightly above 0%, which allow us to generate plots on figure 4.22, 4.23 and 4.24.

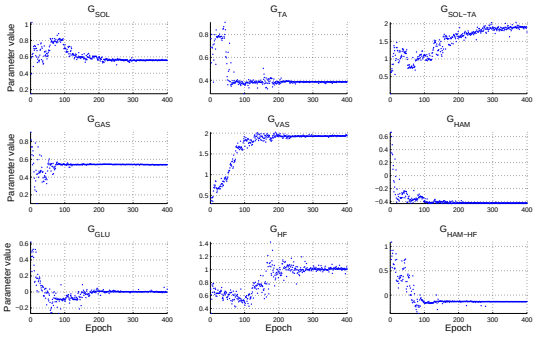
Figure 4.25b shows the convergence to the limit cycle and the regularity of the gait. For some parameter set, it has not been possible to collect enough data because the lack of stability.



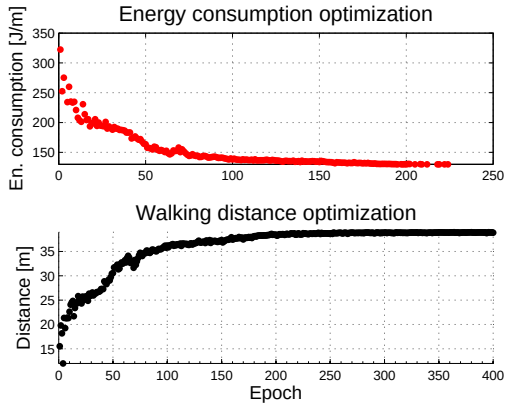
(a) Optimization of parameters



(b) Optimization of parameters



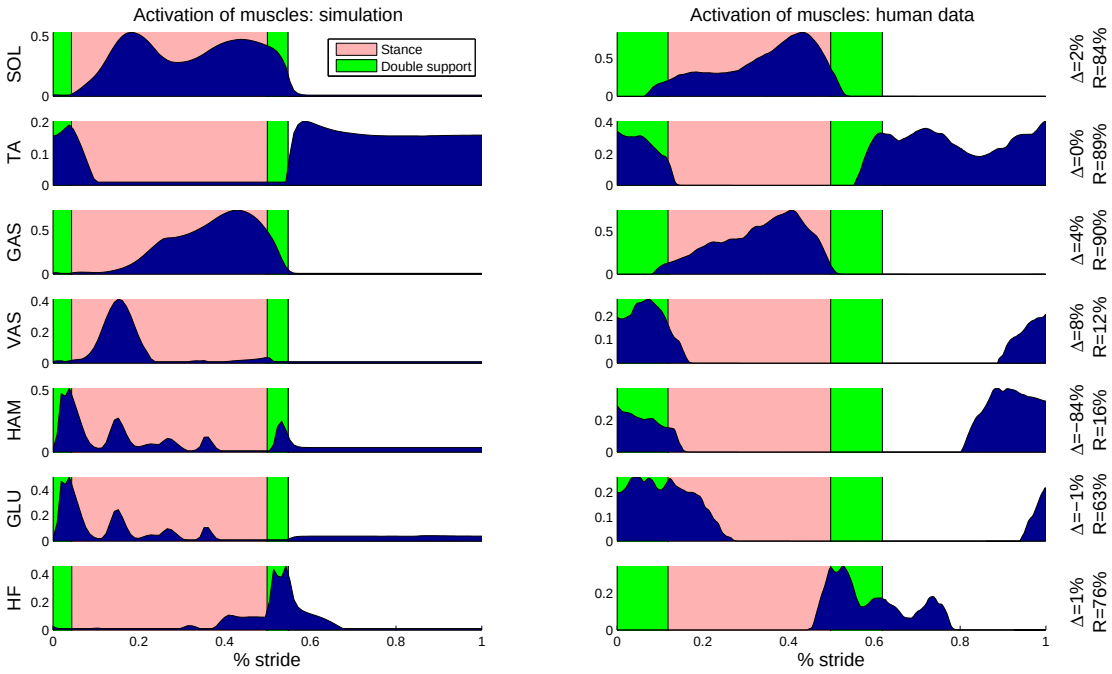
(c) Optimization of parameters



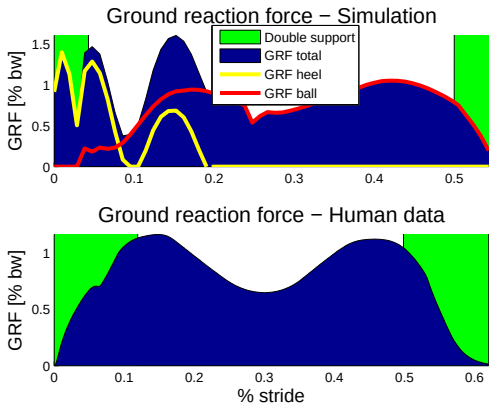
(d) Energy consumption optimization and the distance walked with 5000[J].

Figure 4.22: Experiment 4: Optimization of the parameter set number 3.

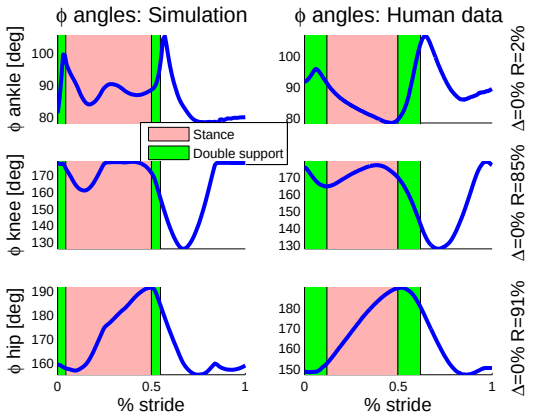
Figure 4.23: Experiment 4: Walking gait of final parameter set 3. Each picture is sample every 50[ms].



(a) Activation of muscles over one step.

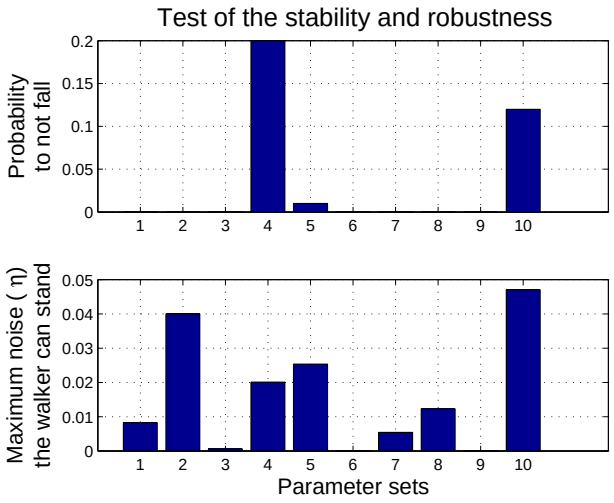


(b) The ground reaction force in the direction of the gravitation, measured from the foot sensors.

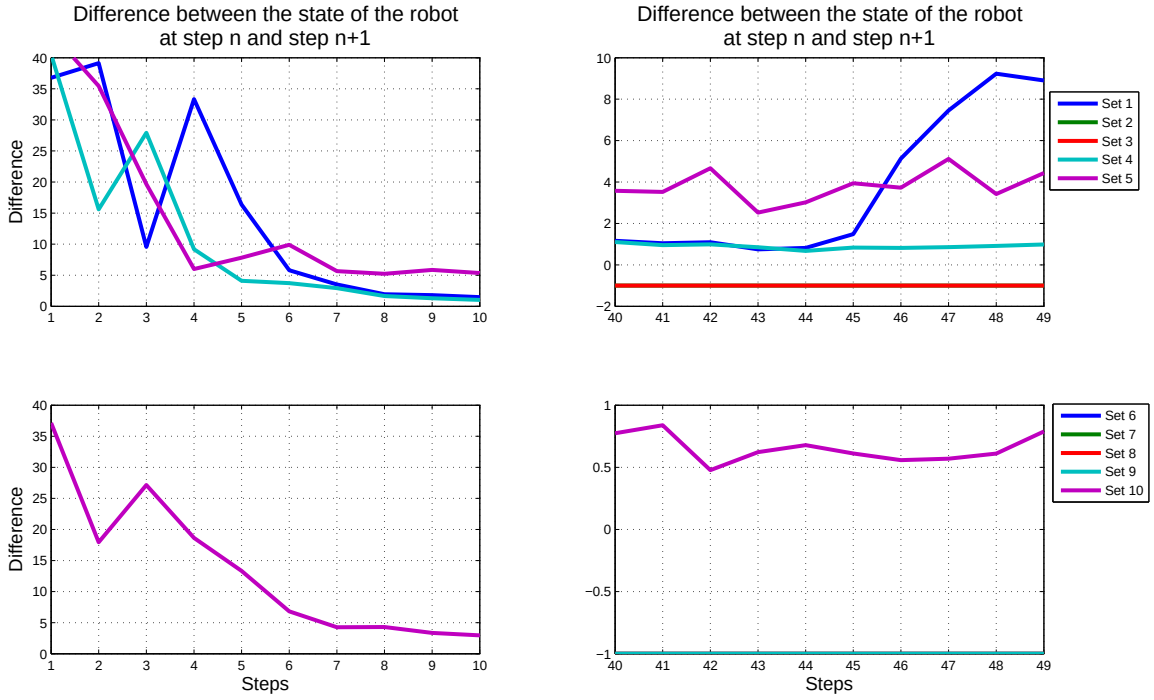


(c) Angle in degree of the right hip, knee and ankle.

Figure 4.24: Experiment4: Walking gait analysis of parameter set 3 after optimization with the PSO algorithm.



(a) Up: Stability against the initial condition Down: Robustness against noise in muscles



(b) Convergence to the limit cycle and regularity of the gaits

Figure 4.25: Experiment4: Stability and robustness of parameter set 3.

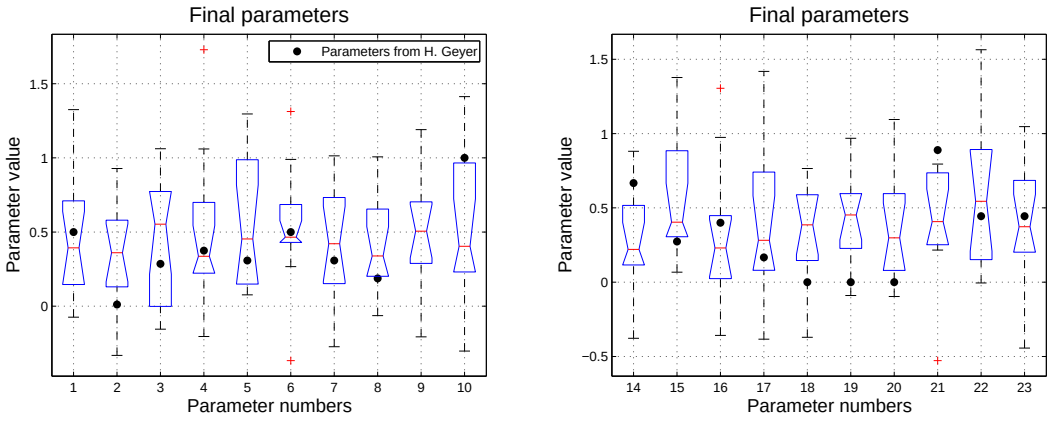


Figure 4.26: Experiment 5: Boxplot showing the statistical value of the final parameters as well as the parameter set of H.Geyer.

Discussion

Running the PSO algorithm allows us to find very good parameter sets in term of energy consumption. The gait analysis of the best parameter set is similar to the gait found with the previous algorithm. On contrary to the previous algorithm, the PSO algorithm converges twice faster (within 12'000 trials versus 26'000 trials) to a good solution.

However, the solutions are not stable at all. It is therefore not possible to efficiently optimize a walking gait with the PSO algorithm without optimizing stability at the same time.

How can we explain the gaits are not stable despite they look similar to the previous optimization algorithm? As we already notice, the final energy consumption is much lower with the PSO algorithm than with the previous optimization algorithm. A gait which minimize the energy consumption will slightly lift the foot above the ground in swing phase. Therefore, a little variation is sufficient to trip the foot on the floor and unbalance the robot.

4.4.3.2 Experiment 5: running the PSO algorithm with stability

In this experiment, we run again 10 times the PSO algorithm focussing on the stability of the solution in a similar fashion as was done in the home made algorithm experiments.. Each particle is evaluated three times before validation, as in experiment 1 with the random generation of parameter sets. If the robot falls before the end of the 3 trials, we return a fitness of 0. The total number of trials of the whole run range from 24'000 to 72'000.

Results of the optimization

Table 7.5 summarizes the final parameters and fitness of each of the 10 runs. In general, these 10 runs can not optimize the energy consumption as much as the previous experiment (average of 28.4 [m] walked with 5000[J] against 34.8[m] in experiment 4).

Figure 4.26 shows boxplot of our final 10 parameter sets.

