

Balance improvement of a humanoid robot while walking

Dissertation presented by
Esther LECOMPTE

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

Supervisor
Renaud RONSSE

Readers
Julien HENDRICKX, Auke IJSPEERT , Nicolas VAN DER NOOT

Academic year 2017-2018

Acknowledgements

First of all, I would like to thank Prof. Renaud Ronsse for the supervision and guidance of my master thesis. His availability and numerous useful advice allowed me to methodologically progress in my work. His remarks concerning some parts of my report were also very valuable.

During the first semester of this year, I have had the opportunity to discover a new university, the Ecole Polytechnique Fédérale de Lausanne in Switzerland. As part of my Erasmus program, I could benefit from the advice and expertise of Prof. Auke Ijspeert and the members of the BioRob Laboratory to begin my master thesis. Thanks for your welcome at the BioRob Laboratory and particularly to Dr. Amy Wu who followed my work during the first semester.

I would like to also express my sincere gratitude to Dr. Nicolas Van der Noot and his availability whose my master thesis mainly constitutes the continuity of his work during his master thesis and thesis. His supervising, guidance and helpful advice although remotely were very precious. I am still impressed by all his contribution and achievements for the bio-inspired locomotion of COMAN robot.

Last but not least, I would like to express my gratitude to my family and my friends for their support and encouragements during this year but also during all my studies.

Special thanks to Philippe Verbist, working also on the robot COMAN as part of his master thesis, for his ideas, encouragements and for all the moments shared together.

And finally, to my Mom for giving me the strength to continue day after day and who I miss so much...

Esther Lecompte
Louvain-la-Neuve, June 2018

Abstract

Walking is a daily movement employed since the dawn of time by most humans but falls may nevertheless occur. Research linked to the management and the avoidance of falling are still carried out since we do not yet understand how the human brain recognizes falling prediction patterns, and reacts accordingly.

These knowledge of human balance would strengthen the robustness of biped humanoid robots and especially bio-inspired robots. Since a robot that falls can harm not only its environment but also itself, implementing a balance system becomes crucially relevant. This will allow the robot to interact in various environments where perturbations could be encountered. However, humanoid robots are far from reaching human balance abilities since they are frequently unstable during locomotion due to their relatively high centre of mass and limited support polygon.

This master thesis is devoted to the development of a balance controller as complement to the existing bio-inspired controller for the compliant robot COMAN, in simulation [Van der Noot, 2017b]. Two main goals guide this work. Firstly, perturbations must be detected to trigger the use of the additional balance controller. And secondly, the implementation of a balance controller to face disturbances during walking is achieved.

The detection of perturbation is realized by comparing specific data of the robot with a reference robot that is not facing perturbations. If the error between these data exceeds a certain threshold, then a perturbation is sensed. During the tests, all perturbations are sensed after around 0.12s. Concerning the implemented balance controller, it relies on a force-based controller and allows to improve the performance of the robot against lateral pushes.

Explanations on the implemented methods, detailed results, comparisons with human responses and discussions are provided in this report.

List of abbreviations

BoS	Base of Support
CoM	Center of Mass
CoP	Center of Pressure
DOF	Degree Of Freedom
CPG	Central Pattern Generator
GAF	Ground Applied Force
GRF	Ground Reaction Force
LHS	Left Heel Strike
LTO	Left Toe Off
RHS	Right Heel Strike
RG	Rhythm Generator (neuron)
RTO	Right Toe Off
XCoM	eXtrapolated Center of Mass
ZMP	Zero-Moment Point

Leg muscle names

EVE	Eversion
GAS	Gastrocnemius
GLU	Gluteus
HAB	Hip Abductors
HAD	Hip Adductors
HAM	Hamstring
HER	Hip External
HFL	Hip Flexors
HIR	Hip Internal Rotator
INV	Inversion
RF	Rectus Femoris
SOL	Soleus
TA	Tibialis Anterior
VAS	Vastus

Contents

Acknowledgements	i
Abstract	iii
List of abbreviations	v
1 Introduction	1
1.1 Bipedal locomotion	2
1.1.1 Gait cycle	3
1.1.2 Anatomical planes	3
1.2 Structure	4
2 Review of the literature	5
2.1 Human balance overview	5
2.1.1 Sensory strategies	5
2.1.2 Movement strategies	6
2.1.3 Human responses to lateral perturbations while walking	6
2.2 Conditions for static and dynamic stability	7
2.2.1 Conditions for static stability	7
2.2.2 Conditions for dynamic stability	8
2.3 Fall detection in biped humanoid robots	9
2.4 Balance controllers	10
3 Existing controller and platform	11
3.1 COMAN	11
3.2 Robotran simulation environment	12
3.3 Bio-inspired controller overview	12
3.3.1 High-level controller	12
3.3.2 Middle-level neuromuscular model	12
3.3.3 Low-level controller	15
3.3.4 Inverse muscular model	15
3.4 Initial robustness against perturbations	16
3.4.1 Types of perturbations	16
3.4.2 Falling state	16
3.4.3 Falling analysis	16
4 General architecture & Perturbation detection	21
4.1 General architecture	21
4.2 Internal model	23
4.3 Gait cycle phase estimator	25
4.3.1 Rhythm generation neurons	25
4.3.2 Estimator method	25

4.3.3	Validation	27
4.4	Stability criterion based on internal model	29
4.4.1	Criteria establishment	29
4.4.2	Determination of the criteria expressions	31
4.4.3	Other tested criteria	34
4.5	Assessment of the performance of the criteria	36
4.5.1	Detection of pushes applied on the torso	36
4.5.2	Detection of slippery grounds	40
4.6	Discussion	41
5	Balance controller	43
5.1	Triggering of the balance controller	43
5.2	Implemented methods	44
5.2.1	General principle	44
5.2.2	Virtual control force	44
5.2.3	Quasi-static force-torque relationship	45
5.2.4	Force-torque relationship based on ground forces	46
5.2.5	From torques to stimulations	50
5.2.6	Parameters determination	50
5.3	Validation method	50
5.4	Results	50
5.4.1	Reaction to inward lateral pushes at heel strike	50
5.4.2	Influence of the gait cycle phase with lateral pushes	54
5.4.3	General performance compared to the initial robustness	57
5.5	Discussion	58
6	Future perspectives & Conclusion	61
6.1	Future perspectives	61
6.2	Conclusion	62
	Bibliography	63
A	State variables sent to the internal models	67
B	Stability criteria	69
B.1	Gait cycle phase estimator equations	69
B.2	Stability criteria equations	70

Chapter 1

Introduction

Walking and moving are daily movements constantly employed by most humans. However, humans are not immune to falls. Indeed, the majority of human falls occurs during walking and with age the number of falls considerably increases [Tinetti et al., 1988, Chen and Chou, 2010]. The domain linked to the management and the avoidance of falling thus seems interesting to analyze, especially since we do not yet understand how the human brain recognizes falling prediction patterns, and reacts accordingly.

These knowledge of human balance would strengthen the robustness of biped humanoid robots and especially bio-inspired robots. Since a robot that falls can harm not only its environment but also itself, implementing a balance system becomes crucially relevant. This will allow the robot to interact in environments where perturbations such as pushes or obstacles on the ground could be encountered.

In this way, humanoid robots are able to interact in the same environments as humans. They can climb steps, sit on a chair and manipulate objects in a similar way as humans. Nevertheless, guaranteeing a stability against large disturbances is not straightforward. Indeed, although humanoid robots have the benefit of being more flexible in comparison to wheeled or multi-legged robots, they are frequently unstable during locomotion due to their relatively high centre of mass and limited support polygon.

During one of the biggest international robotics competitions, DARPA Robotics Challenge [Darpa, 2015], even the robots developed by the leading research centers fell down when faced to unknown environments, as depicted in this video <https://www.youtube.com/watch?v=g0TaYhjp0fo> [IEEE Spectrum, 2015]. Robots are thus far from reaching human capacities although increasingly advanced robots are developed such as Atlas performing somersaults or getting back up after a fall [Boston Dynamics, 2018].

The objective of this master thesis consists in the implementation of a balance controller as complement to the existing bio-inspired controller for the compliant robot COMAN mainly developed by Nicolas Van der Noot [Van der Noot, 2017b]. The simulated robot was already able to walk in a 3D environment and to cope with small perturbations. The balance controller will be activated during limited periods of time after a perturbation is detected on the robot. Two main goals thus guide this work : (i) detecting perturbations occurring on the robot and (ii) implementing a balance controller to face large disturbances during walking.

