Université catholique de Louvain

École polytechnique de Louvain (EPL)

Bio-inspired walking controllers for a variety of robots having different scales

Dissertation presented by
Adrien De Coninck

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

Supervisor
Renaud Ronsse

Readers
Nicolas van der Noot, Philippe Chatelain

Academic year 2015-2016

Acknowledgements

I wish to express my gratitude to **Prof. Renaud Ronsse**, supervisor of this master's thesis, for his expert advice throughout this semester.

I also would like to address a special thank to **Nicolas Van der Noot** for his continuous support throughout the project, for his precious advice, for explaining me the details of his walking controller, for sharing his experience with C++, for answering my dozens numerous by email and for his availability.

I also thank **Bruno Somers**, for explaining me how to use the optimization framework and for the help he brought me to include the upper-body controller.

Eventually, I thank my parents, my girlfriend, and the rest of my family for their moral support.

Adrien De Coninck
Louvain-la-Neuve, June 2016

List of abbreviations

General

DOF	degree of freedom
GCM	ground contact model
SEA	series elastic actuator
IMU	inertial measurement unit
MBS	multibody system
URDF	unified robot description format
ZMP	zero-moment point
BMI	body mass index
CMA-ES	covariance matrix adaptation evolution strategy algorithm

Hill-type muscle model

MTU	muscle-tendon unit
CE	contractile element of the MTU
SE	series elastic element of the MTU
PE	parallel elastic element of the MTU
BE	buffer elastic element of the MTU
SOL	soleus muscle group
TA	tibialis anterior muscle group
GAS	gastrocnemius muscle group
VAS	vasti muscle group
HAM	biarticular hamstring muscle group
GLU	gluteus muscle group
HFL	hip flexor muscle group
m_{MTU}	mass of the MTU
l_{MTU}	length of the MT
l_{CE}	length of the CE
$\tilde{l}_{CE}$	normalized muscle fiber length of the CE
v_{CE}	contraction velocity of the CE
F_m	forces developed by the muscle
F_{CE}	forces developed in the CE
S_m	neural stimulation level
A_m	muscle activation level

Metabolic energy model

$\dot{E}$	rate of metabolic energy expenditure
$\dot{H}$	rate at which heat is liberated
$\dot{W}$	rate at which work is done
$\dot{A}$	muscle activation heat rate
$\dot{M}$	muscle maintenance heat rate
$\dot{S}$	muscle shortening heat rate
$\dot{B}$	basal metabolic heat rate

Contents

Chapter 1

Introduction

1.1 Context

1.1.1 Why humanoid robots ?

Legged humanoid robots are gaining much interest nowadays. They are needed in a large variety of fields. But why would they be preferred to conventional wheeled robots which are more stable and much easier to design and control ?

Our environment is adapted to humans. A key advantage of biped humanoid robots is that they are able to work directly in the human's environment including obstacles, stairs and uneven grounds. The great interest that they receive is then partly motivated by the ability of such robots to replace humans in hazardous environments being specifically designed for humans, such as natural disaster scenarios [10].

The drawbacks of legged humanoid robots is that they are not easy to design, expensive to construct and, above all, difficult to control. The walking control of a variety of legged humanoid robots having different scales and morphologies is the topic of this master's thesis.

The existing walking controllers can be divided into two types : static and dynamic walking controllers.
When a humanoid robot is controlled by a static walking controller, the projection of its center of gravity on the ground is maintained inside the feet support area. Hence, it is in equilibrium at every moment of the gait.
However, if it is controlled by a dynamic walking controller, the robot is only stable in motion so that if its joints get stopped at a given moment, the robot will likely fall [19]. On the one hand, that kind of walking controllers offers better performances in terms of walking speed and energy-efficiency. On the other hand, it is more difficult to implement because of the inertia effects that must be taken into account.

1.1.2 Bio-inspired dynamic walking of humanoid robots

A broadly used concept to generate legged humanoid robots control algorithms is the Zero-Moment Point (ZMP) stability criterion [22]. The ZMP is the generalization of the center of mass, in dynamic walking. As written above, maintaining the center of mass inside of the feet support area is a sufficient condition to keep balance in static walking. Similarly, maintaining the ZMP inside of the feet support area is a sufficient condition to keep balance in dynamic walking. The ZMP stability criterion is very popular although it presents lots of drawbacks regarding energy-efficiency, maximum speed and resistance against external perturbations [19].

Considering the qualities of the human dynamic walking gait (i.e., energy-efficiency, balance, speed), some researchers have tried to follow a bio-inspired approach to design dynamic gait controllers.
Among them, Nicolas Van der Noot[1] developed a bio-inspired dynamic walking controller for Co-Man, a one meter tall robot designed by the Italian Institute of Technology (IIT). This controller is based on a muscle-reflex model developed in [8]. This dynamic walking controller was tested both in simulation and in reality and provided more human-like gaits, compared to traditional walking controllers.

The present work aims at generalizing this controller to a variety of robots having different scales and morphologies. Namely, we want to generalize the controller to NAO (58 cm tall and designed by Aldebaran Robotics), WalkMan (1.85 m tall and designed by the IIT) and Atlas (1.95 m tall and designed by Boston Dynamics). To reach this goal, we first develop a software model of each robot in the Robotran simulation environment. Indeed, working directly on the real robots would take too much time and would be dangerous (both for the robots and the human). Then we use these models to adapt, test and optimize the controller to the different robots. Test on the real robots will not be conducted in this thesis but could be a next step, once good results are achieved in simulations.

1.2 Executive summary

Now that the goals and the context of this work are introduced, this section gives an overview of the contents of the different chapters.

Chapter 2 - Investigated robots

The goal of this chapter is to introduce the different robots that are taken into consideration in this thesis. An overview of each robot is given and a comparison of their technical characteristics (dynamics, kinematics, actuation, sensors, etc.) is made.

Chapter 3 - Robot Models & Tools

In this chapter, the three developed robot models are introduced. These models somehow simulate the physical behavior of the robots. Robot models are needed because the gait controller has to be trained through an optimization process for each robot. For safety (both for the robots and the human) and time consumption (running simultaneous optimizations on a single real robot is impossible) reasons, this optimization phase can hardly be run on the real robot.
This chapter is completed with a presentation of the tools and the evolutionary algorithm used to perform this optimization process.

Chapter 4 - Bio-inspired Walking Controller

This chapter is devoted to the development of the controller itself. We describe how the original controller was developed and how we adapted it to the different robots.

Firstly, the human gait is briefly analyzed. Since we want the robots to adopt bio-inspired gaits, it is useful to specify the main characteristics of the human way of walking.

[1]Center for Research in Mechatronics, Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, Belgium

The 2D bio-inspired walking controller is then introduced. By 2D walking controller, we mean that "the floating base is constrained so that the robots cannot fall aside" [19].

To finish the chapter, we discuss how the optimizer is guided toward an optimal solution in the optimization process.

Chapter 5 - Simulation Results & Comparison

In this chapter, the simulation results are presented and analyzed. We start by analyzing several walking gaits which are the results of an optimization phase without speed constraint. We then compare one optimized controller of each robot with human data. We finally compare the results of several optimization phases with different target speeds and analyze the evolution of the robots gaits across a range of speeds.

Chapter 2

Investigated robots

The goal of this chapter is to introduce the different robots that are taken into consideration in this project. We will present an overview of each robot and compare their technical characteristics (dynamics, kinematics, actuation, sensors, etc.).

2.1 NAO

NAO (Figure 2.1) is a programmable humanoid robot developed by Aldebaran Robotics[1], a French-Japanese robotics company. The NAO robot is one of the most famous humanoid robots. It has been used for various research and education purposes since its release date in 2006.

Figure 2.1: Picture of the NAO taken from `http://robotica.unileon.es`

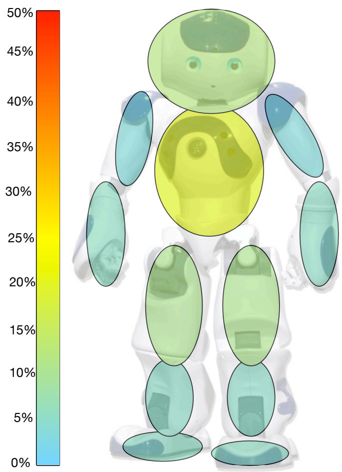
Dimensions and mass NAO is only 58 cm tall and 27 cm wide[2]. Its components are made of plastics and carbon fiber (ABS-PC/PA-66/XCF-30) and it weighs around 5 kg. It then has a BMI[3] (Body Mass Index) of about 15 kg/m^2 which means that it is very light compared to existing robots of the same height (see Table 2.1).

[1] `https://www.aldebaran.com/en`
[2] The width is measured along the Y axis
[3] As a reminder, the BMI formula is $BMI = \frac{Mass}{Height^2}$ and is expressed in kg/m^2

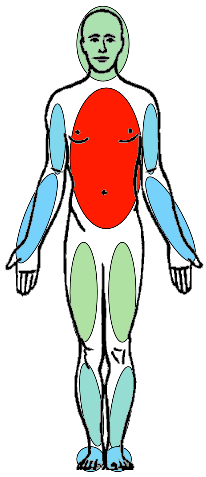
Table 2.1: Characteristics of functional humanoids (Adapted from [9])

	Height $[m]$	Weight $[kg]$	BMI $[kg/m^2]$
KHR-2HV (Kondo Kagaku)	0.34	1.3	10.9
HOAP (Fujitsu Automation)	0.5	7.0	28
NAO (Aldebaran)	**0.57**	**5**	**15**
QRIO (SONY)	0.58	6.5	19
COMAN (IIT)	**0.95**	**31**	**34.3**
ICUB (IIT)	1.04	22	20.3
ASIMO (HONDA)	1.30	54	32
ROMEO (Aldebaran)	1.40	40	20.4
HRP-2 (AIST)	1.54	58	24.5
WALKMAN (IIT)	**1.85**	**142**	**41.49**
ATLAS (Boston Dynamics)	**1.95**	**172**	**45.23**
Human	1.5-2	50-100	18-25

The mass is distributed as shown in Figure 2.2a. The typical mass distribution for human is presented in Figure 2.2b for a qualitative comparison. The mass distributions for NAO and for a human are quite similar except that the human torso represents a bigger proportion of the whole body weight (around 50%).



(a) NAO - Segments weight as a percentage of the whole body weight



(b) Human - Segments weight as a percentage of the whole body weight (based on the values found in [18])

Figure 2.2

The legs and feet of NAO are respectively 33 and 16 cm long. This gives, on the one hand, a leg/body length ratio of around 0.55 and, on the other hand, a foot/leg length ratio of around 0.5 which is really disproportionate compared to human (foot/leg length ratio of around 0.3, toes included). As shown in Figure 2.3, the feet are not centered on the longitudinal axis X (the reference is placed on the ground, at the projection of the main Z axis of the corresponding leg).

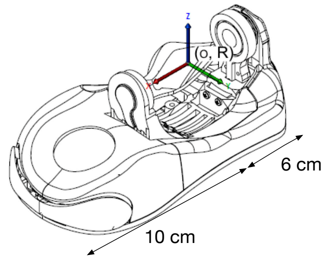


Figure 2.3: Right foot of NAO (adapted from `http://doc.aldebaran.com`)

Kinematic design NAO has 24 degrees of freedom (DOFs) and 23 independent DOFs (11 for the lower part and 12 for the upper part). As depicted in red in Figure 2.4, a special pelvis kinematic design makes NAO different from the other existing humanoid robots :

- The joints in red are not rotating about an axis aligned with one of the main world axes (XYZ). Instead, they are both rotating about a rotation axis inclined at 45° towards the body
- These joints are coupled. They are actuated by a single motor as shown in Figure 2.5a

This specificity is an economic alternative to the classical set of three independent revolute joints encountered in most humanoid robots (Figure 2.5b). This saves two motors at the hip level without reducing the overall mobility [9]. The mechanism is useful for bending over and sitting down because it allows NAO to bend the body forward while spreading the legs.

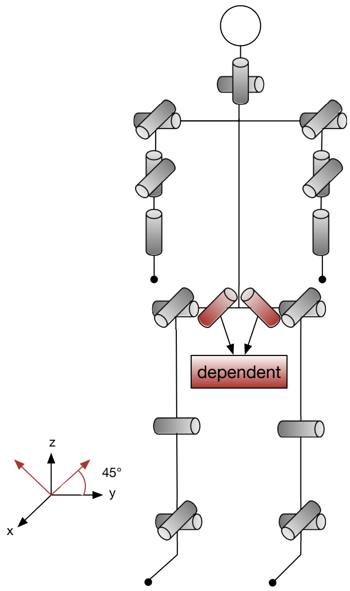
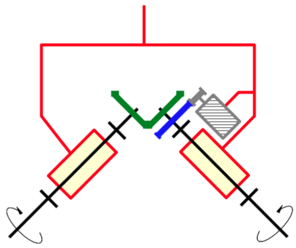
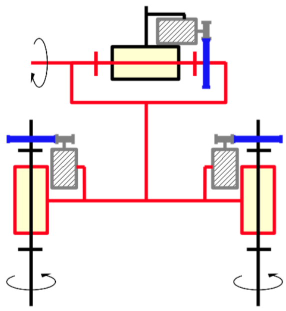


Figure 2.4: The 23 independent DOFs of the NAO. The 2 joints depicted in red are actuated by a single motor and thus constitute only one independent DOF



(a) Coupled inclined axis revolute joints for the NAO pelvis [9]



(b) Classical set of three revolute joints, one pitch joint at the waist and one yaw joint at each hip [9]

Figure 2.5

Actuators To actuate its stiff joints, the NAO robot is equipped with high quality brushed DC motors whose specifications are given in Table B.1 of Appendix B.

Sensors Each joint has a 12-bit precision position sensor using Hall-effect technology. Moreover, each motorboard has a current sensor for each motor made of a shunt resistor and that gives an absolute value of the current in Ampere. However, active stiffness control is impossible since the current measurement is not accurate enough to derive the torque and since no other torque sensor is integrated in the joints.
In order to measure the ground reaction forces, NAO has 4 force sensitive resistors (sensors measuring a resistance change as a function of the pressure applied) under each foot. These sensors have a working range between 0 and 25 N. On top of that, NAO is equipped with an Inertial Measurement Unit (IMU) located in its torso. The output data of the IMU is an estimation of the torso rotational speed along each DOF.

Others The robot battery is a Lithium ion battery that can power the robot during up to 1 hour of operation. The "brain" of the robot is a monocore Intel Atom Processor Z530 and is located in the head. This CPU runs a Linux kernel with a secondary CPU located in the chest. NAO is also equipped with microphones (for voice recognition), loudspeakers, Infra-Red, Sonars, contact and tactile sensors. Further information about the technical specifications of NAO can be found in [1].

2.2 CoMan

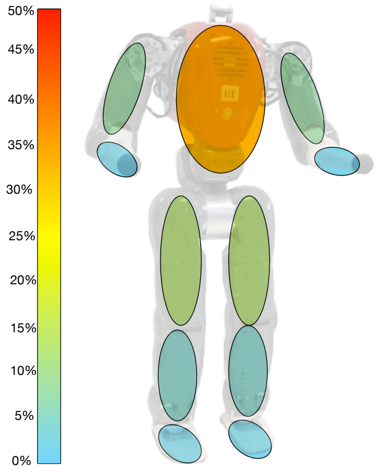
CoMan (Figure 2.6), standing for "COmpliant HuMANoid robot", is a robotic platform developed by the Italian Institute of Technology (IIT)[4].

Figure 2.6: Picture of the CoMan taken from `http://spectrum.ieee.org/automaton/robotics/humanoids/iit-coman-humanoid-robot`

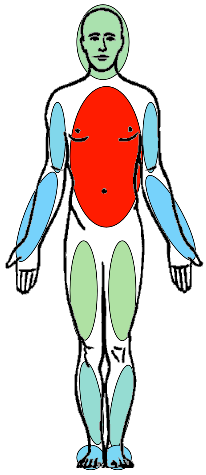
Dimensions and mass CoMan has a morphology similar to the one of a 5-year-old child. It is 95 cm tall and 40 cm wide. Its segments are made of titanium alloy, stainless steel and aluminium alloy and it weighs around 31 kg [19]. It then has a BMI of about 34 $kg/m2$ which means that it is quite heavy compared to existing robots of the same height (see Table 2.1). As done for NAO in section 2.1, the mass distributions for the CoMan robot and for a human are qualitatively compared in Figure 2.7. Since the CoMan has no head, the mass is quite differently distributed. The thighs and the upper arms represent a bigger percentage of the whole body weight (24% and 17% respectively), likely due to the fact that they embed several actuators. The torso represents 34% of the whole body weight. It is the heaviest part of the body even though it does not represent a weight as high as the human torso.

The legs and feet of CoMan are respectively 52 and 18.5 cm long. This gives, on the one hand, a leg/body length ratio of around 0.55 and, on the other hand, a foot/leg length ratio of around 0.35 which seems to be more human-like than the robot NAO. As shown in Figure 2.8b, the feet are not centered on the longitudinal axis X.

[4] `https://www.iit.it`



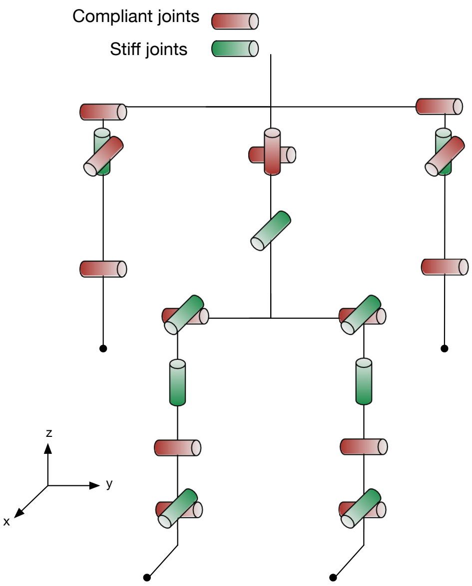
(a) CoMan - Segments weight as a percentage of the whole body weight



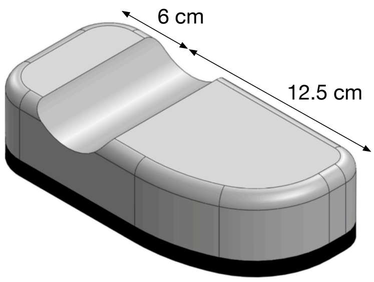
(b) Human - Segments weight as a percentage of the whole body weight (based on the values found in [18])

Figure 2.7

Kinematic design As shown in Figure 2.8a, CoMan has 23 independent DOFs, 12 for the lower part (legs) and 11 for the upper part (torso and arms). It has a classical kinematic design, the joints being all rotating about axes aligned with one of the main world axes (XYZ) and are all independently actuated.



(a) The 23 independent DOFs of the CoMan. CoMan has compliant (red) and stiff (green) joints



(b) Right foot of CoMan (adapted from [19]). The reference is placed on the ground, at the projection of the main Z axis of the corresponding leg

Figure 2.8

Actuators The CoMan is one of the first humanoid robots equipped with compliant and stiff joints (respectively depicted in red and green in Figure 2.8a). All the pitch joints controlled with the bio-inspired approach in Chapter 4, are compliant. The compliant joints are implemented using serie elastic actuators (SEA) as presented in Figure 2.9 [19]. A SEA is an actuator incor-

porating series elasticity (a spring, typically) as a purposeful element (see [15] for more details). The specifications of the motors of CoMan are given in Table B.2 of Appendix B.
In robotics, the interface between an actuator and the load is, traditionnally, as stiff as possible (increased stiffness improves the precision, stability and bandwidth of position control). However, more recent applications tend to deviate from this "stiffer is better" rule of thumb. Indeed, reducing the interface stiffness offers a number of advantages including a greater schock tolerance, a more accurate and stable force control, a safer interaction with the environment and the capacity to store and release energy [15]. These features are particularly desirable for applications requiring the interaction with humans, other robots, and/or uncontrolled environments.

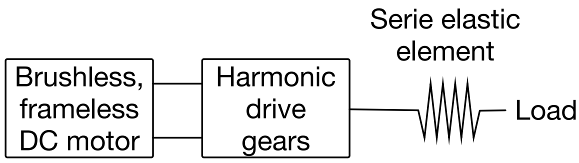


Figure 2.9: Block diagram of the serie elastic actuators used for CoMan

Regarding locomotion, the main advantages of using compliant joints (e.g., with SEA) are the following [19] :

- Reduced impact forces when the foot is hitting the ground, because the impact energy is absorbed by the elastic element (the spring).
- Possibility to store energy in the springs and to release it at different moments in the gait. This allows a more energy-efficient gait.
- A more human-like gait. Indeed, the elasticity of the human muscles and tendons is an important feature of the human gait. We talk about "elasticity" in both case but the human's tendons configuration is totally different as the one of the CoMan.
- Better robustness by passively adapting the foot to an uneven floor.

Sensors Each joint is equipped with a position sensor and a torque sensor, so that an active stiffness control is possible. To measure the ground reaction forces, 6-DOF strain gage sensors are included at the ankle level. The CoMan is also equipped with an IMU attached to its waist.

Others The battery, located in the torso, can power the robot up to two and a half hours of operation. The "brain" of the robot is a dual-core Pentium PC104 and is located in the torso. This CPU runs a Linux kernel.

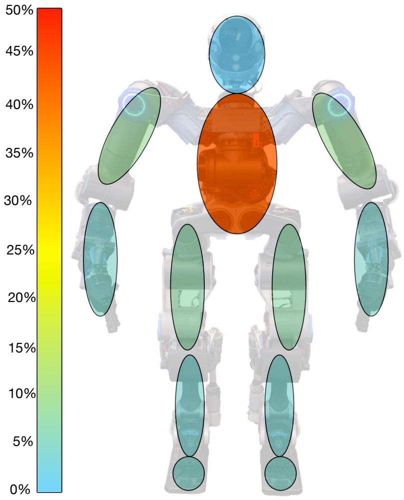
2.3 WalkMan

WalkMan, standing for "Whole-body Adaptive Locomotion and MANipulation" is a robotic platform developed by the Italian Institute of Technology (IIT) in the context of the DRC[5]. The project started in September 2013 and has the goal to develop a humanoid robot which can operate outside the laboratory space in unstructured environments and work spaces as a result of natural and man-made disasters [12].

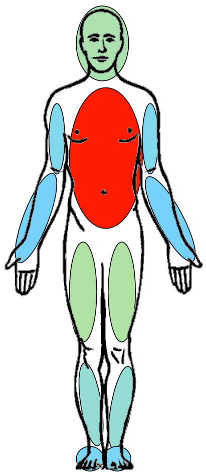
[5] The DARPA Robotics Challenge (DRC) is a competition funded by the US Defense Advanced Research Projects Agency. The last DRC competition was held from 2012 to 2015, it aimed at developing semi-autonomous ground robots that could do "complex tasks in dangerous, degraded, human-engineered environments". See `https://en.wikipedia.org/wiki/DARPA_Robotics_Challenge`

Figure 2.10: Picture of the WalkMan taken from `http://pop.h-cdn.co/assets/15/23/walkman_main.jpg`

Dimensions and mass The WalkMan is 185 cm tall, 80 cm wide and weighs around 142 kg. It then has a BMI of about 41.5 kg/m^2 which means that it is very heavy compared to a human of the same height (see Table 2.1). As done for NAO and CoMan in sections 2.1 and 2.2, the mass distributions for the WalkMan robot and for a human are qualitatively compared in Figure 2.11. It can be observed that the mass distribution for the WalkMan is pretty similar to the mass distribution for a human, except that the head is heavier and the upper arms are lighter in the latter.



(a) WalkMan - Segments weight as a percentage of the whole body weight



(b) Human - Segments weight as a percentage of the whole body weight (based on the values found in [18])

Figure 2.11

The legs and feet of WalkMan are respectively 110 and 32 cm long. This gives, on the one hand, a leg/body length ratio of around 0.6 and, on the other hand, a foot/leg length ratio of around 0.3 which is similar as for CoMan. Unlike humans' feet, the feet of the WalkMan are centered on the longitudinal axis (X) as shown in Figure 2.12. Again, the reference is placed on the ground, at the projection of the main Z axis of the corresponding leg.

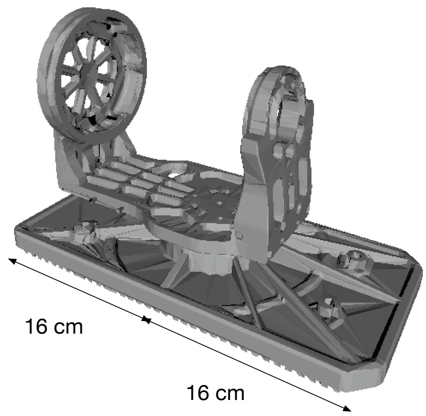


Figure 2.12: Right foot of WalkMan (3D file from `https://gitlab.robotran.be/walkman/walkman_robotran`)

Kinematic design As shown in Figure 2.13, WalkMan has 31 independent DOFs, 12 for the lower part (legs) and 19 for the upper part (torso, head and arms). As depicted in red and green in Figure 2.13, three joints are a bit special compared to the classical kinematic design for humanoid robots (e.g., compared to the CoMan). These joints do not rotate about "regular" axes (XYZ):

- The pitch shoulder joints (depicted in red in Figure 2.13) are both rotating about a rotation axis inclined at 41.7° away from the body.
- The roll torso joint (depicted in green in Figure 2.13) is rotating about a rotation axis inclined at 25.7° in the (z,x) plane.

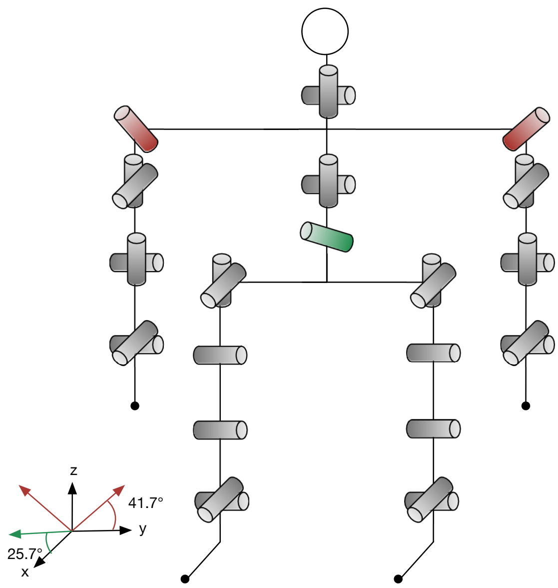


Figure 2.13: The 31 independent DOFs of the WalkMan

Actuators Like the CoMan, WalkMan is equipped with compliant joints implemented using serie elastic actuators (SEA), for the same reasons as in section 2.2. As shown in Figure 2.14, the actuator unit consists of a frameless brushless DC motor, an Harmonic Drive (HD) gearbox (with reduction ratios comprised between 50:1 and 150:1) and a flexible element (a torsion bar) connecting the output of the gearbox to the output flange of the actuator [14]. The specifications of the motors of WalkMan are given in Table B.3 of Appendix B.

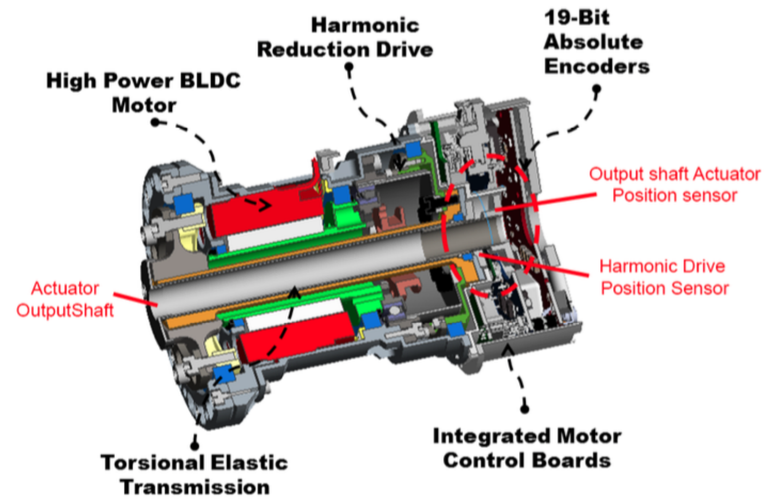


Figure 2.14: 3D section of the high power actuation unit for WalkMan [14]

Sensors Each SEA is equipped with embedded driver electronics and a variety of sensors including position, torque and temperature sensors permitting the implementation of multiple control strategies [14].

2.4 Atlas

Atlas is a robotic platform developed by Boston Dynamics[6] in the context of the DARPA Robotics Challenge.