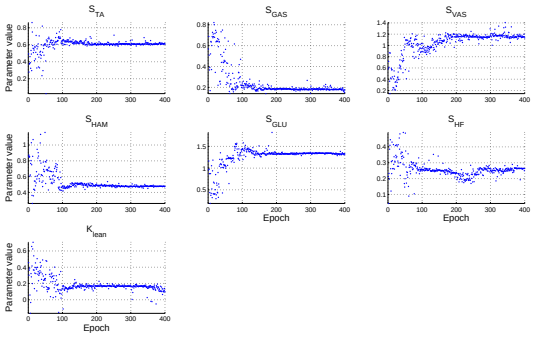
Figure 4.27a, 4.27b and 4.27c show the optimization of each parameter separately for the parameter set 1. At each epoch, we plot the parameter set of the best particle of the population. Figure 4.27d shows the energy optimization. We can see the algorithm converge within 100 epochs with a final walked distance of 31 meters.

Walking gait analysis

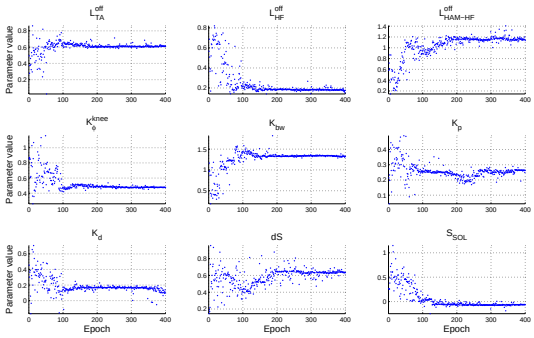
When we look at the gait, they do not seems to be very natural. They are all either walking on the heel or have a kicking like behavior. Figure 4.28 illustrates these two typical gaits. On the top stripe we can see the gait of the parameter set 1 which tend to kick at each step. In the bottom stripe, the humanoid is mainly walking on the heel, slightly crouch forward.

Figure 4.29a shows the activation of muscles.

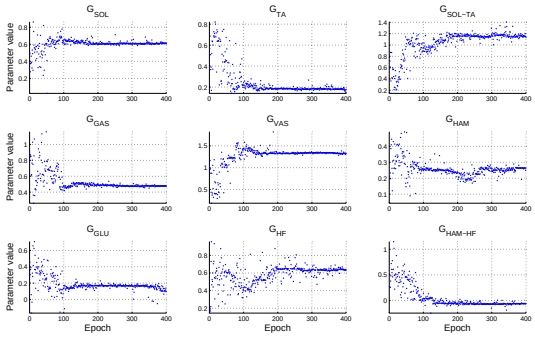
Figure 4.29b shows the ground reaction force which seems to be flat because of the high pick of force (3 times the body weight) in double support phase.



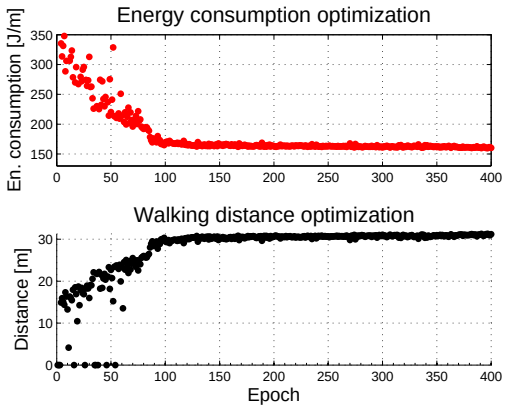
(a) Optimization of parameters.



(b) Optimization of parameters.



(c) Optimization of parameters.



(d) Energy consumption optimization and the distance walked with 5000[J].

Figure 4.27: Experiment 5: Optimization of parameter set 1.

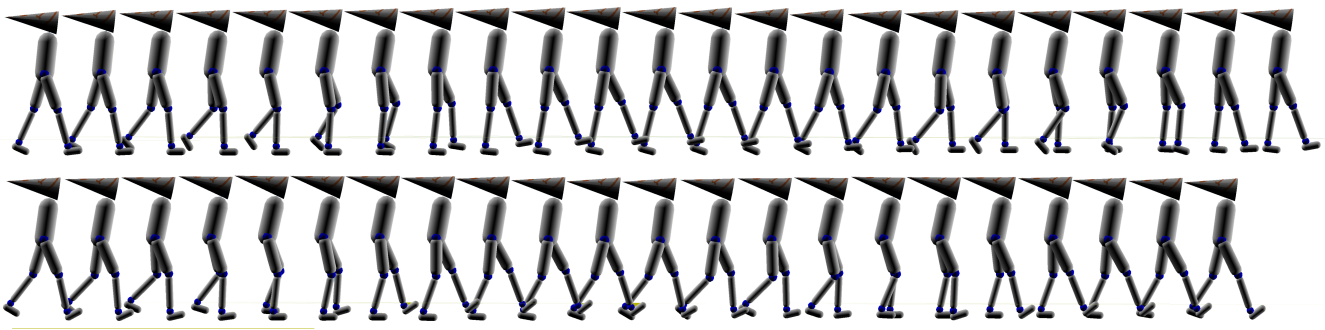


Figure 4.28: Experiment 5: Up: Parameter set 1 is the best optimization but kicks while walking. Down: Parameter set 3 is an example of walking gait on the heels.

Figure 4.29c shows the angle of each joint. The knee still reaches its soft limit, however it does not stuck to it as in the previous experiment.

Robustness analysis

Figure 4.30a shows the results of stability and robustness against noise in muscle. It is obvious our final gaits are much more stable than in the previous experiment. Furthermore, the robustness against noise is improved as well.

Figure 4.30b shows the convergence to the limit cycle and the regularity of the gait. We do not observe a correlation between stability and convergence to the limit cycle/regularity.

Discussion

In this experiment, we run 10 times the PSO algorithm and test three times each trial to add pressure of selection on stability. We found the stability is therefore highly improved, however the energy optimization could not perform as well as before. It sounds logical that there exists a trade off between energy consumption and stability. Being very energetically efficient means using the lowest muscle activation possible. It is therefore not possible to cope with disturbance without using higher muscle activation. An other example of the trade off between energy and stability is the foot clearance in swing. Lifting the foot above the ground in swing cost energy. Optimizing energy would tend to lift the foot as little as possible which increase to probability to hit the ground and to become unstable.

By watching the humanoid walk at the end of these optimizations, we observe the gaits do not look like very natural.

4.4.3.3 Experiment 6: running the PSO algorithm with stability and noise in muscles

In this experiment, we run 10 times the PSO algorithm as in the previous experiment, but we add signal dependent noise in muscle during the whole optimization.

In 1998, C.M.Harris and D.M.Wolpert have shown how signal dependent noise determines motor planning. In [26] they *"present a unifying theory of eye and arm movements based on the single physiological assumption that the neural control signals are corrupted by noise whose variance increases with the size of the control signal"*. Therefore, minimizing the neural control signals would minimize the noise and therefore minimize the error of any movement. In our case, we apply a signal dependent noise directly on the muscle activations with equation 4.1 ($\eta = 0.15$). Such noise is closely related with energy consumption: minimizing the noise means minimizing muscle activation and therefore minimizing energy consumption. Furthermore, minimizing noise would increase stability and robustness.

Results of the optimization

Figure 4.31 shows the H.Geyer parameter set as well as boxplots showing the statistical distribution of our final parameters. Note we are close to the H.Geyer parameters, except for parameter 9 G_{HAM-HF}.

Figures 4.32a, 4.32b and 4.32c show the optimization of each parameter. Figure 4.32d shows the optimization of energy consumption. We can see the optimization reach a threshold after 250 epochs with a final walked distance of 35.1[m] which is higher than in experiment 5.

Walking gait analysis

We observe a walking gait which looks natural, as illustrated on figure 4.33.

Figure 4.34a shows the muscle activation. We still observe a double bump for the soleus muscle which should not be activated so early.

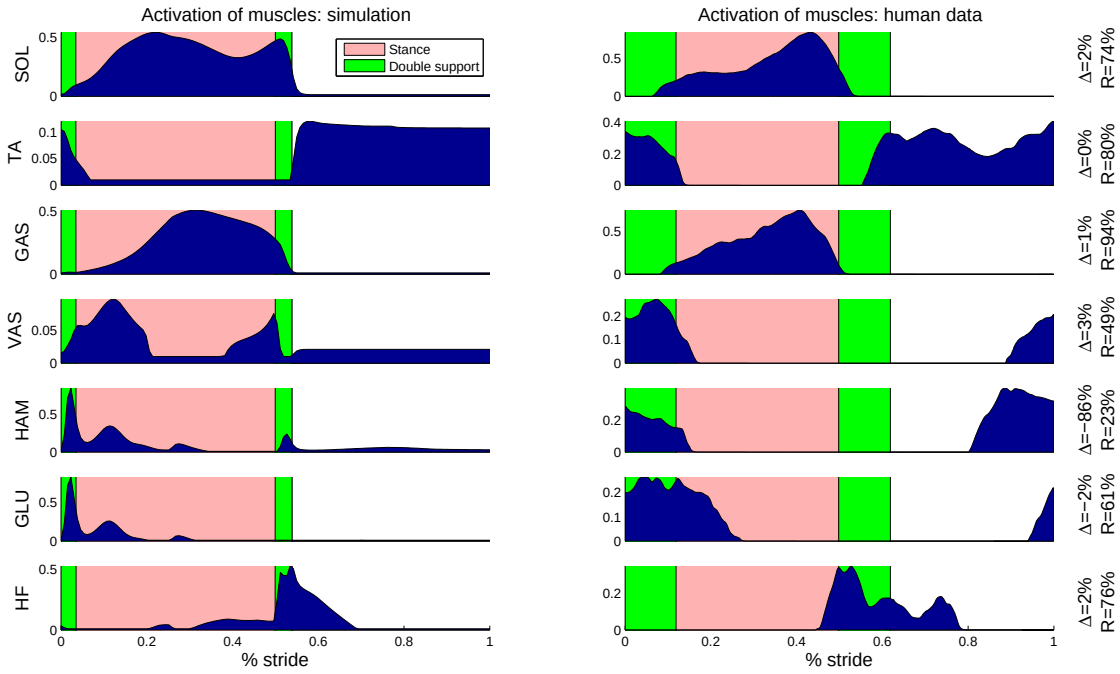
On figure 4.34b we can see the ground reaction force which still begins with a first jerk and which is followed by two bumps which are representative of the human GRF.

Figure 4.34c shows the joint angles. We still observe the saturation of the knee at the soft limit.

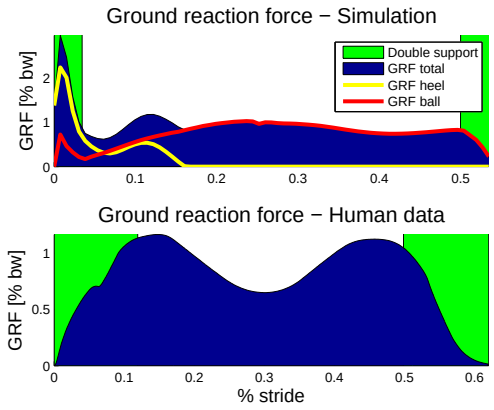
Robustness analysis

Figure 4.35a illustrate the stability and robustness of the solutions. This time, as in experiment 1, the solution are very stable, except for solution 5, 7, 9 and 10. Furthermore, we can see how the gait is robust against noise in muscles.

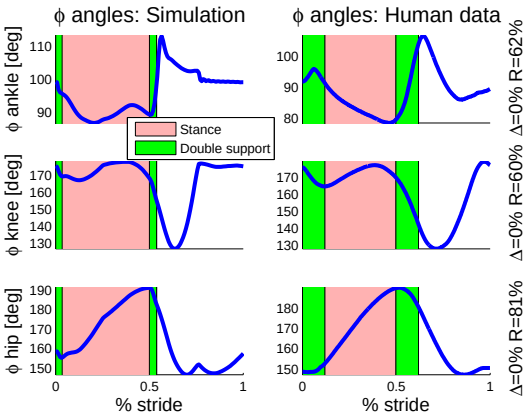
In figure 4.35b we can see the unstable solutions 5, 9 and 10 have more difficulty to reach their limit cycle at the beginning of each trial. However, we do not observe a correlation between stability, robustness and regularity of the gait.



(a) Activation of muscles over one step.

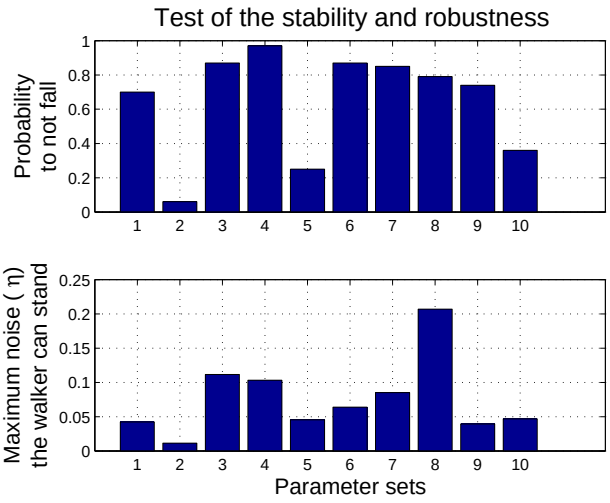


(b) The ground reaction force in the direction of the gravitation, measured from the foot sensors.