1.1 Bipedal locomotion

A biped robot is a robot with two legs and, if its body shape resembles to the human body, it is called a humanoid robot. Generally, bipeds are supposed to mimic human walking.

The fields related to biped locomotion and balance refer to different specific terms as summarized in [Winter, 1995, Vukobratović et al., 2007, Dekker, 2009]. Some key terms are defined in this section :

Walk is defined as the "movement by putting forward each foot in turn, not having both feet off the ground at once" [Vukobratović et al., 2007].

Gait is specific to every individual, hence it represents the "manner of walking or running" [Vukobratović et al., 2007].

Balance is "a generic term describing the dynamics of body to prevent falling. It is related to the inertial forces acting on the body and the inertial characteristics of body segments" [Winter, 1995].

Centre of mass (CoM) is "a point equivalent of the body mass in the global reference system and is the weighted average of the CoM of each body segment in 3D space" [Winter, 1995]. The vertical projection of the CoM on the ground is sometimes called the centre of gravity (CoG).

Double support describes situations where the biped is supported by both feet on the ground. It is not necessary that both feet are fully supported [Dekker, 2009].

Single support corresponds to situations where the biped has only one foot on the ground [Dekker, 2009].

Base of Support (BoS) or support polygon is "the surface determined by the contact of the foot on the ground. In single-support phase, the base of support coincides with the area of the foot in contact with the ground whereas in double-support phase, it is a convex area determined by the areas of the feet on the ground and common tangents, so that the area is maximized" [Vukobratović et al., 2007].

Centre of pressure (CoP) is "the point of application of the ground reaction force. It represents a weighted average of all the pressures over the surface of the area in contact with the ground" [Winter, 1995]. It is located under the stance foot during single support and between the two feet in double support.

Swing leg describes the leg moving forward in the air for performing a step [Dekker, 2009].

Stance leg corresponds to the leg in contact with the ground, supporting the weight of the biped [Dekker, 2009].

A Fall is defined "when a point on the biped, other than a point on the feet of the biped, touches the ground" [Pratt and Tedrake, 2006].

However, the concept of falling down is difficult to precisely define. For example, sitting down on the floor and slipping down onto the floor might result in the exact same trajectories and end state but one is considered as falling and the other as sitting, with the only difference being intent. Therefore, for similar reasons, the notion of stability is difficult to define.

1.1.1 Gait cycle

When a biped walks with a periodic gait, its gait phase can be divided from 0 to 100 % into four phases as shown in Figure 1.1 where approximated periods of single and double support phases are given.

1. **Double support phase** By convention, a stride begins with right heel contact (or right heel strike RHS). Then, a first double support phase occurs where both feet are in contact with the ground.
2. **Left swing phase** After the left toes lift from the ground (left toe-off LTO), the left leg is in swing while the right leg is in single support phase.
3. **Double support phase** When the left foot reaches the ground at left heel contact (or left heel strike LHS), a second double support phase begins. It corresponds to the half of the stride.
4. **Right swing phase** Finally, the right leg is in swing from right toe-off RTO until right heel strike RHS, whereas the left leg is in stance.

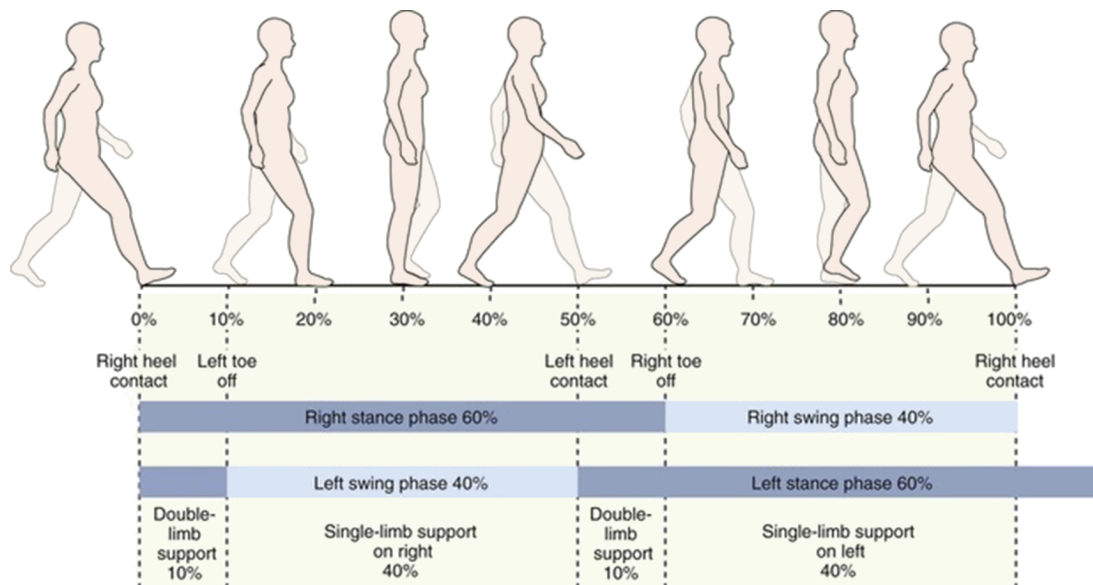


Figure 1.1: Phases of a normal gait cycle. With the courtesy of [Themes, 2016]

1.1.2 Anatomical planes

Conventions are defined to position a biped or human in three dimensional space. The x-axis points forward, the y-axis points to the left-hand side and the z-axis points upwards. Figure 1.2 represents the three perpendicular anatomical planes, the axis and the rotations used for bipeds and humans.

- The frontal or coronal plane is perpendicular to the ground and separates the anterior part from the posterior part of the body and is parallel to the yz-plane. A rotation in this plane is called roll.
- The sagittal plane is also perpendicular to the ground, separating the left part from the right part of the body and is parallel to the xz-plane. A rotation in this plane is called pitch.

- The transverse or horizontal plane is parallel to the ground and separates the superior part from the inferior part of the body and is parallel to the xy-plane. A rotation in this plane is called yaw.

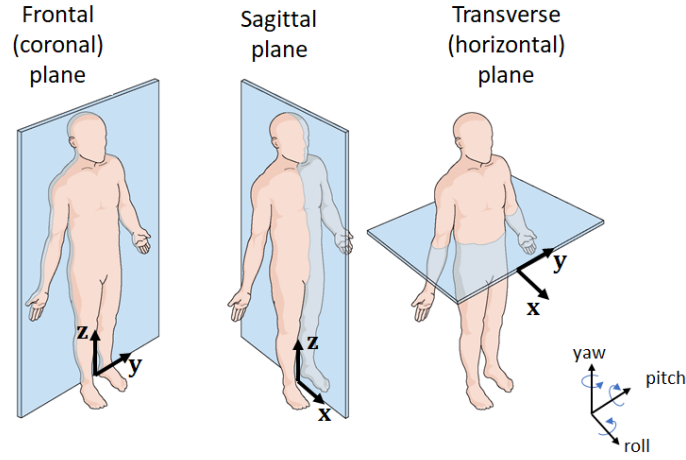


Figure 1.2: Anatomical planes, axis and rotations for a biped, adapted from [Lateral Edge, 2017].

1.2 Structure

This report is divided into a total of six chapters.

Chapter 1 is dedicated to the introduction and objectives of this report. Specific terms related to biped and balance are also defined.

Chapter 2 presents a review of the literature on different aspects related to balance both for humans and biped robots. Existing balance controllers are also described.

Chapter 3 is devoted to the presentation of the existing controller and platform which constitute the starting point of this work and allow already the control of some parameters during locomotion such as the speed and the steering direction of the robot. The initial performances of the robot facing small, medium and large pushes are also analyzed.

Afterwards, Chapter 4 gives an overview of the additional balance controller and details the implementation of some blocks such as the use of an internal model and a gait cycle phase estimator. These elements will be useful for detecting perturbations. The rest of this chapter is then devoted to perturbation detection criteria which will trigger the use of the balance controller.

Chapter 5 focuses on the balance controller itself. Its implementation, the validating methodology and the derived results are presented. A comparison with the initial robustness of the robot is also provided. A discussion on the results closes this chapter.