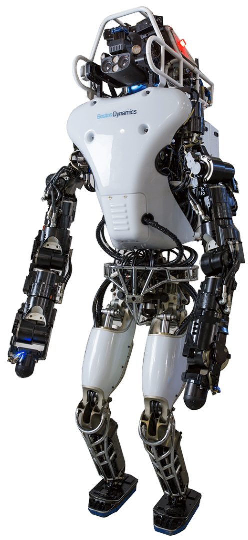
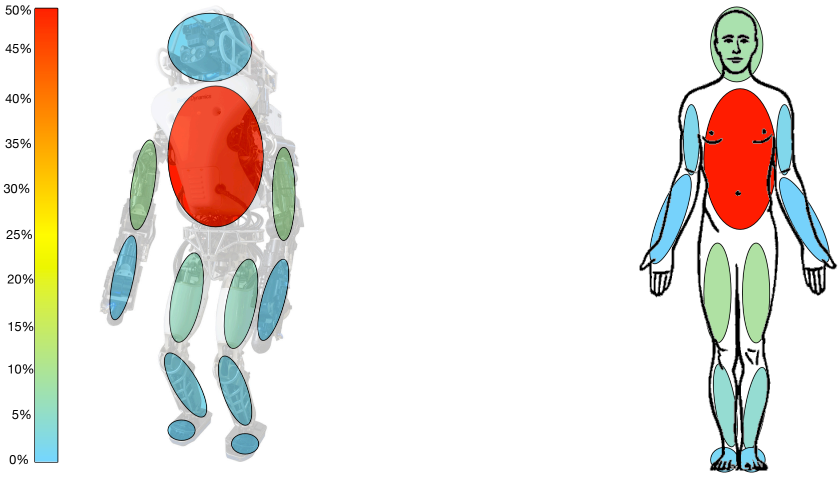


Figure 2.15: Picture of the Atlas taken from https://www.washingtonpost.com/graphics/business/robots/img/atlas.png

[6]http://www.bostondynamics.com/index.html

Dimensions and mass The Atlas is 195 cm tall and 60 cm wide[7] . Its components are made of aluminium and titanium and it weighs around 172 kg. It then has a BMI of about 45 kg/m^2 which means that it is very heavy compared to a human of the same height (see Table 2.1). However, compared to WalkMan, i.e. a robot of the same height, it has a comparable BMI. As done for NAO, CoMan and WalkMan in sections 2.1, 2.2 and 2.3, the mass distributions for the Atlas robot and for a human are qualitatively compared in Figure 2.16. As for WalkMan, it can be observed that the mass distribution for the Atlas is pretty similar to the mass distribution for a human, except that the head is heavier and the upper arms are lighter in the latter.



(a) Atlas - Segments weight as a percentage of the whole body weight

(b) Human - Segments weight as a percentage of the whole body weight (based on the values found in [18])

Figure 2.16

The legs and feet of Atlas are respectively 94 cm and 27 cm long. This gives, on the one hand, a leg/body length ratio of around 0.5 and, on the other hand, a foot/leg length ratio of around 0.3 which is in the same proportions as for CoMan and WalkMan. As shown in Figure 2.17, the feet are not centered on the longitudinal axis (X).

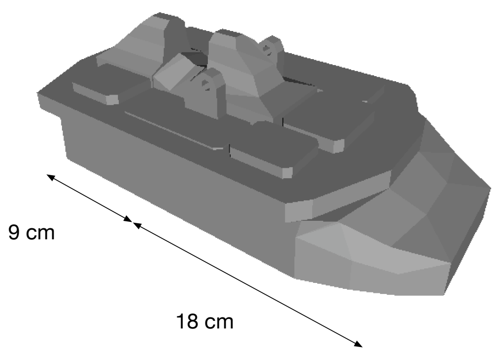


Figure 2.17: Right foot of Atlas (3D file from `https://bitbucket.org/osrf/drcsim`)

Kinematic design As shown in Figure 2.18, Atlas has 28 independent DOFs, 12 for the lower part (legs) and 16 for the upper part (torso and arms). It has a classical kinematic design, the joints are all rotating about "regular" axes and are all independent. The joints are all

[7] A smaller version (1.75 m and 82 kg) was introduced in 2016, after the beginning of this project

hydraulically-actuated except the neck and the last three joints on each arm being electrically-actuated.

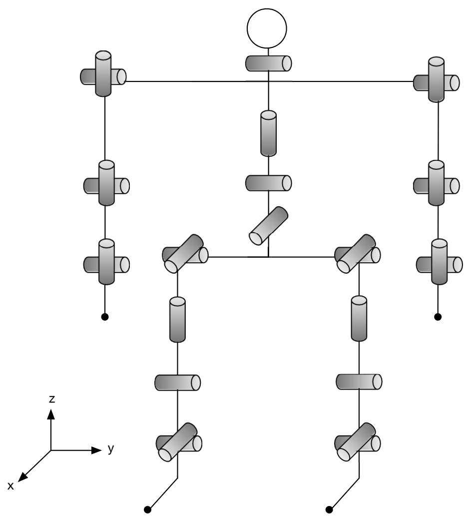


Figure 2.18: The 28 independent DOFs of the Atlas

Sensors Joint position and torque are directly measured on the actuator with Linear Variable Differential Transformers (LVDTs) and pressure sensors inside the pistons, respectively. These positions and torques are then transformed to joint space[8]. Velocity is computed by filtering the finite difference of positions [7].

2.5 Summary and comparison tables

Considering Table 2.2, we can conclude that, in terms of proportions, Atlas and WalkMan are the robots that are the most similar to the human adult standards. However, both robots are very heavy compared to a human of the same height (a BMI above 30 kg/m^2 is synonymous of obesity). CoMan has the same proportions as a 5-year-old child but is also a bit heavier. NAO is very small and light (its proportions are comparable to a 5-month-old baby) but its feet are huge and disproportionate.

Table 2.2: Comparison table : Dimensions

	Height [cm]	Width [cm]	Mass [kg]	BMI [kg/m^2]	Leg/Body ratio	Foot/Leg ratio
NAO	58	27	5	15	0.55	0.5
CoMan	95	40	31	34	0.55	0.35
WalkMan	185	80	142	41.5	0.6	0.3
Atlas	195	60	172	45	0.5	0.3
Human	150-200	35-60	50-100	18-25	≈0.5	≈0.3

Regarding the mass distributions, Figure 2.19 shows that the mass distributions for Atlas and WalkMan are the closest to the one for a human, even though the head is heavier and the upper arms are lighter in the latter. NAO has the less human-like mass distribution. Its torso and arms are very light compared to humans' ones. In general, the differences between robots and human models regarding the mass distribution can be explained by the necessity to add motors at specific locations, given mechanical constraints.

[8]The actuators are not always directly connected to the joint

Moreover, it can be stated that, for the four robots, the mass distributions for the legs and feet are very similar to the human ones.

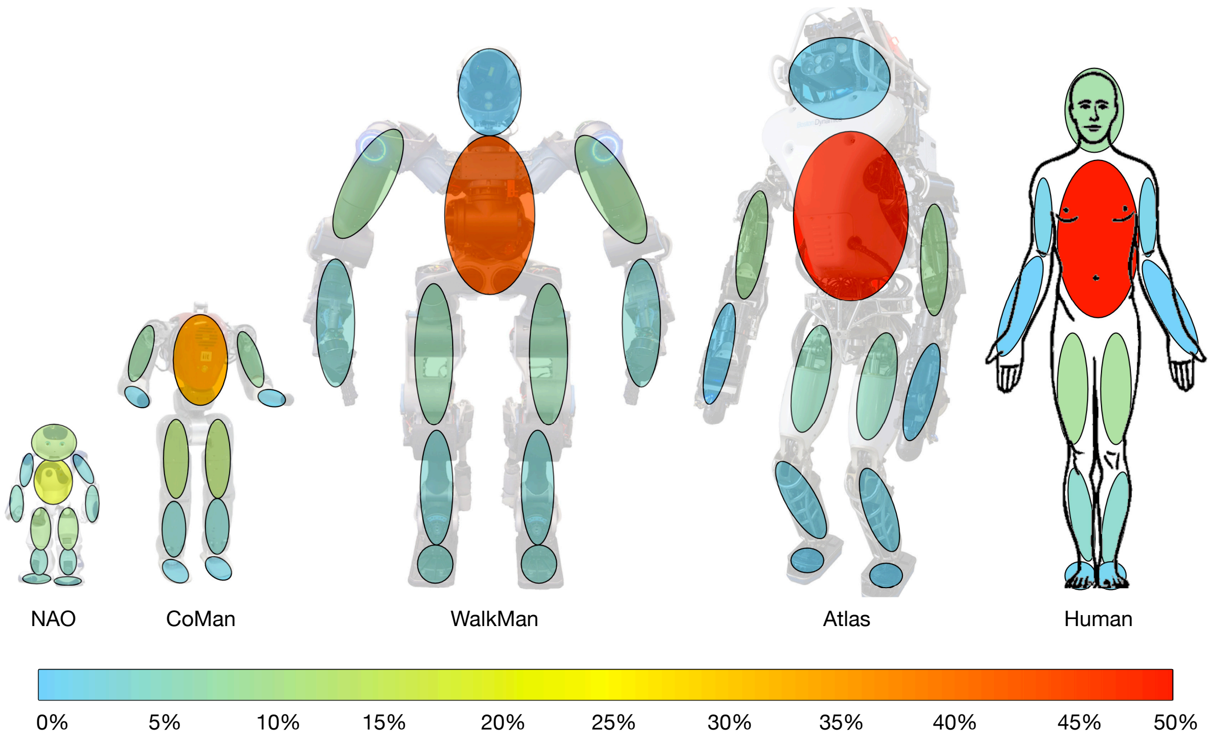


Figure 2.19: Mass distributions for NAO, CoMan, WalkMan, Atlas and a human - Segments weight as a percentage of the whole body weight.

It is stated in [26] that the human body has about 244 DOFs. As written in Table 2.3, the robots investigated in this project have, of course, far less DOFs. This is not a surprising observation because a human is way more compliant than a robot. Regarding this thesis, the main differences with humans in terms of number of DOFs are located inside the feet.

Table 2.3: Comparison table : Kinematics and Dynamics

	Total number of DOFs	Total number of DOFs for the legs	Type of Actuation
NAO	23	11	DC motors (stiff joints)
CoMan	23	12	Serie elastic actuators & DC motors
WalkMan	31	12	Serie elastic actuators
Atlas	28	12	Hydraulic & electric actuators

Chapter 3

Robot Models & Tools

This project aims at generalizing an existing bio-inspired gait controller (designed for the robot CoMan) to three other humanoid robots : NAO, WalkMan and Atlas.

We will first introduce the robot models that we developed. These models somehow simulate the physical behavior of the robots. Robot models are needed because the gait controller has to be trained through an optimization process for each robot. For safety (both for the robots and the human) and time consumption (running simultaneous optimizations on a single real robot is impossible) reasons, this optimization phase can hardly be run on the real robot.

We complete the chapter with a presentation of the tools and the evolutionary algorithm used to perform this optimization process.

3.1 Robot Models

We cannot directly adapt the existing controller to the real robots[1]. Indeed, the gait controller includes up to 20 open parameters (see section 4.3.2). The open parameters must be tuned, through an optimization phase, to allow the robot to walk. As written above, a software model for each robot is required for this optimization phase.

These models are developed with the Robotran multibody simulation environment developed in the Center for Research in Mechatronics (CEREM) at the Université catholique de Louvain (UCL). Robotran is a symbolic software to model and analyze MultiBody Systems (MBS). A MBS is a mechanical system composed of rigid bodies that are connected together by joints which can be subject to rotational and/or translational displacements. More detailed information can be found in [4].

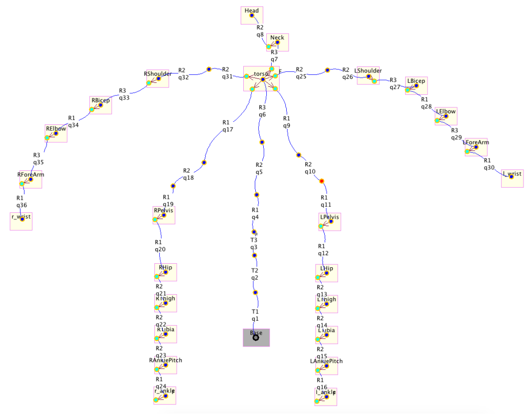
To simulate the physical behavior of the robots, the models must describe the kinematics and dynamics of the robots, the actuators dynamics and must include a ground contact model to simulate the ground reaction forces acting on the feet of the robots.

3.1.1 Kinematic and dynamic descriptions of the robots

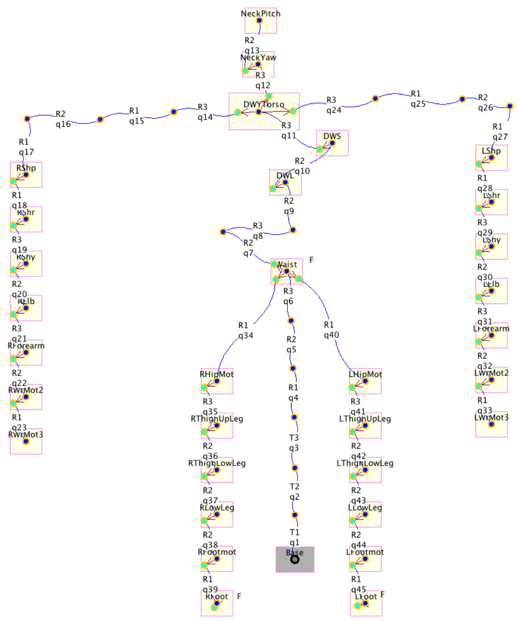
The MBS file format (.mbs) is an XML file format supported by Robotran. It describes the kinematics and dynamics of the robots. This robot description can be either manually edited or generated thanks to MBSysPad, the graphical user interface of Robotran. It is, in any case, a long and tedious work.

[1]Anyway, at the Université catholique de Louvain (UCL), NAO is the only real robot that we have at our disposal

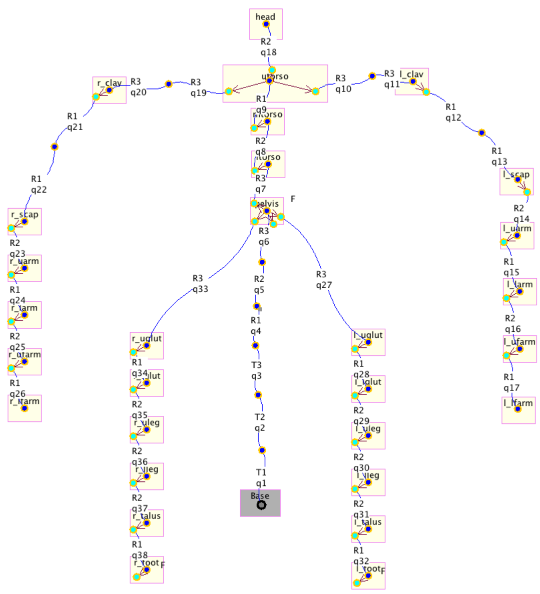
The Universal Robotic Description Format (URDF) is an XML file format used in ROS. Like the MBS file format, it describes the kinematics and dynamics of robots but it follows different conventions. The Appendix A presents the development of a tool converting available URDF files, containing all the desired data, into MBS files.



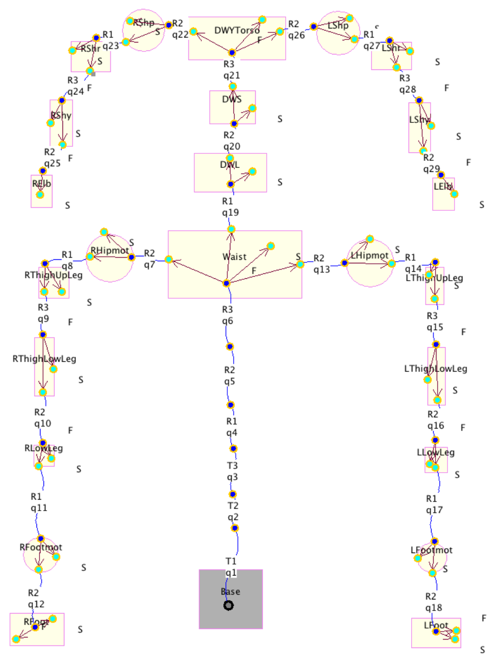
(a) Floating base model of NAO in MBSysPad



(b) Floating base model of WalkMan in MBSysPad



(c) Floating base model of Atlas in MBSysPad



(d) Floating base model of CoMan in MBSysPad taken from [19]

Figure 3.1: The robots are attached to the world inertial frame via a 6-DOFs floating base joint composed of 3 prismatic and 3 revolute joints. Although we adapted it a bit (fingers removed and graphical 2D diagram rearranged to give it a more human-like aspect for the sake of clarity), the model of NAO, WalkMan and Atlas were generated by the converting tool.

Thanks to this tool[2], we generated the MBS descriptions of NAO (Figure 3.1a), WalkMan (Figure 3.1b) and Atlas (Figure 3.1c). The MBS description of CoMan (Figure 3.1d) is taken from [19]. These models are all based on a floating base representation of legged robots. A floating base representation is characterized by the free motion of the base body. The free motion of the base is represented by a 6-DOFs joint (3 prismatic and 3 revolute joints about the XYZ axes) attached between the humanoid robot and the world inertial frame [6].

3.1.2 Actuators dynamics

As written in Chapter 2, the joints of Nao are actuated by DC motors, the ones of CoMan and WalkMan are actuated by serie elastic actuators (SEA) and the ones of Atlas are hydraulically-actuated. To be as close as possible to the reality, the actuators dynamics are simulated as follows :

DC motor We derive the equations of the DC motor by applying Kirchoff's voltage law and Newton's second law on the equivalent circuit of Figure 3.2. We get equations 3.1 and 3.2.

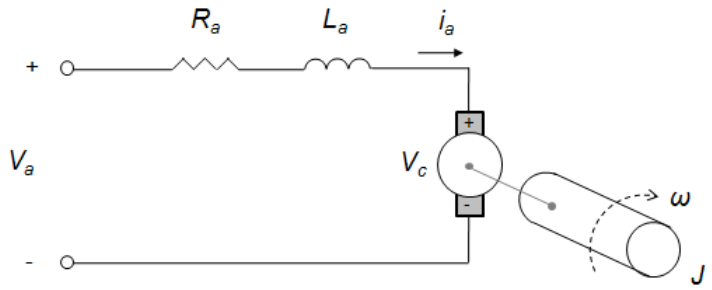


Figure 3.2: Electric equivalent circuit of the armature (figure taken from `http://nl.mathworks.com/products/symbolic/code-examples.html?file=/products/demos/symbolictlbx/DC_motor/DC_motor.html`)

$$V_a = R_a I_a + L_a \frac{dI_a}{dt} + V_c \tag{3.1}$$

$$J_m \frac{d\omega}{dt} = T - T_{load} \tag{3.2}$$

In 3.1, V_a is the supplied voltage, R_a and L_a are the armature resistance and inductance, I_a is the armature current and V_c is the back electromotive force.
In 3.2, ω is the angular speed of the motor, J_m is the moment of inertia of the rotor (in kg/m^2) and T is the electromagnetic torque generated by the DC motor.

V_c is proportional to the angular speed ω and the stator magnetic field B_s. If we assume that B_s is constant, we get :

$$V_c = k_e \omega \tag{3.3}$$

with k_e, the back-emf constant ($\frac{V}{rad/s}$).

In general, T is proportional to the armature current I_a and the stator magnetic field B_s. If we assume that B_s is constant, we get :

$$T = k_t I_a \tag{3.4}$$

[2]In Appendix A, it was also tested on iCub (for validation purposes), even if we do not study it for locomotion.

with k_t, the torque constant (Nm/A).
For DC motors, $k_t = k_e$. Indeed, at constant operating point (constant speed, voltage, current and torque),

$$V_a = R_a I_a + k_e \omega$$

$$T = k_t I_a$$

Hence, the electrical power is :

$$P_{elec} = V_a I_a = R_a I_a^2 + k_e \omega I_a$$

where $R_a I_a^2$ represents the losses. The mechanical output power is :

$$P_{meca} = T\omega = k_t I_a \omega$$

From the conservation of energy, P_{elec}=P_{meca}+ Losses. This can only be true if

$$k_t = k_e = k\phi$$

Neglecting the inductive effects, equations 3.1 and 3.2 become :

$$V_a = R_a I_a + k\phi\omega \tag{3.5}$$

$$T = k\phi I_a \tag{3.6}$$

From 3.5 and 3.6, we get :

$$T = \frac{k\phi}{R_a} V_a - \frac{(k\phi)^2}{R_a}\omega \tag{3.7}$$

The parameters $k\phi$ ("Torque Constant") and R_a ("Terminal Resistance") for the brushed DC motors of NAO are listed in Table B.1 of Appendix B. In the "Low-level controller" paragraph below, the way V_a is controlled to track a torque or position reference will be explained.

Serie elastic actuators (SEA) As mentionned in section 2.2, a SEA is an actuator incorporating series elasticity as a purposeful element. Equations are immediately forthcoming from diagram of Figure 3.3, by applying Newton's second law.

$$T_m - T_{load} = J_m \frac{d^2\theta_m}{dt^2} = J_m \frac{d\omega_m}{dt} \tag{3.8}$$

T_m is the electromagnetic torque generated by the DC motor, T_{load} is the load torque, J_m is the moment of inertia of the rotor, θ_m is the angular position of the motor, and ω_m is the angular speed of the motor.

$$T_m = \frac{k}{R_a} V_a - \frac{k^2}{R_a}\omega \tag{3.9}$$

$$T_{load} = K_s(\theta_m - \theta_{load}) + D_s(\omega_m - \omega_{load}) \tag{3.10}$$

with θ_{load} and ω_{load}, the angular position and speed of the load and with K_s ($\frac{Nm}{rad}$) and D_s ($\frac{Nm}{rad/s}$), the stiffness and damping coefficients of the serie elastic element.

Hence, we get :

$$\frac{1}{J_m}(\frac{k}{R_a}V_a - \frac{k^2}{R_a}\omega - K_s(\theta_m - \theta_{load}) - D_s(\omega_m - \omega_{load})) = \frac{d^2\theta_m}{dt^2} = \frac{d\omega_m}{dt} \tag{3.11}$$

The parameters k ("Torque Constant"), R_a ("Terminal Resistance"), K_s ("Stiffness Coefficient") and D_s ("Damping Coefficient") for the SEA of CoMan and WalkMan are listed in Tables B.2

and B.3 of Appendix B. In the "Low-level controller" paragraph below, the way V_a is controlled to track a torque or position reference will be explained.

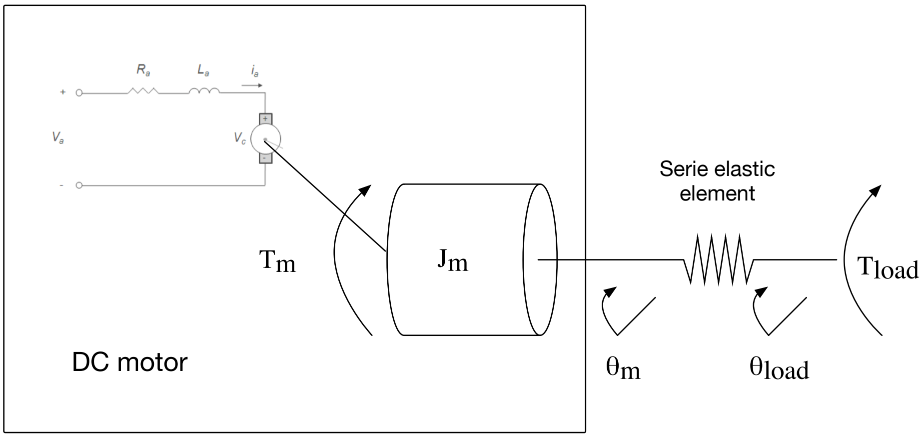


Figure 3.3: Block diagram of the serie elastic actuators used for CoMan and WalkMan.

Hydraulic actuators Atlas is equipped with hydraulically-actuated joints. Since we had no further information about its motors, we decided to use the same SEA as WalkMan, i.e. a robot of the same height, in the Atlas simulation model. The joint-actuator mapping is applied as defined in Table 3.1.

Table 3.1

Joints	Actuator type (relative to Table B.3 of Appendix B)
NeckPitch	1
ElbowYaw	1
WrsitPitch	1
WristYaw	1
TorsoRoll	2
HipYaw	3
TorsoYaw	3
ShoulderPitch	3
ShoulderYaw	3
ElbowPitch	3
HipRoll	4
KneePitch	4
HipPitch	4
AnklePitch	4
AnkleRoll	4

Low-level controller In our simulation models, each motor is locally controlled by its own low-level controller (similarly to [19]). The block diagram of the proposed controller is given in Figure 3.4.

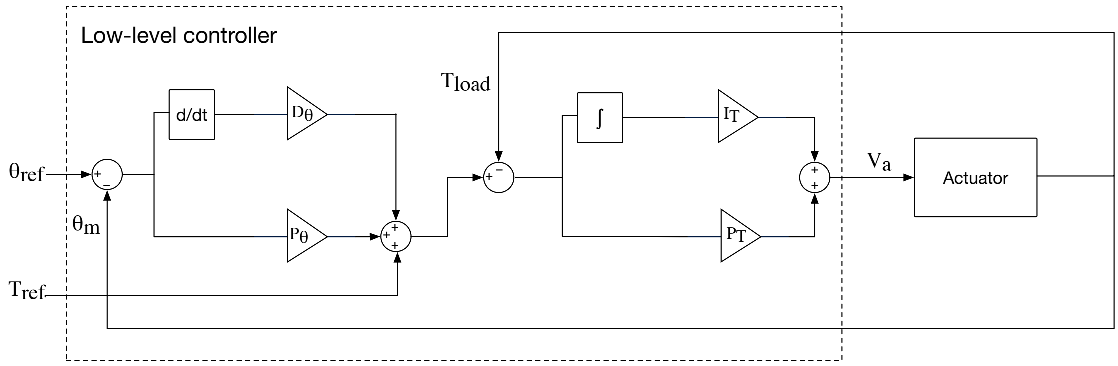


Figure 3.4: Block diagram of the proposed low-level controller

By implementing a back electromotive force compensation, we would simplify the synthesis of the controller and improve its performance. Unfortunately, because of a lack of time, we did not compensate the back electromotive force.

3.1.3 Ground contact model

This section briefly describes how the ground reaction forces, acting on the feet of the robots, were simulated in Robotran using a mesh-based ground contact model (GCM).
This GCM was developed in [19] and is based on the one introduced in [6]. As stated in [19], "the key principle is to compute the elementary ground reaction forces at each point of a user-defined mesh". The advantage of this GCM is the possibility to model complex foot geometries and the possibility to simulate uneven grounds.

Figure 3.5 shows the meshes used to model the robots feet. As written in section 2.1, the original feet of NAO are disproportionate. We decided to use shorter feet in simulation in order to facilitate the emergence of a human-like gait for this robot. The dimensions of the shorter feet and the associated mesh are shown in Figure 3.5a.
The same kind of modifications were done for CoMan in [20] and [21]. The dimensions of the shorter feet and the associated mesh are given in Figure 3.5b.

The ground-feet interaction forces have three components along each axis of the world inertial frame (F_x, F_y and F_z). F_x, F_y and F_z will be computed at each point of the meshes presented in Figure 3.5.

F_z, the normal component of the ground reaction force, is modeled using a nonlinear spring-damper model, as proposed in [8]. The normal ground reaction force (positive upwards), at each point of the mesh, is computed as follows :

$$F_z = -k_z \Delta z (1 - d_z \dot{z}) \qquad (\Delta z > 0)(\dot{z} < \frac{1}{d_z}) \tag{3.12}$$

where k_z is the normal contact stiffness coefficient, d_z is the normal contact damping coefficient, Δz is the ground penetration of that contact point and $\dot{z}$ is its normal velocity.

F_x and F_y, the tangential components of the ground reaction force, are also modeled using a nonlinear spring-damper model, as proposed in [8]. The tangential ground reaction force, at each point of the mesh, is computed either as a sliding force (equations 3.13 and 3.14) or as a stiction force (equations 3.15 and 3.16). A contact point is in stiction if $|\dot{x}|$ (or $|\dot{y}|$) $<$ 1cm/s and returns to sliding if $F_{x,y,st} \geq \mu F_z$.

$$F_{x,sl} = -\text{sgn}(\dot{x}) \mu F_z \tag{3.13}$$

$$F_{y,sl} = -\text{sgn}(\dot{y})\mu F_z \tag{3.14}$$

where $\dot{x}$ and $\dot{y}$ are the horizontal velocities, μ is the sliding friction coefficient and F_z is the normal component of the ground reaction force.

$$F_{x,st} = -k_{xy}\Delta x(1 + \text{sgn}(\Delta x)d_{xy}\dot{x}) \qquad (\text{sgn}(\Delta x)d_{xy}\dot{x} > -1) \tag{3.15}$$

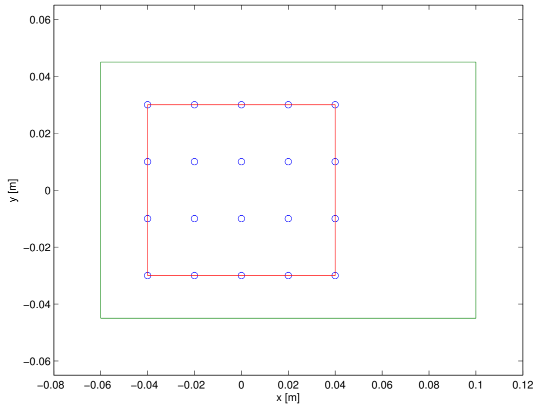
$$F_{y,st} = -k_{xy}\Delta y(1 + \text{sgn}(\Delta y)d_{xy}\dot{y}) \qquad (\text{sgn}(\Delta y)d_{xy}\dot{y} > -1) \tag{3.16}$$

where k_{xy} is the tangential contact stiffness coefficient, d_{xy} is the tangential contact damping coefficient, Δx and Δy are the offsets between the current and initial positions of the contact points, and $\dot{x}$ and $\dot{y}$ are its velocities.

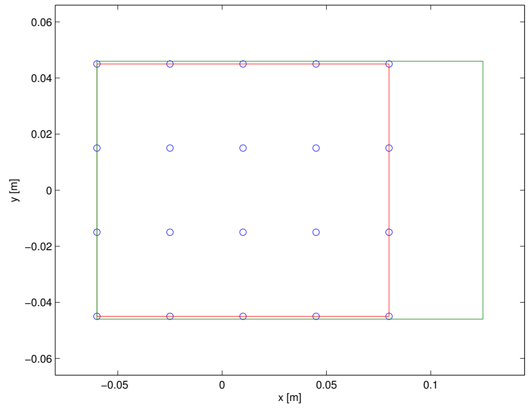
The different parameters, listed for each robot model in Table 3.2, are manually tuned so that they provide a good behavior of the foot on the ground. NAO being far lighter than the three other robots, we need a less stiff spring-damper model for the normal component. For CoMan, WalkMan and Atlas, we take the same values as in [21] and [5].