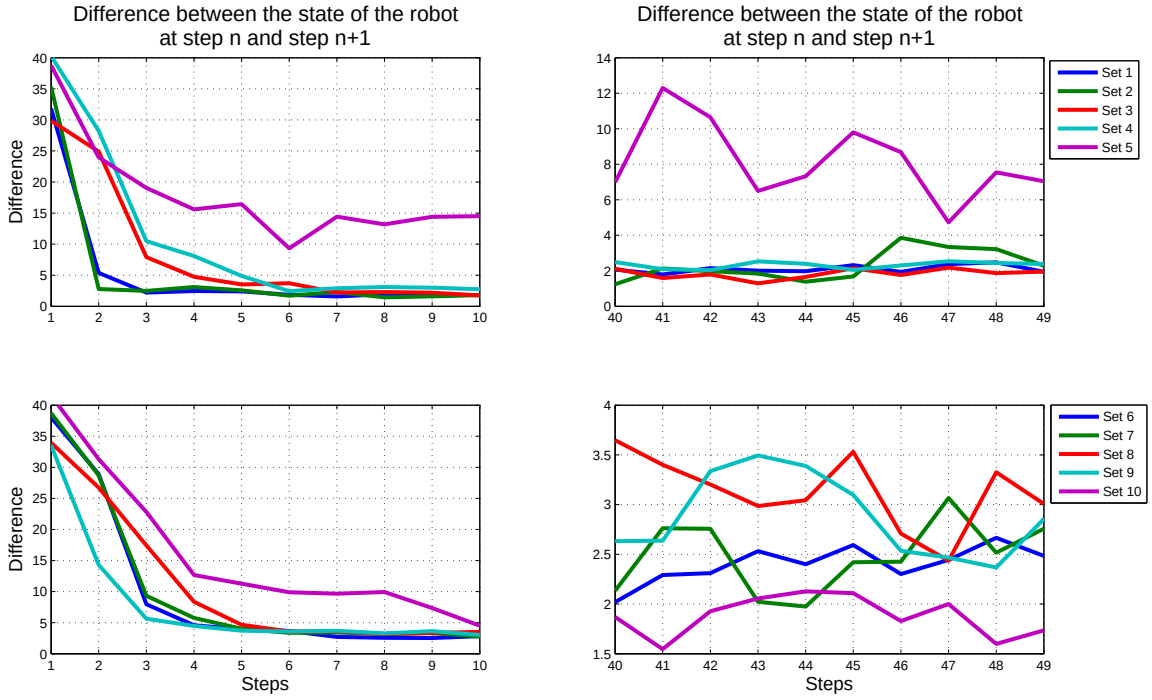


(c) Angle in degree of the right hip, knee and ankle.

Figure 4.29: Experiment 5: Solution 1.



(a) UP: Stability against the initial condition DOWN: Robustness against noise in muscles



(b) Convergence to the limit cycle

Figure 4.30: Experiment 5: Stability and robustness

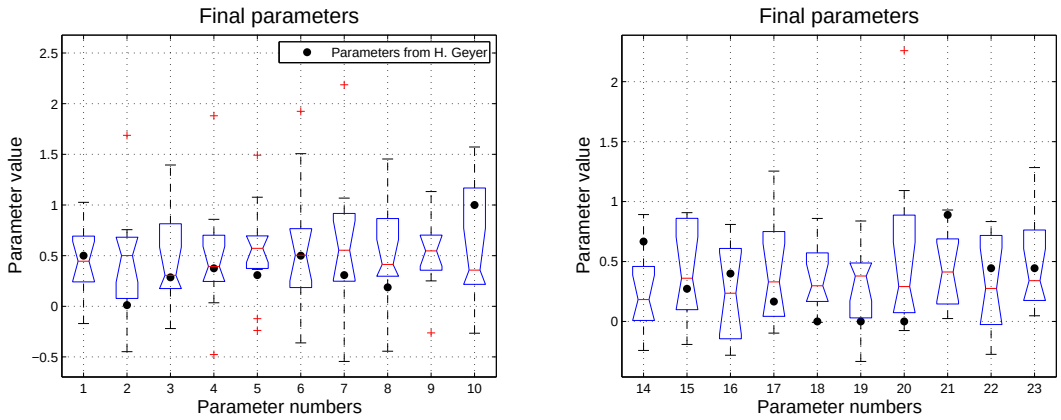
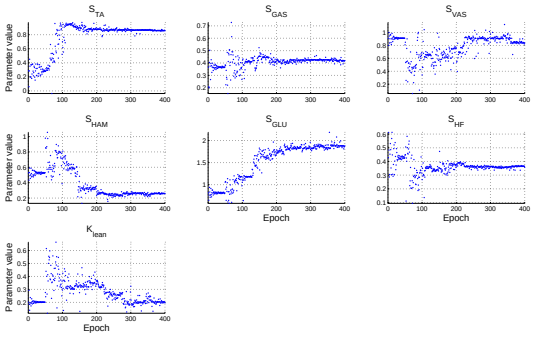
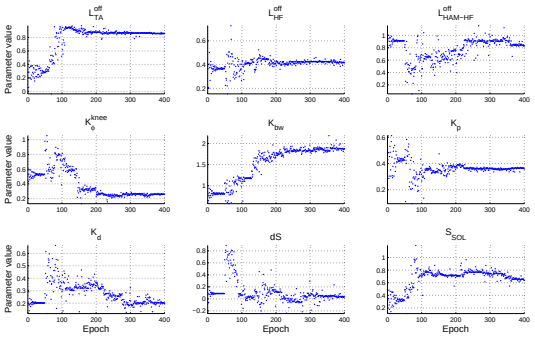


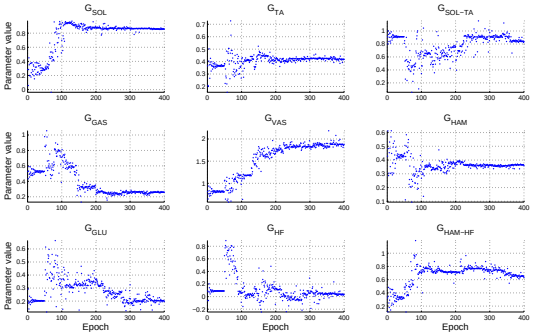
Figure 4.31: Experiment6: Boxplot showing the final parameters after optimization and the H.Geyer parameter set.



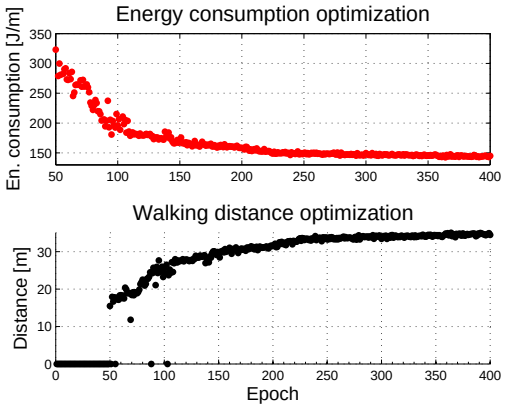
(a) Optimization of parameters.



(b) Optimization of parameters.



(c) Optimization of parameters.



(d) Energy consumption optimization and distance walked with 5000[J].

Figure 4.32: Experiment 6: Optimization of parameter set 4.

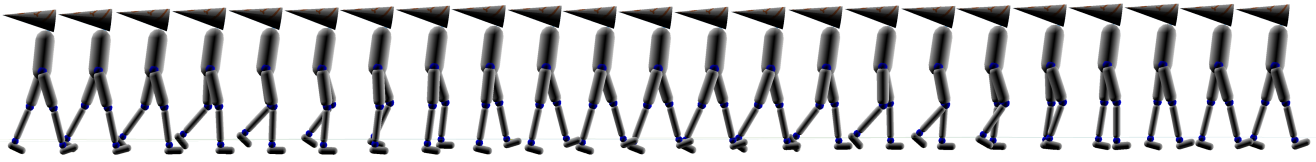
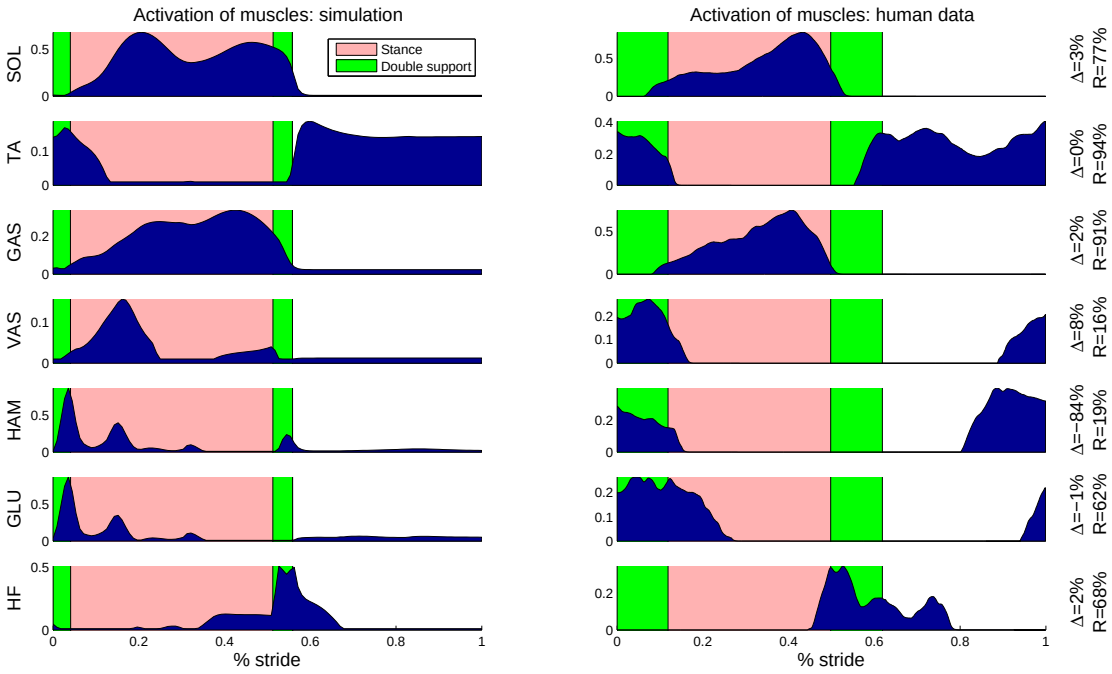
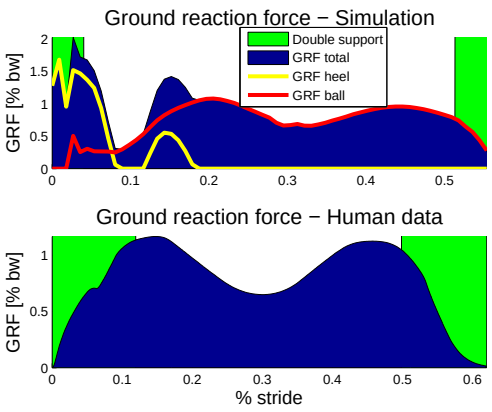


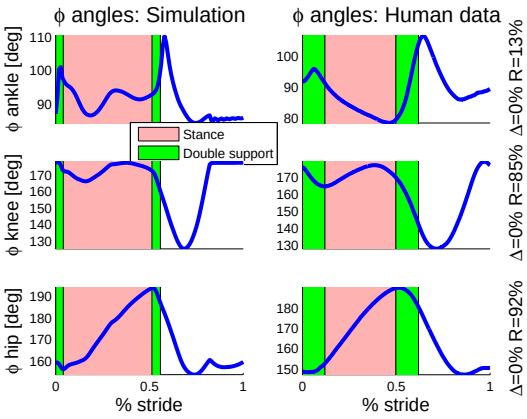
Figure 4.33: Experiment 6: walking gait stripe of the parameter set 4 which is the best optimization. Each picture is sampled every 50[ms].



(a) Activation of muscles over one step.

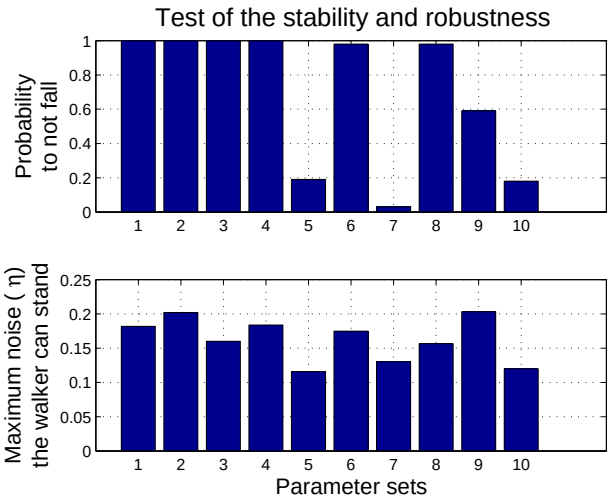


(b) The ground reaction force in the direction of the gravitation, measured from the foot sensors.