Finally, Chapter 6 introduces suggestions for further work and ends this report with a conclusion on the work achieved in the context of this master thesis.

Chapter 2

Review of the literature

At this time, plenty of research has already been conducted in order to improve the stability of humanoid robots. However, no robot has yet succeeded in reaching human capacities with regard to balance.

In this chapter, several studies will be referred to cover different aspects related to balance. First of all, human strategies to manage perturbations when walking will be reviewed. These findings are based on human subjects analysis. Afterwards, conditions for static and dynamic stability will be introduced, followed by methods for fall detection in biped humanoid robots. Finally, several kinds of balance controllers will be described.

2.1 Human balance overview

Most human falls occur when people are standing or walking or more generally during activities involving large displacements of the center of mass (CoM) such as walking, stepping up or down or standing up [Tinetti et al., 1988]. With age, abilities to keep balance are reduced and the probability of falling raises considerably.

The response against a perturbation depends not only on the characteristics of the external disturbance but also on the skills of each individual. Human reactions will differ depending on their experience, intention and expectations. That is the reason why different people will fall in different situations, depending on the systems that are required to complete the task successfully [Horak, 2006].

2.1.1 Sensory strategies

When a human detects an instability from one or several sensory systems, he will trigger diverse strategies to avoid the fall or to reduce the fall damage. [Winter, 1995] suggests that three major sensory systems are involved in balance and posture :

1. the visual system;
2. the vestibular system, which senses linear and angular accelerations;
3. the somatosensory system, which consists of a large amount of sensors that sense the position and velocity of all body segments, their contact with external objects and the orientation of gravity.

Usually, humans base their information mainly on the somatosensory system but, when stability is challenged, vestibular and visual information are more considered [Horak, 2006].

2.1.2 Movement strategies

When an instability is sensed, humans use different strategies to avoid falling. Three main types of movement strategies illustrated in Figure 2.1 have been observed when standing [Winter, 1995, Horak, 2006] :

- the ankle strategy, in which the body moves at the ankle, is generally used for small amounts of perturbations;
- the hip strategy, in which the body exerts torque at the hips to quickly move the body CoM, is generally used when persons stand on narrow or compliant surfaces or when the CoM must be moved quickly.
- the stepping strategy, in which a step is taken, is generally used for large perturbations unlike the two first strategies.

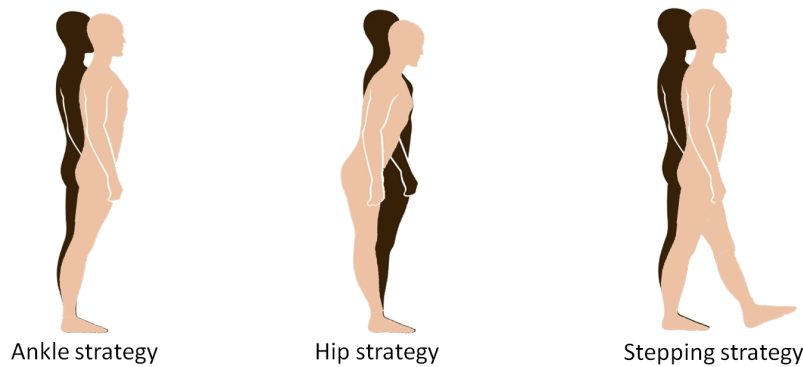


Figure 2.1: Movement strategies to avoid falling.

When walking, the gait can also be adapted in addition to movement strategies. A number of studies have analyzed gait adaptations when the gait stability was perturbed [Hak et al., 2012]. Data reveal that the subjects do not reduce their walking speed in response to balance perturbations. Instead, they decrease the step length whereas they increase the step frequency and the step width.

In these studies, all balance analysis consider young and healthy individuals, assuming that for older subjects, perturbations are corrected less accurately, require more steps to regain balance, and demand more attention [Tinetti et al., 1988, Hak et al., 2012].

2.1.3 Human responses to lateral perturbations while walking

[Hof et al., 2010] analyze balance responses to lateral perturbations in human treadmill walking. Two different strategies are employed for recovering : (i) a stepping strategy and (ii) a lateral ankle strategy which comprises a medial or lateral movement of the centre of pressure under the foot sole.

The ankle strategy is faster than the stepping strategy since it can be used during stance, while the stepping strategy can only be applied at the next step. This is useful when the perturbation is given after or shortly before foot placement when it is too late to apply a step strategy. Moreover, the lateral ankle strategy compensates in part for the inaccuracy of foot placement.

The experimental set-up is represented in Figure 2.2. When a push is applied at heel contact, the initial foot placement is unaltered while the stride time can be reduced in some cases. Indeed, in the case of lateral inward pushes applied at right heel strike, the duration of the right foot in stance is shortened due to a shortening of the swing foot time. In contrast, for lateral outward pushes applied at left heel strike, no shortened stance is observed.

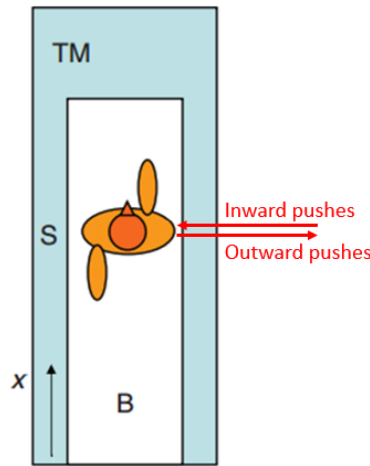


Figure 2.2: Experimental set-up of [Hof et al., 2010]. The subject S walks on the belt B of treadmill TM and is subject to lateral pushes from a device placed on the right.

However, with outward pushes, the trunk rotation shows greater deviations from normal and more apparent lateral ankle strategies compared to inward perturbations.

Moreover, after a push, the CoM position is only slightly changed while the CoM velocity is considerably altered. The reactions in terms of lateral foot placement can be important, almost 20 cm for a 'cross-over' step. Finally, based on these measurements, arm movements do not play a significant role in walking with perturbations [Hof et al., 2010].

2.2 Conditions for static and dynamic stability

Evaluating the stability of a biped allows to predict a potential fall but also the state of the biped. Although static stability is easily defined, dynamic stability is less straightforward.

2.2.1 Conditions for static stability

The well-known condition for standing stability in static situations is that the vertical projection of the centre of mass (CoM), sometimes called the center of gravity (CoG), should be within the base of support (BoS) [Winter, 1995], as represented in Figure 2.3.

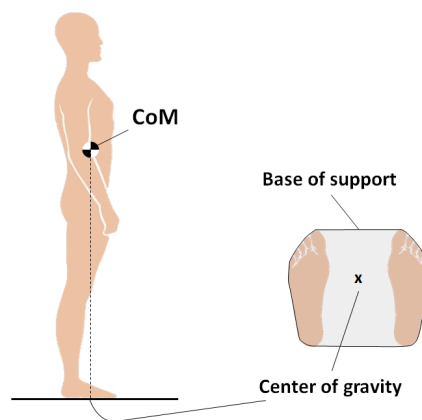


Figure 2.3: The center of gravity is the vertical projection of the CoM on the ground. As long as the CoG is positioned within the base of support of the feet, balance is maintained under static conditions.

2.2.2 Conditions for dynamic stability

Exact computation of stability margins is difficult to achieve since the dynamics of bipedal walking is complex. However, simplified walking models, such as the Inverted Pendulum model (IPM) or the Linear Inverted Pendulum model (LIPM), can be used to obtain approximations.

Stability margins may be characterized as necessary or sufficient conditions :

- Necessary : if the stability margin is outside an acceptable threshold of values, the robot will fall down.
- Sufficient. If the stability margin is inside the acceptable threshold of values, the robot will not fall down.

Zero Moment Point

One of the most known criterion for dynamic stability refers to the Zero-Moment Point (ZMP). The ZMP is defined as the location on the ground where the net moment generated from the ground reaction forces equals has zero moment about the two axes that lie in the plane of the ground [Pratt and Tedrake, 2006, Vukobratović et al., 2007]. As long as the ZMP lies strictly inside the support polygon of the foot, then the desired trajectories of the robot joints are dynamically feasible. If the ZMP lies on the edge of the support polygon or outside, then the trajectories may not be dynamically feasible. A representation is provided in Figure 2.4.