Table 3.2: GCM parameters for the four investigated robots

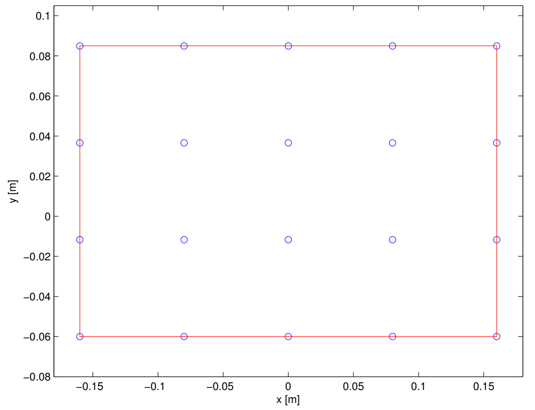
Parameter	Value for NAO	Value for CoMan	Value for WalkMan	Value for Atlas
k_z $[N/m]$	10000	81500	81500	81500
d_z $[s/m]$	5	30	30	30
k_{xy} $[N/m]$	8200	8200	8200	8200
d_{xy} $[s/m]$	30	30	30	30
μ $[-]$	0.8	0.8	0.8	0.8



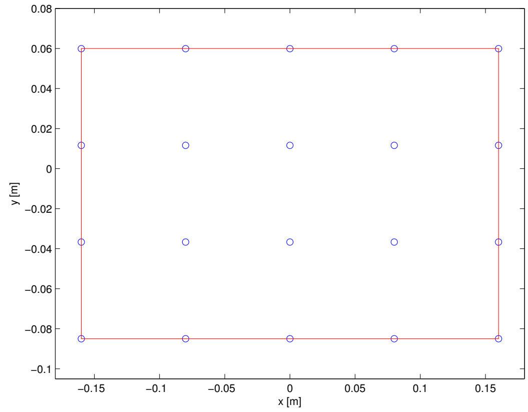
(a) Mesh used to model the feet of NAO. The green box shows the dimensions of the initial feet (presented in section 2.1). The red box shows the dimensions of the shorter feet used in simulations



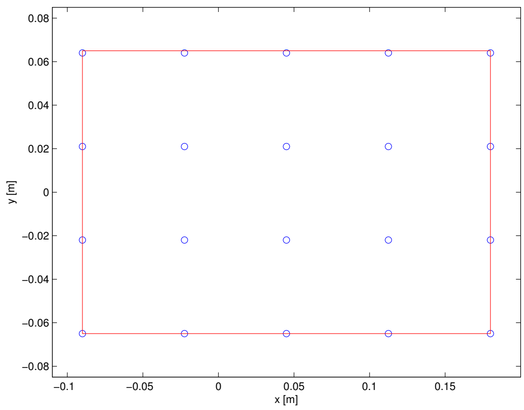
(b) Mesh used to model the feet of CoMan. The green box shows the dimensions of the initial feet (presented in section 2.2). The red box shows the dimensions of the shorter feet used in simulation in [21]. These shorter feet really exist and were used to perform experiments on the real CoMan in [20]



(c) Mesh used to model the left foot of Walk-Man



(d) Mesh used to model the right foot of Walk-Man



(e) Mesh used to model the feet of Atlas

Figure 3.5: Meshes used to model the ground-feet interactions in the GCM.

3.2 Optimization

3.2.1 Problem Statement

The bio-inspired gait controller, presented in Chapter 4, contains up to 20 open parameters. These open parameters must be carefully tuned in order to expect the robots to walk with human-like gaits. This parameter tuning can hardly be done manually so that an optimization phase is required.

Each parameter is bounded in a user-defined interval. The key principle of the optimization phase is the maximization of an objective function that quantitatively evaluates the quality of the gait. This objective function is described in section 4.5.

3.2.2 Covariance Matrix Adaptation Evolution Strategy algorithm (CMA-ES)

We need an appropriate optimization framework to find a solution to the complex problem described above.

We will use the Covariance Matrix Adaptation Evolution Strategy algorithm (CMA-ES). It is a heuristic evolutionary algorithm based on the principle of biological evolution. The choice for a heuristic algorithm is guided by their capacity to find an optimal solution to complex optimization problems (i.e., with a high number of variables and constraints being non-linear). However, the optima being found are not always global and the optimizer can be trapped in possible local optima.

In evolutionary algorithms, from a set of μ "parent" particles, λ "child" particles are produced. To produce each child, ρ parents recombine with each other. The produced child are then mutated, by adding a random value from a normal distribution. The μ particles giving the best value to the objective function are then selected and produce again λ "child" particles, during the next iteration.
As written above, the CMA-ES algorithm is an evolutionary algorithm. The specificity of the CMA-ES algorithm lies in the adaptation, over the iterations, of the covariance matrix of the normal distribution used for the mutation[3].

Running such optimization processes requires a lot of computational ressources both regarding time and power. Fortunately, the "Consortium des Equipements de Calcul Intensif" (CECI) provides researchers and students from 5 universities in Belgium (Université catholique de Louvain, Université de Namur, Université libre de Bruxelles, Université de Mons and Université de Liège) with access to powerful computing equipments (clusters). Clusters are installed and managed locally in the different sites, but they are accessible to all researchers from the member universities via a command-line version of Linux[4]. Basic parallel computing libraries, such as Message Passing Interface (MPI), are installed. Heuristic algorithms like CMA-ES are well adapted to run in parallel and thanks to MPI, a considerable amount of time is saved. Indeed, different particles can be simultaneously evaluated on different nodes of the cluster.

[3] For more detailed information about the CMA-ES algorithm, see `https://en.wikipedia.org/wiki/CMA-ES`
[4] For more information about the CECI clusters, see `http://www.ceci-hpc.be/clusters.html`

Chapter 4

Bio-inspired Walking Controller

This thesis aims at generalizing an existing bio-inspired 2D walking controller (designed for the robot CoMan by A. Barrea and N. Van der Noot in [19] and further extended in [20] and [21]) to three other humanoid robots : NAO, WalkMan and Atlas. By 2D walking controller, we mean that "the floating base is constrained so that the robots cannot fall aside" [19]. In simulation, 2D walking gaits are achieved by constraining the lateral position of the waist, along with its roll and yaw angles to stay at a zero position (see section 3.1.1).
The controller general structure has not changed a lot since then. However, modifications were needed to adapt it to robots having different scales and morphologies.

In this chapter, we will first briefly analyze the human gait. Since we want the robots to adopt bio-inspired gaits, it is useful to specify the main characteristics of the human way of walking.

The bio-inspired 2D walking controller is then introduced. The controller presentation is divided into two sections. They are respectively devoted to the control of the leg degrees of freedom (DOFs) and the control of the upper body DOFs.

To finish the chapter, we discuss the staged objective function (called "fitness" function in the remainder of this work) used in the optimization phase described in section 3.2.

4.1 Human gait analysis

In this section, we briefly describe the different phases of the human walking gait. We also introduce, in Figure 4.1, the terminology related to these phases.

In the remainder of this work, a walking gait cycle is defined as the time interval between a *heel strike* of a leg and the next *heel strike* of the same leg. By *heel strike*, we mean the moment when the heel impacts the ground.
As shown in Figure 4.1, a gait cycle is divided in two main phases :

- A leg is in "stance phase" when its foot is in contact with the ground.
- A leg is in "swing phase" when its foot is in the air.

The distinction between swing and stance phases is extremely important because different control rules are applied, depending on this leg state (see section 4.3.2).

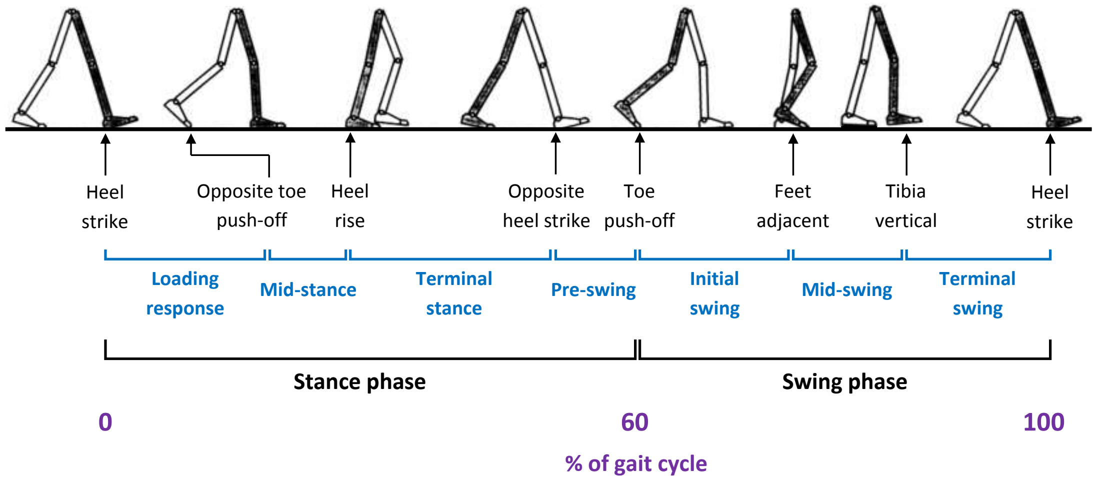


Figure 4.1: A human walking gait cycle is divided into 7 periods punctuated by 8 events. The considered leg is depicted in black. Figure taken from [19]

4.2 Controller overview

This work focus on steady-state walking. The controller described in this chapter is then the controller used after walk initiation. The initiation consists in bending the robots forward so that the x component of its center of mass tracks a position reference (x_ref_end) which is tuned during the optimization phase. In order to track this position reference, we use the impedance controller with inverse muscular model, developed in [10], during the two first seconds of the simulation.

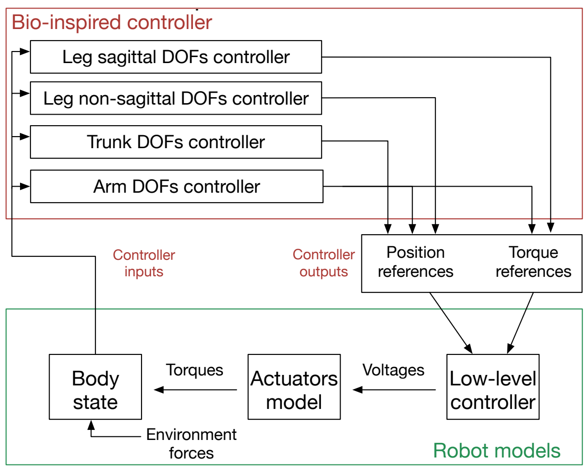


Figure 4.2: Interactions between the bio-inspired controller and the robot models developed in section 3.1. Figure inspired from [19]

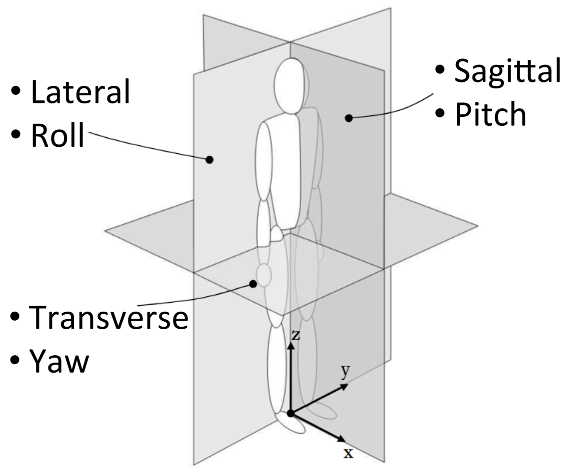


Figure 4.3: Definition of the world reference planes. Figure taken from [19]

As presented in Figure 4.2, the bio-inspired controller receives inputs from the robots and their environment in Robotran (see section 3.1). The controller inputs are :

- Ground reaction forces measured under each foot
- Position, velocity and torque for each joint

- Absolute orientation matrix provided by the IMU

Based on these inputs, the controller computes the output position or torque references (depending on the joints), for each joint. The leg sagittal DOFs are always controlled with torque references[1], the leg non-sagittal and the trunk DOFs are always controlled with position references and the arm DOFs are controlled with position or torque references. The definition of the sagittal (pitch), lateral (roll) and transverse (yaw) planes is given in Figure 4.3 (a sagittal joint being defined as a joint whose revolute axis is normal to the sagittal plane).

4.3 Leg DOFs

The leg non-sagittal joints are useful to maintain lateral dynamic stability. Since the robots cannot fall aside (because we study 2D walking gait), the control of all non-sagittal joints simply consists in tracking zero-position references.

The leg sagittal joints (i.e., hip, knee and ankle) are the most important joints for walking. These joints are controlled using a bio-inspired approach described in [8]. The key idea is that "human locomotion is controlled by a chain of reflexes encoding principles of legged mechanics" [8].

As presented in Figure 4.4, seven muscle groups are identified for each leg [8]. SOL, TA, GAS, VAS, HAM, GLU, and HFL respectively stand for soleus, tibialis anterior, gastrocnemius, vasti, biarticular hamstring, gluteus and hip flexor. These muscles react, by contracting, to neuronal signals (generated, in this case, by reflexes). This causes the legs to move and allows humans to walk.

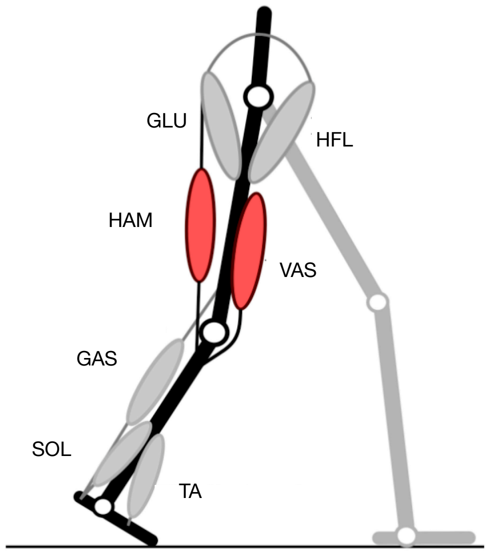


Figure 4.4: Seven muscle groups are identified in each leg in [8]. Figure adapted from [16]

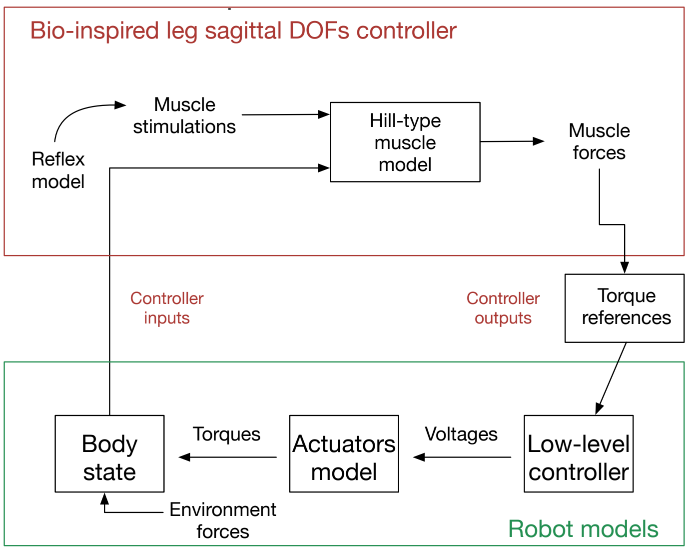


Figure 4.5: Schematic of the legs sagittal DOFs controller. Figure inspired from [19]

Of course, the robots do not have any muscle. Therefore, we need to simulate virtual muscles whose behavior is pretty similar to biological muscles. More precisely, we use the Hill-type model. To control these muscles, we derive stimulations from the reflex model developed in [8]. Both the Hill-type model and the reflex model are detailed below. A schematic of the bio-inspired leg sagittal DOFs controller is presented in Figure 4.5.

[1]We suppose, throughout this thesis, that the joints of all the robots are equipped with torque sensors. In Chapter 2, we stated that it was not the case for NAO in reality. This means that the controller developed for NAO, in this master's thesis, cannot be directly applied on the real NAO.

4.3.1 Biological muscle model (Hill-model)

As written above, we use the Hill-type model [11] to simulate virtual muscles. Each one of the seven muscle groups is represented using a muscle-tendon unit (MTU). In normal operation, each MTU is composed of two main elements : a contractile element (CE) and a serie elastic element (SE). Outside its normal range of operation, the muscle state is also affected by two other elements : a parallel-elastic element (PE) and a buffer elasticity element (BE). This is illustrated in Figure 4.6. The following paragraph is based on [20].
Since we know all the joints positions (ϕ) of the robots, the length l_{mtu} can be computed from the joint angles. As shown in Figure 4.6, the length l_{ce} is dependent of the muscle activation A (more details below). The length l_{se} is directly related to F_m, the force produced by the muscle. This length is computed as $l_{se} = l_{mtu} - l_{ce}$. Each muscle then generates a torque contribution on one or two (for the bi-articular muscles HAM and GAS) joints as $\tau_m = r_m(\phi) * F_m$, where $r_m(\phi)$ is the lever arm (dependent on the joints positions). These torque contributions are then summed for each joint and the resulting torque references are sent to the low-level controller of Figure 4.5. For more detailed information, see [8].

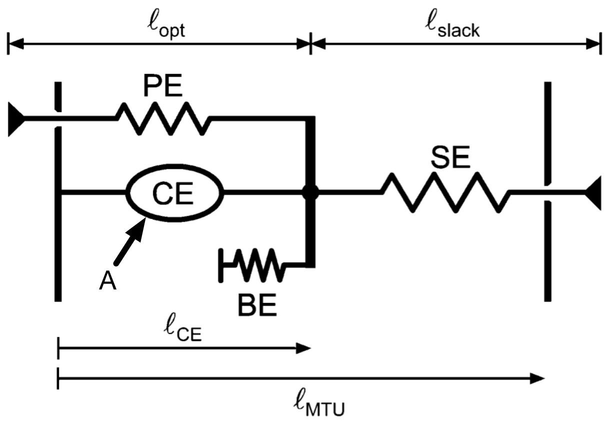


Figure 4.6: Hill-type model muscle-tendon unit (MTU). Figure adapted from [8]

Dynamic scaling of the muscles parameters

We want to implement muscles on the four robots, using the Hill-model presented above. To this purpose, we need the following muscles characteristics :

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

We get these characteristics from [8], except for m and λ whose values are given as explained in [17]. Parameters m and λ are only useful in section 4.5 to compute the metabolic energy consumption for each muscle.

The data given in [8] are the results for an adult-like model of 1.8 m and 80 kg. Since we want to implement muscles on four robots having different scales and morphologies, a scaling of the muscle parameters is required. To this end, we use dynamic scaling which is a scaling method presented in [16] and [2]. We will present the equations needed to scale the muscle properties for the size and the mass of the four considered robots (NAO, CoMan, WalkMan and Atlas).

Suppose we have two bodies :

- Body 1, for which we know all the muscle properties, with length L_1, mass M_1, and time reference T_1
- Body 2, for which we want to scale the muscle properties based on the ones of body 1, with length L_2, mass M_2, and time reference[2] T_2

Since gravity (in $[m/s^2]$) must be the same for both bodies :

$$\frac{G_2}{G_1} = \frac{L_2/T_2^2}{L_1/T_1^2} = \frac{L_2}{L_1}\frac{T_1}{T_2}^2 = 1 \rightarrow \frac{T_2}{T_1} = \sqrt{\frac{L_2}{L_1}} \tag{4.1}$$

From Newton's law and equation 4.1, we get the forces ratio :

$$\frac{F_2}{F_1} = \frac{M_2}{M_1}\frac{L_2/T_2^2}{L_1/T_1^2} = \frac{M_2}{M_1} \tag{4.2}$$

From the definition of a torque and equation 4.2, we get the torques ratio :

$$\frac{\tau_2}{\tau_1} = \frac{L_2 F_2}{L_1 F_1} = \frac{L_2}{L_1}\frac{M_2}{M_1} \tag{4.3}$$

We then compute the velocity ratio (4.4) and the velocity ratio normalized by the length (4.5). From equation 4.1, we get :

$$\frac{V_2}{V_1} = \frac{L_2/T_2}{L_1/T_1} = \frac{L_2}{L_1}\frac{T_1}{T_2} = \frac{L_2}{L_1}\sqrt{\frac{L_1}{L_2}} = \sqrt{\frac{L_2}{L_1}} \tag{4.4}$$

$$\frac{v_2}{v_1} = \frac{1/T_2}{1/T_1} = \frac{T_1}{T_2} = \sqrt{\frac{L_1}{L_2}} \tag{4.5}$$

In order to evaluate these ratios, we then need to compute the length and the mass ratios. Therefore, we compare the leg length and the total mass of the adult-like model (L_H = 1m and M_H = 80 kg) of [8] with the ones of the four robots considered in this thesis (L_R and M_R). Considering the values of Chapter 2, we get the results of Table 4.1.

Table 4.1: Comparison between the leg length and mass of the adult-like model of [8] and the ones of the four robots considered in this thesis

Robot	L_R/L_H	M_R/M_H
NAO	0.33	0.0625
CoMan	0.52	0.3875
WalkMan	1.10	1.77
Atlas	0.94	2.15

[2] Although it may seem counter-intuitive, two bodies can have different time references in the dynamic scaling theory. For instance, small animals usually have higher motion frequencies than bigger animals with a similar body shape.

From equations 4.1 to 4.5, we get the results of Table 4.2 for the different ratios.

Table 4.2: Ratios used to scale the muscle properties for the size and the mass of the different robots

Robot	L_R/L_H	M_R/M_H	F_R/F_H	τ_R/τ_H	V_R/V_H	v_R/v_H
NAO	0.33	0.0625	0.0625	0.0206	0.5744	1.7409
CoMan	0.52	0.3875	0.3875	0.2015	0.7211	1.3867
WalkMan	1.10	1.77	1.77	1.947	1.048	0.9541
Atlas	0.94	2.15	2.15	2.021	0.9695	1.0314

Based on these ratios, the muscle properties for the different robots are computed as follows :

- $F_{max,R} = \frac{F_R}{F_H} \cdot F_H$
- $v_{max,R} = \frac{V_R}{V_H} \cdot v_{max,H}$
- $l_{opt,R} = \frac{L_R}{L_H} \cdot l_{opt,H}$
- $l_{slack,R} = \frac{L_R}{L_H} \cdot l_{slack,H}$
- $r_{0,R} = \frac{L_R}{L_H} \cdot r_{0,H}$
- $\phi_{max,R} = \phi_{ref,H}$
- $\phi_{ref,R} = \phi_{ref,H}$
- $\rho_R = \rho_H$
- $m_R = \frac{M_R}{M_H} \cdot m_H$
- $\lambda_R = \lambda_H$

The muscle properties values given in [8] for an adult-like model and the scaled values for the different robots are listed in Appendix C.

The dynamic scaling theory will also be used to scale the arms and torso muscles properties (see section 4.4) and to analyse and compare the simulation results in section 5.

4.3.2 Bio-inspired reflex rules

This section will give the basic principles of the reflex model developed in [8] and used to control the virtual muscles.
As written above, the length l_{ce} of CE depends on the muscle activation signal A (Figure 4.6). As stated in [20], that "the muscle activations A are related to neural inputs S called muscle stimulations, using a first-order low-pass filter capturing the excitation-contraction coupling, presented in equation 4.6, where τ is a time constant".

$$\tau \frac{dA}{dt} = S - A \tag{4.6}$$

The computation of these stimulations S can be achieved with many different approaches. In this thesis, we use the reflex-based approach proposed by [8]. The key idea is that the signals S are combinations of different weighted inputs such as the muscle states (F_m and l_{ce}), the ground reaction forces, the knee angle and its derivative or the absolute trunk sagittal angle and its derivative.

The control rules depends on the current phase of the leg (swing or stance phase) and are given in Appendix D. To determine if the legs are in swing or in stance phase, we use a finite state machine.

A leg enters the swing state if :

- The force acting on its foot is under a defined force threshold. This threshold equals 20 N for CoMan, WalkMan and Atlas and 10 N for NAO.
- The current stance phase was long enough (> 0.2 s). This is more a numerical safety condition.

A leg enters the stance state if :

- The force acting on its foot is greater or equal to the previously defined force thresholds.
- The current swing phase was long enough (> 0.15 s). Again, this is more a numerical safety condition.
- The ankle pitch angle < 0.6 [rad]

Open parameters to optimize

As presented in Appendix D, the reflex rules are parametrized with constant values that must be tuned. These 19 constants constitute the main part of the 20 open parameters to optimize that we were talking about in the problem statement of section 3.2. The 20th parameter is related to walk initiation (see section 4.2). The twenty open parameters and their optimization bounds are listed in Table 4.3. Because the control rules parameters are normalized, we do not need to adapt the bounds to the different robots. Only x_ref_end, the parameter related to walk initiation, needs to be adapted.

Suggested improvement

In [19], the ground reaction forces sent to the leg sagittal DOFs controller were limited, for safety reasons, to 700 N. When we adapted the controller (initially designed for the robot CoMan) to the other robots, we forgot to increase this limit for Atlas and WalkMan. The parameters for these two robots were then optimized with wrong ground reaction forces. Only three control rules are affected by the ground reaction forces : S_{HAM}, S_{GLU} and S_{HFL} in stance phase. The HAM, GLU and HFL muscles contribute to the trunk's balance control, only during the stance phase (see the equations below, where $\theta(t)$ is the trunk sagittal angle at time t). The ground reaction forces are used to modulate the stimulations of these muscles proportionally to the amount of body weight it bears. This allows smoother transitions between single and double support phases.
Fortunately, the open parameter k_{bw} was there to compensate somehow the erroneous value of $\frac{|F_{leg}^{ipsi}(t_s)|}{F_g}$ during the optimization phase (see the control rules for HAM, GLU and HFL in stance phase below). The performance of the controller is not very impacted by this error but running new optimizations, with the correct ground reaction forces, might give even better results.

$$(5) \quad S_{HAM} = S_{MIN} + [k_p \cdot (\theta(t_s) - \theta_{ref}) + k_d \cdot \dot{\theta}(t_s)]_{\{+\}} \cdot k_{bw} \cdot \frac{|F_{leg}^{ipsi}(t_s)|}{F_g}$$

$$(6) \quad S_{GLU} = [S_{MIN} + [0.68 \cdot k_p \cdot (\theta(t_s) - \theta_{ref}) + k_d \cdot \dot{\theta}(t_s)]_{\{+\}} \cdot k_{bw} \cdot \frac{|F_{leg}^{ipsi}(t_s)|}{F_g}]_{\{\text{SSup}\}} - [S_{MIN}]_{\{\text{DSup}\}}$$

$$(7) \quad S_{HFL} = [S_{MIN} - [k_p \cdot (\theta(t_s) - \theta_{ref}) + k_d \cdot \dot{\theta}(t_s)]_{\{-\}} \cdot k_{bw} \cdot \frac{|F_{leg}^{ipsi}(t_s)|}{F_g}]_{\{\text{SSup}\}} + [S_{MAX}]_{\{\text{DSup}\}}$$

Table 4.3: Optimized parameters bounds

		min	max
1	$S_{0,VAS}$	0.01	0.02
2	G_{SOL}	0.6	3
3	G_{GAS}	0	2
4	G_{VAS}	0.6	15
5	G_{HAM}	0	0.9
6	G_{GLU}	0	0.7
7	G_{HFL}	0.1	4
8	$G_{HAM-HFL}$	0	150
9	$G_{TA,SW}$	2.5	5
10	$G_{TA,ST}$	2.5	5
11	l_{HFL}^{off}	0	0.8
12	l_{HAM}^{off}	0.75	2
13	φ_{knee}^{off}	2.5	3.14
14	$l_{TA,SW}^{off}$	0.5	0.95
15	$l_{TA,ST}^{off}$	0.5	0.95
16	k_d	0.05	1
17	k_p	1.4	24
18	k_{bw}	0.05	6
19	θ_{ref}	0.01	0.2
20	x_ref_end	0	0.04 (NAO)
		0	0.08 (CoMan)
		0	0.16 (WalkMan)
		0	0.12 (Atlas)

4.4 Arm and Trunk DOFs

In general, the upper-body is useful to provide lateral balance to the walker. Because we study 2D walking, this lateral balance is not needed and the trunk joints[3] are simply controlled by tracking constant position references.

Regarding the arms, only the shoulder and the elbow sagittal DOFs are controlled with torque references. The other joints do not impact sagittal robot motion and are then controlled with constant position references.

As done for the leg sagittal DOFs, we implemented virtual muscles for the arm sagittal DOFs using the Hill-type model. Namely, for each leg :

- 2 muscles at the shoulder level, in the sagittal plane : SET and SFL
- 2 muscles at the elbow level, in the sagittal plane : EFL and EET

In [17], the characteristics of these muscles are given for an adult-like model. Using the exact same method as in section 4.3, these characteristics are dynamically scaled for the four robots considered in this thesis.

To control the elbow virtual muscles, we send stimulations to the EFL and EET muscles. These stimulations are computed using a proportional controller which tracks a constant position ref-

[3] 3 trunk joints for CoMan, WalkMan and Atlas, 1 trunk joint for NAO

erence.