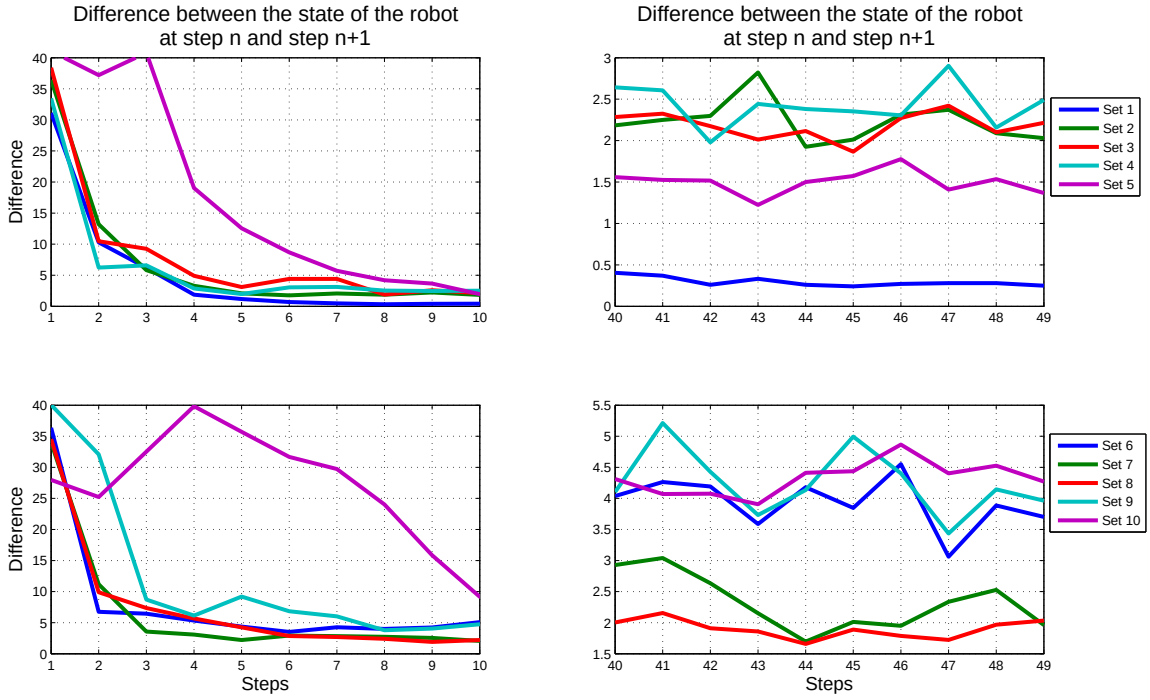


(c) Angle in degree of the right hip, knee and ankle.

Figure 4.34: Experiment 6: Final solution of parameter set 4.



(a) UP: Stability against the initial condition DOWN: Robustness against noise in muscles



(b) Convergence to the limit cycle

Figure 4.35: Experiment 6: Stability and robustness

Discussion

In this last optimization experiment, we add signal dependent noise in muscle activation. The solutions found are statistically more stable against initial condition, more robust against noise in muscle activation and look more natural. It seems therefore very important to take signal dependent noise in account in the optimization.

Looking at the muscle activation, joint angles and ground reaction force shows we do not perfectly match human data.

Chapter 5

Stumbling corrective response

5.1 Introduction

If someone hits an obstacle with his foot while walking (during swing phase) while walking, a series of reflexes will compensate the unexpected perturbation. Such reflexes has been termed a *stumbling corrective reaction* by Forssberg in 1975 [25] because this reaction helps animals in maintaining their balance despite the perturbation. Such reflexes has been invasively studied in cats [50] and non-invasively studied in man [43, 39, 22].

Here, we are interested in the human corrective response to stumbling. The chain of reflexes triggered by tripping the swing foot in an obstacle while walking is phase dependent. The reflexes will not be the same if the foot hits the obstacle in early or late swing. Basically, if the obstacle is met in early swing, the foot will be retracted, elevated above the obstacle and bring back on the floor. In the other case, the foot will be directed to the floor. These two strategies respond to two different states: in early swing, the body center of mass is ahead the swinging foot, thus the need to bring the swing foot forward. In late swing, the foot is already forward and the body center of mass is accelerating downward, so the best way to avoid falling is to put the foot as quickly as possible on the floor.

We are interested in the early swing stumbling correction strategy and we did not try to create a controller for the late swing strategy.

5.1.1 Early swing stumbling response

We will define the time zero as the very first time the obstacle is detected. Furthermore, we will define the ipsilateral leg as the leg which meets the obstacle (in swing phase).

First, the foot has to be removed from the obstacle. It has been observed in cats [50] the soleus muscle is very briefly excited right after hitting the obstacle and the tibialis anterior muscle is inhibited (see fig. 5.1). Such activation/inhibition leads to an extension of the ankle which drives the foot away from the obstacle. At the same time, the biarticular hamstring muscle is activated which flexes the knee and extends the hip. This muscle activation will take the whole swing leg away from the obstacle.

Second, the foot has to be brought above the obstacle. To this end, every flexor muscle is recruited. The tibialis anterior muscle is activated and the soleus muscle inhibited, which flexes the ankle. The hip flexor muscles and knee flexor muscles are activated to drive the leg forward and elevate it above the obstacle. To prevent the whole trunk to rotate forward under the activation of the hip flexor, an extra stimulation of the gluteus muscle of the contralateral leg has been observed in humans [24]. At the same time, the contralateral leg is extended to increase the body height.

Third, the foot has to be brought back on the floor with the help of extensor muscle activation. The extension of the contralateral leg (and thus the elevation of the whole body) provides an extra time to extend the swing limb in preparation for the landing.

5.2 Modeling the stumbling corrective response

In these simulations, the right leg will be the one which hits a box during the swing phase (10 cm height). Therefore, we embedded a force sensor on the toe of the right foot which can sense if the foot hits an obstacle. If so, three signals become active (SCR_1, SCR_2, SCR_3 are switched to 1) and decay as:

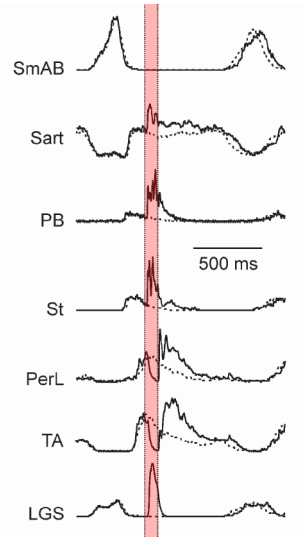


Figure 5.1: Records of intracellular membrane potential of motoneurons in cat. Artificial stimulation of superficial peroneal nerve (which provoke the SCR) is shown with pink area. HIP MOTONEURONS: sartorius (Sart) and semimembranosus and anterior biceps (SmAB). KNEE MOTONEURONS: posterior biceps and semitendinosus (PBSt). ANKLE MOTONEURONS: tibialis anterior (TA) and lateral gastrocnemius and soleus (LGS). Taken from [50].

$$\frac{dSCR_x}{dt} = -\frac{SCR_x}{\tau_x}$$

where τ_x is a time constant to be determined and $\tau_1 < \tau_2 < \tau_3$.

We choose to use 3 different signals to represent the three phase of the stumbling reaction: retraction of the foot from the obstacle, elevation of the foot above the obstacle and landing of the foot. Furthermore, we choose to decay these three signals with a first order differential equation because we can easily observe on figure 5.1 such decaying process in motoneuron membrane potential.

The previously descrived feedback (see section 2.4) used for the walking gait are still active (table 2.1). We will therefore use the three above signals to modulate the stimulation of some muscles. The total stimulation of a given muscle will be the sum of the walking gait stimulation (S_m, where m is a muscle) and the stumbling corrective reaction stimulation (S_m^{SCR}): $S_m^{tot} = S_m + S_m^{SCR}$.

The SCR stimulation rules are shown on table 5.1 and explained below.

Retraction of the foot from the obstacle. The first reflex is to retract the foot from the obstacle. To do so, we activate briefly the soleus $\left(S_{SOL}^{SCR} = x_1 \cdot SCR_1\right)$ and we inhibit the tibialis anterior muscle

$\left(S_{TA}^{SCR} = x_3 \cdot SCR_2 - x_2 \cdot SCR_1\right)$ to extend the ankle. At the same time, we inhibit the vastus muscle $\left(S_{VAS}^{SCR} = x_6 \cdot SCR_3 - x_5 \cdot SCR_2\right)$ to flex the knee and activate the gastrocnemius $\left(S_{GAS}^{SCR} = x_4 \cdot SCR_2\right)$ and the hamstring muscle

$\left(S_{HAM}^{SCR} = x_7 \cdot SCR_2 - x_8 \cdot SCR_3 \cdot S_{VAS}\right)$. Activation of the gluteus muscle $\left(S_{GLU}^{SCR} = x_9 \cdot SCR_1\right)$ briefly extend the hip.

To summarize, knee flexion coupled with ankle and hip extension contribute to drive the leg backward and to clear the foot from the obstacle.

Driving the foot above the obstacle The second reflex is used to drive the foot above the obstacle. The first SCR signal (SCR_1) decays rapidly to zero, therefore the initial stimulation of the soleus and gluteus decays to zero as well. The tibialis anterior starts to be excited by the SCR_2 signal $\left(S_{TA}^{SCR} = x_3 \cdot SCR_2 - x_2 \cdot SCR_1\right)$, which flexes the ankle. Since the vastus stays inhibited by the SCR_2 signal $\left(S_{VAS}^{SCR} = x_6 \cdot SCR_3 - x_5 \cdot SCR_2\right)$ and both gastrocnemius $\left(S_{GAS}^{SCR} = x_4 \cdot SCR_2\right)$ and hamstring $\left(S_{HAM}^{SCR} = x_7 \cdot SCR_2 - x_8 \cdot SCR_3 \cdot S_{VAS}\right)$ are excited, the knee will continue its flexion. Finally, under the absence of activity of the gluteus muscle and the increasing stimulation of the hip flexor muscle, the hip will flex and will drive the foot forward.

Swing (ipsilateral leg)

SOL	$S^{SCR}_{SOL} = x_1 \cdot SCR_1$
TA	$S^{SCR}_{TA} = x_3 \cdot SCR_2 - x_2 \cdot SCR_1$
GAS	$S^{SCR}_{GAS} = x_4 \cdot SCR_2$
VAS	$S^{SCR}_{VAS} = x_6 \cdot SCR_3 - x_5 \cdot SCR_2$
HAM	$S^{SCR}_{HAM} = x_7 \cdot SCR_2 - x_8 \cdot SCR_3 \cdot S_{VAS}$
GLU	$S^{SCR}_{GLU} = x_9 \cdot SCR_1$
HF	$S^{SCR}_{HF} = x_{11} \cdot SCR_3 - x_{10} \cdot SCR_1$

Swing (contralateral leg)

HF	$S^{SCR}_{HF} = x_{12} \cdot SCR_3$

Stance (ipsilateral leg)

GLU	$S^{SCR}_{GLU} = x_{13} \cdot SCR_3$

Table 5.1: Feedback stumbling corrective reaction rules

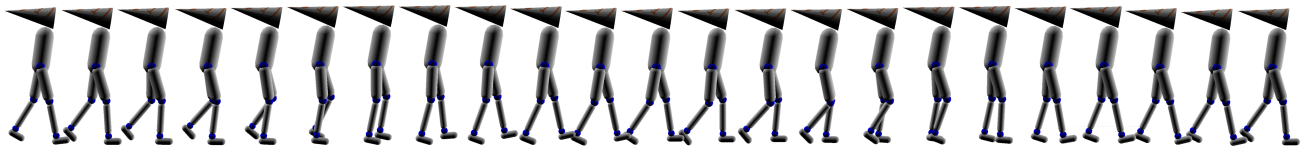


Figure 5.2: Walking gait of the robot used for the SCR experiment under normal condition (no obstacles, no noise). Each picture is sampled every 50[ms].

Bringing the foot on the floor The third reflex is to bring back the foot on the floor. Since the SCR_2 signal decays to zero, the gastrocnemius stimulation becomes zero $\left(S^{SCR}_{GAS} = x_4 \cdot SCR_2\right)$, the vastus muscle group starts to be excited $\left(S^{SCR}_{VAS} = x_6 \cdot SCR_3 - x_5 \cdot SCR_2\right)$ and the hamstring muscle is therefore inhibited

$\left(S^{SCR}_{HAM} = x_7 \cdot SCR_2 - x_8 \cdot SCR_3 \cdot S_{VAS}\right)$. This whole pattern of muscle activation extends the knee.

Decreasing the forward velocity of the body Finally, since the whole body is falling forward, we need to apply an extra stimulation of the hip flexor $\left(S^{SCR}_{HF} = x_{12} \cdot SCR_3\right)$ during the next swing phase of the contralateral leg. This extra stimulation brings the contralateral leg forward faster, which helps the robot to avoid falling. The contralateral foot is therefore placed ahead of its heel strike position under a normal walking cycle, which contributes to decelerate the body and helps the robot to recover a stable walking gait. To avoid the forward rotation of the trunk under this contralateral hip flexion torque, the gluteus muscle of the ipsilateral leg (stance leg) is excited $\left(S^{SCR}_{GLU} = x_{13} \cdot SCR_3\right)$.