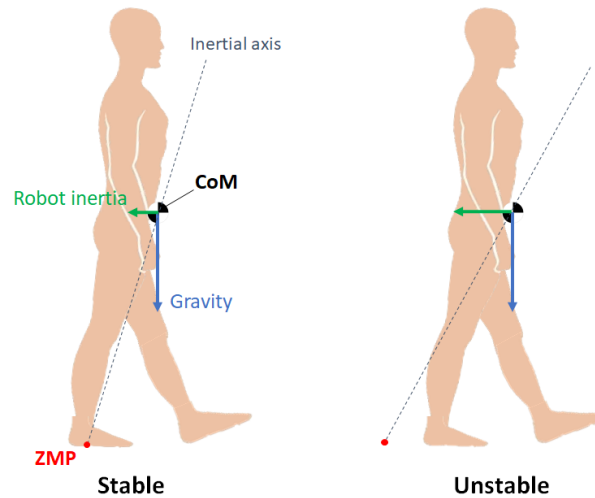


Figure 2.4: Illustration of the ZMP criterion. If the ZMP lies inside the base of support, the biped is dynamically stable whereas if the ZMP lies outside, the biped is dynamically unstable.

The ZMP is equivalent to the Center of Pressure (CoP) only if the ZMP lies inside the support polygon. Nevertheless, the ZMP is commonly used to mean the analytically computed point based on the state and acceleration of the robot whereas the CoP is commonly used to mean the point measured from ground reaction forces (GRF).

Maintaining the ZMP inside the support polygon is not a necessary condition for stable walking since during toe-off in human walking, the ZMP stability criterion is violated. However, it becomes a sufficient criterion when associated with some algorithms such as trajectory tracking control algorithms [Pratt and Tedrake, 2006].

CoM velocity

It has been proposed by [Pai and Patton, 1997] that the horizontal velocity of the CoM should be considered when describing the feasible movements for balance control. Indeed, in the case of a large horizontal CoM velocity, standing will probably not be maintained even though the CoM is located inside the BoS. Inversely, it is possible that the CoM is initially located outside the BoS and that upright standing is still achievable, provided that sufficient horizontal CoM velocity is directed toward the BoS. Therefore, the magnitude and direction toward the BoS of the CoM velocity may provide critical information.

Extrapolated centre of mass (XCoM)

The extrapolated centre of mass (XCoM) proposed by [Hof et al., 2005] takes into account the CoM velocity. They propose an extension of the condition for standing stability expressed in section 2.2.1 for dynamical situations. They define the XCoM as the vertical projection of the CoM plus its velocity multiplied by a parameter related to the subject's leg length $\sqrt{l/g}$ where l is the leg length and g represents the gravity.

$$x_{\text{XCoM}} = x_{\text{CoM}} + \sqrt{\frac{l}{g}} \dot{x}_{\text{CoM}} \quad (2.1)$$

As long as the XCoM lies inside the base of support, dynamic stability is ensured.

However, due to the several assumptions made to obtain this condition, it is only valid at foot placement.

Capture Point

The Capture Point, introduced by [Pratt et al., 2006] is defined as "a point that can be stepped to in order to stop" while the Capture Region is "the set of all Capture Points" [Pratt and Tedrake, 2006]. They extend this notion to a broader concept, the "N-step capturability", which is "the ability of a legged system to come to a stop without falling by taking N or fewer steps" [Koolen et al., 2012]. They use this concept to establish a condition : "if a walking biped cannot stop within several steps, it is probably close enough to falling to consider it unstable" [Pratt and Tedrake, 2006]. Therefore, they define stability margins to estimate the likelihood that a biped can stop over a given number of steps. These notions are used in several controllers for maintaining balance, as mentioned in section 2.4.

2.3 Fall detection in biped humanoid robots

Other methods have been conducted to detect unstable situations in humanoid robots.

As referred in [Pratt and Tedrake, 2006], it has been recognized that traditional stability margins such as zero-moment point (ZMP) described in 2.2.2, foot rotation indicator, centre of pressure (CoP), and zero-rate change of angular momentum, are not general and insufficient for predicting instabilities under all situations.

Methods derived from a simplified model such as the linear inverted pendulum have a limited applicability to a specific posture [Subburaman et al., 2016].

For these reasons, other techniques using the full model of the robot are proposed to detect falls in walking conditions. In [Ruiz-del Solar et al., 2010, Renner and Behnke, 2006], the trunk attitude and its derivative in the lateral and sagittal planes are used as sensor measurements and are compared with a sensor model which corresponds to the coefficients of the sensor measurements in simulated walking situations. [Rebula et al., 2007] proposes velocity-based techniques such as a capture region, [Xinjilefu et al., 2015] uses methods with a centre of mass estimator and [Li

et al., 2015] the total mechanical energy of the system.

[Subburaman et al., 2016] use data from multiple sensors such as CoM, inertial measurement unit (IMU), ZMP and foot pressure sensors to detect a fall. They obtain better performances than using a single type of sensor since they obtain less instances of fall positives and earlier fall prediction.

There are further extensions with predictor models using probabilistic [O. Höhn and W. Gerth, 2009] and machine learning techniques [Kalyanakrishnan and Goswami, 2011].

2.4 Balance controllers

Most balance controllers implemented for biped are based on the control of the position and speed of the CoM. It can be done by adjusting the ground reaction force through the CoP following the same strategies as humans : (i) ankle strategy, (ii) hip strategy and (iii) stepping strategy as illustrated in Figure 2.1.

In the case of stepping strategies, the placement of the foot is critical. Two types of controller are generally employed : (i) position-based and (ii) force-based controllers.

[Koolen et al., 2012] calculate the N-step capture region through different simple models of legged locomotion before determining the adequate foot placement and using trajectory planning to reach it. [Faraji et al., 2018] use a time-projection control to adjust the next footstep location based on an improved linear model that simulates swing and torso dynamics. On the other hand, force-based controllers have the advantage to be able to quickly adapt to sudden environmental changes [Stephens and Atkeson, 2010]. [Hyon et al., 2007] manage unknown perturbations by transforming desired contact forces into joint torques using a Jacobian-transpose mapping, i.e. $\tau = J^T F$. In case of small disturbances, force feedback can also be used to modify the inverse kinematics and directly control the contact forces. However, this requires force measurement at the point of contact [Stephens and Atkeson, 2010].

Other methods are based on human motions. [Kudoh et al., 2006] capture first human motions when maintaining balance, build then a simple model to extract global parameters such as the CoM and ZMP. And finally, they use these data to generate adequate trajectories when the balance of the robot is challenged.

Specific mechanisms against fall have also been implemented. In [Renner and Behnke, 2006], the robot slows down in case of a small instability and in case of a large perturbation, it stops and lowers its centre of mass. In [O. Höhn and W. Gerth, 2009], reflex steps are used to avoid the fall whereas crouching mechanisms are triggered when a fall is unavoidable.

Indeed, inspired from martial arts, falling techniques (e.g. ukemi) allow to limit the impact. They consist in a sequence of movements that vary the geometry of the body to spread the kinetic energy through a wider contact area and a longer lapse of time in order to lower the damage of falls [Ruiz-del Solar et al., 2010].

Chapter 3

Existing controller and platform

The starting point of this master thesis relies on a bio-inspired controller developed on the robot COMAN mainly by [Van der Noot, 2017b] and former students as part of their master thesis. The most relevant ones for this project are : [Van der Noot and Barrea, 2013], [Heremans, 2015] and [Greiner, 2017]. The work is achieved in the Robotran simulation environment. At the beginning of this master thesis, the robot was already able to generate a human-like gait with speed and steering adaptations and to cope with small perturbations due to the robustness of the implemented bio-inspired controller, even if no balance module had been developed yet.

In this chapter, the general functioning of the different tools will be outlined with first a presentation of the robot COMAM, followed by a short description of the simulation platform Robotran. Then, an overview of the controller and its sub-components will be provided. Finally, an analysis of the initial performance of the robot facing perturbations will be given.

3.1 COMAN

This project is based on the robot COMAN (**CO**mpliant **HU**MANoid Platform) developed by the Italian Institute of Technology (IIT) and represented in Figure 3.1.