To control the shoulder virtual muscles, we send stimulations to the SET and SFL muscles. These stimulations are computed using a proportional controller which tracks a position reference. The position references for the left and right shoulder are both proportional to the sagittal hip angles difference but are of opposite signs. This rule is used to achieve asynchronous arm motion, compared to the corresponding leg position.

In [19], the position references for the shoulder were also proportional to the sagittal hip angles difference but the shoulder sagittal DOFs were controlled with position references. The muscular model and the torque control for these DOFs give smoother and more human-like movements.

4.5 Fitness function for steady-state walking

As presented in section 4.3.2, the bio-inspired gait controller includes 20 open parameters that needs to be tuned through an optimization phase. As developed in section 3.2, a fitness function (also called objective function) is used to quantitatively evaluate the quality of the gait and to guide the optimizer toward an optimal solution.

In this section, we explain how the quality of the gait is evaluated by the fitness function. Because the minimization of the metabolic energy consumption is one of the criteria, the metabolic energy model is also briefly introduced.

4.5.1 Staged fitness function

The fitness function used to optimize the controller is exactly the same as the one used in [19]. It consists of several stages. The key principle is that the stages are entered one after another, a stage being accessible only if the previous one was completed. The four stages of the fitness function and their entrance conditions are listed below.

1. The robot is able to walk a minimal distance in 60 s (the simulation time), without falling. The minimal distance is smaller for NAO and CoMan (9 and 15 m, respectively) than for WalkMan and Atlas (18 m). Since it is the first stage of the fitness function, there is no entrance condition.

2. The robot is able to walk, without falling, during the simulation time. To enter this stage, the robot must have walked the minimal distance defined in the first stage.

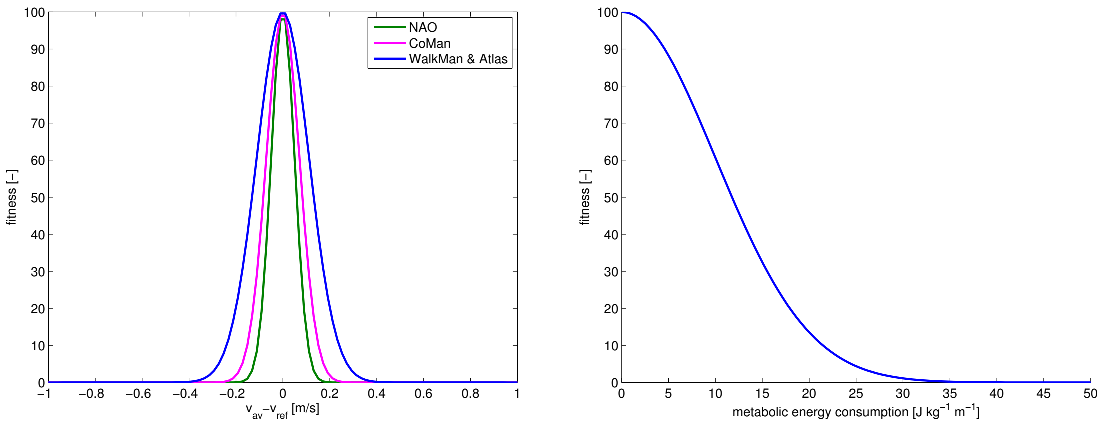
3. The robot is able to walk at a given speed. Te enter this stage, the robot must have walked for 60 s. It is important to note that this stage is optional. Results without and with this stage are presented in sections 5.3 and 5.4, respectively.

4. The robot minimizes its metabolic energy consumption per unit distance walked. This stage is entered if the robot average walking speed is close enough to the target walking speed. More precisely, this stage is entered if $|v_{av} - v_{ref}| < v_{margin}$ with v_{av}, the average speed of the robot, v_{ref}, the target speed of stage 3 and v_{margin} equal to 0.02 [m/s] for NAO, 0.05 [m/s] for CoMan and 0.1 [m/s] for WalkMan and Atlas. If the 3rd stage is not activated, the entrance condition is a walking time of 60 s.

If the third stage is not activated, the robot "naturally" converges to an energy-efficient speed (see section 5.2).

Each stage is associated with a quantitative fitness bounded between 0 and 100, as follows :

1. the fitness is proportional to the fraction of the minimal distance traveled by the robot.
2. the fitness is proportional to the time the robot walked without falling
3. the fitness is evaluated using a gaussian distribution so that it is maximal when the average speed is exactly equal to the target speed. See Figure 4.7a.
4. the fitness is evaluated using a gaussian distribution so that it is maximal when the metabolic energy consumption is equal to 0 (which is of course unrealistic). See Figure 4.7b.



(a) Gaussian function associated with the 3rd fitness stage. Nao and CoMan being smaller than WalkMan and Atlas, $v_{av} - v_{ref}$ must be smaller for these robots as well. This is why we use thiner gaussian distributions for them.

(b) Gaussian function associated with the 4th fitness stage. The metabolic energy consumption being normalized by the mass of the robots, we use the same gaussian distribution for the four robots.

Figure 4.7

4.5.2 Metabolic energy consumption model

The metabolic energy that we are evaluating in the fitness function is the energy consumed by the virtual muscles developed for the leg and arm sagittal DOFs in section 4.3 and 4.4. Although it is different from the electrical energy used by the actuators, we will compare these two energetic consumptions in section 5.

The energy consumption model used in this project is based on [3]. It is exactly the same as the one used in [19], [20] and [21]. That is why we will not expand on it too much. The following description is based on [19].

From the first law of thermodynamics,

$$\dot{E} = \dot{H} + \dot{W} = (\dot{A} + \dot{M} + \dot{S} + \dot{B}) + \dot{W} \tag{4.7}$$

where

$\dot{E}$ is the rate of metabolic energy expenditure for a given MTU.

$\dot{H}$ is the rate at which heat is liberated.

$\dot{W}$ is the rate at which work is done. It is related to the contraction velocity of the MTU (v_{CE}) and the active force produced by its contractile element (F_{CE}).

$\dot{A}$ is the muscle activation rate. It models the rate of energy that a MTU converts in heat. It is related to the mass of the MTU (m_{MTU}) and its level of stimulation S.

$\dot{M}$ is the muscle maintenance heat rate. It models the rate of energy that a MTU converts in heat to maintain contraction at a certain level. It is related to the mass of the MTU (m_{MTU}), its activation level (A_m) and its normalized fiber length ($\tilde{l}_{CE} = \frac{l_{CE}}{l_{opt}}$).

$\dot{S}$ is the muscle fibers shortening heat rate. It is related to the contraction velocity of the MTU (v_{CE}) and the total force that it produces.

$\dot{B}$ is the basal metabolic rate. It is estimated to $0.0225 \cdot m_{MTU}$ in [3].

Chapter 5

Simulation Results & Comparison

In Chapter 2 and Chapter 3, we described the different robots and the models used to simulate their physical behavior in Robotran. Chapter 4 was then dedicated to the description of the bio-inspired controller and its adaptation to the different robots.

In this chapter, the simulation results are presented and analyzed. We start by analyzing several walking gaits which are the results of an optimization phase without speed constraint. We then compare one optimized controller of each robot with human data and analyze the results with the dynamic scaling theory. We finally compare the results of several optimization phases with different target speeds and analyze the evolution of the robots gaits across a speed range.

5.1 Optimizations with different fitness functions

As developed in section 4.3, twenty open parameters need to be tuned through an optimization phase. Different combinations of these parameters can make the robot walk. These multiple possibilities correspond to local optima in the search space. As written in section 3.2, a drawback of the optimization framework used is the possibility to get trapped in local optima.

Consequently, two different sets of parameters can be found as results of two similar optimization runs. It is then important to run several similar optimizations and to consider the different solutions found.

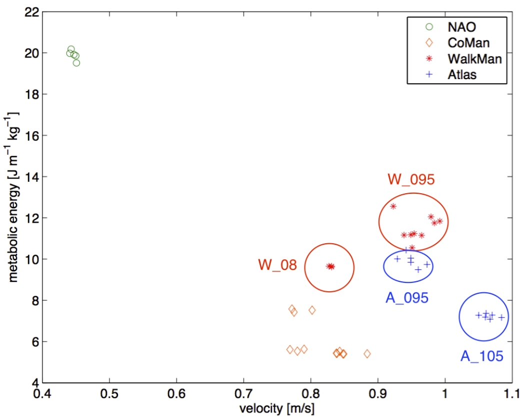
Note that, during the optimization phase, we use a specific ground with "obstacles" next to the stance foot to increase foot clearance with the ground, see [5].

5.1.1 Three-stage fitness function

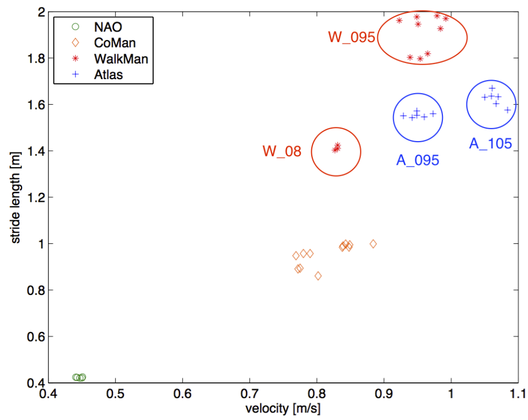
We start by running 4 optimizations for each robot with exactly the same parameters (same optimization bounds as in Table 4.3 of section 4.3.2, 100 particles and 200 generations in the CMA-ES framework and same optimization time). The fitness function used for these optimizations is the three-stage fitness function without speed constraint, described in section 4.5. As a reminder, this fitness function aims at guiding the optimizer toward solutions which allow the robots to walk without falling while trying to minimize their metabolic energy consumption. We then expect the optimizer to converge toward the *natural* and optimal walking speeds of the robots.

We get four sets of parameters for each robot corresponding to sixteen "different" controllers (four for each robot). We evaluated each controller three times in simulation and present the results in Figure 5.1 in terms of metabolic energy consumption, stride period and stride length.

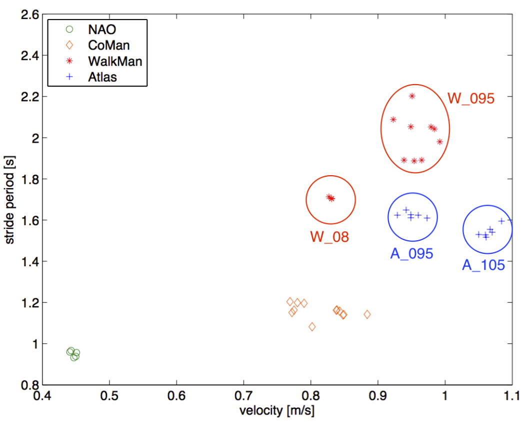
By stride length (resp. period), we mean the distance (resp. time) between two successive *heel strike* events of the same foot. The stride period is then equal to the gait cycle period.



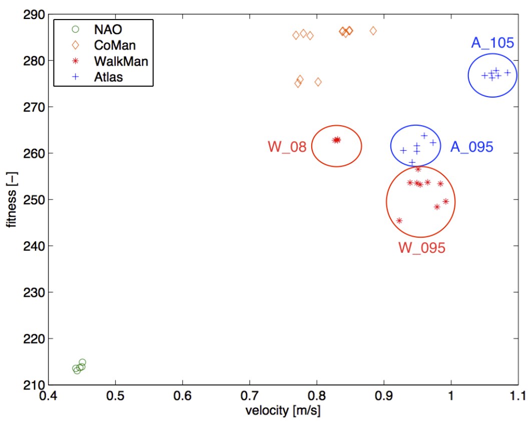
(a) Metabolic energy consumption per unit distance walked corresponding to the different solutions found



(b) Stride length corresponding to the different solutions found



(c) Stride period corresponding to the different solutions found



(d) Fitness value returned by the fitness function corresponding to the different solutions found

Figure 5.1: Results corresponding to 4 optimizations for each robot. Each optimized solution was run 3 times.

By analyzing the results of Figure 5.1d for **NAO**, we observe that all the optimizations converged to the exact same set of parameters and gait features. This set of parameters allows the robot to walk without falling, at an average speed of 0.45 m/s with a metabolic energy consumption of around 20 $Jm^{-1}kg^{-1}$. In Figure 5.1a, we see that for this "working" set of parameters, the metabolic energy consumption is way bigger than for the other robots. We ran tens of optimizations with different optimization bounds and modified control rules for the leg sagittal DOFs but it never gave better results than the ones presented in Figure 5.1 and analyzed in section 5.2.1. We will try to explain these observations in section 5.2.1.

Considering the results of Figure 5.1 for **CoMan**, we observe that all the optimizations converged to the same kind of solutions featuring an average speed of around 0.8 m/s and a good minimization of the metabolic energy consumption. To further analyze these results in sections 5.2.2 and 5.3, we chose the set of parameters giving the best fitness value which correspond to a speed of around 0.8 m/s and a metabolic energy consumption of around 6 $Jm^{-1}kg^{-1}$.

As shown in Figure 5.1, two clusters can clearly be identified for **WalkMan**. We call them W_095 and W_08 in the remainder of this work. W_095 and W_08 respectively converged to an average speed of around 0.95 m/s and 0.8 m/s. W_08 features a slightly better minimization of its metabolic energy consumption. Therefore, we selected a sample from W_08, corresponding to a speed of around 0.8 m/s and a metabolic energy consumption of around 10 $Jm^{-1}kg^{-1}$, to further analyze the results for WalkMan in sections 5.2.3 and 5.3. Both gaits are presented in Figure 5.12 and 5.13.

As for WalkMan, two clusters can be identified for **Atlas** in Figure 5.1. We call them A_105 and A_095 in the remainder of this work. A_105 and A_095 respectively converged to an average speed of around 1.05 m/s and 0.95 m/s. Both clusters are represented by two sets of parameters. A_105 features a better minimization of its metabolic energy consumption. Therefore, we chose a sample from A_105, corresponding to a speed of around 1.05 m/s and a metabolic energy consumption of around 7 $Jm^{-1}kg^{-1}$, to further analyze the results in sections 5.2.4 and 5.3. Both gaits are presented in Figure 5.17 and 5.18.

5.1.2 Four-stage fitness function

As explained in section 4.5, it is possible to constrain the optimizer to converge toward a target walking speed. In order to analyze the evolution of the robots gaits across a speed range, we ran the following optimizations (we decided not to analyze the evolution of the NAO gait across a speed range since it was impossible to get a converging speed constrained optimization for this robot on a sufficient range) :

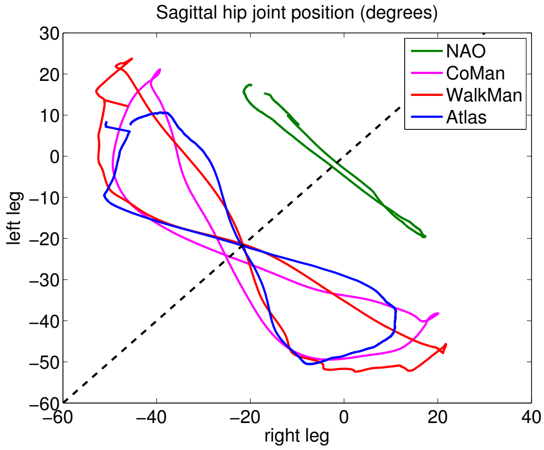
- CoMan : speed constrained optimizations with the following target speeds : 0.5, 0.6, 0.7, 0.8 and 0.9 m/s
- WalkMan : speed constrained optimizations with the following target speeds : 0.6, 0.7, 0.8, 0.9, 1.0 and 1.1 m/s
- Atlas : speed constrained optimizations with the following target speeds : 0.6, 0.7, 0.8, 0.9, 1.0 and 1.1 m/s

Four similar and independent optimizations are launched for each target speed.

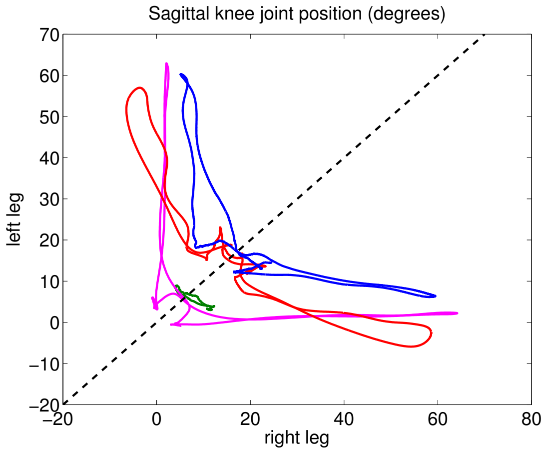
The results of these speed constrained optimizations are analyzed in section 5.4.

5.2 Robots gaits analysis

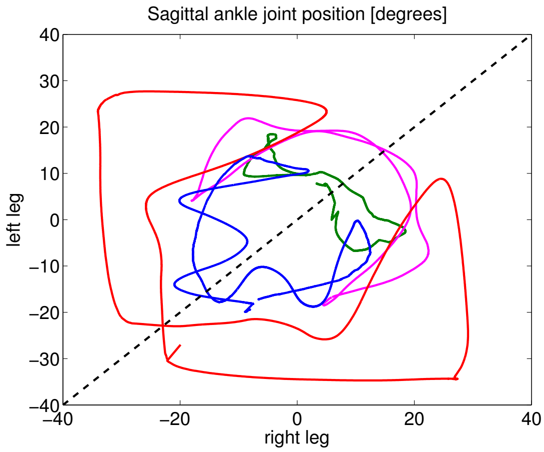
In this section, we analyze the gaits produced by the four sets of parameters selected in section 5.1.1.
For the sake of conciseness, we will focus on the robots right leg to analyze their gait. Anyway, Figure 5.2 shows that the left leg kinematics is symmetric to the right leg one.



(a) Kinematics of the hip sagittal joints



(b) Kinematics of the knee sagittal joints



(c) Kinematics of the ankle sagittal joints

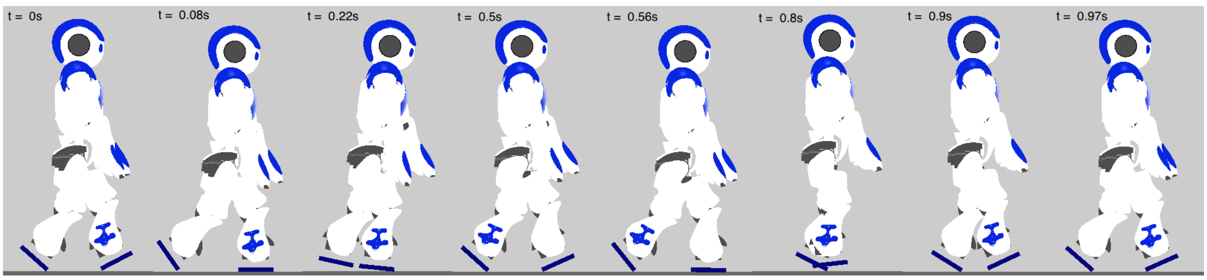
Figure 5.2: Left leg kinematics as a function of the right leg one. The data are averaged over 10 gait cycles.

5.2.1 NAO

Figure 5.3 shows nine snapshots of the simulated NAO with the optimal set of parameters found in section 5.1.1. A time frame of 0.97 s is spanned, corresponding to one gait cycle starting at right *heel strike* (and so, finishing at the next right *heel strike*). These snapshots correspond to the nine moments presented in Figure 4.1 of section 4.1. Several observations can be made :

- The trunk is never tilted. This is most likely due to the special pelvis kinematic design described in section 2.1 and to the very small mass of its torso (see Figure 2.19).
- The *heel rise* event (Figure 5.3c) happens very quickly after the *opposite toe push off* (Figure 5.3b) event. Consequently, NAO seems to adopt a gait on tiptoes.
- NAO does not bend his knee enough during the swing phase. This can clearly be observed in Figure 5.3f.

The last point highlights one of the big problems we encountered with NAO. The majority of the optimizations results were giving either a gait with straight legs, or a gait with swing feet scraping the ground. We tried to modify the control rules for the knee sagittal DOF to force NAO to bend his knees during the swing phase but the results were very poor (the optimizer was simply trying to reduce the effect of the additional knee reflexes). With more time we could add a stage to the fitness function guiding the optimizer toward solutions with bended knees during the swing phase.
As a reminder, the joints of NAO are stiff joints actuated with DC motors. As stated in section 2.2, the use of compliant joints can offer several advantages regarding locomotion. With more time, it would have been interesting to equip NAO with compliant joints (i.e., with serie elastic actuators) and see the optimization results.
Another characteristic of NAO is its very small mass (5 kg). This makes the transitions between stance and swing phases more difficult to detect (see section 4.3.2). More carefully tuned swing/stance thresholds and ground contact model could benefit the controller.



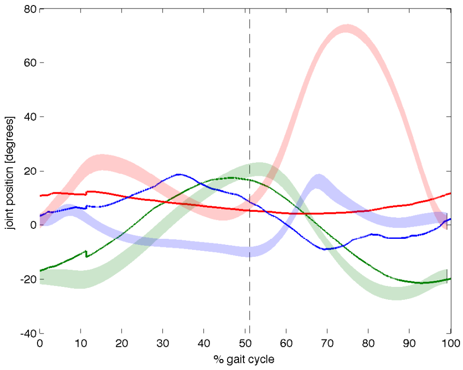
(a) 0% (b) 8% (c) 22% (d) 51% (e) 57% (f) 82% (g) 90% (h) 100%

Figure 5.3: Snapshots of NAO in the Robotran simulation environment. The caption corresponds to the percentage of the gait cycle.

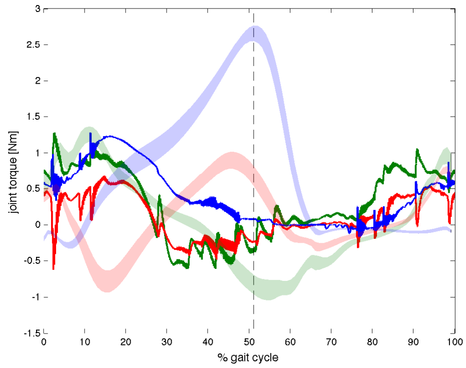
Leg sagittal joints positions, torques and mechanical powers

Figure 5.4 presents the positions, torques and mechanical powers for the right leg sagittal joints of NAO. Sagittal joints positions are compared with ground truth human data at 1.25 m/s (shaded areas) in Figure 5.4a. The hip joint position curve for NAO is quite similar to the human data. The remarks concerning the knees and the *heel rise* event can again be observed in this figure. The knee is straight during the whole gait cycle and the *heel rise* happens quickly after the *heel strike*.

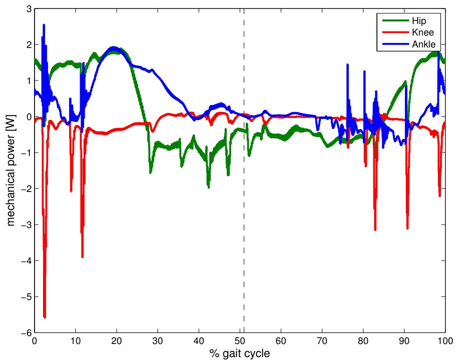
To compare the sagittal joints torques of the NAO with the human ones (Figure 5.4b), we multiplied the human data by the dynamic scaling factor for torques, derived in section 4.3.1 (Table 4.2). As for the joints positions, only the hip joint torque curve for NAO is quite similar to the human data. Note that the imperfections of the curves (i.e., the curves width seems to be bigger due to fast oscillations around the real value) are due to a too high time step and have no physical meaning. NAO already has a simulation time step of 125 μs compared to the time step of 500 μs of the other robots. This is mainly due to the DC motor equations.



(a) Sagittal joints positions [degree]



(b) Sagittal joints torques [Nm]



(c) Sagittal joints mechanical powers [W]

Figure 5.4: Positions, torques and mechanical powers for the right leg sagittal joints of NAO. The graphs start at right *heel strike* and span over a gait cyle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles. The shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. For qualitative comparison, these data are multiplied by a dynamic scaling factor, derived in Table 4.2 of section 4.3.1.

Metabolic and electric energy consumption

Figure 5.5 compares the metabolic and electric energy consumption of NAO. The electric energy consumption is proportional to $\sum_i T_i^2$ where T_i is the electromagnetic torque generated by the DC motor i of NAO. Indeed, as presented in section 3.1.2, $T = k\phi I_a$ for each motor.
Then,

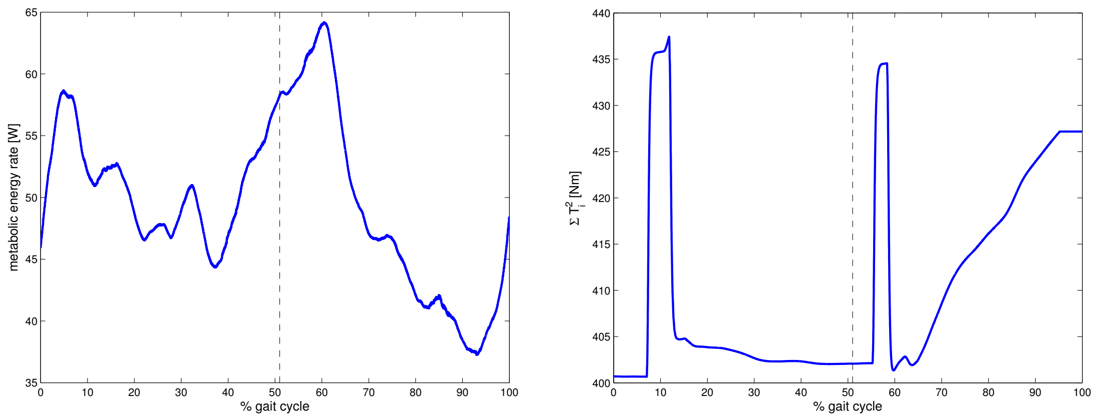
$$T^2 = (k\phi)^2 I_a^2 = C_1 I_a^2$$

where C_1 is a constant. And

$$P = R I_a^2$$

Since R is a constant, the electric power and T^2 are proportionally related.
In both figures 5.5a and 5.5b, we can see two major peaks occuring at around 10% and 60%

of the gait cycle. This corresponds to the moment just after heel strike, where huge energy is required to transfer the body weight to the other leg and prevent the robot from collapsing.



(a) Metabolic energy rate

(b) Electric energy rate

Figure 5.5: Qualitative comparison between the metabolic power consumed by the virtual muscles implemented for NAO and the electric power consumed by its joints actuators, during one gait cycle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles.

Ground reaction forces

Figure 5.6 shows the vertical ground reaction forces acting under the feet of NAO during one gait cycle. The right foot (in stance phase), supports 100% of the body weight, until the left *heel strike* (represented by the vertical dashed line) after which the body weight is transferred to the left leg.

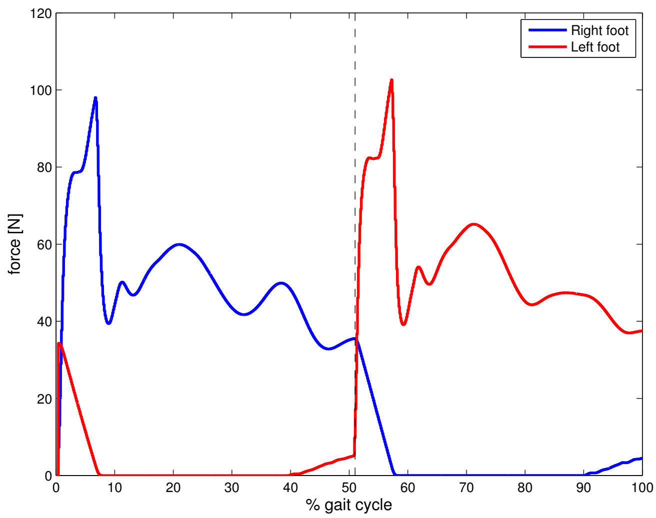
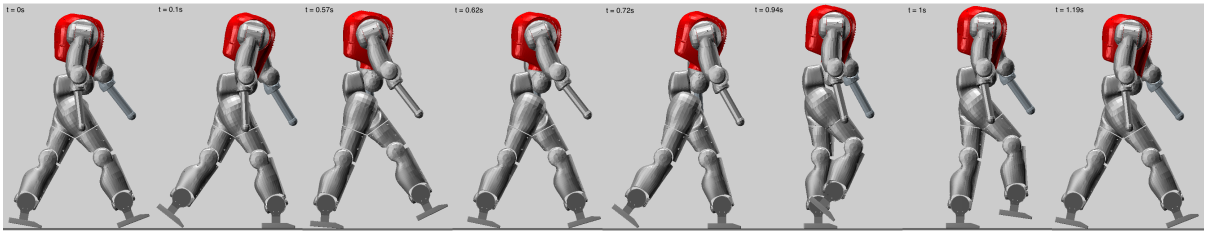


Figure 5.6: Vertical ground reaction forces acting under the feet of NAO during one gait cycle. The Vertical dashed line represent the left *heel strike*. The data are averaged over 10 gait cycles.

5.2.2 CoMan

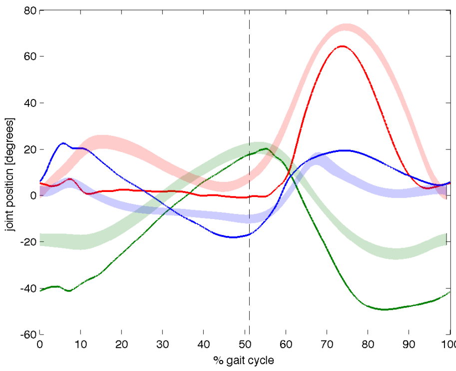
Figure 5.7 shows nine snapshots of the simulated CoMan with the optimal set of parameters found in section 5.1.1. A time frame of 1.19 s is spanned, corresponding to one gait cycle starting at right *heel strike*. These snapshots correspond to the nine moments presented in Figure 4.1 of section 4.1.