5.3 Method

We use the walking parameter set 1 found in experiment 6 (see table 7.6) because it is among the best parameter sets found so far and because this gait looks natural. Figure 5.2 shows the gait of the robot under normal conditions (no obstacles, no noise in muscles and at steady state).

It is well known that the stumbling corrective reflex is modulated by the phase of the stride. If the foot hits the obstacle in mid swing, the reflex has to be sligthly different than in early swing. Therefore, we subdivide the swing phase into 10 parts and optimize parameter sets for each of those parts.

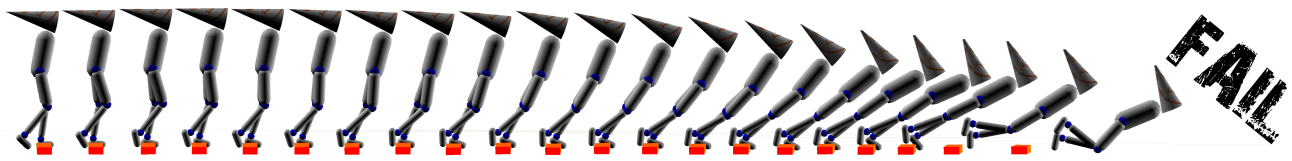


Figure 5.3: Without SCR feedback. Each picture is sample every 50[ms].

Parameter set	1	2	3	4	5	6	7	8	9	10
Range of action [% of stride]	50-53	53-56	56-59	59-62	62-65	65-68	68-71	71-74	74-77	77-80

Initially, we randomly search a *single* SCR parameter set which allows the robot to perform the SCR without falling (the obstacle is met between 62% to 68% of the stride). We then copy-past this set to the 10 sets above. This is our initial condition.

We now launch the robot such as to hit the obstacle 13 meters further (it is long enough to be certain the robot is in a steady state) between 50% to 80% of the stride (which is in swing phase). When the obstacle is met, the model uses the corresponding SCR parameter set $\overrightarrow{p}$ and adds some signal dependent noise:

$$p_i^{noised} = p_i + \delta \cdot r \cdot p_i$$

where r is a random number in the range $[-1; 1]$, p_i is the value of the parameter number i and $\delta = 0.2$ is an open parameter.

If the robot falls, we do nothing. We say the robot succeeds if it is able to walk 15 meters after the obstacle without falling. In this case, we update our parameter set as:

$$\overrightarrow{p} = \overrightarrow{p} + \eta \cdot \left(\overrightarrow{p^{noised}} - \overrightarrow{p} \right)$$

where $\eta = 0.05$ is a learning rate.

Such random search algorithm is very (very) long to run, but slowly converges to a solution. We can observe no convergence if the initial condition is too far away from the optimal solution. Therefore, we should increase the noise added to the parameters to search in a wider region of the parameter space.

5.4 Results

Control experiment

A control experiment has been done to see how the humanoid reacts when hitting an obstacle in swing without any SCR feedback. On figure 5.3 we can see the humanoid tripping the obstacle and falling. The leg does not even react to the obstacle, showing the walking feedback rules can not cope with stumbling.

SCR parameter sets

The final value of the SCR parameters are given in table 7.7 and on figure 5.4. We can see how some parameters seem to follow a smooth change from one parameter set to another. For example, τ_1 decreases if the obstacle is hit later in the swing phase. τ_2 and τ_3 do not seem to follow a regular change. The same holds for other parameters: some seem to follow more or less a smooth change and others do not. We try to match on these data some regression curves and to use them as signals for the SCR in the simulation, but it failed to give any good result. Therefore, it seems each parameter set is specific to its task.

We believe however two neighbors parameter sets should be linked in one way or another. It could be interesting to try to add some constraints on the parameter sets to keep a smooth parameter value change from one set to its neighbors. As far as now, we did not try any experiment in this way.

Testing the efficiency of the final parameter sets

Figure 5.5 shows in the upper subplot the number of times each parameter set has been tested (blue dot). If the robot does not fall after hitting the obstacle, we say it successes (red dot shows the number of successes). We run this test experiment until we get at least 100 trials per set. The bottom subplot shows the probability of success

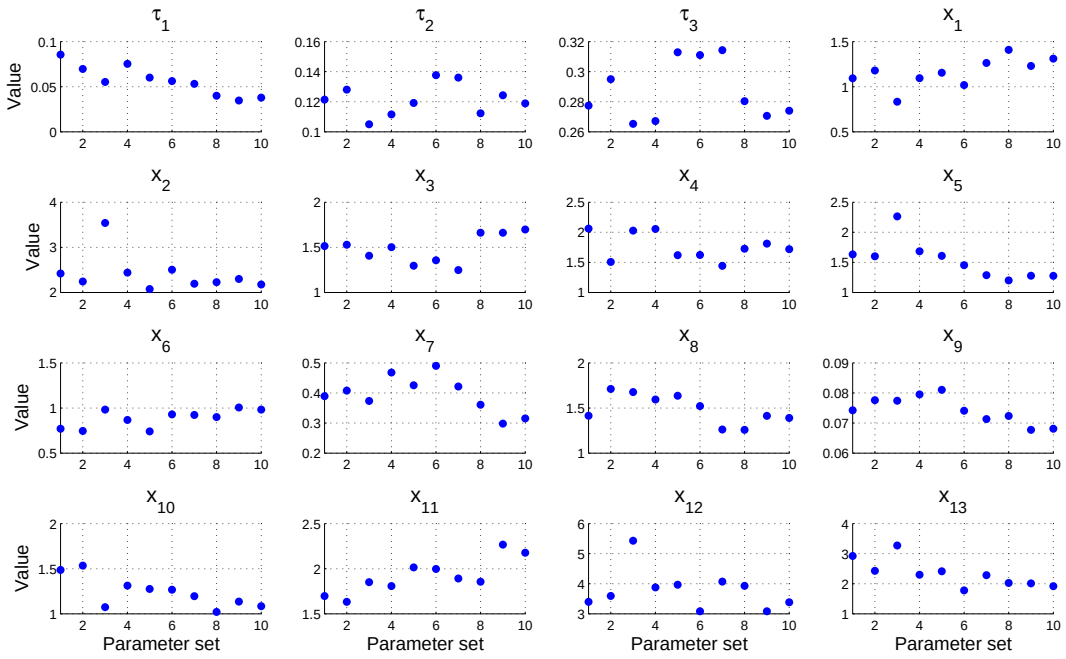


Figure 5.4: Final value of each parameter.

per set. We can see we obtain good results which allow the robot to avoid falling in 95.4% of the case (without taking in account parameter set 10).

If we observe the parameter set 10 in action, the swing phase is 77% to 80% of the total stride and lifting the foot in the air increase critically the step length. We are reaching the limit where the first SCR strategy is efficient and from 77% to 100% of the stride we should implement the second SCR strategy which bring the foot directly on the floor. This explains why we obtain only 1% of success with set 10.

Gait analysis

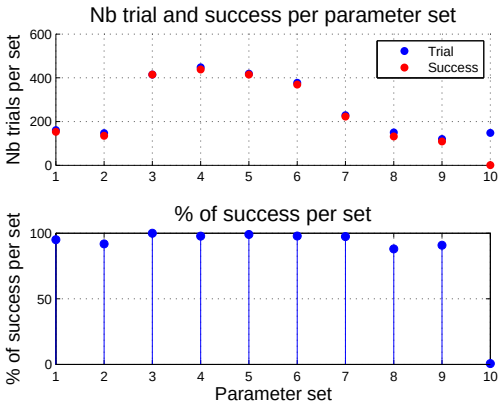
Early swing case: parameter set 1 Figure 5.8 shows the SCR when the obstacle is hit in early swing. We clearly see how the foot is lifted above the obstacle and brought back on the floor. The gait stabilizes again after the SCR.

Figure 5.7 shows the muscle activation and the extra SCR stimulation for the right leg which hits the obstacle in early swing. To avoid a long term influence of the SCR stimulation, we cut it off as soon as the SCR_3 variable is below 0.2. In the early swing case, the SCR feedback becomes inactive even before the end of the swing phase, showing it is not necessary to extra stimulate some muscles at the next step as initially thought.

Mid swing case: parameter set 9 Figure 5.8 shows the SCR if the obstacle is met in middle swing. We clearly see how the foot is lifted above the obstacle and brought back on the floor, faster than in the early swing case. The gait stabilizes again after the SCR.

In figure 5.9 we can see the muscle activation and SCR stimulation for the right leg. Again, the SCR stimulation has been cut off as soon as the SCR_3 variable reaches its threshold of 0.2. We can see this happen at the beginning of the stance phase of the right leg. It is therefore again not necessary to extra stimulate muscles after the end of the swing phase.

It is clearly visible that the time constant τ_1 is much smaller in mid swing than in early swing. This can be understood by the fact that the whole reflex has to be performed faster, since the body center of mass is already ahead of the foot support.



	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
% success	95	92	100	98	99	98	98	88	91	1

Figure 5.5: Percentage of success per parameter set. UP: blue dot show the number of time we test a given set and the red dot the number of time the test is positive. BOTTOM: for each parameter set, we compute the probability of success.

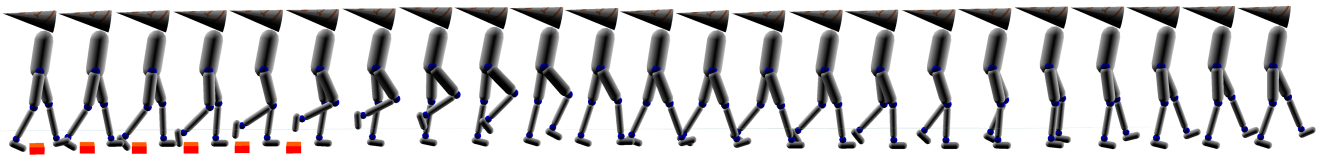


Figure 5.6: Hitting the obstacle in early swing. Each picture is sample every 50[ms].

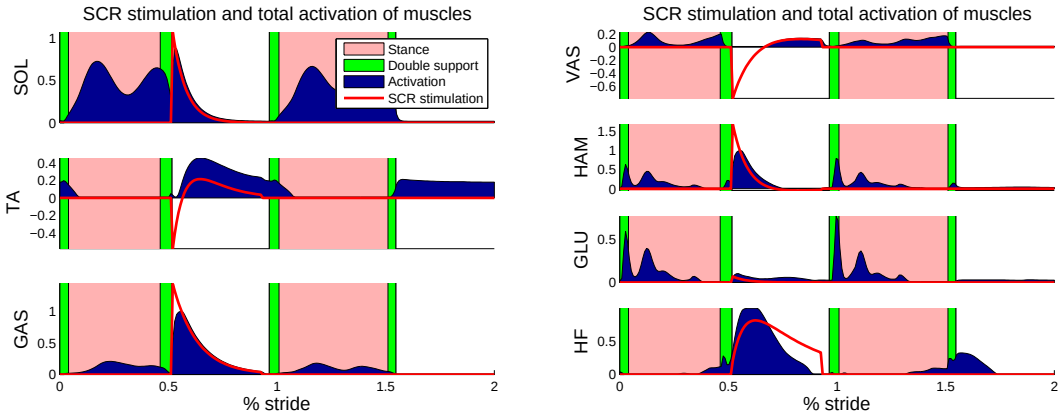


Figure 5.7: Early swing SCR: Muscle activation and SCR extra stimulation are shown.

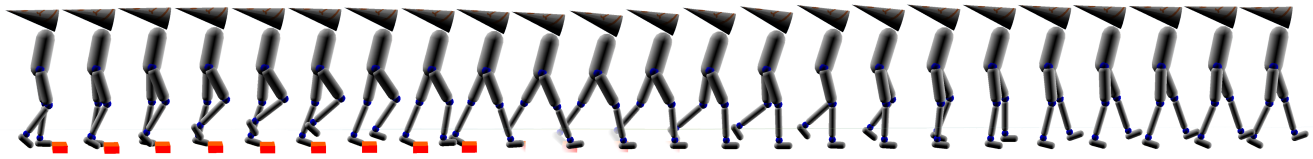


Figure 5.8: Hittiong the obstacle in middle swing. Each picture is sample every 50[ms].

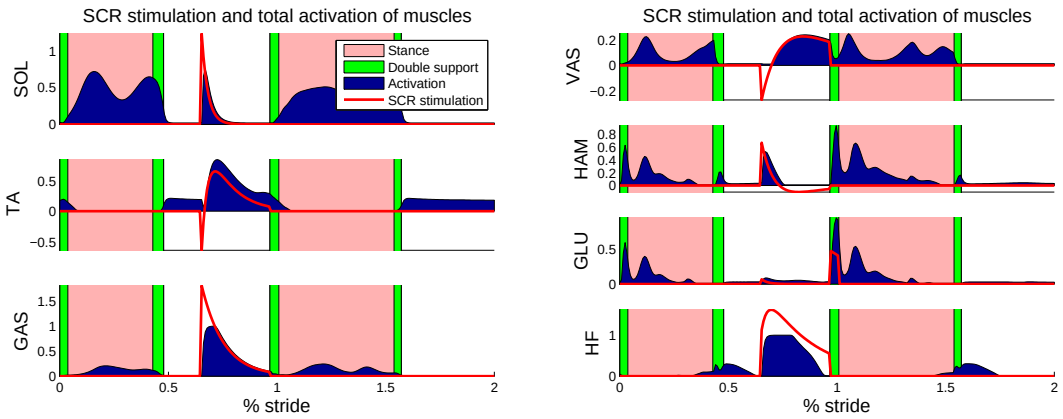


Figure 5.9: Middle swing SCR: Muscle activation and SCR extra stimulation are shown.