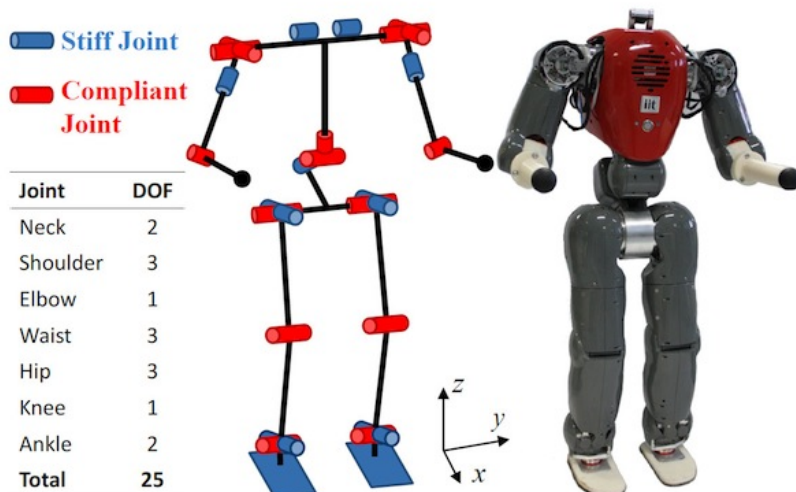


Figure 3.1: Representation of COMAN and its 23 degrees of freedom [Falconer, 2013].

The robot is modeled based on a 4-year-old child, is 95 cm tall, weights 31 kg, possesses 23 degrees of freedom (DOF) and is equipped with compliant joints using series elastic actuators (SEA). These latter are able to store and release elastic energy and allow a soft actuation, a high

shock tolerance and a safe interaction with the environment [Van der Noot, 2017b]. However, the robot have no head nor hands.

Regarding the robot sensors, all joints have position encoders and torque sensors. Below the ankle joint, 6 DOFs force and torque sensors are inserted to measure the interaction forces between each foot and the ground [Tsagarakis et al., 2013]. An inertial measurement unit (IMU) is attached to the robot waist for measuring the body orientation relative to the ground.

3.2 Robotran simulation environment

Robotran is a symbolic software to model and analyze MultiBody Systems (MBS). It is developed by the Multibody research group of the Center for Research in Mechatronics (CEREM) which is part of the Université catholique de Louvain (UCL) in Belgium.

By using Robotran software, symbolic equations of the robot dynamics are generated and can be used for simulations in C++.

The employed Robotran model was developed by [Van der Noot and Barrea, 2013] and further refined in [Van der Noot, 2017b].

The simulated robot was modeled so that it fit with a realistic behaviour. The sensory inputs are restricted to the ones available on the existing robot and noise is added to the torque readings of the simulated robot, according to measurements realized on the CoMan robot in [Van der Noot and Barrea, 2013].

More information on the model development and the use of Robotran can be found in [Van der Noot, 2017b].

3.3 Bio-inspired controller overview

An overview of the initial controller [Van der Noot, 2017b] is given in Figure 3.2. The changes made in the context of this work are presented in the next chapter in Figure 4.1 but the general functioning of the controller is kept.

Three parts comprise the controller : (i) high-level controller, (ii) middle-level neuromuscular model, (iii) low-level impedance controller. The final goal of the controller is to provide appropriate voltages to the robot motors. The time step of the global controller is 1 ms.

3.3.1 High-level controller

Two values constitute the global inputs of the controller : (i) the speed of the robot v_{ref} , (ii) the steering direction h_{ref} . Different specific parameters are modulated through linear and quadratic equations depending on these two inputs and are injected to the middle-level controller.

3.3.2 Middle-level neuromuscular model

The objective of this part is to calculate reference torques τ_{ref} and reference joint positions q_{ref} in order to satisfy the global inputs, while taking into account the state of the robot. The controller is considered as bio-inspired since it aims to generate human-like walking patterns by reproducing the behaviour of virtual muscles.

The torques references are computed from forces of virtual muscles after that the muscles have received appropriate stimulation. A combination of reflexes and a Central Pattern Generator (CPG) allows to obtain the stimulation to apply to the virtual muscles.

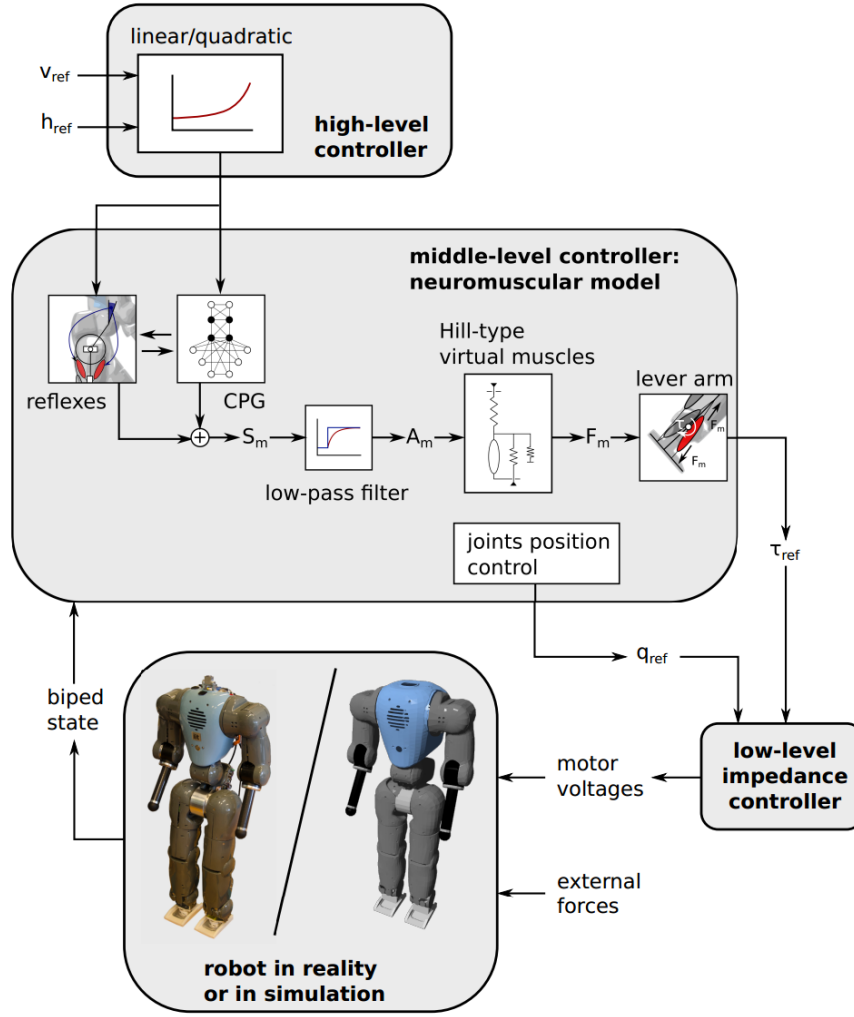


Figure 3.2: Overview of the initial controller [Van der Noot, 2017b].

Virtual Hill-type muscles + Lever arm

Although this work only focuses on the joint torques applied to the lower limbs, Figure 3.3 shows the full muscular configuration of the robot. Each leg is modeled by thirteen virtual Hill-type muscles [Geyer and Herr, 2010].

The Hill muscle model is represented by a set of equations modeling the mechanical response of a muscle unit in response to a simulation. As represented in Figure 3.3c, each muscle-tendon unit (MTU) consists of a contractile element (CE), the active element responsible for the contraction of the muscle, and a series elastic element (SE) that transfers the muscular force to the muscle anchor point. Two other elements, the parallel elastic element (PE) and the buffer elastic element (BE) are used to prevent the muscle from respectively overstretch or collapse.

The MTU length l_{mtu} is computed through geometric relationships with joint positions of the robot, while the contractile element l_{CE} is found by integration on its velocity v_{CE} which depends on l_{mtu} and on the muscle activation A_m . Then, the force of the contractile element and of the series elastic element are computed. This latter corresponds to the force applied by the muscle F_m .

As each joint depends on several MTUs, the contributions of each muscle force related to one joint multiplied by the corresponding lever arm are summed to obtain the total reference torque τ_{ref} to be applied for each joint.