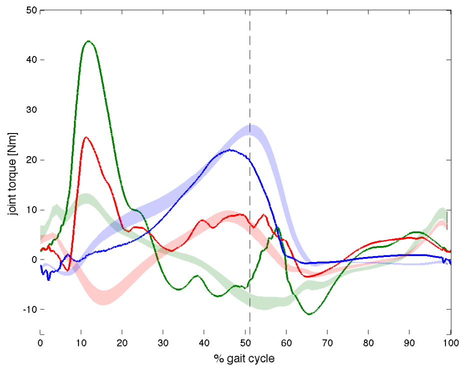
(a) 0% (b) 8% (c) 48% (d) 52% (e) 60% (f) 79% (g) 84% (h) 100%

Figure 5.7: Snapshots of CoMan in the Robotran simulation environment. The caption corresponds to the percentage of the gait cycle.

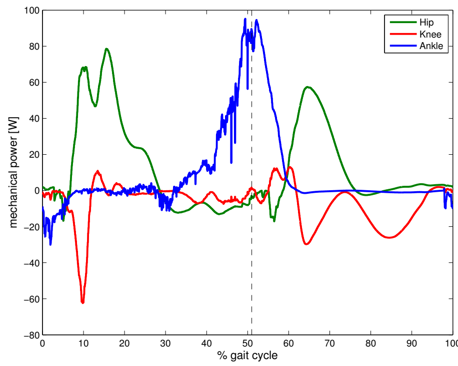
Leg sagittal joints positions, torques and mechanical powers



(a) Sagittal joints positions [degree]



(b) Sagittal joints torques [Nm]



(c) Sagittal joints mechanical powers [W]

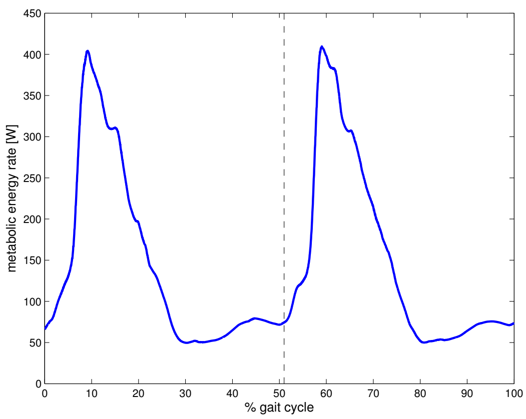
Figure 5.8: Positions, torques and mechanical powers for the right leg sagittal joints of CoMan. The graphs start at right *heel strike* and span over a gait cyle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles. The shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. For qualitative comparison, these data are multiplied by a dynamic scaling factor, derived in Table 4.2 of section 4.3.1.

Figure 5.8 presents the positions, torques and mechanical powers for the right leg sagittal joints of CoMan. Sagittal joints positions are compared with ground truth human data at 1.25 m/s (shaded areas) in Figure 5.8a. The hip, knee and ankle joints positions curves for CoMan are very similar to the human data. However, for the CoMan, the knee is kept a bit straight during the *loading response* period, just after *heel strike*. The risk is knee hyperextension.

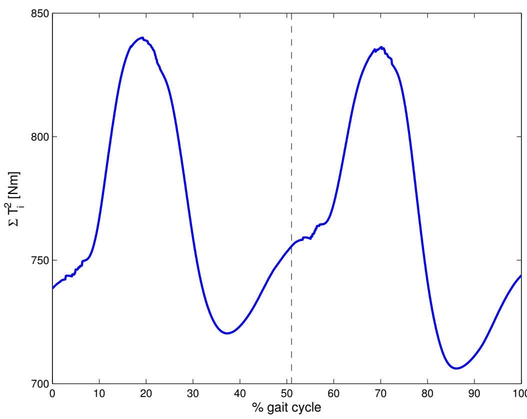
To compare the sagittal joints torques of the CoMan with the human ones (Figure 5.8b), we multiplied the human data by the dynamic scaling factor for torques, derived in section 4.3.1 (Table 4.2), as done for NAO in the previous section. The torque curves follow a similar evolution except for the knee at the beginning of the gait cycle. For CoMan, the positive knee peak torque at 10% counteracts the knee hyperextension just after the *heel strike* while for human, the negative knee peak torque at 10% counteracts the knee flexion happening after the *heel strike*. As for humans, we can observe two major torque peaks occurring at 15% and 50% of the gait cycle. The first one is due to the effort produced by the supporting hip to move the body forward. The second one is due to the effort produced by the ankle to propel the robot forward.

The qualitative matching between the sagittal joints position and torques of the CoMan and the human corroborates the human-like gait of Figure 5.7.

Metabolic and electric energy consumption



(a) Metabolic energy rate



(b) Electric energy rate

Figure 5.9: Qualitative comparison between the metabolic power consumed by the virtual muscles implemented for CoMan and the electric power consumed by its joints actuators, during one gait cycle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles.

As done in section 5.2.1, the metabolic and electric energy consumption of CoMan are compared in Figure 5.9. Again, we use $\sum_i T_i^2$ to represent the evolution of the electric power (T_i being the torque generated by the SEA i of CoMan).
In both figures 5.9a and 5.9b, we can see two major peaks occurring at around 15% and 70% of the gait cycle. These peaks are related to the efforts produced by the hip joints, during the *mid-stance* and the *initial swing*, to

- propel the swing leg forward
- maintain the trunk stable after the strike of the opposite leg (muscles HAM and GLU of

the new stance leg act a lot just after the *strike* of the former stance leg to prevent the trunk from collapsing).

The hip muscles seem to have more impact on the metabolic energy than the ankle muscles. Indeed, the ankle torque and mechanical power peaks at 50% appearing in Figures 5.8b and 5.8c do not appear in Figure 5.9a. This can be explained by the fact that the hip muscles are the most heavy ones. Each muscle consuming an energy proportional to its mass, the heavier the muscle, the more impacted the metabolic energy.

Ground reaction forces

Figure 5.10 shows the vertical ground reaction forces acting under the feet of CoMan during one gait cycle. The right foot (in stance phase), supports 100% of the body weight, until the left *heel strike* (represented by the vertical dashed line) after which the body weight is transferred to the left leg.

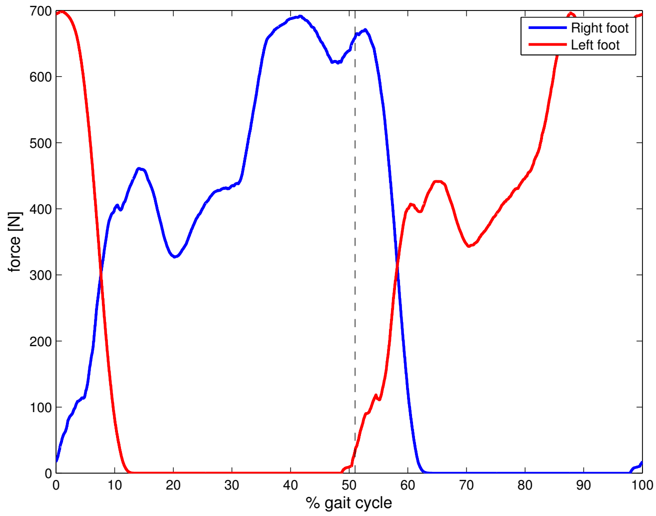


Figure 5.10: Vertical ground reaction forces acting under the feet of CoMan during one gait cycle. The Vertical dashed line represent the left *heel strike*. The data are averaged over 10 gait cycles.

It is interesting to compare the ground reaction forces profile of CoMan (Figure 5.10) with the human one (Figure 5.11). Both present two maxima at around 15 % and 50%.

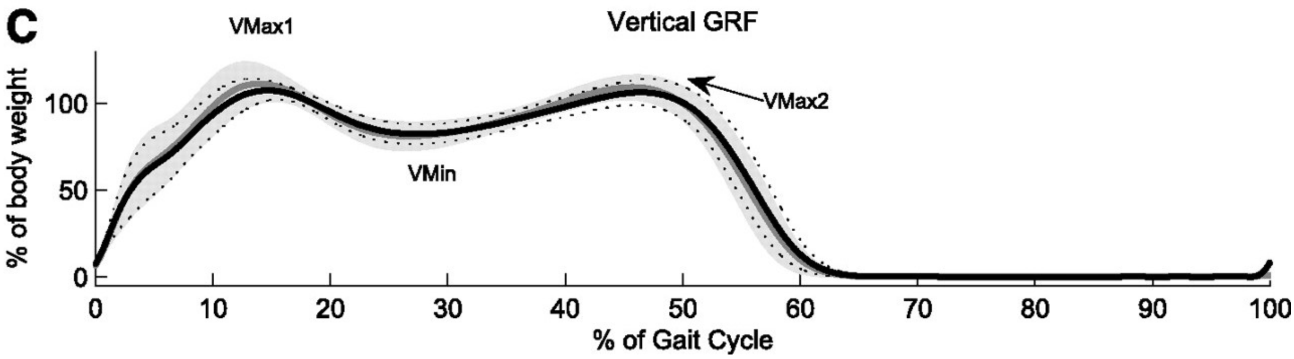
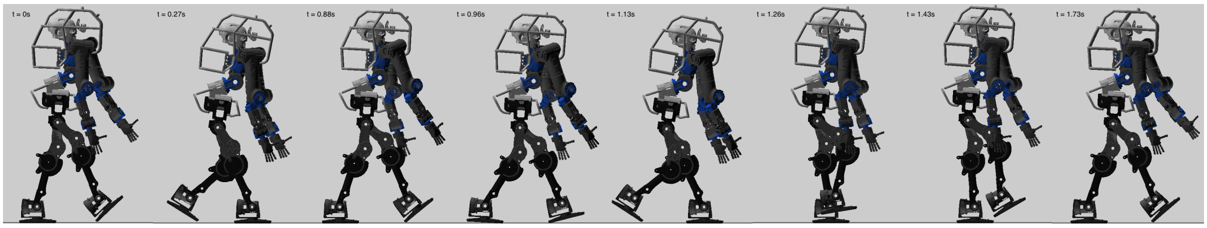


Figure 5.11: Average vertical ground reaction forces acting on a human's foot in stance phase. [13]

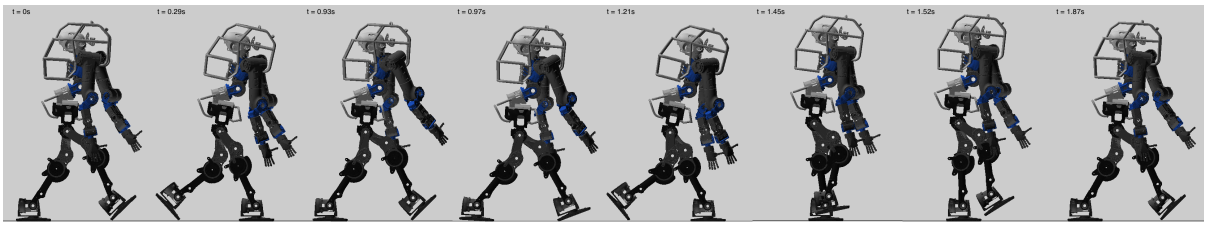
5.2.3 WalkMan

As explained in section 5.1.1, two clusters (W_08 and W_095) were identified among the optimized solutions. Figures 5.12 and 5.13 show nine snapshots of the simulated WalkMan with the optimal set of parameters corresponding to the best sample of W_08 and W_095, respectively. These figures correspond to one gait cycle starting at right *heel strike*. These snapshots correspond to the nine moments presented in Figure 4.1 of section 4.1.
The same observation can be made for both gaits : unlike the human gait, the *opposite heel strike* event (c) happens just before the *heel rise* (d) event.
In both gaits, WalkMan seems to bend too much forward compared to a human gait. This is not really the case. Indeed, as a reminder from section 2.3, WalkMan has a special torso lateral joint which gives him this aspect.



(a) 0% (b) 15% (c) 50% (d) 55% (e) 65% (f) 72% (g) 82% (h) 100%

Figure 5.12: Snapshots of WalkMan, walking with the W_08 controller, in the Robotran simulation environment. The caption corresponds to the percentage of the gait cycle.



(a) 0% (b) 15% (c) 49% (d) 51% (e) 64% (f) 77% (g) 81% (h) 100%

Figure 5.13: Snapshots of WalkMan, walking with the W_095 controller, in the Robotran simulation environment. The caption corresponds to the percentage of the gait cycle.

The W_08 and W_095 controllers allow the WalkMan to walk at 0.8 m/s and 0.95 m/s respectively. The stride period is longer with W_095 (1.87s) than with W_08 (1.73s). The stride length is bigger with W_095 (1.9m) than with W_08 (1.4m). With W_08, WalkMan walk with shorter steps but with a higher step frequency giving him a more human-like gait than with W_095.

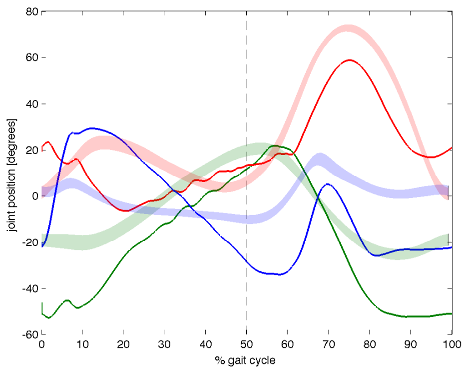
W_08 is the reference controller for WalkMan in the remainder of this chapter.

Leg sagittal joints positions, torques and mechanical powers

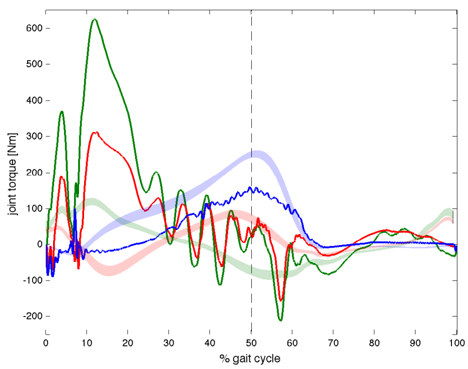
Figure 5.14 presents the positions, torques and mechanical powers for the right leg sagittal joints of WalkMan. Sagittal joints positions are compared with ground truth human data at 1.25 m/s (shaded areas) in Figure 5.14a. The hip, knee and ankle joints positions curves for WalkMan are pretty similar to the human data. However, for the WalkMan, the knee is kept a bit straight during the *loading response* period, just after *heel strike*. A bending knee during this period would lead to a smoother gait and would eliminate the risk of knee hyperextension.

To compare the sagittal joints torques of the WalkMan with the human ones (Figure 5.14b), we multiplied the human data by the dynamic scaling factor for torques, derived in section 4.3.1 (Table 4.2), as done for the other robots in the previous sections.
The ankle torque curve follows an evolution similar to the one of humans. However, the knee and the hip torque curves present large oscillations in the *mid-stance* and *terminal stance* periods. This is due to the biarticular muscle HAM (acting on the knee and the hip, but not on the ankle). As stated in Appendix D, the HAM is controlled, in stance phase, by a stimulation resulting from a proportional-derivative controller on the trunk sagittal angle. The trunk is shaking a bit during the *mid-stance* and *terminal stance* periods because these are single support phases. The proportional-derivative gains, tuned through the optimization phase, are maybe too high and make the system instable during these periods. This has a very light impact on the joints positions (see Figure 5.14a).
As for CoMan in the previous section, a positive knee peak torque occurs at 10% and counteracts the knee hyperextension just after the *heel strike* while for human, a negative knee peak torque occurs at 10% and counteracts the knee flexion happening after the *heel strike*.

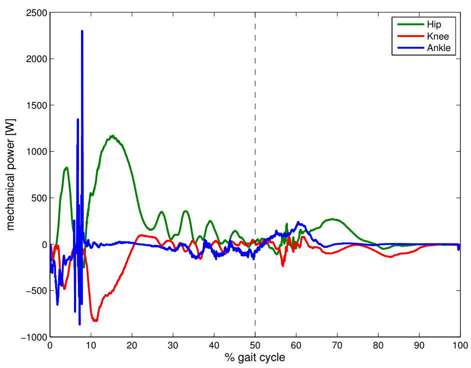
As for humans and the other robots so far, we can observe two major torque peaks occurring at 15% and 50% of the gait cycle. Again, the first one is due to the effort produced by the supporting hip to move the body forward. The second one is due to the effort produced by the ankle to propel the robot forward.



(a) Sagittal joints positions [degree]



(b) Sagittal joints torques [Nm]

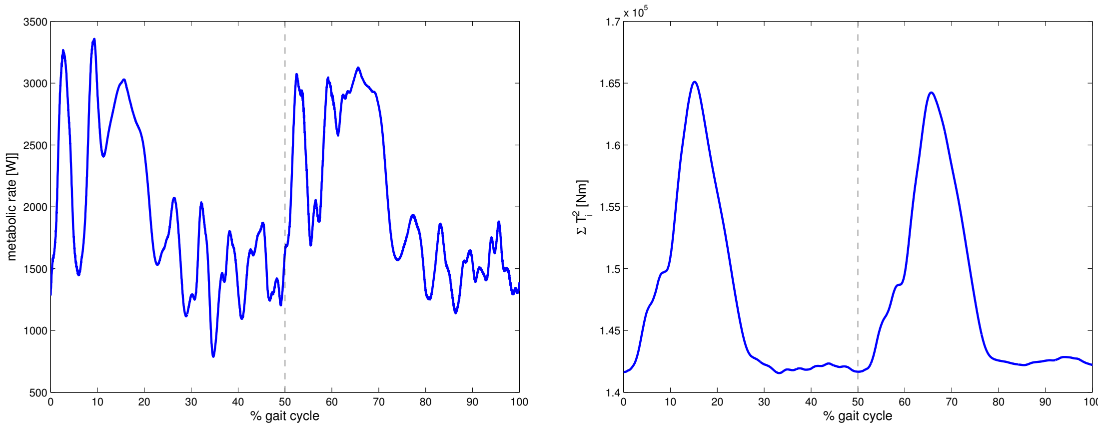


(c) Sagittal joints mechanical powers [W]

Figure 5.14: Positions, torques and mechanical powers for the right leg sagittal joints of WalkMan. The graphs start at right *heel strike* and span over a gait cyle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles. The shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. For qualitative comparison, these data are multiplied by a dynamic scaling factor, derived in Table 4.2 of section 4.3.1.

Metabolic and electric energy consumption

As done in section 5.2.1 and 5.2.2, the metabolic and electric energy consumption of WalkMan are compared in Figure 5.15. Again, we use $\sum_i T_i^2$ to represent the evolution of the electric power (T_i being the torque generated by the SEA i of WalkMan).
In both figures 5.15a and 5.15b, we can see two major peaks occurring at around 15% and 70% of the gait cycle. These peaks are related to the efforts produced by the hip joints during the *mid-stance* and the *initial swing* periods (see section 5.2.2).



(a) Metabolic energy rate (b) Electric energy rate

Figure 5.15: Qualitative comparison between the metabolic power consumed by the virtual muscles implemented for WalkMan and the electric power consumed by its joints actuators, during one gait cycle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles.

Ground reaction forces

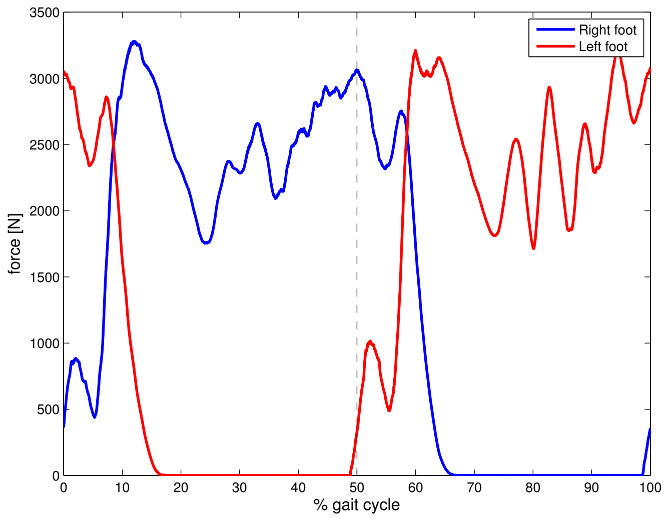
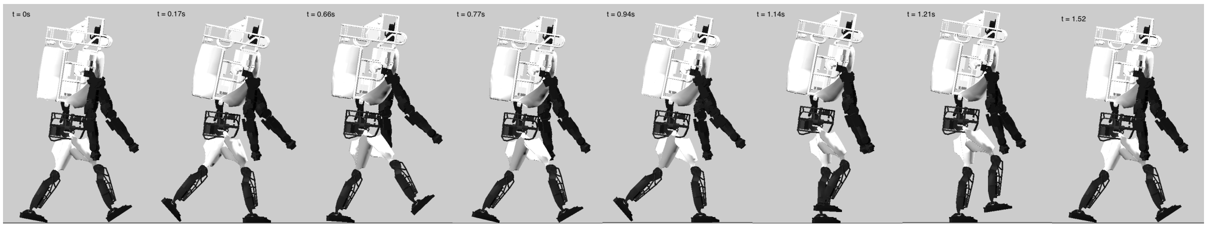


Figure 5.16: Vertical ground reaction forces acting under the feet of WalkMan during one gait cycle. The Vertical dashed line represent the left *heel strike*. The data are averaged over 10 gait cycles.

Figure 5.16 shows the vertical ground reaction forces acting under the feet of WalkMan during one gait cycle. The right foot (in stance phase), supports 100% of the body weight, until the *heel rise* after which the body weight is transferred to the left leg. Indeed, as written above, the *opposite heel strike* (Figure 5.12c) happens just before the *heel rise* (Figure 5.12d). The body weight transfer is then delayed of around 10% of the gait cycle compared to human gait. This is clear in Figure 5.16, the blue curve decreases at 60% while the left *heel strike* happens at 50%. Again, it is interesting to observe the two maxima at 15 % and 50% in both Figures 5.16 and 5.11.

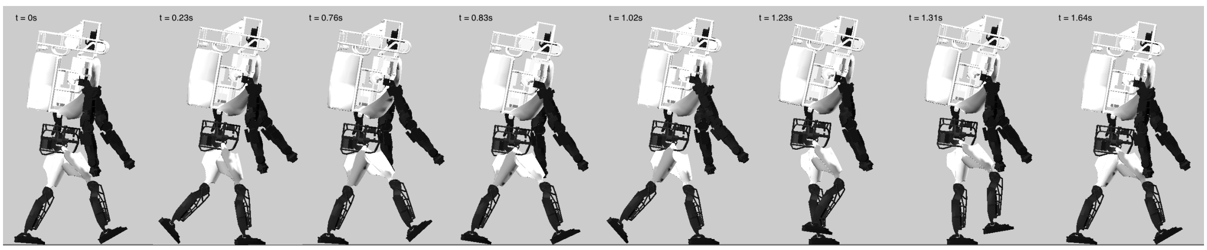
5.2.4 Atlas

As explained in section 5.1.1, two clusters (A_105 and A_095) were identified among the optimized solutions. Figures 5.17 and 5.18 show nine snapshots of the simulated Atlas with the optimal set of parameters corresponding to the best sample of A_105 and A_095, respectively. These figures correspond to one gait cycle starting at right *heel strike*. These snapshots correspond to the nine moments presented in Figure 4.1 of section 4.1.



(a) 0% (b) 11% (c) 43% (d) 50% (e) 61% (f) 75% (g) 79% (h) 100%

Figure 5.17: Snapshots of Atlas, walking with the A_105 controller, in the Robotran simulation environment. The caption corresponds to the percentage of the gait cycle



(a) 0% (b) 14% (c) 46% (d) 50% (e) 62% (f) 75% (g) 79% (h) 100%

Figure 5.18: Snapshots of Atlas, walking with the A_095 controller, in the Robotran simulation environment. The caption corresponds to the percentage of the gait cycle

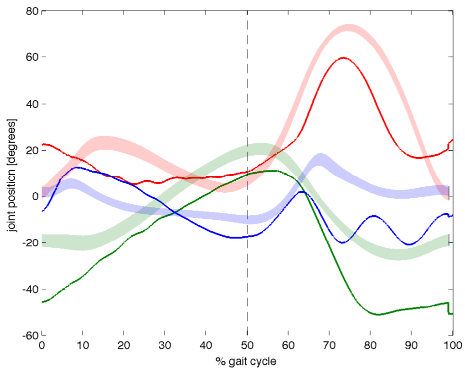
Both gaits are very similar regarding the attitude and the timing. The A_105 and A_095 controllers allow the Atlas to walk at 1.05 m/s and 0.95 m/s respectively. The stride period is a bit longer with A_095 (1.64s) than with A_105 (1.52s). The stride length is a bit bigger with A_105 (1.65m) than with A_095 (1.55m).

A_105 is the reference controller for Atlas in the remainder of this chapter.

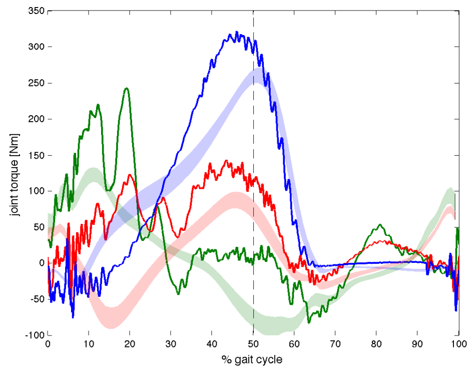
Leg sagittal joints positions, torques and mechanical powers

Figure 5.19 presents the positions, torques and mechanical powers for the right leg sagittal joints of Atlas. Sagittal joints positions are compared with ground truth human data at 1.25 m/s (shaded areas) in Figure 5.19a. The hip, knee and ankle joints positions curves for Atlas are pretty similar to the human ones. However, for the Atlas, the ankle position curve presents oscillations during the *terminal swing* period. This is simply due to the low-level controller (Figure 3.4 of section 3.1.2), associated with the the ankle joints actuators, which was not tuned carefully enough. The ankle being very light, these oscillations have no impact on the quality of the gait.

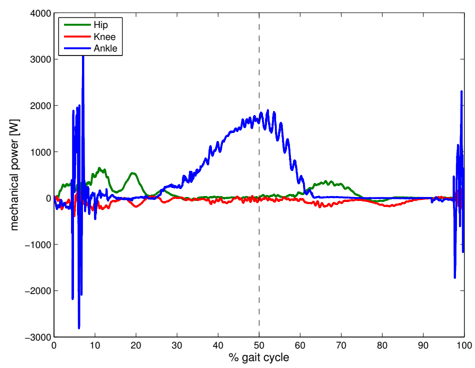
To compare the sagittal joints torques of the Atlas with the human ones (Figure 5.19b), we multiplied the human data by the dynamic scaling factor for torques, derived in section 4.3.1 (Table 4.2), as done for the other robots in the previous sections.
The hip, knee and ankle torque curves follow an evolution similar to the one of humans. As for humans and the other robots, we can observe two major torque peaks occurring at 15% and 50% of the gait cycle and respectively corresponding to the efforts produced, to move the body forward, by the supporting hip and the pushing-off ankle.



(a) Sagittal joints positions [degree]



(b) Sagittal joints torques [Nm]



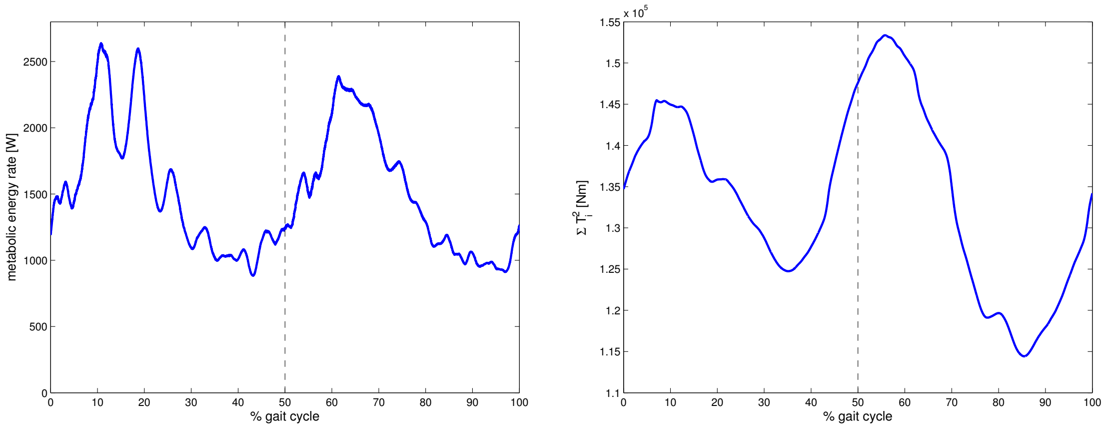
(c) Sagittal joints mechanical powers [W]

Figure 5.19: Positions, torques and mechanical powers for the right leg sagittal joints of Atlas. The graphs start at right *heel strike* and span over a gait cyle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles. The shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. For qualitative comparison, these data are multiplied by a dynamic scaling factor, derived in Table 4.2 of section 4.3.1.

Metabolic and electric energy consumption

As done in section 5.2.1, 5.2.2 and 5.2.3, the metabolic and electric energy consumption of Atlas are compared in Figure 5.20. Again, we use $\sum_i T_i^2$ to represent the evolution of the electric

power (T_i being the torque generated by the SEA i of Atlas).[1].
In both figures 5.20a and 5.20b, we can see two major peaks occurring at around 15% and 60% of the gait cycle. These peaks are related to the efforts produced by the hip joints during the *mid-stance* and the *initial swing* periods (see section 5.2.2).