Discussion

We designed and implemented SCR feedback rules as well as a method to tune the parameter sets. Our simulations shows the method used to tune the parameters is very slow. However, we obtain reflexes which allow the robot to avoid falling when hitting its foot on an obstacle in 95% of the case (if the obstacle is met between 50% to 77% of the stride).

We do not systematically observe a smooth change in the parameter value of two neighbor parameter set. We explain this lack of elegance by the fact we did not link two neighbor set with any constrains. However, we strongly believe these constrains should exist and could even help the learning process. For example, we could try to optimize a continuous function which depend on the phase.

By mistake, we cut off the SCR feedback if the SCR_3 variable drop below a threshold of 0.2. However, the robot is still able to perform the SCR nicely in 95% of the case. Therefore, we do not need to extra stimulate any muscles after the end of the swing phase. In other words, we should be able to perform the same results without parameters x_{12} and x_{13}.

Chapter 6

Conclusion

Conclusion

We showed how to model the musculo-tendon complex (MTC) with their non-linear properties, how to model sensory-motor feedbacks to actuate the MTCs and how to model a humanoid as in [31]. This purely feedback controller is known to generate a stable cyclic walking gait and had the advantage of adapting motor command output from the state of the robot, which is not the case in a purely feedforward controller.

We needed to find feedback parameter sets to make our humanoid walk properly. We first generated a random parameter set and observed they allowed a stable walking gait. However, they were not looking natural, indicating such a parameter set needed further fine tuning.

Taking energy consumption as an optimization criterion, we tested two different optimization algorithms. The first one was a gradient descent-like algorithm. We first showed that any run of the optimization starting from the same initial parameter set ended on a different result, suggesting several local maximum should exist. We then optimized 12 different initial parameter sets and we observed they all ended up with a smooth walking gait. However, the stability of the gaits has been decreased by the optimization because we did not optimize explicitly for stability.

We then used PSO as a second optimization algorithm. In a first experiment, we showed that this algorithm was able to find a much better solution in terms of energy consumption, but a much poorer solution in terms of stability. To improve this, we added stability selection while optimizing the parameter sets and obtained therefore a much more stable solution. However, we note these solutions do not look natural. Therefore, it seems not sufficient to add pressure only on stability.

To improve robustness of the gait, we add signal dependent noise in muscles and we finally obtained a robust parameter set. In this last experiment of walking gait optimization, we showed a robust parameter set gives a much more natural walking gait. Since minimizing signal dependent noise is perfectly matching with energy optimization (both required to keep the activation of muscles as low as possible) it seems possible this simple property of motor command helps the learning of walking gait in biological systems.

Finally, we design, optimize and test the stumbling corrective responses. We show it is possible to avoid falling in 95% of the time the robot hits its foot on an obstacle in early- or mid-swing. There are several points on which the results can be improved further. First, the designed controller should be compared with the biological processes which yield the SCR. Second, the learning phase of the parameters should be improved (we would like to have a faster optimization phase). Third, the SCR should be extended to the late-swing which requires a different reflex.

In a more general point of view, could we design a general principle which would allow a bipedal agent to develop several reflexes to avoid to fall? We are still far from being able to take a robot out of the box and let him learn (or help him learn) to walk. Running computer-assisted optimization would be too heavy to design systematic legged agent and we believe the future belongs to online learning processes, where daily adaptation to changing mechanical properties, to unexpected breakdown or any unpredictable constraints would be more suited to our real world.

Future works

If we look at the evolution of vertebrates, there were first lamprey-like animals which slowly moved out of water with the help of evolving limbs. These higher tetrapods, as the salamanders or the newts, further evolved always more robust limbs to walk and run on unpredictable terrain, which means they had to evolve their legs, muscles, etc... as well as the way to control them.

If legs appeared only at the end of a long evolution of sensory-motor systems [44], could we find some trace of ancestral motor principles in actual mammals, surely reconfigured in a different manner? Or do the legged bodies evolve a completely new system of locomotion?

Works on lamprey allow scientific to better understand the principles of locomotion for anguilliform bodies, based on central pattern generators (CPG). [36, 35] show that such motor systems can be extended to legged animals, like salamanders, and can drive both swimming gait and walking gait within a single framework. Others studies show how it is possible to extend CPGs to quadruped [38] and biped robots [37, 40], giving them the ability to converge toward stable gaits. From a biological point of view, it is well established today mammals (and humans) possess such central pattern generators [23].

However, if the system is perturbed by external and unpredicted forces (for example in the stumbling corrective response), the walker needs fast reflexes to restore its equilibrium, which are not provided by a CPG controller. We can easily imagine such reflexes are not really useful for swimming lamprey. Indeed, water is a more stable environment than earth or air. But early creatures which evolved limbs to cope with terrestrial environment needed a much more stable controller able to deal with unpredicted terrains and a higher gravity force. Maybe these living organisms evolved reflexes on top of already existing CPG controllers. [51] shows how the helmeted guinea fowl *Numida meleagris* manages to run on unpredictable land, by controlling its proximal joints (hip and knee) mainly with a CPG feedforward strategy and its distal joints (ankle and metatarsophalangeal) mainly with a feedback strategy. Such results hypothesize a proximo-distal gradient of feedforward control.

Therefore, it seems very clear a good bio-inspired controller should integrate both feedback and feedforward strategies to, maybe, understand definitively the principles of legged locomotion?

Chapter 7

Annexes

Experiment 1: Set of feedback parameters obtained with the random generation of parameter method

N°	Name	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10	Set 11	Set 12
1	G_{SOL}	0.447	0.392	0.221	0.155	0.326	0.318	0.194	0.326	0.677	0.690	0.006	0.446
2	G_{TA}	0.577	0.365	0.434	0.769	0.877	0.380	0.597	0.962	0.819	0.065	0.356	0.111
3	G_{SOL-TA}	0.755	0.574	0.499	0.172	0.198	0.449	0.918	0.600	0.312	0.474	0.310	0.667
4	G_{GAS}	0.835	0.676	0.587	0.781	0.997	0.384	0.150	0.581	0.589	0.620	0.216	0.214
5	G_{VAS}	0.312	0.328	0.792	0.349	0.769	0.944	0.018	0.867	0.449	0.627	0.341	0.827
6	G_{HAM}	0.531	0.551	0.302	0.411	0.096	0.280	0.772	0.076	0.409	0.091	0.485	0.030
7	G_{GLU}	0.001	0.698	0.316	0.566	0.239	0.376	0.798	0.470	0.553	0.289	0.865	0.388
8	G_{HF}	0.487	0.262	0.397	0.652	0.765	0.833	0.255	0.060	0.622	0.301	0.691	0.614
9	G_{HAM-HF}	0.019	0.694	0.006	0.521	0.834	0.578	0.582	0.972	0.707	0.808	0.614	0.931
10	l_{TA}^{off}	0.909	0.099	0.461	0.700	0.671	0.445	0.105	0.691	0.683	0.064	0.242	0.823
11	l_{HF}^{off}	0.916	0.064	0.248	0.815	0.576	0.676	0.727	0.165	0.232	0.246	0.870	0.646
12	l_{HAM-HF}^{off}	0.973	0.294	0.930	0.983	0.465	0.560	0.677	0.364	0.156	0.499	0.842	0.093
13	k_{φ}^{knee}	0.734	0.735	0.750	0.019	0.920	0.451	0.706	0.866	0.177	0.890	0.681	0.264
14	k_{bw}	0.637	0.990	0.202	0.542	0.883	0.815	0.506	0.679	0.710	0.138	0.428	0.937
15	k_p	0.934	0.076	0.072	0.831	0.426	0.124	0.360	0.929	0.887	0.033	0.306	0.693
16	k_d	0.847	0.016	0.401	0.640	0.760	0.967	0.795	0.686	0.829	0.229	0.072	0.599
17	$\triangle S$	0.735	0.850	0.244	0.948	0.607	0.713	0.722	0.693	0.679	0.097	0.799	0.365
18	S_{SOL}^{0}	0.571	0.126	0.795	0.029	0.796	0.397	0.080	0.946	0.320	0.325	0.370	0.106
19	S_{TA}^{0}	0.930	0.227	0.704	0.636	0.696	0.832	0.587	0.530	0.666	0.037	0.402	0.781
20	S_{GAS}^{0}	0.653	0.850	0.402	0.160	0.118	0.811	0.824	0.746	0.806	0.534	0.576	0.526
21	S_{VAS}^{0}	0.363	0.575	0.199	0.369	0.408	0.290	0.831	0.705	0.402	0.174	0.238	0.100
22	S_{HAM}^{0}	0.251	0.013	0.129	0.447	0.611	0.311	0.174	0.984	0.398	0.281	0.212	0.089
23	S_{GLU}^{0}	0.076	0.180	0.886	0.458	0.056	0.237	0.016	0.244	0.045	0.141	0.374	0.054
24	S_{HF}^{0}	0.999	0.779	0.953	0.103	0.235	0.888	0.488	0.971	0.740	0.7403	0.257	0.662
25	k_{lean}	0.939	0.545	0.303	0.383	0.630	0.968	0.721	0.359	0.641	0.738	0.144	0.942

Table 7.1: 12 sets of parameters generated randomly

Set of feedback parameters obtained with the home made optimization algorithm

Experiment 2: running the home made optimization algorithm from a single initial parameter set

Nb epochs = 1000	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10
Distance walked [m]	27.1	27.2	27.0	26.2	27.3	27.6	27.6	30.9	28.8	27.4
G_{SOL}	0.321	0.447	0.320	0.361	0.351	0.401	0.488	0.558	0.508	0.527
G_{TA}	0.576	0.509	0.524	0.267	0.446	0.420	0.743	0.491	0.543	0.446
G_{SOL-TA}	0.796	0.603	0.640	0.679	0.744	0.859	0.786	0.647	0.745	0.659
G_{GAS}	0.649	0.807	0.670	0.900	0.831	0.671	0.668	0.436	0.598	0.711
G_{VAS}	0.725	0.733	0.740	0.767	0.833	0.834	0.957	1.081	1.030	0.816
G_{HAM}	0.575	0.607	0.549	0.546	0.628	0.620	0.585	0.467	0.624	0.602
G_{GLU}	-0.125	-0.001	-0.250	0.119	0.072	-0.002	-0.029	-0.200	-0.128	0.079
G_{HF}	0.379	0.500	0.436	0.611	0.296	0.610	0.572	0.563	0.504	0.467
G_{HAM-HF}	-0.638	-0.632	-0.672	-0.675	-0.646	-0.607	-0.650	-0.485	-0.538	-0.597
l_{TA}^{off}	0.899	0.805	0.807	0.743	0.954	0.963	0.919	0.933	0.861	0.837
l_{HF}^{off}	1.001	1.165	1.129	0.953	1.043	1.061	1.043	1.317	0.994	1.025
l_{HAM-HF}^{off}	0.391	0.258	0.327	0.427	0.182	0.162	0.083	0.015	0.047	0.166
k_{φ}^{knee}	0.795	0.844	0.957	0.801	0.896	0.930	0.650	0.758	0.659	0.892
k_{bw}	0.301	0.467	0.419	0.565	0.444	0.284	0.350	0.240	0.459	0.516
k_p	0.701	0.651	0.667	0.606	0.770	0.806	0.906	0.404	0.692	0.782
k_d	0.670	0.805	0.900	0.729	0.872	0.838	0.838	0.976	0.783	0.797
$\triangle S$	0.344	0.513	0.490	0.585	0.322	0.287	0.624	0.289	0.356	0.471
S_{SOL}^{0}	0.594	0.505	0.553	0.527	0.471	0.248	0.449	0.364	0.391	0.276
S_{TA}^{0}	0.702	0.910	1.008	0.985	0.853	0.923	0.823	0.866	0.902	0.873
S_{GAS}^{0}	0.369	0.394	0.694	0.684	0.421	0.741	0.519	0.775	0.636	0.660
S_{VAS}^{0}	0.048	0.049	0.075	0.089	0.022	0.030	0.062	0.007	0.022	0.064
S_{HAM}^{0}	0.018	-0.001	0.036	0.038	-0.031	-0.028	0.016	0.004	-0.032	0.003
S_{GLU}^{0}	-0.023	-0.103	-0.110	-0.092	-0.110	-0.091	-0.094	-0.104	-0.085	-0.077
S_{HF}^{0}	0.548	0.579	0.555	0.650	0.565	0.563	0.519	0.639	0.518	0.524
k_{lean}	0.899	0.646	0.712	0.857	0.946	0.945	0.588	0.176	0.576	0.736

Table 7.2: Fitness and best parameters for 10 trials of 1000 epochs all starting from the same initial set of parameters. We obtain of total walked distance of 27.7[m] in average.