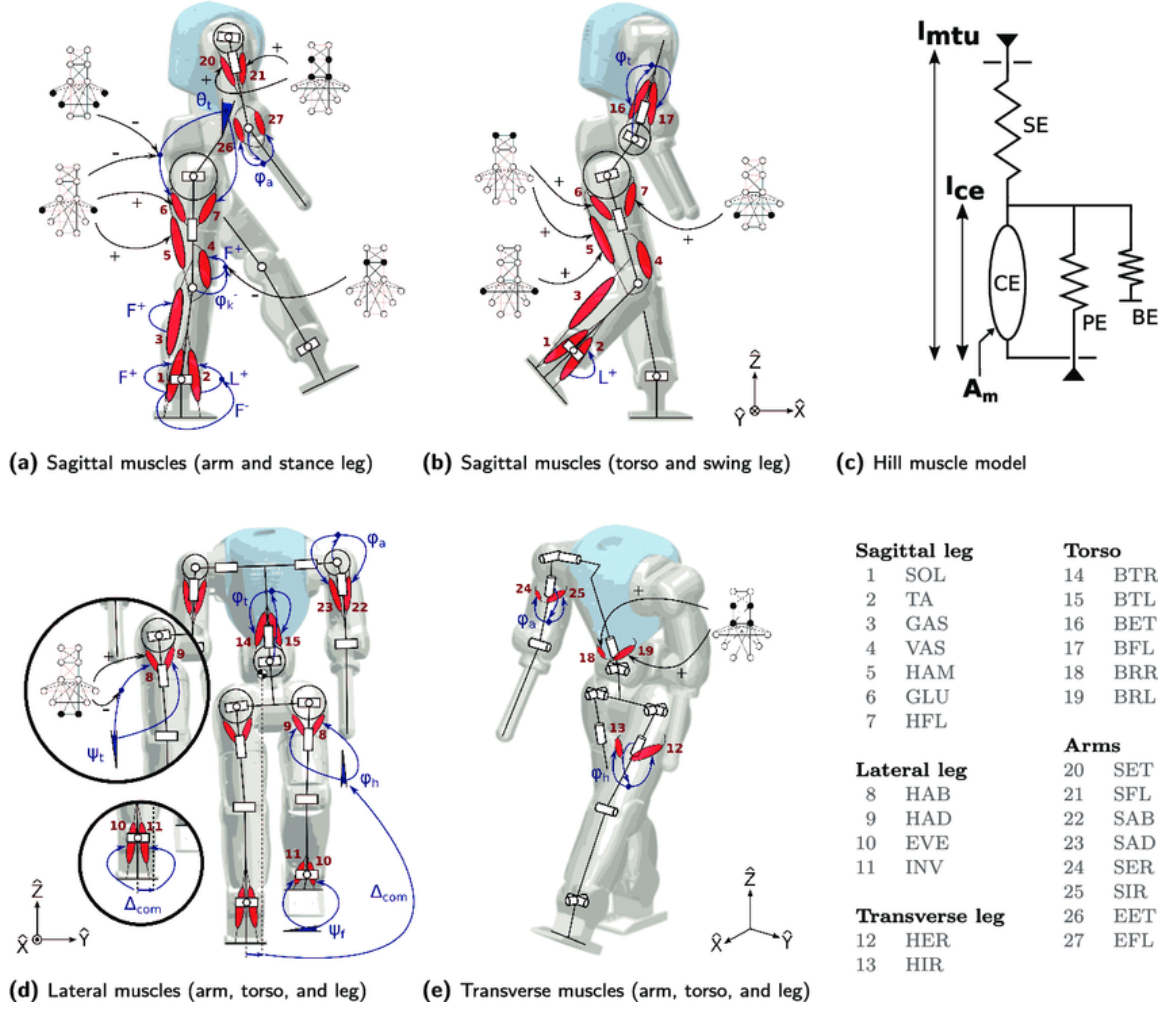


Figure 3.3: Muscular configuration of the robot in the different planes [Van der Noot et al., 2018].

Low-pass filter

To get rid of undesirable high-frequencies, each stimulation S_m goes through a first order low-pass filter to become activation A_m .

Reflexes and CPG model

The muscle stimulations are obtained by combining reflex feedback mechanisms and signals generated by the central pattern generator. A CPG is a neural circuit capable of producing rhythmic patterns without receiving rhythmic inputs [Ijspeert, 2008].

The combination of these two types of signals is such that muscles close to the hips are mainly controlled by CPG signals, while the muscles close to the feet are mainly driven by reflexes since these latter are more impacted by external perturbations such as ground contacts [Dzeladini et al., 2014, Van der Noot et al., 2018].

The CPG model described in [Van der Noot, 2017b] uses a twelve-neurons network of Matsuoka oscillators to produce its outputs. Four neurons are called Rhythm Generation (RG) neurons and eight other neurons, Pattern Formation (PF) neurons. Figure 3.4a shows the CPG network and 3.4b the firing rates of the neurons for straight walking. In Figure 3.4a, x_i is the firing rate, v_i is the self-inhibition modulated by an adaptation constant β_j , η_k are the mutual inhibition factors and u_i is the external input. Equations and more explanations on the model used as CPG can

be found in [Van der Noot, 2017b]. In this work, no modification on the CPG has been achieved.

To briefly give an overview of the CPG, the four rhythm generator neurons are responsible for providing the main frequency and phasing during the gait. They are not influenced by the eight other neurons which are used to form the stimulation patterns for the muscles. A given neuron is active when its neuron firing rate is positive.

To ensure synchronization between the CPG phase and the state of the robot, the neurons RG_1 and RG_3 are triggered at heel strikes, RG_1 at RHS (0%) and RG_3 at LHS (50%), while RG_2 and RG_4 approximately start during mid-stance [Van der Noot, 2017b].

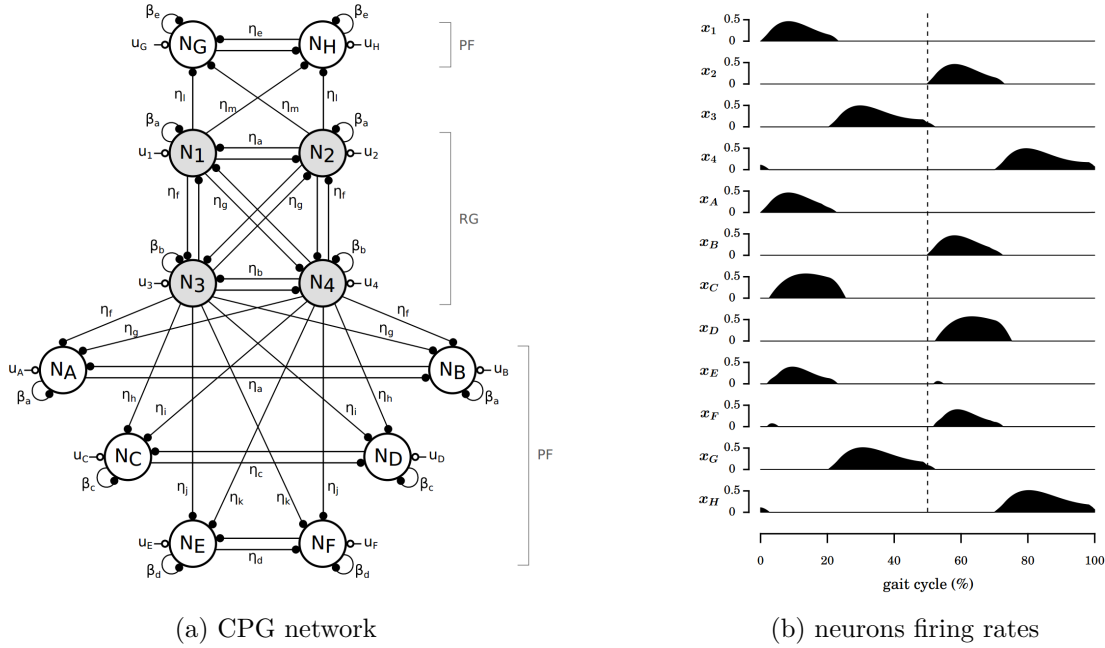


Figure 3.4: Representation of the CPG and the neurons firing rates for straight walking [Van der Noot et al., 2018].

3.3.3 Low-level controller

The low-level impedance controller receives the reference torques and positions and aims to compute appropriate motor voltages that will be applied on the real or simulated robot.

3.3.4 Inverse muscular model

Another module that is not represented in Figure 3.2 but has been developed in [Heremans, 2015, Van der Noot, 2017b], concerns an inverse muscular model that will be used in this work. The objective of this model is to transform desired torques into muscle stimulations. Since each leg possesses six joints but thirteen muscles, the problem is said over-actuated. Indeed, multiple muscles actuate the same joint, making the system redundant.