(a) Metabolic energy rate

(b) Electric energy rate

Figure 5.20: Qualitative comparison between the metabolic power consumed by the virtual muscles implemented for WalkMan and the electric power consumed by its joints actuators, during one gait cycle. The vertical dashed lines represent the left *heel strike*. The data are averaged over 10 gait cycles.

Ground reaction forces

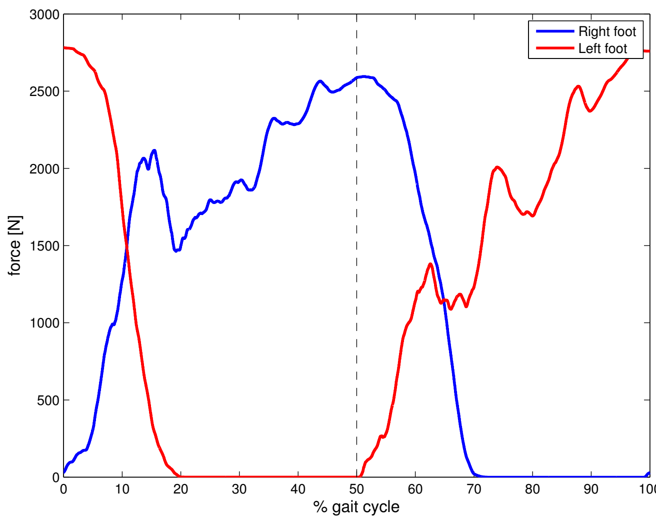


Figure 5.21: Vertical ground reaction forces acting under the feet of Atlas during one gait cycle. The Vertical dashed line represent the left *heel strike*. The data are averaged over 10 gait cycles.

Figure 5.21 shows the vertical ground reaction forces acting under the feet of Atlas during one gait cycle. The right foot (in stance phase), supports 100% of the body weight, until the left *heel strike* (represented by the vertical dashed line) after which the body weight is transferred to the

[1] As a reminder, the real Atlas is not equipped with SEA. Having no information on its hydraulic actuators, we equipped his joints with SEA in section 3.1.2

left leg.
Again, it is interesting to observe the two maxima at 15 % and 50% in both Figures 5.21 and 5.11.

5.3 Comparison at optimal speed

In this section, we compare the reference controller of each robot with human data and analyze the results with the dynamic scaling theory.

Hip sagittal joints positions and torques

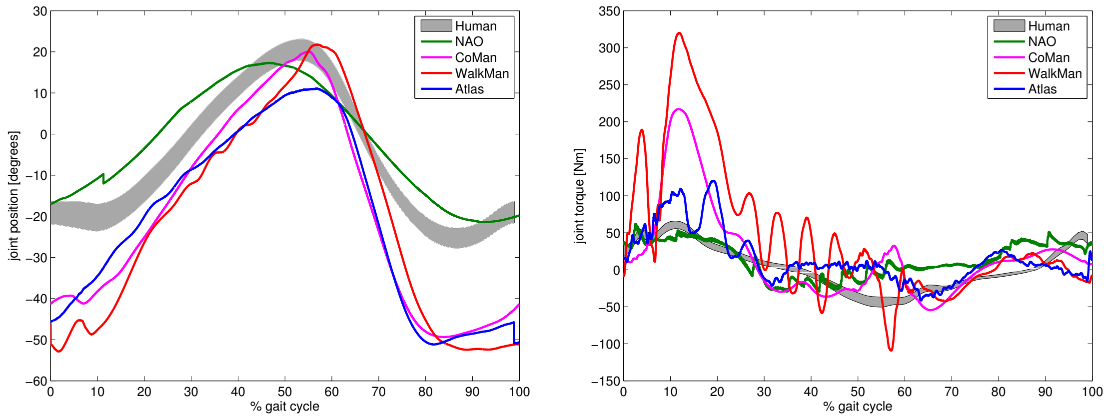
Figure 5.22a shows the qualitative matching between NAO, CoMan, WalkMan, Atlas and the human, regarding the positions of the right hip sagittal joints over a gait cycle.

Figure 5.22b compares the hip sagittal joints torques of the different robots. The robots torques are divided by the dynamic scaling factor derived in section 4.3.1 so that a "quantitative" comparison is possible. In other words, for each robot, the torque is transferred on a human scale by dividing the torque value by the corresponding ratio in the last column of Table 5.1.

Figure 5.22b shows that the hip sagittal joints torques for NAO and Atlas are very close to the ones for a human. However, they are four or five times larger for CoMan and WalkMan especially at the beginning of the gait cycle, when all the body weight is supported by the right leg.

Table 5.1: Ratios used to transfer the robots torques on a human scale according to the dynamic scaling theory

Robot	L_R/L_H	M_R/M_H	τ_R/τ_H
NAO	0.33	0.0625	0.0625
CoMan	0.52	0.3875	0.3875
WalkMan	1.10	1.77	1.77
Atlas	0.94	2.15	2.15



(a) Hip sagittal joints positions (b) Hip sagittal joints torques

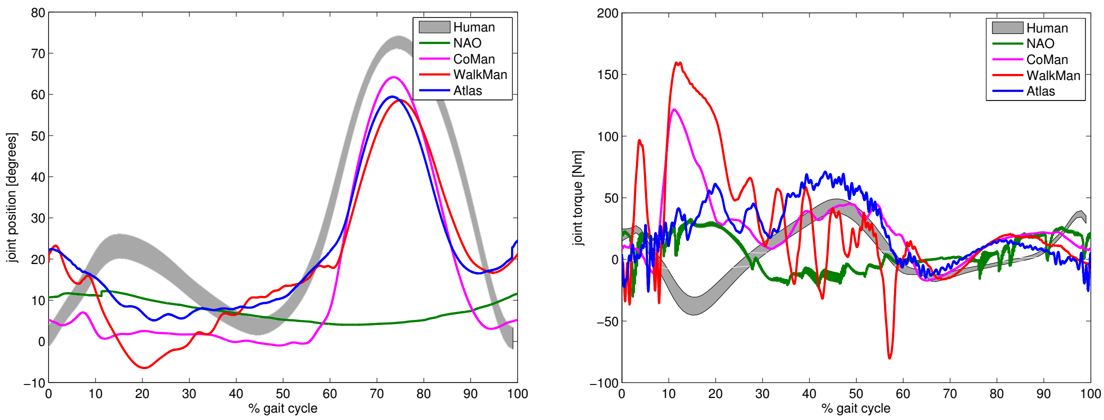
Figure 5.22: Positions and torques for the right hip sagittal joints of NAO, CoMan, WalkMan and Atlas. The graphs start at right *heel strike* and span over a gait cycle. The data are averaged over 10 gait cycles. The grey shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. In (b), the robots torques are multiplied by the dynamic scaling factor $(M_H L_H)/(M_R L_R)$, see Table 5.1.

Knee sagittal joints positions and torques

Figure 5.23a shows the qualitative matching between CoMan, WalkMan, Atlas and the human, regarding the positions of the right knee sagittal joints over a gait cycle. As written above, the robots knees are too straight during the *loading response* and the *mid-stance* periods compared to the human gait.

Figure 5.23b compares the knee sagittal joints torques of the different robots. For each robot, the torque is transferred on a human scale by dividing the torque value by the corresponding ratio in the last column of Table 5.1.

We observe that the knee sagittal joint torques for Atlas are very close to the ones for a human. For CoMan and WalkMan, the knee torque curves are very different from the human ones at the beginning of the gait cycle (during the *loading response* period). They are two or three times larger in amplitude and are of opposite sign compared to the human one. We explained these two torque peaks by the necessity to counteract knee hyperextension, see sections 5.2.2 and 5.2.3. During the swing phase however, the knee sagittal joint torques for CoMan and WalkMan are very close to the ones for a human.



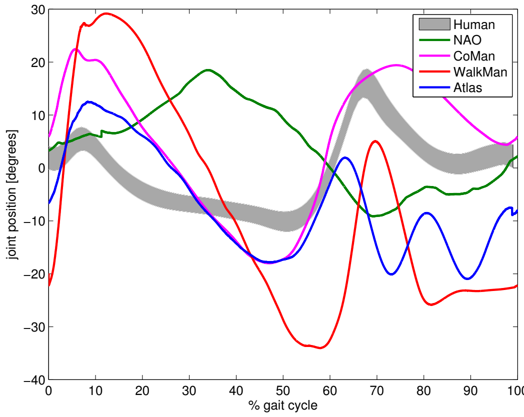
(a) Knee sagittal joints positions (b) Knee sagittal joints torques

Figure 5.23: Positions and torques for the right knee sagittal joints of NAO, CoMan, WalkMan and Atlas. The graphs start at right *heel strike* and span over a gait cycle. The data are averaged over 10 gait cycles. The grey shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. In (b), the robots torques are multiplied by the dynamic scaling factor $(M_H L_H)/(M_R L_R)$, see Table 5.1.

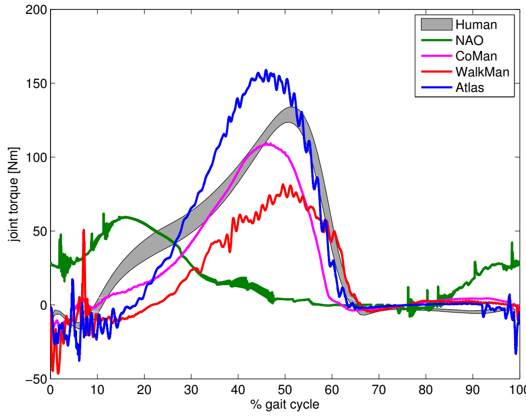
Ankle sagittal joints positions and torques

Figure 5.24a shows the qualitative matching between CoMan, WalkMan, Atlas and the human, regarding the positions of the right ankle sagittal joints over a gait cycle. The different robots present a bigger amplitude in position for the ankle joints (especially WalkMan).

Figure 5.24b compares the ankle sagittal joints torques of the different robots. For each robot, the torque is transferred on a human scale by dividing the torque value by the corresponding ratio in the last column of Table 5.1. We observe that the ankle sagittal joints torques for CoMan, WalkMan and Atlas are very close to the ones for a human, both quantitatively and qualitatively.



(a) Ankle sagittal joints positions



(b) Ankle sagittal joints torques

Figure 5.24: Positions and torques for the right ankle sagittal joints of NAO, CoMan, WalkMan and Atlas. The graphs start at right *heel strike* and span over a gait cyle. The data are averaged over 10 gait cycles. The grey shaded areas represent one standard deviation of the ground truth human data at 1.25 m/s [24]. In (b), the robots torques are multiplied by the dynamic scaling factor $(M_H L_H)/(M_R L_R)$, see Table 5.1.

Metabolic energy consumption

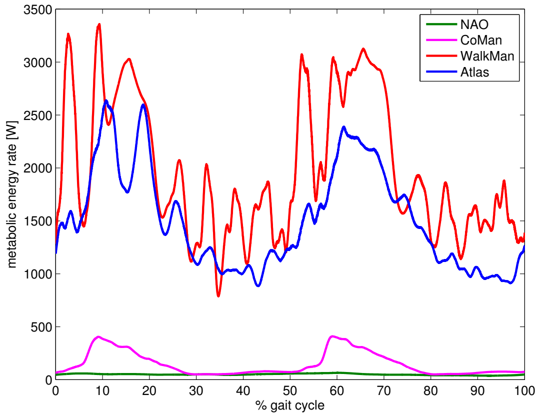
In the previous sections, the general comment about the metabolic energy consumption of the robots was the presence of two peaks occurring at 15% and 60 %. We explained these peaks by the efforts produced by the hip joints, during the *mid-stance* and the *initial swing*, to

- propel the swing leg forward
- maintain the trunk stable after the strike of the opposite leg (muscles HAM and GLU of the new stance leg act a lot just after the *strike* of the former stance leg to prevent the trunk from collapsing).

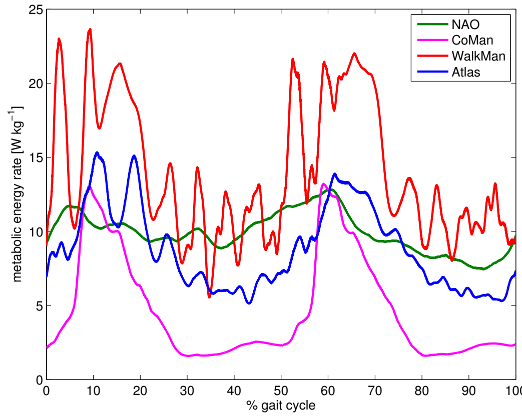
Figure 5.25 compares the metabolic energy rates over a gait cycle for NAO, CoMan, WalkMan and Atlas. It is difficult to compare the energy-efficiencies of the four robots based on the raw data (Figure 5.25a). Then, the energy rate of each robot is normalized by its mass (Figure 5.25b) or by the product of its mass and height (Figure 5.25c). This is, somehow, a dynamic scaling.

Figure 5.25c shows that NAO is not energy-efficient compared to CoMan, WalkMan and Atlas, whose energy-efficiencies are proportionally close. The inefficient gait of NAO was already observed in Figure 5.3.

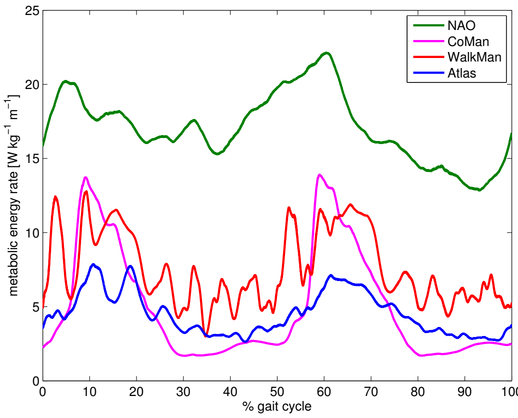
If the height of the robot is taken into account (5.25c), then Atlas is the most energy efficient. If height is not taken into account (5.25b), then CoMan is more energy efficient than Atlas.



(a) Metabolic energy rates



(b) Metabolic energy rates normalized by the mass



(c) Metabolic energy rates normalized by the mass·height product

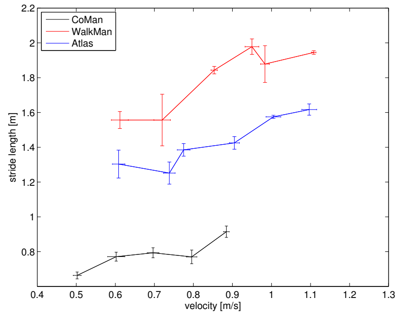
Figure 5.25: Metabolic energy rates over a gait cycle for NAO, CoMan, WalkMan and Atlas. (b) is normalized by the mass and (c) is normalized by the mass · height product. The data are averaged over 10 gait cycles.

5.4 Comparison across a speed range

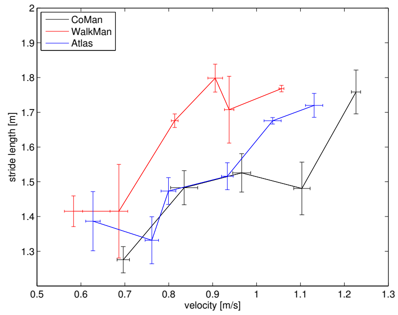
As written in section 5.1.2, we want to analyze the evolution of the gaits across a speed range, for CoMan, WalkMan and Atlas. NAO giving very bad results with speed constrained optimizations, we decided not to analyze the evolution of its gait.

CoMan was optimized for target speeds ranging from 0.5 m/s to 0.9 m/s with a step of 0.1 m/s. WalkMan and Atlas were optimized for target speeds ranging from 0.6 m/s to 1.1 m/s with a step of 0.1 m/s.

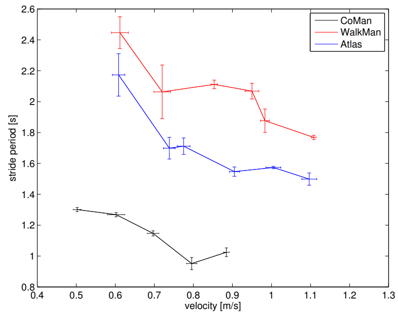
For each target speed, four similar and independent optimizations were performed. The evolutions of the gaits in terms of stride period, stride length and energy efficiency are presented in Figure 5.26.



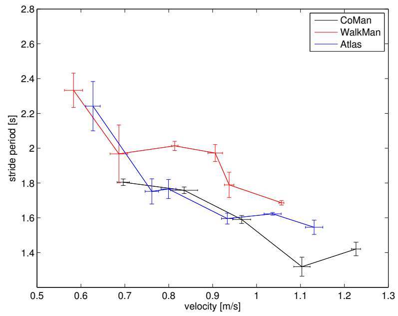
(a) Stride length



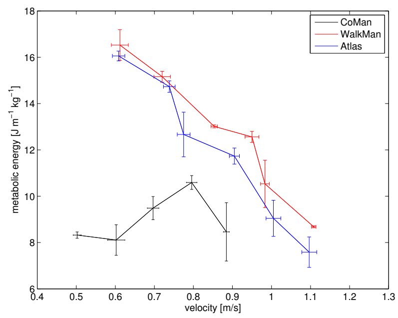
(b) Stride length



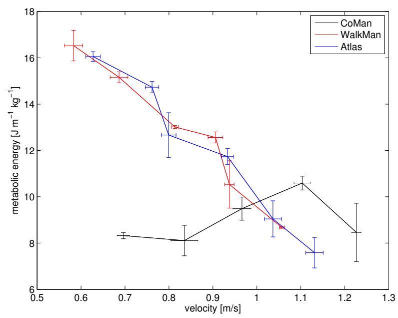
(c) Stride period



(d) Stride period



(e) Metabolic energy per unit distance walked



(f) Metabolic energy per unit distance walked

Figure 5.26: For each target speed, we ran four optimizations with CoMan, WalkMan and Atlas. For each set of four optimizations, the mean and the standard deviation are depicted with errorbars. The errorbars correspond to the half of the standard deviation, for graph legibility. (a), (c) and (e) are the real data. In (b), (d) and (f), the speeds are scaled by a factor V_H/V_R, in (b), the stride lengths are scaled by a factor L_H/L_R and in (d), the stride periods are scaled by a factor T_H/T_R. See Table 5.2 for the values of these scaling factors.

Figures 5.26a and 5.26c show the evolution of the stride length and period across a range of speeds, for the three studied robots. As expected, the robots increase their stride length and frequency (i.e., decreasing stride period) in order to walk faster.

To quantitatively compare these values, we use the dynamic scaling theory (see section 4.3.1). In Figures 5.26b and 5.26d, the robots speeds, stride periods and stride lengths are transferred on a human scale :

- The speeds are multiplied by V_H/V_R.
- The stride periods are multiplied by T_H/T_R
- The stride lengths are multiplied by L_H/L_R

The dynamic scaling factors for each robot are listed in Table 5.2.

In Figures 5.26b and 5.26d, we can observe that WalkMan features higher stride periods and higher stride lengths, so favoring bigger steps with a slower frequency. This is most likely due to the length of its legs (i.e., 20 cm longer than the ones of Atlas).

Regarding energy efficiency (Figures 5.26e and 5.26f), the metabolic energy consumption decreases with increasing speed, for WalkMan and Atlas. However, the curve for CoMan is not as we expected. Indeed, the same experiments in [21] leads to a decreasing metabolic energy consumption with increasing speed.
Moreover, 0.8 m/s (i.e., the optimal speed found for CoMan in section 5.1.1) is the less efficient speed. The four optimizations for this target speed most likely got trapped in poor local optima. This is proved by Figure 5.1a which shows that it is possible to get lower energy consumption, at some of the requested speeds.

Table 5.2: Ratios used to scale the muscle properties for the size and the mass of the different robots

Robot	L_R/L_H	M_R/M_H	V_R/V_H	T_R/T_H
NAO	0.33	0.0625	0.5744	0.57
CoMan	0.52	0.3875	0.7211	0.72
WalkMan	1.10	1.77	1.048	1.04
Atlas	0.94	2.15	0.9695	0.96

Chapter 6

Conclusion

Context

Implementing a dynamic walking controller is not an easy task. Lots of controller have been developed so far but none of them made a legged humanoid robot walk with performances (in terms of energy-efficiency, balance and speed) comparable to the ones of the humans.

The present work aimed at generalizing an existing bio-inspired dynamic walking controller to a variety of robots having different scales and morphologies.

Since it is unpractical to work directly on the real robots, the first step was to develop software robot models simulating somehow the physical behavior of the additional robots and their environment. To this end, we developed a tool generating robot models and compatible with the Robotran simulation environment.

Thanks to these models and the simulation environment, we were able to optimize and validate the adapted controller for the three additional robots.

Simulation results

Most simulation results focus on the analysis and comparison of the optimal 2D walking gaits obtained for NAO, CoMan, WalkMan and Atlas, at the end of the optimization phase.

For CoMan, WalkMan and Atlas, the results were very interesting. For each robot, the optimizations converged to human-like gaits with straight knees and with real *heel strike* and *push-off* events. The comparison between the leg sagittal joints positions of each robot shows a qualitative and quantitative matching. These positions are also very similar to the leg sagittal joints positions of humans. The same observations can be made for most of the leg sagittal joints torques. According to the dynamic scaling theory, these torques are qualitatively and quantitatively matching. To sum up, the optimizations for CoMan, WalkMan and Atlas resulted in proportionally comparable human-like gaits.

For NAO, the results were not as positive as for CoMan, WalkMan and Atlas. The optimizations resulted in a dynamic walking gait with straight knees and with real *heel strike* and *push-off* events. However, the knees are kept straight during the whole gait cycle and the metabolic energy consumption is absolutely not minimized. We do not see clear reasons to this failure. Some possible reasons might be related to the robot mass, its morphology (including joints configurations), its motors or even a more difficult gait initiation, affecting the rest of the gait.

Another interesting result is the qualitative matching between the metabolic energy rate and the electric energy rate for the four robots.

We also studied the evolution of the gaits across a range of speeds for CoMan, WalkMan and Atlas and observed an increasing energy-efficiency with an increasing speed.

Limits and suggested improvements

To complete this conclusion, we identify the limits of our work and suggest several improvements.

Firstly, we optimized our controllers in simulation without taking into account the multiple non-idealities and external perturbations of the real world. Many developments could be realized to improve the robustness of our controllers.

Only the 2D walking gait was adapted to NAO, WalkMan and Atlas. It would be interesting to adapt the 3D walking gait controller as well.

The existing controller includes the possibility to modify the speed of the CoMan during its gait with a unique set of optimized parameters. To achieve it, the reflexes used in this thesis are combined with central pattern generators. This feature might be included in our generalized controller.

Bibliography

[1] Aldebaran. Nao - technical overview. `http://doc.aldebaran.com/2-1/family/robots/index_robots.html#all-robots`, May 2016.

[2] A. Bejan. Unifying constructal theory for scale effects in running, swimming and flying. *Journal of Experimental Biology*, 209(2):238–248, Jan. 2006.

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

[4] CEREM. Robotran - symbolic generator of multibody systems. `http://www.robotran.be`, May 2016.

[5] L. Colasanto, N. V. d. Noot, and A. J. Ijspeert. Bio-inspired walking for humanoid robots using feet with human-like compliance and neuromuscular control. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 26–32, Nov. 2015.

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

[7] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim. Optimization based controller design and implementation for the Atlas robot in the DARPA Robotics Challenge Finals. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 1028–1035. IEEE, 2015.

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

[9] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier. The nao humanoid: a combination of performance and affordability. *CoRR abs/0807.3223*, 2008.

[10] F. Heremans, N. Van der Noot, A. J. Ijspeert, and R. Ronsse. Bio-inspired balance controller for a humanoid robot, 2015.

[11] A. V. Hill. The Heat of Shortening and the Dynamic Constants of Muscle. *Proceedings of the Royal Society of London B: Biological Sciences*, 126(843):136–195, Oct. 1938.

[12] IIT. Walk-man - whole-body adaptive locomotion and manipulation. `https://www.walk-man.eu`, May 2016.

[13] S. J. Lee and J. Hidler. Biomechanics of overground vs. treadmill walking in healthy individuals. *Journal of Applied Physiology*, 104(3):747–755, Mar. 2008.

[14] F. Negrello, M. Garabini, M. G. Catalano, J. Malzahn, D. G. Caldwell, A. Bicchi, and N. G. Tsagarakis. A modular compliant actuator for emerging high performance and fall-resilient humanoids. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 414–420. IEEE, 2015.

[15] G. A. Pratt and M. M. Williamson. Series elastic actuators. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 399 – 406, Pittsburgh, PA, Aug. 1995. IEEE.

[16] A. Schepelmann, H. Geyer, and M. Taylor. Development of a Testbed for Robotic Neuromuscular Controllers. In *Robotics: Science and Systems*, 2012.

[17] B. Somers. Course bio-inspirée pour un robot humanoïde [in progress], 2016.

[18] A. Tözeren. *Human Body Dynamics: Classical Mechanics and Human Movement*. Springer Science & Business Media, Apr. 2006.

[19] N. Van der Noot and A. Barrea. Real-time control of a humanoid robot: Dynamic walking using a bio-inspired approach, 2013.

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

[21] N. Van der Noot, A. J. Ijspeert, and R. Ronsse. Biped gait controller for large speed variations, combining reflexes and a central pattern generator in a neuromuscular model. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6267–6274. IEEE, 2015.

[22] M. Vukobratović and B. Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2005.

[23] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM transactions on graphics*, 31(4), 2012.

[24] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM transactions on graphics*, 31(4), 2012.

[25] Y. Xu. *From simulation to reality.* PhD thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2014.

[26] V. M. Zatsiorsky and B. I. Prilutsky. *Biomechanics of skeletal muscles.* Human Kinetics, Champaign, IL, 2012.

Appendix A

URDF to MBS Converter

As discussed in section 3.1, the gait controllers cannot directly be developed on the real robots. A software model describing the kinematics and dynamics of the robots is then required. The Universal Robotic Description Format (URDF) is an XML file format used in ROS which describes robots. However, we would like this model to be supported by Robotran (the simulation environment used throughout this thesis) which is not the case for the URDF format. Therefore, the software model should be represented by another XML format called MBS (MultiBody System). This MBS description can be either manually edited or generated thanks to MBSysPad, a graphical user interface. It is, in any case, a long and tedious work. The development of a tool converting an available description format, containing all the desired data, into an MBS file would be a considerable saving in time.
We will first present the URDF specification. Then it will be compared to the MBS format. After that, the conversion logic will be explained with an emphasis given to the adaptations needed. We will finally validate the converter by testing it with multiple examples.

A.1 Unified Robot Description Format (URDF)

The Unified Robot Description Format (URDF) is an XML format used in ROS[1] that represent a robot model. This specification is quite general, but it cannot describe all robots. The main limitation is that only tree structures can be represented, ruling out all parallel robots. The specification also assumes that the robot only consists of rigid links connected by joints. Flexible elements are thus not supported. Humanoid robots are a good example of robots that can easily be described with the URDF format. The URDF description of a lot of humanoid robots (NAO, Atlas, CoMan and WalkMan included) can be found on the internet.

The specification covers:

- Kinematics and dynamics of the robot
- Collision model of the robot
- Visual representation of the robot

The URDF description of a robot consists of a set of joint elements connecting a set of links together. Because URDF can only represent tree structures, a typical URDF robot description looks like illustrated in Figure A.2.

[1] Robot Operating System (ROS) is a collection of software frameworks for robot software development. For more information, see `http://www.ros.org`

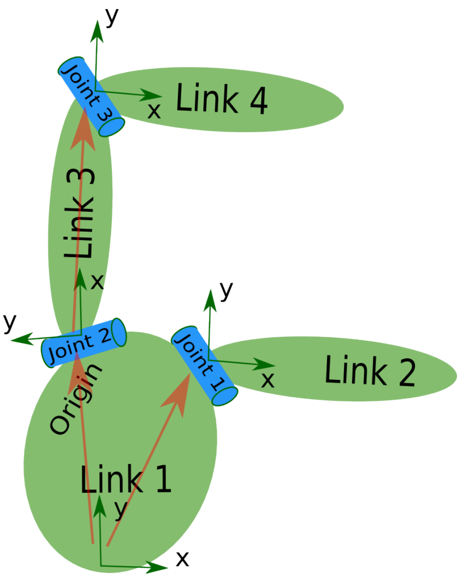
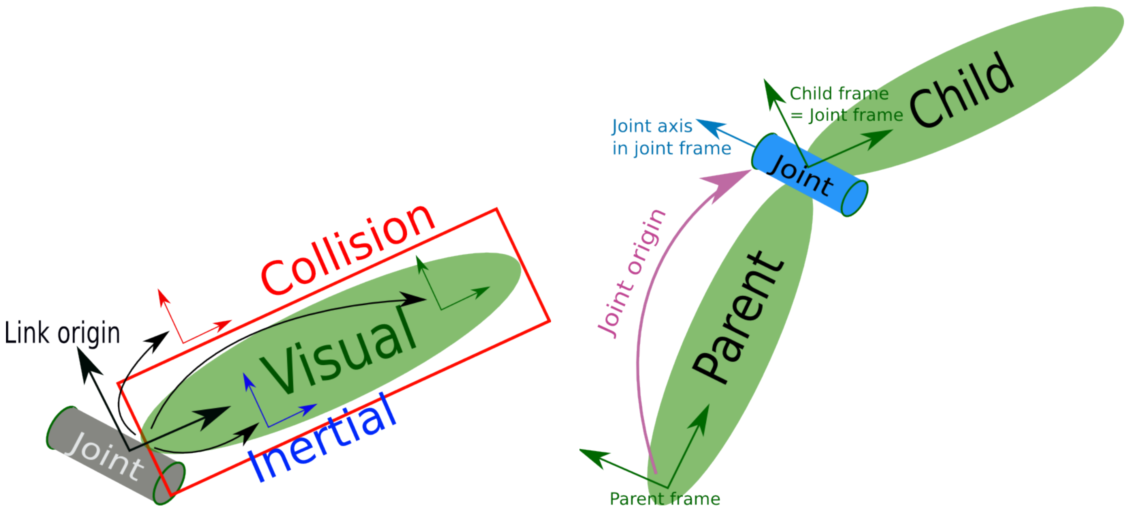


Figure A.1: Basic schematic example of a tree structure URDF specification (Figure taken from `http://wiki.ros.org/urdf`)

A.1.1 Link elements

The inertial, visual and collision properties of each rigid body of the multibody system are described by a link element. As shown in Figure A.2a, inertial, visual and collision properties are defined in 3 different frames. As presented in Listing A.1, three subelements ("inertial", "visual" and "collision") describe the corresponding type of properties.