Experiment 3: running the home made optimization algorithm from different initial parameter sets

Nb epochs = 1000	Set 1	Set2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10	Set 11	Set 12
Distance walked [m]	26.3	29.6	30.0	31.7	33.1	36.9	28.0	34.6	34.1	30.4	30.1	34.7
G_{SOL}	0.344	0.404	0.667	0.379	0.757	0.763	0.393	0.707	0.851	0.725	0.172	0.559
G_{TA}	0.432	0.416	0.134	0.817	1.207	0.349	0.809	0.864	0.624	0.112	0.372	0.411
G_{SOL-TA}	0.726	0.789	0.571	0.389	0.550	0.641	1.139	1.050	0.585	0.537	0.792	0.563
G_{GAS}	0.779	0.442	0.463	0.093	0.666	0.435	-0.200	0.299	0.268	0.504	-0.027	0.259
G_{VAS}	0.682	0.520	1.513	1.138	1.169	1.915	0.689	1.553	1.288	1.074	0.839	1.221
G_{HAM}	0.538	0.490	0.416	0.405	-0.027	0.104	0.517	0.120	0.302	0.437	0.609	0.031
G_{GLU}	0.097	0.656	-0.060	0.333	0.230	0.470	0.673	0.286	0.622	0.374	0.539	0.310
G_{HF}	0.461	0.380	0.354	0.526	0.888	0.923	0.091	0.072	0.737	-0.030	0.682	0.688
G_{HAM-HF}	-0.669	0.516	-0.558	-0.658	0.483	0.479	-0.184	0.849	0.509	0.677	-0.670	0.865
l_{TA}^{off}	0.916	0.161	0.590	0.380	0.965	0.440	0.035	0.693	0.794	0.180	0.283	0.604
l_{HF}^{off}	1.092	0.305	0.278	0.896	0.939	0.893	0.581	0.516	0.505	0.370	1.029	0.389
l_{HAM-HF}^{off}	0.607	-0.207	0.233	0.403	-0.379	-0.755	0.095	-0.625	-0.565	0.115	0.934	-0.521
k_{φ}^{knee}	0.727	0.810	0.636	0.058	0.840	0.311	0.966	0.690	0.480	1.070	0.679	0.176
k_{bw}	0.262	0.899	0.246	0.976	0.723	0.754	0.627	0.886	0.506	0.269	0.294	1.066
k_p	0.812	-0.123	0.108	0.806	0.145	0.087	0.101	0.726	0.709	-0.331	0.309	0.346
k_d	0.798	0.047	0.611	0.888	0.939	0.874	1.124	0.764	1.286	0.197	-0.061	0.937
$\triangle S$	0.463	0.626	0.426	0.668	0.605	0.653	0.532	0.640	0.453	-0.010	0.311	0.309
S_{SOL}^0	0.433	-0.123	0.557	-0.029	0.551	0.016	-0.091	0.681	0.025	0.093	0.012	-0.113
S_{TA}^0	1.078	0.408	0.783	0.616	0.569	0.804	0.822	0.444	0.705	0.298	0.436	0.989
S_{GAS}^0	0.781	0.325	0.327	-0.391	-0.268	0.469	0.553	0.200	0.762	0.316	0.613	0.154
S_{VAS}^0	0.073	0.374	0.036	0.080	-0.005	-0.008	0.240	0.014	0.011	0.010	0.213	0.042
S_{HAM}^0	0.043	-0.094	0.041	0.041	0.165	0.078	0.011	0.108	0.015	-0.111	-0.005	0.153
S_{GLU}^0	-0.104	0.092	0.360	-0.095	-0.097	0.110	0.040	0.111	-0.011	0.102	-0.111	0.057
S_{HF}^0	0.556	0.781	0.256	-0.008	0.127	0.325	0.575	0.909	0.462	0.423	0.082	0.427
k_{lean}	0.748	0.691	0.094	0.451	0.462	1.012	0.720	0.088	0.639	0.539	0.267	0.946

Table 7.3: Fitness and best parameters for 10 different initial parameter sets of 1000 epochs each. We obtain of total walked distance of 31.6[m] in average.

Set of feedback parameters obtained with the PSO algorithm

Experiment 4: Without robustness

Nb epochs = 1000	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
Distance walked [m]	35.2	31.2	39.0	31.7	31.1	34.8	33.2	37.3	36.9	37.3
G_{SOL}	0.700	0.791	0.560	0.400	0.888	0.658	0.371	0.844	0.445	0.428
G_{TA}	0.457	0.905	0.385	0.626	0.566	0.794	1.382	1.043	0.406	0.491
G_{SOL-TA}	0.725	0.587	1.898	0.721	0.784	0.713	1.846	0.836	0.709	0.997
G_{GAS}	0.472	0.465	0.540	0.280	0.460	0.222	0.037	0.417	0.240	0.415
G_{VAS}	1.836	0.890	1.950	1.029	0.763	1.182	1.125	1.576	1.806	2.054
G_{HAM}	0.183	0.284	-0.422	-0.098	0.439	0.246	0.271	-0.022	0.110	-0.208
G_{GLU}	0.343	0.659	-0.001	0.155	-0.156	0.449	-0.059	0.078	0.216	0.134
G_{HF}	1.317	0.713	1.010	0.375	0.146	0.607	0.776	-0.141	-0.013	0.489
G_{HAM-HF}	0.444	0.046	-0.134	0.012	0.079	0.688	-0.845	-0.009	0.434	-0.172
l_{TA}^{off}	0.431	0.755	0.178	0.300	0.593	0.605	0.567	0.870	0.411	0.229
l_{HF}^{off}	0.923	1.143	1.270	0.754	0.399	1.937	1.159	0.514	1.118	0.243
l_{HAM-HF}^{off}	-0.462	0.056	-0.371	-0.143	-0.046	-0.442	0.511	-0.556	-0.563	-0.602
k_{φ}^{knee}	0.244	0.182	0.324	0.413	0.545	0.905	0.788	0.639	1.204	0.211
k_{bw}	0.698	0.383	1.057	0.518	1.109	0.624	1.753	0.627	0.624	0.425
k_p	-0.024	0.204	0.232	-0.023	0.408	0.201	0.215	0.288	-0.047	0.100
k_d	1.075	0.848	1.677	0.509	0.445	0.953	1.077	1.352	1.126	1.323
$\triangle S$	0.743	0.676	0.563	0.890	0.675	0.406	0.627	0.571	0.574	0.566
S_{SOL}^{0}	0.855	0.185	-0.066	0.328	-0.257	0.642	-0.422	-0.108	-0.023	0.254
S_{TA}^{0}	0.327	0.007	-0.415	0.365	0.735	-0.394	0.539	0.425	0.105	-0.299
S_{GAS}^{0}	0.372	0.379	-0.459	0.162	0.704	0.444	-0.264	0.295	0.627	0.081
S_{VAS}^{0}	0.070	0.039	-0.001	0.322	0.217	0.017	0.143	0.010	-0.006	-0.001
S_{HAM}^{0}	0.159	-0.026	0.287	0.266	0.009	0.033	0.141	0.102	0.063	0.129
S_{GLU}^{0}	0.076	0.226	0.242	0.385	0.225	-0.010	-0.079	0.228	0.250	0.219
S_{HF}^{0}	0.130	0.739	-0.088	0.366	0.550	0.537	0.182	0.228	0.304	0.479
k_{lean}	0.071	0.163	0.021	0.328	-0.272	0.589	0.351	0.041	0.621	-0.223

Table 7.4: Fitness and best parameters for 10 runs of PSO of 1000 epochs each. We obtain of total walked distance of 34.8[m] in average.

Experiment 5: With robustness

Nb epochs = 1000	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
Distance walked [m]	31.3	29.4	26.0	27.0	28.2	27.4	26.7	26.6	30.5	31.0
G_{SOL}	0.610	0.569	0.769	0.666	0.567	0.466	0.619	0.792	0.607	0.760
G_{TA}	0.163	0.416	1.061	1.060	1.000	0.511	0.730	0.525	0.419	0.921
G_{SOL-TA}	1.154	0.612	0.784	1.729	0.983	0.699	1.013	0.587	0.681	1.099
G_{GAS}	0.478	0.339	-0.022	-0.166	0.453	0.446	-0.147	0.208	0.489	0.403
G_{VAS}	1.325	0.928	0.553	0.558	1.296	0.994	0.886	0.790	0.881	1.281
G_{HAM}	0.263	0.360	0.014	0.314	0.351	0.264	0.200	0.209	0.305	0.255
G_{GLU}	0.094	-0.297	-0.155	-0.204	0.109	0.431	-0.273	0.294	0.105	0.158
G_{HF}	0.633	0.328	0.983	0.241	0.857	0.394	0.741	0.995	0.592	0.126
G_{HAM-HF}	-0.060	-0.331	-0.109	0.346	0.076	-0.353	0.250	0.177	0.235	0.341
l_{TA}^{off}	0.328	0.524	0.005	0.166	0.200	0.457	0.006	0.328	0.505	0.491
l_{HF}^{off}	0.393	0.131	0.612	0.800	1.032	1.312	0.501	-0.055	0.768	0.270
l_{HAM-HF}^{off}	-0.073	0.126	0.733	0.336	0.087	0.470	0.352	0.094	-0.206	-0.302
k_{φ}^{knee}	0.941	0.628	0.373	0.286	0.162	0.681	0.421	0.605	1.190	1.413
k_{bw}	0.880	0.573	0.292	0.814	0.291	0.273	0.351	0.722	0.684	1.046
k_p	0.868	1.377	-0.169	-0.383	0.588	0.436	0.046	-0.526	1.003	0.408
k_d	0.365	1.198	1.304	1.418	0.678	0.956	1.094	0.784	1.563	0.646
$\triangle S$	0.591	0.821	0.974	0.848	0.479	0.673	0.839	0.756	0.781	0.861
S_{SOL}^{0}	0.156	0.409	-0.358	-0.261	-0.096	0.454	-0.067	0.214	0.609	0.322
S_{TA}^{0}	0.137	0.397	0.266	0.097	0.518	0.864	0.631	0.608	1.068	-0.443
S_{GAS}^{0}	-0.377	0.947	0.084	0.320	-0.370	0.500	0.203	0.781	0.441	0.722
S_{VAS}^{0}	0.122	0.225	0.424	0.244	0.283	0.219	0.244	0.288	0.101	0.285
S_{HAM}^{0}	0.109	0.067	0.471	0.101	0.033	-0.036	0.111	0.234	0.054	0.118
S_{GLU}^{0}	-0.094	0.083	0.098	0.062	0.258	-0.088	-0.096	0.266	0.201	-0.081
S_{HF}^{0}	0.442	0.386	0.193	0.358	0.588	0.228	0.560	0.495	0.478	0.365
k_{lean}	0.284	0.390	-0.037	0.668	0.764	0.518	0.510	0.358	-0.005	0.380

Table 7.5: Fitness and best parameters for 10 runs of PSO of 1000 epochs each. We obtain of total walked distance of 28.4[m] in average.

Experiment 6: With robustness and noise in muscle

Nb epochs = 1000	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
Distance walked [m]	31.5	32.7	27.5	35.1	30.6	34.5	35.0	33.3	33.7	25.7
G_{SOL}	0.626	0.707	0.279	0.858	0.721	0.690	0.644	0.845	0.415	0.430
G_{TA}	0.444	0.369	1.290	0.418	0.686	0.385	1.067	0.412	0.546	1.118
G_{SOL-TA}	1.025	0.673	0.811	0.841	1.076	0.582	0.553	1.165	0.815	1.349
G_{GAS}	0.486	0.532	0.305	0.258	0.436	0.405	0.294	0.270	0.348	0.232
G_{VAS}	1.023	1.687	1.394	1.881	1.490	1.924	2.186	1.454	2.524	1.573
G_{HAM}	0.055	0.116	0.379	0.365	0.432	0.244	0.483	0.305	0.400	0.335
G_{GLU}	0.126	-0.037	-0.069	0.203	0.364	0.005	0.104	-0.060	0.250	0.347
G_{HF}	0.424	0.419	0.256	0.034	0.376	0.994	0.362	0.663	0.588	1.411
G_{HAM-HF}	0.279	0.756	-0.220	0.655	-0.238	0.517	0.623	0.328	0.620	-0.269
l_{TA}^{off}	0.343	0.500	0.271	0.449	0.625	0.503	0.930	0.346	0.664	0.322
l_{HF}^{off}	0.891	0.599	0.822	0.392	0.572	1.506	0.911	0.926	1.132	0.517
l_{HAM-HF}^{off}	-0.170	-0.448	0.394	-0.476	-0.122	-0.361	-0.544	-0.443	-0.262	0.212
k_{φ}^{knee}	0.611	-0.094	-0.117	0.373	0.589	-0.128	0.059	0.811	0.358	-0.116
k_{bw}	0.559	0.907	0.719	1.112	0.136	0.442	0.940	0.343	0.833	0.341
k_p	-0.015	-0.125	-0.250	1.254	0.201	0.774	2.258	0.774	-0.273	0.575
k_d	0.892	0.883	0.809	0.802	0.859	0.837	0.833	0.929	0.802	1.288
$\triangle S$	0.221	0.675	0.499	0.607	0.316	0.419	0.713	0.481	0.642	0.670
S_{SOL}^0	0.404	0.176	-0.264	-0.097	0.198	0.031	-0.075	0.173	0.212	0.045
S_{TA}^0	0.514	0.848	0.387	0.352	0.704	-0.332	1.092	0.798	0.714	0.935
S_{GAS}^0	-0.198	-0.191	-0.281	0.698	0.548	0.175	0.140	0.504	0.218	0.280
S_{VAS}^0	0.033	0.056	0.443	0.026	0.086	-0.004	0.006	0.024	-0.027	0.307
S_{HAM}^0	0.144	0.141	0.083	0.018	-0.006	0.027	0.004	0.055	-0.023	0.178
S_{GLU}^0	0.113	0.261	0.047	0.307	0.279	0.356	0.253	0.304	0.332	0.055
S_{HF}^0	0.389	0.459	-0.038	0.308	0.595	0.532	0.230	0.603	-0.063	0.212
k_{lean}	-0.241	0.873	0.809	0.060	0.438	0.402	0.329	0.118	0.721	0.873

Table 7.6: Fitness and best parameters for 10 runs of PSO of 1000 epochs each. We obtain of total walked distance of 32.0[m] in average.