Two steps are considered in the inverse muscular model. First of all, the desired torques are transformed into muscle forces using optimization criteria. The objective function consists in minimizing the sum of the muscle forces, normalized to their maximum value, while constraints impose that the force should respect $\tau = R \cdot F$, with R the lever arm, and should be bounded between 0 and F_{max} . Indeed, a muscle can only provide positive force since it can only pull. Then, for each muscle, the muscle stimulation is deduced from the corresponding force by reversing the order of the operations used for computing the muscle force from the stimulation.

3.4 Initial robustness against perturbations

Before implementing any balance controller, the initial robustness of the robot against perturbations is evaluated. This preliminary study allows to obtain a reference performance set that can be used as comparison with the additional implemented controller.

3.4.1 Types of perturbations

Perturbations can be of different kinds : (i) pushes, (ii) obstacles on the ground, (iii) slopes, (iv) slippery ground. During this initial analysis, only pushes applied to a robot walking at a speed of 0.65 m/s will be considered to obtain a general picture of its performance.

Description of the pushes

Pushes only occur when the robot is walking for a minimal time of 8 seconds to ensure a steady-state walking.

Pushes are applied on the torso of the robot, horizontally, thus in the transverse (or horizontal) plane, as represented in Figure 3.5 and during a period of 0.2 second. During two pushes, a resting time between 5 to 6 seconds is imposed. The robot is considered as having survived to a push when it has not fallen in the interval of five seconds after the push.

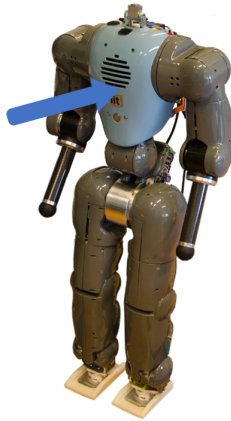


Figure 3.5: Representation of a push applied to the torso of the robot.

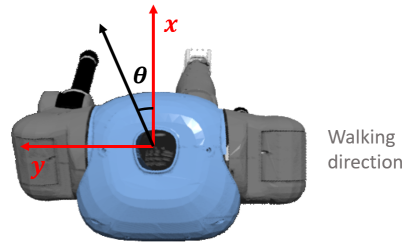


Figure 3.6: Push direction θ in the robot coordinate system.

3.4.2 Falling state

As stated in [Pratt and Tedrake, 2006], a fall is defined when a point on the biped, other than a point on the feet of the biped, touches the ground.

However, in the simulation, another condition is considered. Indeed, due to the limited number of sensors on the robot, detecting that another part of the body than the feet has touched the ground is not a straightforward computation. Therefore, an earlier condition leading always to a fall is used. When the distance between the middle of the waist and the lowest foot drops down below 15 cm or when the ground reaction force becomes too high, then a fall is considered. Most of the time, the first condition is used as threshold and the waist to feet distance can be easily computed with a limited number of sensors.

3.4.3 Falling analysis

Three different parameters have been identified that could influence the robot behaviour against pushes : (i) the amplitude of the pushes, (ii) the direction of the pushes, (iii) the moment in

the gait cycle at which the perturbation occurs. The following analysis aimed to evaluate the influence of these parameters.

Falling probabilities in function of the push amplitude

This first analysis evaluates the falling probability for five different push amplitudes. The push direction and the moment in the gait cycle are random values taken from uniform distributions. Between 100 and 490 pushes were applied for each push amplitude. This number varies because when a simulation is launched, it stops only if the robot has fallen or if 60 seconds have elapsed. Thus, since the robot is more resistant for lower push amplitudes, it can handle a higher number of pushes before stopping.

As shown in Figure 3.7, the larger the push amplitude, the higher the probability to fall. However, the trend between the five amplitudes is not exactly linear since there is a higher probability difference between 30 and 40 N than between 40 and 50 N.

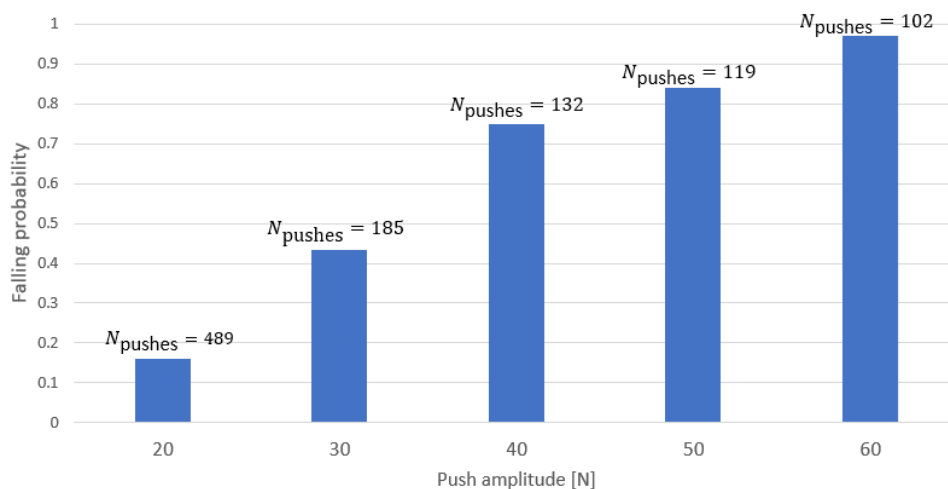


Figure 3.7: Falling probabilities for five push amplitudes. The larger the amplitude, the higher the probability to fall.

Falling probabilities in function of the push amplitude and push direction

Considering that the direction of the pushes can have an influence on the robot response, Figure 3.8 highlights the falling probabilities for different amplitudes and directions. The 360° around the robot have been divided in 16 different directions, each separated by 22.5° . The robot is walking in the x-direction when the push is applied and the moment in the gait cycle is not specified. For each considered case, a minimum number of 20 simulations have been executed. The mean number of pushes is situated around 50.

Globally, we observe that the robot is more sensitive to lateral perturbations than to forward/backward pushes or to pushes oriented in 45° or -45° . Again, larger push amplitudes provoke more falls.

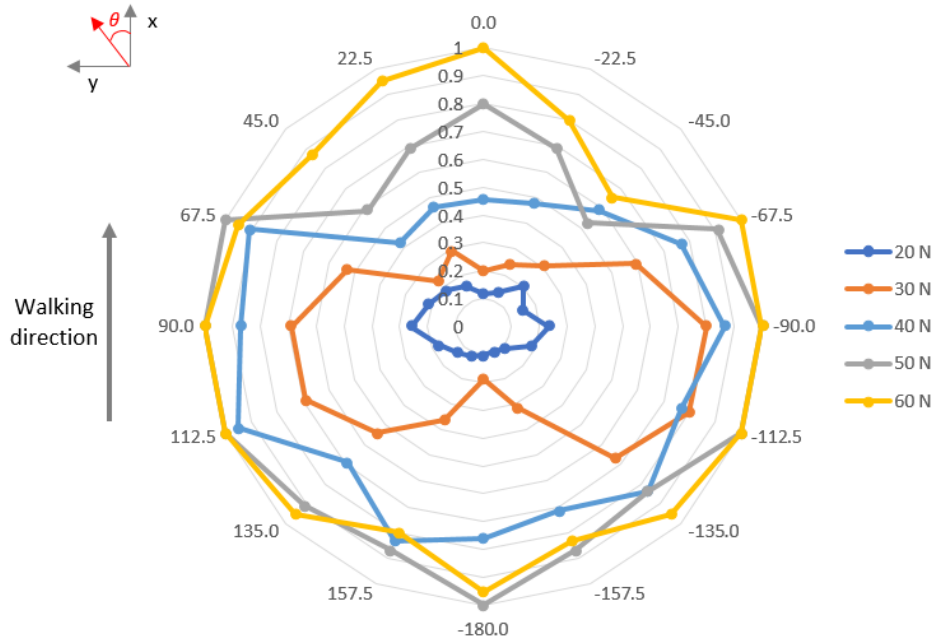


Figure 3.8: Falling probabilities for five push amplitudes and 16 push directions. A fall is more probable for lateral pushes than for forward or backward pushes.