(a) Schematic example of a link element with its different frames

(b) Schematic example of a joint element with its different frames

Figure A.2: Figures taken from `http://wiki.ros.org/urdf`

Inertial element

The inertial element has three subelements :

- origin : this is the pose of the inertial reference frame, relative to the link reference frame. This subelement has two attributes :
  - xyz : the center of gravity described in the link reference frame

  - rpy : the roll, pitch and yaw angles (in radians) describing the transformation from the link reference frame to the inertial reference frame. This transformation is a "fixed frame" transformation

- mass : the mass of the body
- inertia : the rotational inertia matrix, represented in the inertial frame

Visual element

Two subelements of the visual element are important in this case :

- origin : this is the pose of the visual element with respect to the reference frame of the link. This subelement has two attributes :
  - xyz : the position of the visual element along the x, y and z axis in the reference frame of the link
  - rpy : the roll, pitch and yaw angles (in radians) describing the orientation of the visual element relative to the link reference frame. This transformation is a "fixed frame" transformation
- geometry : the visual appearance of the element. This can be a geometric figure or a mesh file (a Collada .dae file or a even .stl file).

Collision element

This element has the same structure as the visual one and is not of great interest for the rest of this chapter.

Listing A.1: Example of a URDF link element

```
<link name="name">
   <inertial>
     <origin xyz="x y z" rpy="r p y"/>
     <mass value="m"/>
     <inertia ixx="ixx"  ixy="ixy"  ixz="ixz" iyy="iyy" iyz="iyz" izz="izz" />
   </inertial>

   <visual>
     <origin xyz="x y z" rpy="r p y" />
     <geometry>
        <mesh filename="path" scale = "x y z"/>
     </geometry>
   </visual>

   <collision>
     <origin xyz="x y z" rpy="r p y"/>
     <geometry>
       <cylinder radius="r" length="l"/>
     </geometry>
   </collision>
 </link>
```

A.1.2 Joint elements

The kinematics of each joint is described by a joint element whose type can be :

- revolute/continuous : joint that rotates along the axis
- prismatic : sliding joint that slides along the axis

- fixed : not really a joint because it cannot move. All degrees of freedom are locked
- floating : joint allowing motion for all 6 degrees of freedom
- planar : joint allowing motion in a plane perpendicular to the axis

The first three types are the most common. As presented in Listing A.2, a joint element has four important subelements.

- origin : this is the transform from the parent link to the child link. The joint is located at the origin of the child link, as shown in Figure A.2b. It has two attributes :
  - xyz : the anchor point of the joint on the parent link
  - rpy : the roll, pitch and yaw angles (in radians) describing the transformation from the parent link to the child link. This transformation is a "fixed frame" transformation
- parent : the parent link
- child : the child link
- axis : the joint axis specified in the child frame

Listing A.2: Example of a URDF joint element

```
<joint name="name" type="type">
    <origin xyz="x y z" rpy="r p y"/>
    <parent link="parent"/>
    <child link="child"/>
    <axis xyz="x y z"/>
 </joint>
```

A.2 Robotran MultiBody System (MBS)

As written above, the MBS (MultiBody System) format is the one supported by Robotran, the multibody simulation environment used throughout this thesis. Like Robotran, this description format was developed at the Université catholique de Louvain (UCL) within the Center for Research in Mechatronics (CEREM).
MBS files can be generated either manually by editing the XML type file or thanks to a graphical user interface : MBSysPad. MBSysPad allows the user to draw a diagram of segments linked by joints as shown in Figure A.3.
A MBS specification describes a mechanical system composed of rigid bodies that are connected together by joints which can be subject to rotational and/or translational displacements.

The MBS format covers :

- Kinematic and dynamic description of the robot
- Visual representation of the robot

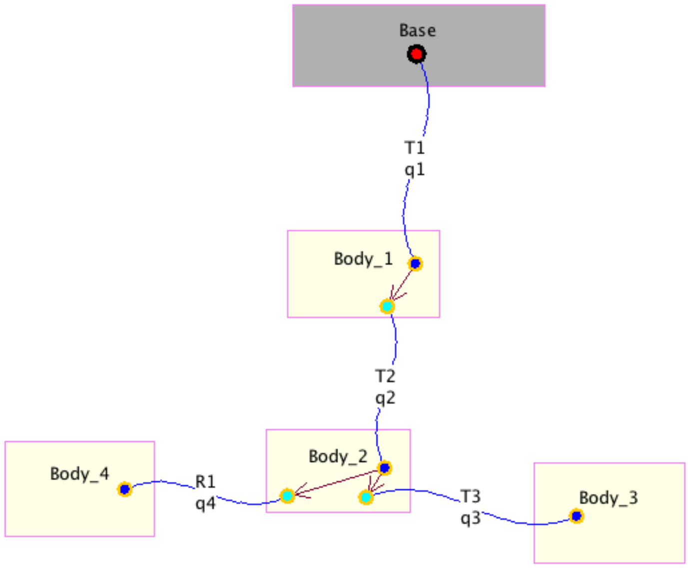


Figure A.3: Example of MBS made with MBSysPad

The MBS description, like the URDF description, consists of a set of bodies connected together by a set of joints in a tree-like structure. Each body has a unique parent body but can have several child bodies. Unlike URDF structure, MBS structure allows a joint to be attached to another joint.
For each body, there is a body-fixed frame. This frame moves with the body and is fixed at the position of the parent joint.
The MBS bodytree is only based on one type of element : the bodies.
A body element generally has eight important subelements :

- bodyname : the name of the body
- parent : the parent of the body. It has two subelements :
  - bodyname : the name of the parent body
  - pointname : the point, on the parent body, where the joint binding both bodies is attached (the anchor point)
- joint : the joint(s) binding the body to its parent. There are four subelements for each joint :
  - jointname : the name of the joint
  - type : the type of joint, revolute (R1, R2, R3) or prismatic (T1, T2, T3)
  - nature : the nature of the joint :
    - dependent
    - independent
    - forced driven
  - graphics : this field has no physical meaning, it defines the way the joint is drawn in the MBSysPad graphical interface
- mass : the mass of the body
- com : the center of mass of the body relative to the body-fixed frame

- inertia : the rotational inertia matrix, represented in the body-fixed frame
- point : the anchor point(s) of the body relative to the body-fixed frame
- graphics : this field has no physical meaning. It has two subelements :
  - x2D : the way the body is drawn in the MBSysPad graphical interface
  - x3D : the visual properties of the body. Some precisions about three fields (the others are trivial)
    * position : the pose of the visual element in the body-fixed frame
    * rotation : the roll, pitch and yaw angles (in radians) describing the orientation of the visual element relative to the body-fixed frame. This transformation is a "current frame" transformation.
    * url : the path leading to the mesh file (a .wrl file).

The root of this bodytree is always a base element whose structure consists of three elements :

- gravity : the x, y and z components of the gravity
- baseBodyName : the name of the base
- graphics : this field has no physical meaning. It has 2 subelements :
  - x2D : the way the base is drawn in the MBSysPad graphical interface
  - x3D : the visual properties of the base

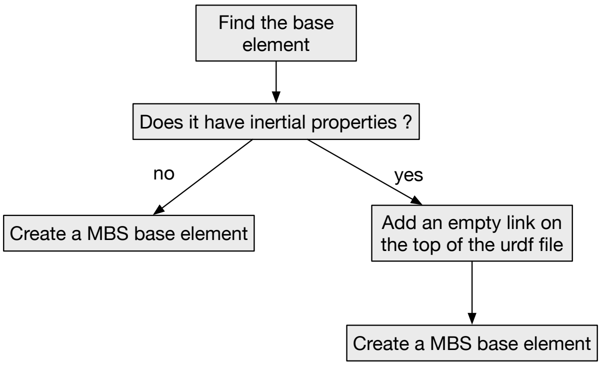
A.3 Python tool

As presented in the two previous sections, URDF and MBS are two different ways of specifying the kinematic and dynamic description of robots (both of them also cover the visual representation of the robot). All the information needed to create a MBS description can be found in the corresponding URDF description but sometimes needs to be adapted due to conventions differences.
We developed a converting tool in Python aiming to create a MBS file, to retrieve the needed data in the URDF file and to translate these data following the MBS conventions.
This converter is as general as possible but it was particularly developed for robots having a tree-like structure, typically for humanoid robots.

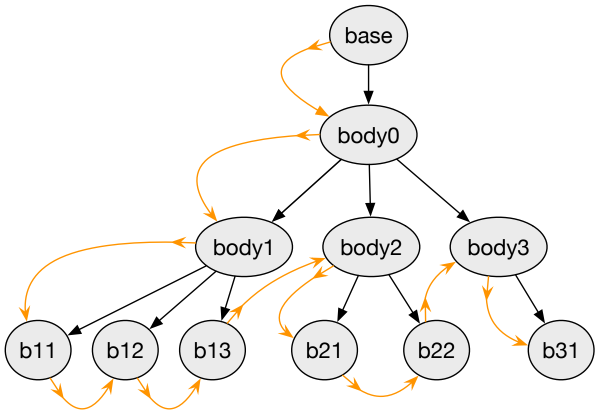
A.3.1 Converter development

Creation of the base element

As written above, the root of a MBS bodytree is always a base element. A proper MBS base element does not have any parent and does not have any inertial property. In the URDF format, the root link sometimes has inertial properties. The converter must, in that case, modify the initial URDF file by adding a URDF link without inertial properties on top of the tree structure. The base creation process is summed up in Figure A.4a.
As stated in section 3.1, the humanoid robots models are based on a floating base representation. The base body is then free to move and this is represented by a 6-DOF joint (three prismatic and three revolute joints about the XYZ axes) between the robot and the world inertial frame. These joints will appear when creating the first body element of the bodytree (the one attached to the base element).



(a) Base creation process



(b) Illustration of the descending order in which the URDF tree structure is browsed

Creation of the bodytree

To create the bodytree, the URDF tree structure is browsed in descending order, in the same way as in Figure A.4b. For each URDF link found (except the null mass links which are neglected), a MBS body is added in the bodytree following the steps below :

1. Retrieve the inertial properties of the URDF link and adapt them to the MBS body conventions

In both descriptions, the mass is expressed in kilograms and the center of mass is described in the body frame. There is then no need to adapt these data.
However, as stated in section A.1.1, the URDF rotational inertia matrix is represented in the inertial frame while the MBS rotational inertia matrix is represented in the body reference frame. The inertial frame is the frame obtained by applying a "roll-pitch-yaw" transformation in "fixed frame" to the body reference frame[2]. Therefore, the components of the MBS rotational inertia matrix need to be adapted.
Considering $\{\hat{I}\}$ and $\{\hat{B}\}$, the inertial and body frames respectively,

$$[\hat{B}] = R[\hat{I}]$$

$$[\hat{I}] = R^T[\hat{B}]$$

R being the transformation matrix for the roll-pitch-yaw transformation in fixed frame.

$$R = R(Z, y)R(Y, p)R(X, r)$$

$$R = \begin{pmatrix} cos(y)cos(p) & -sin(y)cos(r) + cos(y)sin(p)sin(r) & sin(y)sin(r) + cos(y)sin(p)cos(r) \\ sin(y)cos(p) & cos(y)cos(r) + sin(y)sin(p)sin(r) & -cos(y)sin(r) + sin(y)sin(p)cos(r) \\ -sin(p) & cos(p)sin(r) & cos(p)cos(r) \end{pmatrix}$$

Considering $T_{\{\hat{I}\}}$, the rotational inertia matrix described in frame $\{\hat{I}\}$,

$$T_{\{\hat{I}\}} = \begin{pmatrix} i_{xx} & i_{xy} & i_{xz} \\ i_{xy} & i_{yy} & i_{yz} \\ i_{xz} & i_{yz} & i_{zz} \end{pmatrix}$$

then, $T_{\{\hat{B}\}}$, the corresponding rotational inertia matrix described in frame $\{\hat{B}\}$ can be computed as follows :

$$T_{\{\hat{B}\}} = RT_{\{\hat{I}\}}R^T$$

[2]The roll, pitch and yaw angles (in radians) can be found in the URDF link element, under *inertial>origin>rpy*

2. Retrieve information on the joint binding the URDF link to its parent

The next thing the converter does is to check if the transformation between the URDF parent frame and the URDF joint frame has rotation components[3]. If it is the case, then additional "orientation" forced driven joints are needed before the real DOF because the axis of rotation is described in the joint frame (which is equivalent to the child frame as shown in Figure A.2b) and not in the parent frame.

The URDF description gives the angles r (for roll), p (for pitch) and y (for yaw) for the transformation in fixed frame (rotation of r around X, p around Y and y around Z) . However, successive joints in the MBS description correspond to transformations in current frame. Therefore, we need to find the equivalent angles α, β and γ (rotation of α around X, β around Y and γ around Z) for a transformation in current frame.

In fixed frame[4], we have :

$$R = R(Z, y)R(Y, p)R(X, r)$$

$$R = \begin{pmatrix} cos(y)cos(p) & -sin(y)cos(r) + cos(y)sin(p)sin(r) & sin(y)sin(r) + cos(y)sin(p)cos(r) \\ sin(y)cos(p) & cos(y)cos(r) + sin(y)sin(p)sin(r) & -cos(y)sin(r) + sin(y)sin(p)cos(r) \\ -sin(p) & cos(p)sin(r) & cos(p)cos(r) \end{pmatrix}$$

In current frame[5], we have :

$$R = R(X, \alpha)R(Y, \beta)R(Z, \gamma)$$

$$R = \begin{pmatrix} cos(\beta)cos(\gamma) & -sin(\gamma)cos(\beta) & sin(\beta) \\ sin(\alpha)sin(\beta)cos(\gamma) + cos(\alpha)sin(\gamma) & -sin(\alpha)sin(\beta)sin(\gamma) + cos(\alpha)cos(\gamma) & -sin(\alpha)cos(\beta) \\ -sin(\beta)cos(\alpha)cos(\gamma) + sin(\alpha)sin(\gamma) & sin(\beta)sin(\gamma)cos(\alpha) + sin(\alpha)cos(\gamma) & c(\alpha)c(\beta) \end{pmatrix}$$

Using these two matrices it can be stated that :

$$\frac{-r_{23}}{r_{33}} = tan(\alpha) \rightarrow \alpha = atan2(cos(y)sin(r) - sin(y)sin(p)cos(r), cos(p)cos(r))$$

$$\frac{-r_{12}}{r_{11}} = tan(\gamma) \rightarrow \gamma = atan2(cos(r)sin(y) - sin(r)sin(p)cos(y), cos(p)cos(y))$$

$$\frac{r_{13}}{\frac{1}{cos(\alpha)}r_{33}} = tan(\beta) \rightarrow \beta = atan2(sin(r)sin(y) + cos(y)sin(p)cos(r), \frac{1}{cos(\alpha)}cos(p)cos(r))$$

Now, the converter can create the real DOF joint by retrieving the type of the joint and its axis. The translation between the URDF and the MBS joints is trivial and is summed up in Table A.1.

Table A.1

URDF (type and axis)	MBS (type and nature)
Revolute/continuous of axis (1, 0, 0)	R1 independent
Revolute/continuous of axis (0, 1, 0)	R2 independent
Revolute/continuous of axis (0, 0, 1)	R3 independent
Prismatic of axis (1, 0, 0)	T1 independent
Prismatic of axis (0, 1, 0)	T2 independent
Prismatic of axis (0, 0, 1)	T3 independent
Floating base	T1T2T3R1R2R3
Fixed	T3 forced driven to 0[m]
Planar	not considered

[3]If, under the URDF joint element, the field *origin>rpy* has non-zero components

[4]In fixed frame, rotation matrices are pre-multiplied

[5]In current frame, rotation matrices are post-multiplied

However, while MBS joints can only move along/around axis XYZ, URDF joints axis can take a large number of values. Therefore, the converter needs one, three or five MBS joints (depending on the case) to convert one URDF joint. The joints chain respects the following structure :

1. One or two "forced driven" joints to allow the real DOF joint to rotate around the desired axis.
2. One independent joint, the real DOF joint.
3. One or two "forced driven" joints to get back to the joint frame.

The different cases are presented in Table A.2.

Table A.2

URDF axis	MBS type	MBS nature	MBS initial value
(-1, 0, 0)	R3 or R2	forced driven	π
	+ R1	independent	
	+ R3 or R2	forced driven	-π
(0, -1, 0)	R1 or R3	forced driven	π
	+ R2	independent	
	+ R1 or R3	forced driven	-π
(0, 0, -1)	R1 or R2	forced driven	π
	+ R3	independent	
	+ R1 or R2	forced driven	-π
(0, y, z)	R1	forced driven	$atan2(z, y)$
	+ R2	independent	
	+ R1	forced driven	-$atan2(z, y)$
(x, 0, z)	R2	forced driven	$atan2(x, z)$
	+ R3	independent	
	+ R2	forced driven	-$atan2(x, z)$
(x, y, 0)	R3	forced driven	$atan2(y, x)$
	+ R1	independent	
	+ R3	forced driven	-$atan2(y, x)$
(x, y, z)	R3	forced driven	$atan2(y, x)$
	+ R2	forced driven	-$atan2(z, x)$
	+ R1	independent	
	+ R2	forced driven	$atan2(z, x)$
	+ R3	forced driven	-$atan2(y, x)$

3. Retrieve the coordinates of the anchor points

The anchor points of the MBS body are the anchor points specified in the URDF joint elements whose parent link is the corresponding URDF link.

4. Retrieve the visual properties of the URDF link and adapt them to the MBS body conventions

In both description, the mesh file, the position, the scaling and the orientation of the visual element are given.

- The mesh file simply needs to be converted (by the user) from .dae/.stl to .wrl.
- For both description, the position and the scaling are described in the body-fixed frame.

- The orientation is described :
  - by three fixed frame rotations (angle r around X, angle p around Y and angle y around Z) for the URDF description.
  - by three current frame rotations (rotation of α around X, β around Y and γ around Z) for the MBS description.

  To find the angles α, β and γ equivalent to the angles r, p and y, the reasoning is exactly the same as in the second step. This gives :

$$\alpha = atan2(cos(y)sin(r) - sin(y)sin(p)cos(r), cos(p)cos(r))$$

$$\gamma = atan2(cos(r)sin(y) - sin(r)sin(p)cos(y), cos(p)cos(y))$$

$$\beta = atan2(sin(r)sin(y) + cos(y)sin(p)cos(r), \frac{1}{cos(\alpha)}cos(p)cos(r))$$

5. Give 2D graphical properties to each MBS element

Each body is represented by a rectangle box in MBSysPad. To obtain an organized and easy to handle diagram in MBSysPad, the position of each rectangle box is assigned in such a way that it never overlaps an other box. To do so, each body is given (x,y) coordinates in an imaginary two-dimensional array. The base body is located in the upper-left cell ((0,0)).
Considering Figure A.5, the (x,y) coordinates evolution is as follows :

- orange arrows (parent to child) : (x,y) $\rightarrow$ (x, y+1)
- purple arrows (brother to brother) : (x,y) $\rightarrow$ (x+1, y)
- green arrows (nephew to uncle) : (x,y) $\rightarrow$ (x+1, y-1)

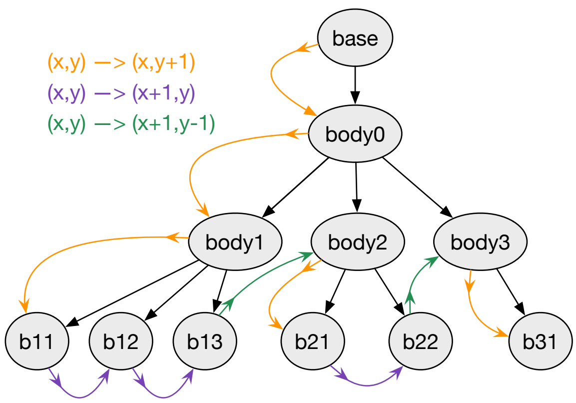


Figure A.5: Attribution rule for the bodies graphical positions in MBSysPad

This gives the result of Figure A.6.

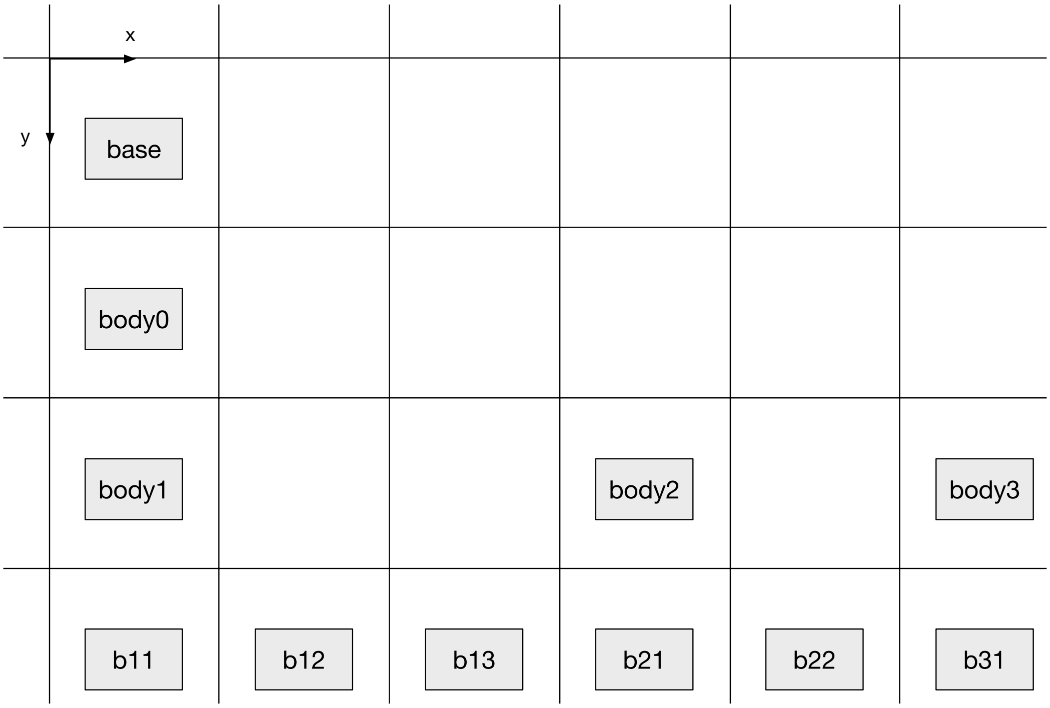


Figure A.6: Typical graphical layout created by the converter for the tree-like structure example of Figure A.5

A.3.2 Converter validation

In order to test and validate the developed converter, we checked the outputs produced by the Python program for four different inputs (the URDF files of robots Atlas, NAO, WalkMan and iCub).
In Figures A.7, A.9, A.12 and A.14, we see that, for all the robots tested, the graphical interface is organized as presented in Figure A.6. This is clear for the user who can rearrange it afterwards (e.g., by drawing a human-like diagram for the sake of clarity, see section 3.1).
In Figures A.8, A.10, A.13 and A.15, we see that the 3D Model of MBSysPad looks coherent with the expected aspects of the different robots.

Results for Atlas :

The URDF file describing the Atlas robot was found on the internet : `https://bitbucket.org/osrf/drcsim/src/583228ebd72677c36a393b3260f0c5f0bc7b0f87?at=default`

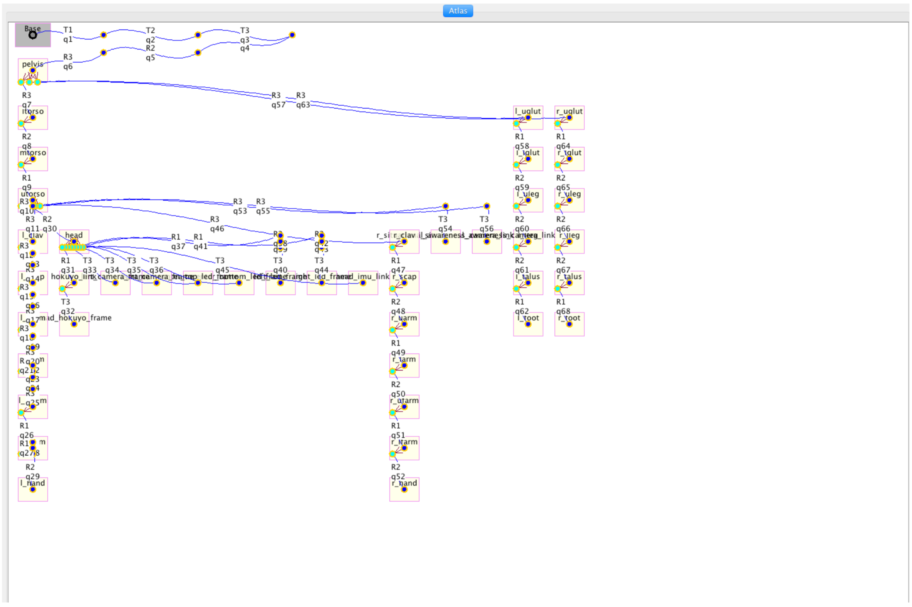


Figure A.7: Screenshot of the MBSysPad graphical interface for the generated Atlas MBS file

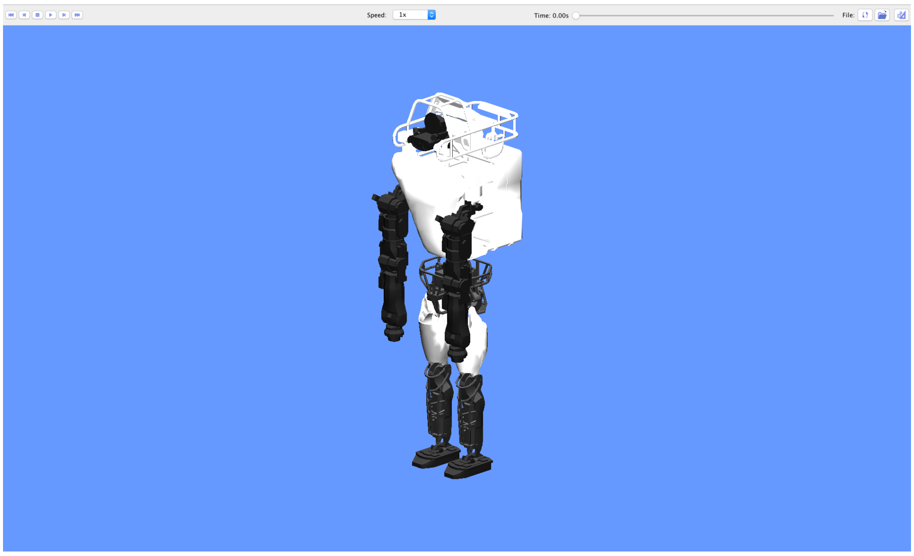


Figure A.8: Screenshot of the MBSysPad Animation 3D Model for the generated Atlas MBS file

Results for NAO :

The URDF file describing the NAO robot was found on the internet : `https://github.com/ros-naoqi/nao_robot/blob/master/nao_description/urdf/naoV50_generated_urdf/nao.urdf`

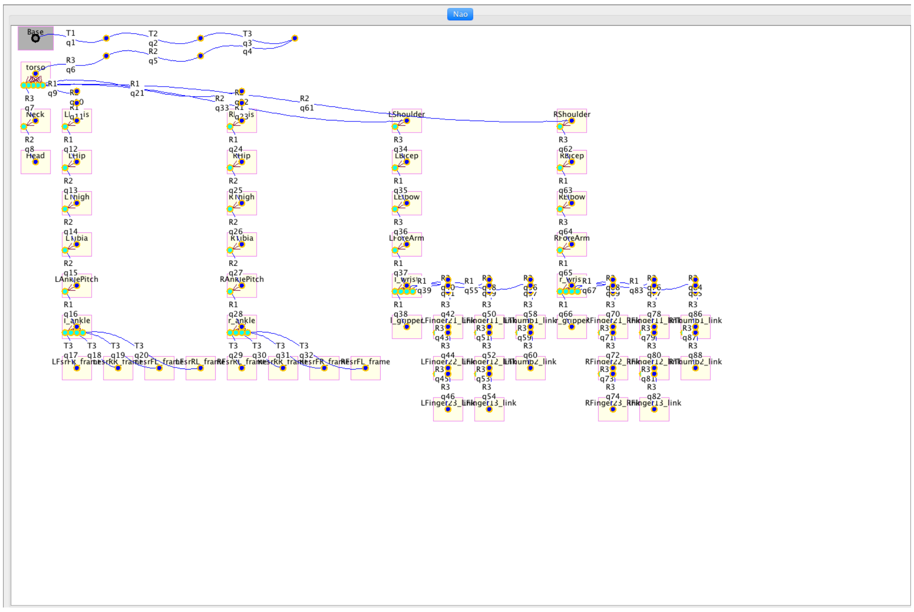


Figure A.9: Screenshot of the MBSysPad graphical interface for the generated NAO MBS file

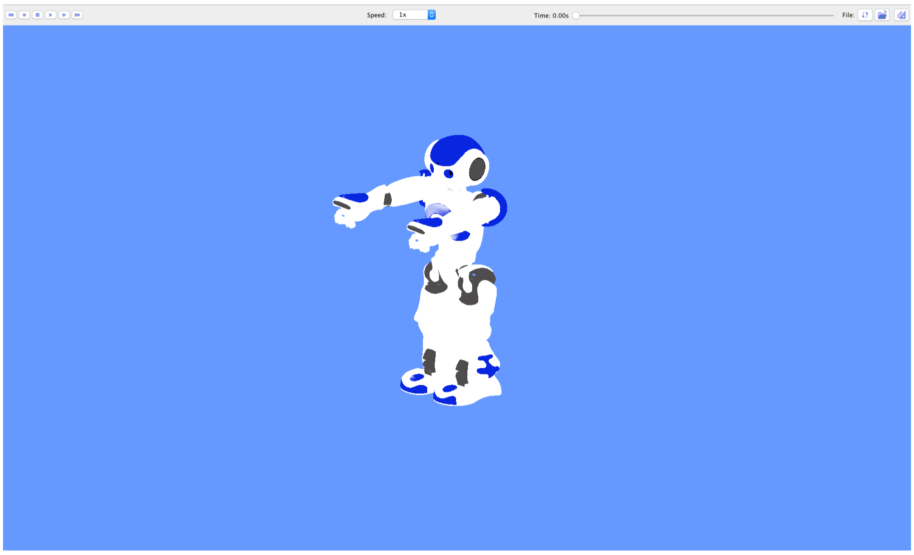
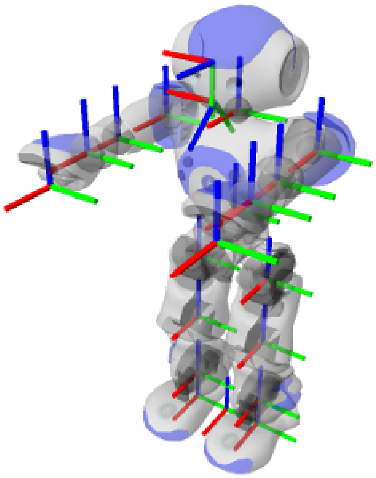
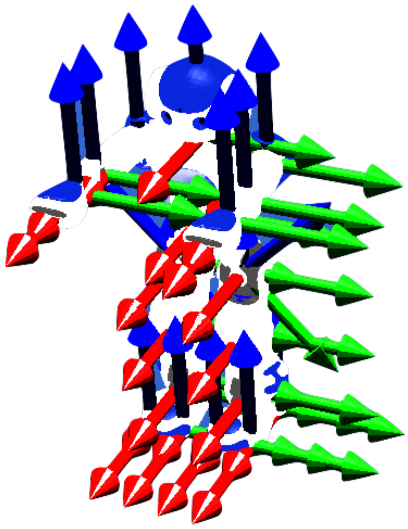


Figure A.10: Screenshot of the MBSysPad Animation 3D Model for the generated NAO MBS file

As an additional check, Figure A.11 compares the expected NAO model (provided by ROS) and the output of the converter. The joint frames definitions are exactly the same.