Set of SCR parameters

Parameter Set	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10
τ_1	0.085	0.069	0.055	0.075	0.060	0.056	0.053	0.039	0.034	0.037
τ_2	0.121	0.128	0.105	0.111	0.119	0.137	0.136	0.112	0.124	0.118
τ_3	0.277	0.294	0.265	0.267	0.312	0.311	0.314	0.280	0.270	0.274
x_1	1.092	1.179	0.833	1.093	1.154	1.018	1.262	1.408	1.229	1.311
x_2	2.420	2.242	3.539	2.439	2.074	2.503	2.194	2.228	2.298	2.175
x_3	1.513	1.528	1.406	1.501	1.296	1.355	1.247	1.661	1.661	1.697
x_4	2.061	1.506	2.028	2.055	1.619	1.624	1.441	1.726	1.812	1.719
x_5	1.632	1.601	2.265	1.684	1.608	1.454	1.286	1.200	1.278	1.275
x_6	0.770	0.745	0.982	0.867	0.741	0.929	0.923	0.899	1.004	0.981
x_7	0.389	0.407	0.373	0.467	0.425	0.490	0.421	0.360	0.298	0.315
x_8	1.414	1.710	1.676	1.594	1.635	1.521	1.261	1.258	1.412	1.390
x_9	0.074	0.077	0.077	0.079	0.081	0.074	0.071	0.072	0.067	0.068
x_{10}	1.485	1.533	1.073	1.311	1.274	1.266	1.194	1.019	1.134	1.084
x_{11}	1.696	1.632	1.850	1.806	2.013	1.996	1.891	1.854	2.266	2.176
x_{12}	3.396	3.593	5.430	3.876	3.966	3.084	4.070	3.927	3.083	3.381
x_{13}	2.921	2.429	3.268	2.297	2.412	1.777	2.284	2.024	2.008	1.917

Table 7.7: Final SCR parameters

Bibliography

[1] Webots, http://www.cyberbotics.com, Commercial Mobile Robot Simulation Software.

[2] Kirschner M. W. & Gerhart J. C., *The Plausibility of Life: Resolving Darwin's Dilemma*, Yale Univ. Press, New Haven, Connecticut, 2005

[3] Perry Jacquelin, *Gait analysis: normal and pathological function*, Thorofare, NJ:SLACK Inc. 1992

[4] E.R.Kandel, S.A.Siegelbaum, *Principles of Neural Science,* 4th edition, chapter 11, pp. 187-206, 2000

[5] C.Ghez, J.Krakauer, *Principles of Neural Science,* 4th edition, chapter 33, pp. 653-673, 2000

[6] G.E.Loeb, C.Ghez, *Principles of Neural Science,* 4th edition, chapter 34, pp. 674-694, 2000

[7] K.Pearson, J.Gordon, *Principles of Neural Science,* 4th edition, chapter 36, pp. 714-736, 2000

[8] K.Pearson, J.Gordon, *Principles of Neural Science,* 4th edition, chapter 37, pp. 737-755, 2000

[9] D.Purves, G.J.Augustine, D.Fitzpatrick, W.C.Hall, A.S. LaMantia, J.O.McNamara, S.M.Williams, *Neuroscience*, 3rd edition, chapter 15, pp. 371-392

[10] B.Alberts, A.Johnson, J.Lewis, M.Raff, K.Roberts, P.Walter, *The molecular biology of the cell*, 5th edition, ed. Garland Science, chapter 11, pp. 651- 694, 2008

[11] B.Alberts, A.Johnson, J.Lewis, M.Raff, K.Roberts, P.Walter, *The molecular biology of the cell*, 5th edition, ed. Garland Science, chapter 16, pp. 965- 1052, 2008

[12] T.A. McMahon, *Muscles, reflexes and locomotion*, Princeton University Press, 1984

[13] http://training.seer.cancer.gov/anatomy/muscular/structure.html

[14] http://obiwan.sandi.net/˜pfowler/grch40-49.html

[15] K.E.Adolph, A.M.Avolio, *walking infants adapt locomotion to changing body dimensions*, Journal of experimental psychology, vol.26, No3, pp.1148-1166, 2000

[16] K.E:Adolph, *Learning in the development of infant locomotion*, Monographs of the society for research in child development, Serial No251, vol.62, No3, 1997

[17] K.E. Adolph, L.Martin, I.Weise, *Locomotor development*, Parenthood in America: an encyclopedia.

[18] Anderson, F. C., and Pandy, M. G., 1999, *A Dynamic Optimization Solution for Vertical Jumping in Three Dimensions*, Comput. Meth. Biomech. Biomed. Eng., 2, pp. 201–231.

[19] F.C. Anderson, M.G.Pandy, *Dynamic optimization of human walking*, Journal of biomechanical engineering, Vol 123, pp. 381-390, 2001.

[20] L.J.Bhargava, M.G.Pandy, F.C.Anderson, *A phenomenological model for estimating metabolic energy consumption in muscle contraction*, Journal of Biomechanics, 37, pp.81.88, 2003.

[21] B. C.ABBOTT, X. M. AUBERT, A. V. Hill, *The absorption of work by a muscle stretched during a single twitch or a short tetanus*. Proc. R. Soc. B 139, 86-104, 1951.

[22] Berger W, Dietz V, Quintern J, *Corrective reactions to stumbling in man: neuronal co-ordination of bilateral leg muscle activity during gait*, Jounal of physiology, 357, pp.109-125, 1984.

[23] Ole Kiehn, *Locomotor circuits in the mammalian spinal cord*, Annual review of neuroscience, vol.29, pp279-306, 2006

[24] Dietz V, Quintern J, Boos G, Berger W, *Obstruction of the swing phase during gait: phase-dependent bilateral leg muscle coordination*, Brain research, 384, pp.166-169, 1986.

[25] Forssberg H, Grillner S & Rossignol S, *Phase dependent reflex reversal during walking in chronic spinal cats*, Brain research, 85, pp.103-107, 1975.

[26] Christopher M.Harris, Daniel M.Wolpert, *Signal-dependent noise determines motor planning*, Nature, vol394, 1998, pp.780-784.

[27] A.V. Hill, *The head of shortening and the dynamic constants of muscle*, Proc. R. Soc. Lond. B 126, 136-195, 1938

[28] H.Geyer, A.Seyfarth, R.Blickhan, *Positive force feedback in bouncing gaits?*, The Royal Society, 270, pp.2173-2183, 2003

[29] H.Geyer, A.Seyfarth, R.Blickhan, *Compliant leg behavior explains basic dynamics of walking and running*, The royal Society, 273, pp.2861-2867, 2006

[30] M.Eilenberg, H.Geyer, H.Herr, *Control of a powered ankle-foot prosthesis based on a neuromuscular model*, IEEE transaction on neural systems and rehabilitation engineering, vol.18, N°2, 2010

[31] H.Geyer, H.Herr, *A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities*, IEEE transaction on neural systems and rehabilitation engineering, vol.18, N°3, pp263-273, 2010

[32] R.Blickhan, A.Seyfarth, H.Geyer, S.Grimmer, H.Wagner, M.Günther, *Intelligence by mechanics*, Philosophical transactions of the royal society, 365, pp199-220, 2006

[33] A.Seyfarth, H.Geyer, H.M.Herr, *Swing-leg retraction: A simple control model for stable running*, J.Exp.Biol., vol.206, pp.2547-2555,2003

[34] G.Venture, K.Yamane, Y.Nakamura, *Identifying musculo-tendon parameters of human body based on the musculo-skeletal dynamics computation and Hill-Stoeve muscle model*, Proceedings of 2005 5th IEEE-RAS International conference on humanoid robots, 2005

[35] A.J. Ijspeert, *A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander*, Biological Cybernetics, Vol.84, pp331-348, 2001

[36] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, Jean-Marie Cabelguen, *From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model*, Science, Vol. 315, pp 1416-1419, 2007

[37] L. Righetti and A. J. Ijspeert. *Programmable Central Pattern Generators: an application to biped locomotion control*. Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006.

[38] Jonas Buchli, Auke Jan Ijspeert, *Self-organized adaptive legged locomotion in a compliant quadruped robot*, Autonomous Robot, vol.25, pp331-347, 2008

[39] Eng Janice J, Winter David A, Patla Aftab E., *Strategies for recovery from a trip in early and late swing during human walking*, Exp Brain research, 102, pp.339-349, 1994.

[40] Jesse van den Kieboom, *Biped locomotion ans stabilization - A practical approach*, Master project at Biorob, 2009

[41] B. R. Jewell, , D. R. Wilkie, *The mechanical properties of relaxing muscle*. J. Physiol. (Lond.). 152: 30-47, 1960

[42] B.Katz, *The relation between force and speed in muscular contraction*, J. Physiol. 96, 45-64, 1939.

[43] Lam Tania, Wolstenholm Claire, van der Linden Marleen, Pang Marco Y.C., Yang Jaynie F., *Stumbling corrective responses during treadmill-elicited stepping in human infants*, Journal of Physiology, 553.1, pp.319-331, 2003.

[44] M. S. Fischer, H. Witte, *Legs evolved only at the end!*, Phil. Trans. R. Soc. A 365, pp. 185–198, 2007

[45] J.A.Nelder, R.Mead, *A simplex method for function minimization*, computer journal, 7, pp.308-313, 1965

[46] Marcus G.Pandy, Thomas P.Andriacchi, *Muscle and joint function in human locomotion, Annual review of biomedical engineering*, 12, 2010, pp.401-433.

[47] K.Pearson and D.Collins, *Reversal of te influence of group Ib afferents from plantaris on activity in medial gastrocnemius muscle during locomotor activity*, J.Neurophys., vol 70, pp.1009-1017, 1993

[48] J. Kennedy, R. Eberhart, *Particle swarm optimization*, Proc. IEEE.Int. Conf. on Neural Networks, pp. 1942-1948, 1995.

[49] Maurice Clerc, James Kennedy, *The particle swarm - explosion, stability and convergence in a multidimensional complex space*, IEEE transactions on evolutionary computation, vol.6, No1, 2002, pp. 58-73

[50] Quevedo Jorge, Stecina Katinka, McCrea David A, *Intracellular analysis of refled pathways underlying the stumbling corrective reaction during fictive locomotion in the cat*, Journal of Neurophysiologie, 94, pp.2053-2062, 2005.

[51] M.A.Daley, G.Felix, A.A.Biewener, *Running stability is enhanced by a proximo-distal gradient in joint neuromechanical control*, Exp.Biol. , 210, pp 383-394, 2007

[52] J.C.Spall, *Implementation of the simultaneous perturbation algorithm for stochastic optimization*, IEEE transactions on aerospace and electronic systems, vol.34, No 3, pp.817-823, July 1998

[53] F.Saibene, A.E.Minetti, *Biomechanical and physiological aspects of legged locomotion in humans,* Eur J Appl Physiol, 88, pp. 297-316, 2003

[54] D.F.B.Haeufle, S.Grimmer, A.Seyfarth, *The role of intrinsic muscle properties for stable hopping - stability is achieved by the force velocity relation*, Bioinspiration and Biomimetics, vol.5, 2010

[55] S.R.Ward, L.H.Smallwood, R.L.Lieber, *Are current measurements of lower extremity muscle architecture accurate?*, Clin Orthop Relat Res, 467, pp. 1074-1082, 2009

[56] http://biorob.epfl.ch/page-65335-en.html

[57] M. A. Daley, G. Felix, and A. A. Biewener, *Running stability is enhanced by a proximo-distal gradient in joint neuromechanical control,* J. Exp. Biol., 210, pp.383–394, 2007.

[58] Kelvin E. Jones, Antonia D. de C. Hamilton, Daniel M. Wolpert, *Sources of signal-dependent noise during isometric force production*, Journal of neurophysiology, 88, pp 1533-1544, 2002.