(a) Expected NAO model provided by ROS (figure taken from `http://wiki.ros.org/nao_description`)



(b) Screenshot of the MBSysPad Animation 3D Model for the generated NAO MBS file with apparent joint frames

Figure A.11: Comparison between the expected NAO model and the output of the converter

Results for WalkMan :

The URDF file describing the WalkMan robot was found on the internet : `https://gitlab.robotran.be/walkman/walkman_robotran/blob/master/dataR/urdf%20files/bigman.urdf`

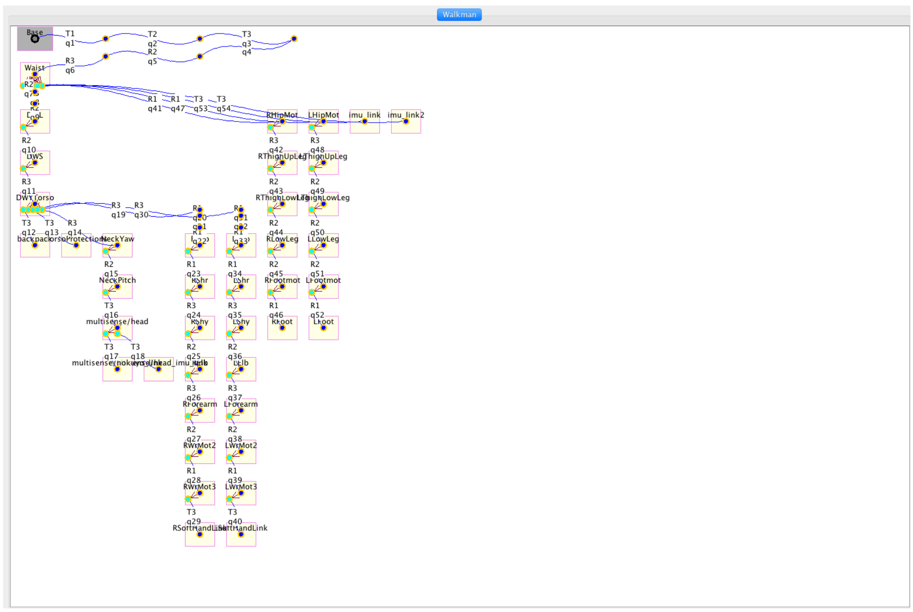


Figure A.12: Screenshot of the MBSysPad graphical interface for the generated WalkMan MBS file

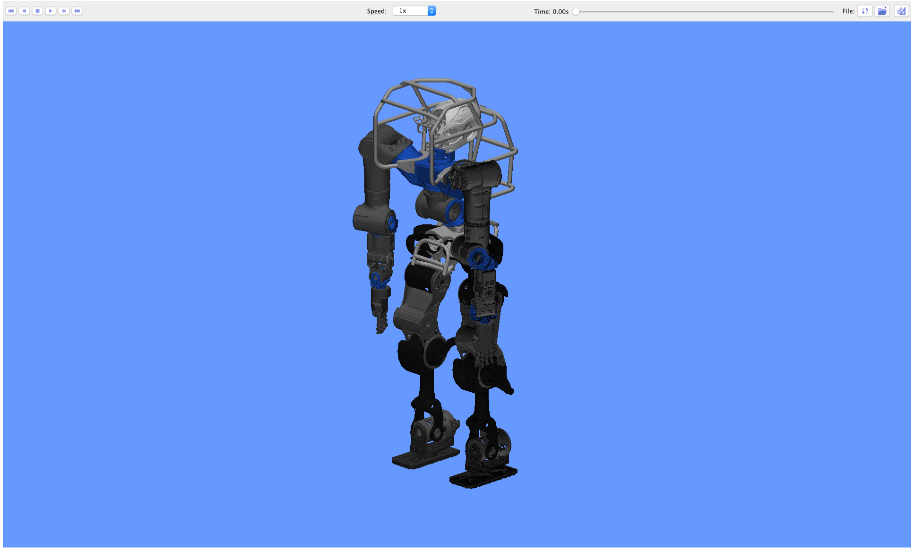


Figure A.13: Screenshot of the MBSysPad Animation 3D Model for the generated WalkMan MBS file

Results for iCub :

The URDF file describing the iCub robot was found on the internet : `https://github.com/robotology-playground/icub-model-generator/blob/master/generated/urdf/iCubParis02/icub.urdf`

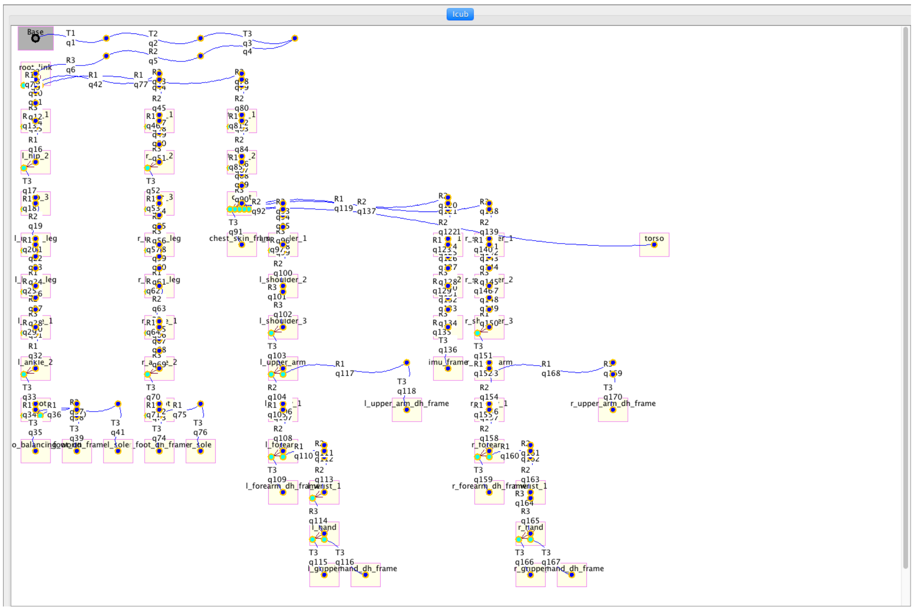


Figure A.14: Screenshot of the MBSysPad graphical interface for the generated iCub MBS file

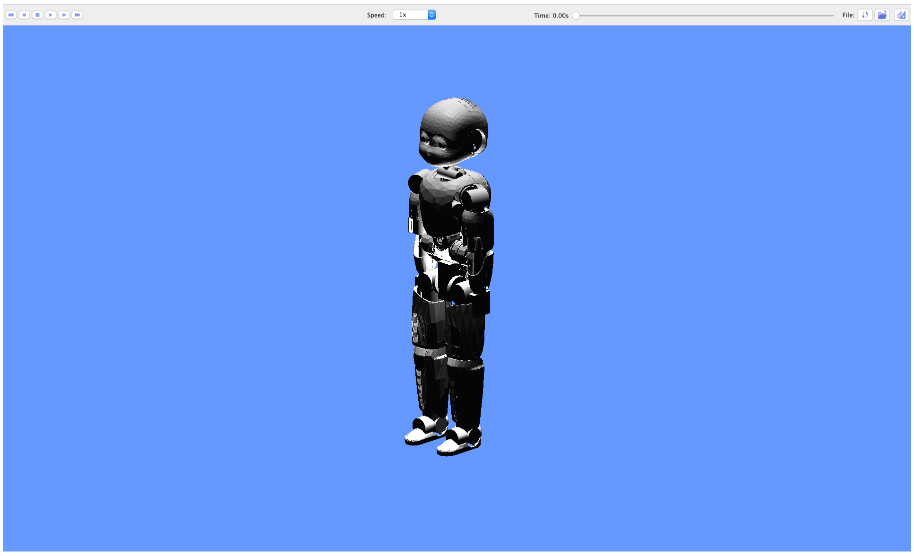


Figure A.15: Screenshot of the MBSysPad Animation 3D Model for the generated iCub MBS file

Appendix B

Motors Specifications

B.1 NAO

Table B.1: Specifications of the four types of motors in NAO (adapted from [25])

Type	1	2	3	4
Maxon motor model	RE-Max 24	RE-Max 17	RE-Max 24	RE-Max 17
Speed reduction ratio	201.3	150.27	130.85	173.22
No load speed $[rpm]$	39.7	79.2	61.1	68.7
Stall torque $[Nm]$	12	2.3	7.8	2.6
Nominal speed $[rpm]$	31.4	58.6	48.4	50.9
Nominal torque $[Nm]$	2.5	0.6	1.6	0.7
Terminal resistance $[\Omega]$	6.44	23.1	6.44	23.1
Torque contant $[Nm/A]$	4.3	2.5	2.8	2.9
Speed constant $[rmp/V]$	2.2	3.8	3.4	3.3
Joints	HipYawPitch HipRoll AnkleRoll	HeadYaw ShoulderPitch ElbowYaw	HipPitch KneePitch AnklePitch	HeadPitch ShoulderRoll ElbowRoll

B.2 CoMan

Table B.2: Specifications of the three types of motors in CoMan (adapted from [19])

Type	1	2	3
Motor inertia $[kg * m^2]$	0.1387	0.1387	0.1387
Motor back EMF constant $[Nm * s/rad]$	25.52	25.52	25.52
Voltage to torque ratio $[Nm/V]$	6.175	6.175	6.175
Torque constant $[Nm/A]$	4.13	4.13	4.13
Terminal resistance $[\Omega]$	0.66	0.66	0.66
Stiffness coefficient $[Nm/rad]$	120	395.46	8400
Damping coefficient $[Nm * s/rad]$	0.198	0.198	4.2
Joints	ShoulderRoll ElbowPitch ShoulderPitch	HipPitch KneePitch TorsoPitch FootPitch TorsoYaw	HipRoll HipYaw FootRoll ShYaw TorsoYaw

B.3 WalkMan

Table B.3: Specifications of the four types of motors in WalkMan (adapted from `https://gitlab.robotran.be/walkman/walkman_robotran`)

Type	1	2	3	4
Motor inertia $[kg * m^2]$	0.1387	0.416	0.30208	0.811
Motor back EMF constant $[Nm * s/rad]$	25.52	385.1575	38.3489	31.17
Voltage to torque ratio $[Nm/V]$	6.175	32.865	5.606	6.852
Torque constant $[Nm/A]$	4.13	11.719	6.84	4.62
Terminal resistance $[\Omega]$	0.66	0.356	1.22	0.6744
Stiffness coefficient $[Nm/rad]$	500	5000	1300	10000
Damping coefficient $[Nm * s/rad]$	1	1	1	1
Joints	NeckYaw NeckPitch ElbowYaw WristPitch WristRoll	TorsoRoll	HipYaw TorsoYaw ShoulderPitch ShoulderRoll ShoulderYaw ElbowPitch	HipRoll HipPitch KneePitch AnklePitch AnkleRoll TorsoPitch

Appendix C

Hill Muscles Properties

The properties of the muscles shown in Figure C.1 are listed below for an adult like model and for the four robots investigated in this thesis.

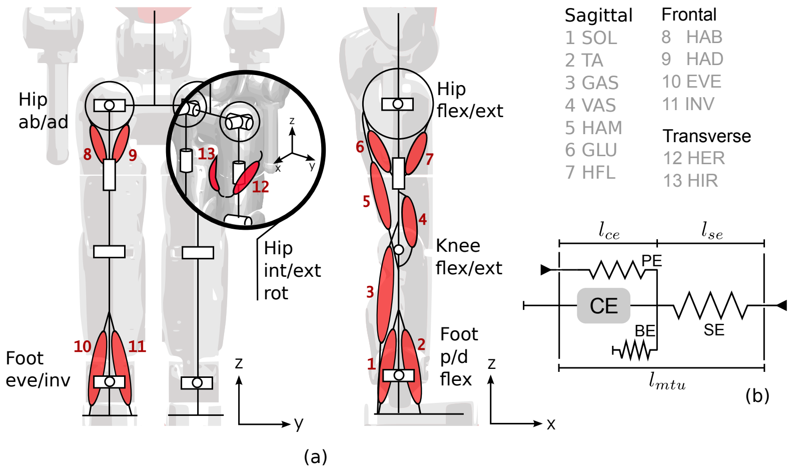


Figure C.1: Muscles acting on the leg joints. [10]

C.1 Adult-like model [8] [23]

Table C.1: Leg muscles properties for an adult-like model (1)

	SOL	TA	GAS	VAS	HAM	GLU
F_{max} [N]	4000	800	1500	6000	3000	1500
v_{max} [l_{opt}/s]	6	12	12	12	12	12
l_{opt} [cm]	4	6	5	8	10	11
l_{slack} [cm]	26	24	40	23	31	13
r_0 [cm]	5	4	5(a) 5(k)	6	5(k) 8(h)	10
ϕ_{max}[deg]	20	-10	20(a) 40(k)	15	0(k) -(h)	-
ϕ_{ref}[deg]	-10	20	-10(a) 15(k)	55	0(k) -25(h)	-30
ρ[-]	0.5	0.7	0.7(a) 0.7(k)	0.7	0.7(k) 0.7(h)	0.5
m [kg]	0.68	0.2	0.32	2.04	1.27	0.7
λ [-]	0.81	0.7	0.54	0.5	0.44	0.5

Table C.2: Leg muscles properties for an adult-like model (2)

	HFL	HAB	HAD	HER	HIR	EVE	INV
F_{max} [N]	2000	3000	4500	500	800	700	900
v_{max} [l_{opt}/s]	12	12	12	12	12	12	12
l_{opt} [cm]	11	9	10	4	8	5	5
l_{slack} [cm]	10	7	18	5	7	25	30
r_0 [cm]	10	6	3	4	3	3	2
ϕ_{max}[deg]	-	-	-	-	-	-10	5
ϕ_{ref}[deg]	0	-10	-15	10	-20	-5	-10
ρ[-]	0.5	0.7	1	1	0.7	0.7	0.7
m [kg]	0.93	1.14	1.91	0.09	0.27	0.15	0.19
λ [-]	0.5	0.5	0.57	0.5	0.5	0.57	0.55

C.2 NAO model

Table C.3: Leg muscles properties for the NAO model

	SOL	TA	GAS	VAS	HAM	GLU
F_{max} [N]	250	50	93.7	375	187.5	93.75
v_{max} [l_{opt}/s]	10.4	20.8	20.8	20.8	20.8	20.8
l_{opt} [cm]	1.3	1.9	1.6	2.6	3.3	3.6
l_{slack} [cm]	8.5	7.9	13.2	7.5	10.2	4.2
r_0 [cm]	1.6	1.3	1.6(a) 1.6(k)	1.9	1.6(k) 2.6(h)	3.3
ϕ_{max}[deg]	20	-10	20(a) 40(k)	15	0(k) -(h)	-
ϕ_{ref}[deg]	-10	20	-10(a) 15(k)	55	0(k) -25(h)	-30
ρ[-]	0.5	0.7	0.7(a) 0.7(k)	0.7	0.7(k) 0.7(h)	0.5
m [kg]	0.04	0.01	0.02	0.12	0.07	0.04
λ [-]	0.81	0.7	0.54	0.5	0.44	0.5

Table C.4: Leg muscles properties for the NAO model (2)

	HFL	HAB	HAD	HER	HIR	EVE	INV
F_{max} [N]	125	187.5	281.2	31.2	50	43.7	56.2
v_{max} [l_{opt}/s]	20.8	20.8	20.8	20.8	20.8	20.8	20.8
l_{opt} [cm]	3.6	2.9	3.3	1.3	2.6	1.6	1.6
l_{slack} [cm]	3.3	2.3	5.9	1.6	2.3	8.2	9.9
r_0 [cm]	3.3	1.9	0.9	1.3	0.9	0.9	0.6
ϕ_{max}[deg]	-	-	-	-	-	-10	5
ϕ_{ref}[deg]	0	-10	-15	10	-20	-5	-10
ρ[-]	0.5	0.7	1	1	0.7	0.7	0.7
m [kg]	0.05	0.07	0.11	0.005	0.01	0.009	0.01
λ [-]	0.5	0.5	0.57	0.5	0.5	0.57	0.55

C.3 CoMan model

Table C.5: Leg muscles properties for the CoMan model (1)

	SOL	TA	GAS	VAS	HAM	GLU
F_{max} [N]	1550	310	581.2	2325	1162.5	581.2
v_{max} [l_{opt}/s]	8.3	16.6	16.6	16.6	16.6	16.6
l_{opt} [cm]	2.0	3.1	2.6	4.1	5.2	5.7
l_{slack} [cm]	13.5	12.4	20.8	11.9	16.1	6.7
r_0 [cm]	2.6	2.0	2.6(a) 2.6(k)	3.1	2.6(k) 4.1(h)	5.2
ϕ_{max}[deg]	20	-10	20(a) 40(k)	15	0(k) -(h)	-
ϕ_{ref}[deg]	-10	20	-10(a) 15(k)	55	0(k) -25(h)	-30
ρ[-]	0.5	0.7	0.7(a) 0.7(k)	0.7	0.7(k) 0.7(h)	0.5
m [kg]	0.26	0.07	0.12	0.79	0.49	0.27
λ [-]	0.81	0.7	0.54	0.5	0.44	0.5

Table C.6: Leg muscles properties for the CoMan model (2)

	HFL	HAB	HAD	HER	HIR	EVE	INV
F_{max} [N]	775	1162.5	1743.7	193.7	310	271.2	348.7
v_{max} [l_{opt}/s]	16.6	16.6	16.6	16.6	16.6	16.6	16.6
l_{opt} [cm]	5.7	4.6	5.2	2.0	4.1	2.6	2.6
l_{slack} [cm]	5.2	3.6	9.3	2.6	3.6	13	15.6
r_0 [cm]	5.2	3.1	1.5	2.0	1.5	1.5	1.0
ϕ_{max}[deg]	-	-	-	-	-	-10	5
ϕ_{ref}[deg]	0	-10	-15	10	-20	-5	-10
ρ[-]	0.5	0.7	1	1	0.7	0.7	0.7
m [kg]	0.36	0.44	0.74	0.03	0.10	0.05	0.07
λ [-]	0.5	0.5	0.57	0.5	0.5	0.57	0.55

C.4 WalkMan model

Table C.7: Leg muscles properties for the WalkMan model (1)

	SOL	TA	GAS	VAS	HAM	GLU
F_{max} [N]	7100	1420	2662.5	10650	5325	2662.5
v_{max} [l_{opt}/s]	5.7	11.4	11.4	11.4	11.4	11.4
l_{opt} [cm]	4.4	6.6	5.5	8.8	11.0	12.1
l_{slack} [cm]	28.6	26.4	44.0	25.3	34.1	14.3
r_0 [cm]	5.5	4.4	5.5	6.6	5.5	11.0
			5.5(k)		8.8(h)	
ϕ_{max}[deg]	20	-10	20(a)	15	0(k)	-
			40(k)		-(h)	
ϕ_{ref}[deg]	-10	20	-10(a)	55	0(k)	-30
			15(k)		-25(h)	
ρ[-]	0.5	0.7	0.7(a)	0.7	0.7(k)	0.5
			0.7(k)		0.7(h)	
m [kg]	1.20	0.35	0.56	3.62	2.25	1.24
λ [-]	0.81	0.7	0.54	0.5	0.44	0.5

Table C.8: Leg muscles properties for the WalkMan model (2)

	HFL	HAB	HAD	HER	HIR	EVE	INV
F_{max} [N]	3550	5325	7987.5	887.5	1420	1242.5	1597.5
v_{max} [l_{opt}/s]	11.4	11.4	11.4	11.4	11.4	11.4	11.4
l_{opt} [cm]	12.1	9.9	11.0	4.4	8.8	5.5	5.5
l_{slack} [cm]	11.0	7.7	19.8	5.5	7.7	27.5	33
r_0 [cm]	11.0	6.6	3.3	4.4	3.3	3.3	2.2
ϕ_{max}[deg]	-	-	-	-	-	-10	5
ϕ_{ref}[deg]	0	-10	-15	10	-20	-5	-10
ρ[-]	0.5	0.7	1	1	0.7	0.7	0.7
m [kg]	1.6	2.0	3.3	0.1	0.4	0.2	0.3
λ [-]	0.5	0.5	0.57	0.5	0.5	0.57	0.55

C.5 Atlas model

Table C.9: Leg muscles properties for the Atlas model (1)

	SOL	TA	GAS	VAS	HAM	GLU
F_{max} [N]	8600	1720	3225	12900	6450	3225
v_{max} [l_{opt}/s]	6.1	12.3	12.3	12.3	12.3	12.3
l_{opt} [cm]	3.7	5.6	4.7	7.5	9.4	10.3
l_{slack} [cm]	24.4	22.5	37.6	21.6	29.1	12.2
r_0 [cm]	4.7	3.7	4.7 4.7(k)	5.6	4.7 7.5(h)	9.4
ϕ_{max}[deg]	20	-10	20(a) 40(k)	15	0(k) -(h)	-
ϕ_{ref}[deg]	-10	20	-10(a) 15(k)	55	0(k) -25(h)	-30
ρ[-]	0.5	0.7	0.7(a) 0.7(k)	0.7	0.7(k) 0.7(h)	0.5
m [kg]	1.4	0.4	0.6	4.3	2.7	1.5
λ [-]	0.81	0.7	0.54	0.5	0.44	0.5

Table C.10: Leg muscles properties for the Atlas model (2)

	HFL	HAB	HAD	HER	HIR	EVE	INV
F_{max} [N]	4300	6450	9675	1075	1720	1505	1935
v_{max} [l_{opt}/s]	12.3	12.3	12.3	12.3	12.3	12.3	12.3
l_{opt} [cm]	10.3	8.4	9.4	3.7	7.5	4.7	4.7
l_{slack} [cm]	9.4	6.5	16.9	4.7	6.5	23.5	28.2
r_0 [cm]	9.4	5.6	2.8	3.7	2.8	2.8	1.8
ϕ_{max}[deg]	-	-	-	-	-	-10	5
ϕ_{ref}[deg]	0	-10	-15	10	-20	-5	-10
ρ[-]	0.5	0.7	1	1	0.7	0.7	0.7
m [kg]	1.9	2.4	4.1	0.1	0.5	0.3	0.4
λ [-]	0.5	0.5	0.57	0.5	0.5	0.57	0.55

Appendix D

Bio inspired control rules for the leg sagittal DOFs

The control rules presented in this appendix are the ones from [8] (sometimes slightly modified). The different equations and explanations are adapted from [19] to match the ones used in this project.

Stance

(1) $S_{SOL} = S_{MIN} + G_{SOL} \cdot \frac{F_{SOL}(t_l)}{F_{SOL}^{max}}$

$F+$ generates compliant behavior

(2) $S_{TA} = S_{MIN} + G_{TA,ST} \cdot (l_{CE,TA}(t_l) - l_{TA,ST}^{off} \cdot l_{opt,TA})$

(3) $S_{GAS} = S_{MIN} + G_{GAS} \cdot \frac{F_{GAS}(t_l)}{F_{GAS}^{max}}$

$F+$ prevents knee overextension + compliant leg behavior

(4) $S_{VAS} = S_{0,VAS} + [G_{VAS} \cdot \frac{F_{VAS}(t_m)}{F_{VAS}^{max}}]_{\{\varphi_k\}}$

$F+$ generates compliant behavior

(5) $S_{HAM} = S_{MIN} + [k_p \cdot (\theta(t_s) - \theta_{ref}) + k_d \cdot \dot{\theta}(t_s)]_{\{+\}} \cdot k_{bw} \cdot \frac{|F_{leg}^{ipsi}(t_s)|}{F_g}$

$\theta+$ is used to balance the trunk

(6) $S_{GLU} = [S_{MIN} + [0.68 \cdot k_p \cdot (\theta(t_s) - \theta_{ref}) + k_d \cdot \dot{\theta}(t_s)]_{\{+\}} \cdot k_{bw} \cdot \frac{|F_{leg}^{ipsi}(t_s)|}{F_g}]_{\{\text{SSup}\}} - [S_{MIN}]_{\{\text{DSup}\}}$

$\theta+$ is used to balance the trunk

$\Delta S-$ further initiates swing in double support

(7) $S_{HFL} = [S_{MIN} - [k_p \cdot (\theta(t_s) - \theta_{ref}) + k_d \cdot \dot{\theta}(t_s)]_{\{-\}} \cdot k_{bw} \cdot \frac{|F_{leg}^{ipsi}(t_s)|}{F_g}]_{\{\text{SSup}\}} + [S_{MAX}]_{\{\text{DSup}\}}$

$\theta - (< 0)$ is used to balance the trunk

$\Delta S+$ further initiates swing in double support

Swing

(1) $S_{SOL} = S_{MIN}$

silent: leg's ballistic motion

(2) $S_{TA} = S_{MIN} + G_{TA,SW} \cdot (l_{CE,TA}(t_l) - l^{off}_{TA,SW} \cdot l_{opt,TA})$

$L+$ prevents the ankle from overextending + foot clearance with the ground

(3) $S_{GAS} = S_{MIN}$

silent: leg's ballistic motion

(4) $S_{VAS} = S_{0,VAS}$

silent: leg's ballistic motion

(5) $S_{HAM} = S_{MIN} + G_{HAM} \cdot \frac{F_{HAM}(t_s)}{F^{max}_{HAM}}$

$F+$ transfers part of the protraction momentum into leg lowering and retraction

(6) $S_{GLU} = S_{MIN} + G_{GLU} \cdot \frac{F_{GLU}(t_s)}{F^{max}_{GLU}}$

$F+$ transfers part of the protraction momentum into leg lowering and retraction

(7) $S_{HFL} = S_{MIN} + G_{HFL} \cdot (l_{CE,HFL}(t_s) - l^{off}_{HFL} \cdot l_{opt,HFL}) - G_{HAM-HFL} \cdot (l_{CE,HAM}(t_s) - l^{off}_{HAM} \cdot l_{opt,HAM})$

$L_{HFL}+$ modulates ballistic motion: facilitating leg protraction during swing

$L_{HAM}-$ enforces swing-leg retraction, compensates for the hip rotation

S_{MIN} is the minimal stimulation and is equal to 0.01. S_{MAX} is the maximal stimulation and is equal to 1. Some terms are only added in special circumstances:

- $[\,]_{DSup}$: $\neq 0$ if trailing leg is in double support (finishing stance)
- $[\,]_{SSup}$: $\neq 0$ if trailing leg is in single support
- $[\,]_{\{+/-\}}$: $\neq 0$ if positive/negative values in $[\,]$
- $[\,]_{\{\varphi_k\}}$: $\neq 0$ if $\varphi_{knee} > \varphi^{off}_{knee}$ & $\varphi_{knee} > 0$

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