Falling probabilities in function of the push direction and the moment in the gait cycle

For the analysis in function of the push direction and the moment in the gait cycle, only push amplitudes of 30 N are considered.

The gait cycle is divided into 8 moments : right and left heel strike (HS), right and left toe-off (TO), and intermediate moments between each HS and TO.

We observe, in Figure 3.9 that when the perturbation occurs at heel strike or slightly after, the robot is not resistant against lateral pushes nor pushes oriented in the opposite backward direction, i.e. to the left backward for RHS and to the right backward for LHS.

On the contrary, when the push occurs at toe-off or slightly after, the falling probability is globally reduced as the robot is in single-support phase and its swing foot is more able to react at the next foot placement. However, this is not true for pushes around 112.5° for LTO nor around -112.5° for RTO.

Discussion

These initial analyses provide an overview of the performance of the robot against pushes and highlight the situations where the biped is more likely to fall. Lateral pushes induce more falls than forward or backward pushes.

A comparison with human falling probabilities could not have been accomplished since no such analysis on the falling likelihood has been found in the literature for humans. As already mentioned, humans react differently depending on the situation and their experience, making it difficult to obtain global probabilities. However, lateral pushes constitute critical perturbations for the humans as well since they are employed in different studies to evaluate the responses of humans against these pushes [Hof and Duysens, 2013, Hof et al., 2010, Rietdyk et al., 1999].

Finally, since the falling probabilities are not exactly 0 or 1 for each case, this means that other parameters also influence the behaviour of the robot when its stability is challenged. This could come from the previous steps, the trunk orientation, the CoM velocity, ... but this would require a further analysis that will not be realized in this work.

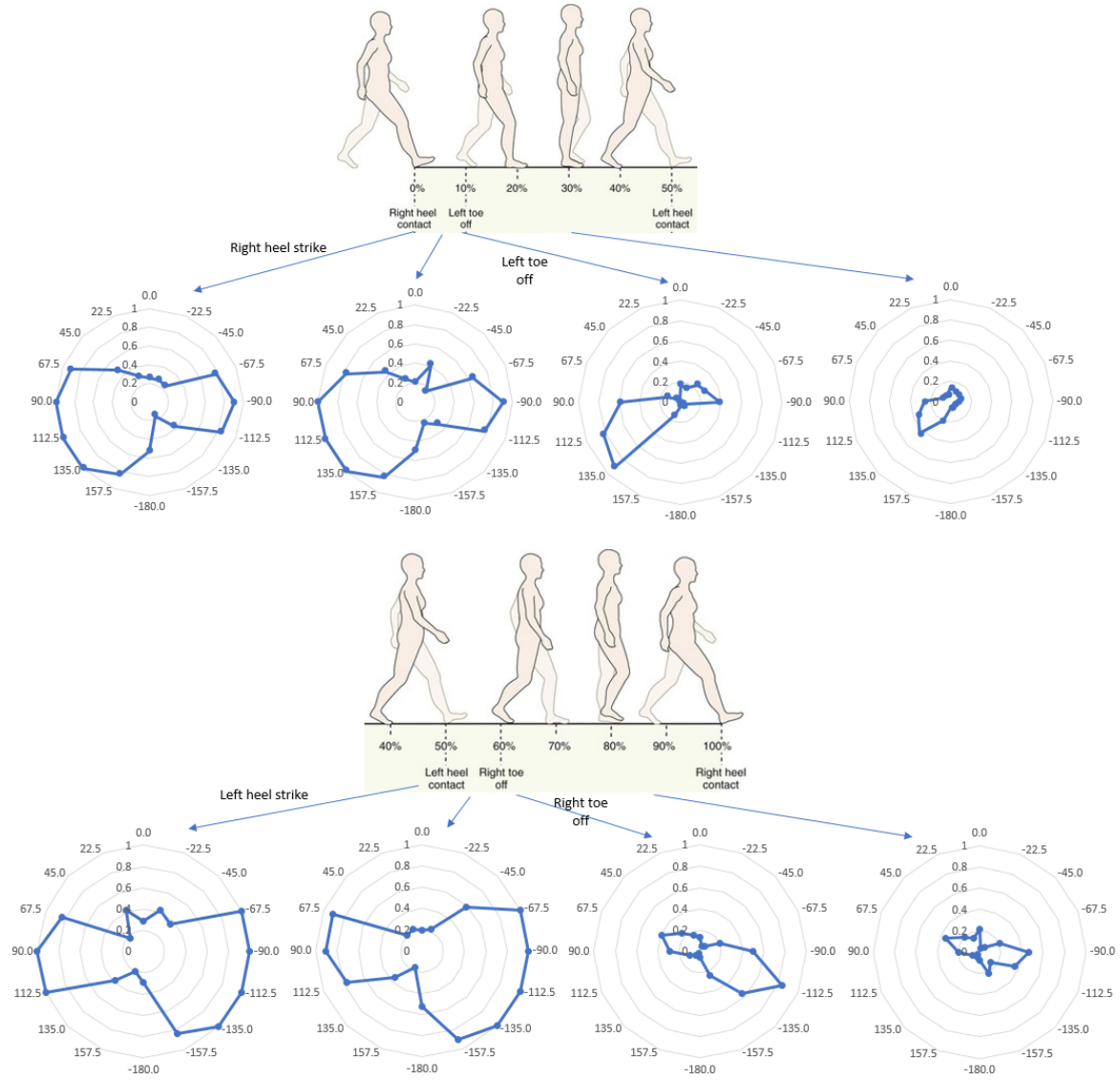


Figure 3.9: Falling probabilities for 30N pushes, 16 push directions and for different moments in the gait. A fall is more probable at heel strike than at toe-off as the foot can adapt more quickly its placement at toe-off.

Chapter 4

General architecture & Perturbation detection

The previous chapters outlined several notions and techniques related to balance recovery taken from the literature, as well as the working environment used for the simulations. By means of the robust controller developed mainly by Nicolas van der Noot [Van der Noot, 2017b], the robot was already able to cope with small and medium perturbations even if no balance recovery was implemented.

This chapter and the next are respectively devoted to the detection of perturbations and to the recovery of balance in order to improve the stability against perturbations.

In this chapter, the general architecture of the additional balance controller will be outlined to provide an overview of the additional balance implementation. Then, each feature will be explained in more details. First, the use of an internal model will be described. Then, a gait cycle phase estimator will be detailed before establishing a stability criterion that will detect perturbations and therefore trigger the use of the additional balance controller.

The balance controller itself will be presented in the next Chapter 5.

4.1 General architecture

In order to visualize the supplementary developments carried out with respect to the additional controller, Figure 4.1 is based on the initial controller architecture from Figure 3.2 and shows the added components. Although each added feature will be explained in details, the following paragraphs give a general description of Figure 4.1.

Internal model The use of internal model is a key element in the following implemented balance method. It consists in a virtual model running in parallel, at the same speed as the real walking robot and simulating the robot facing no external perturbations. Several instances of the same internal model can run simultaneously. In this work, two instances, called "right internal model" and "left internal model", are considered in order to better sense perturbations.

Model selection Each instance of the internal model is considered during the half of a gait cycle. During right foot stance, the right internal model is selected while during left foot stance, the left internal model is chosen.

Reset At regular intervals, data of each instance of the internal model are reset with data from the real robot in order to keep a certain faithfulness regarding the real model.

Two main goals motivate the use of an internal model. (i) Specific data of the real walking robot can be compared with data from a reference walking robot that is not subject to disturbances, in order to detect perturbations applied to the real robot and therefore trigger the balance controller. (ii) When the balance controller is activated, the second goal of the internal model is to use the CoM position of the internal model for attracting the CoM of the real robot in such a way that the real robot regains balance.

More explanations on the internal model will be given in Section 4.2.

Perturbation detection Concerning the triggering of the balance controller, a perturbation is detected when the error of the CoM position, CoM velocity or CoP position between the real robot and the internal model exceeds a certain threshold. This boundary is defined by criteria functions in Section 4.4 which depend on the gait cycle phase. Hence a gait cycle phase estimator is developed in Section 4.3.

Balance controller & inverse muscular model The balance controller computes the joint torques τ as if there is a virtual force attracting the CoM of the real robot to the one of the internal model. Then, an inverse muscular model allows to transform the torques into stimulations S_b which are finally summed to the stimulations required for walking.

The rest of the controller has not been changed. More details on each new sub-block are given in the following sections.

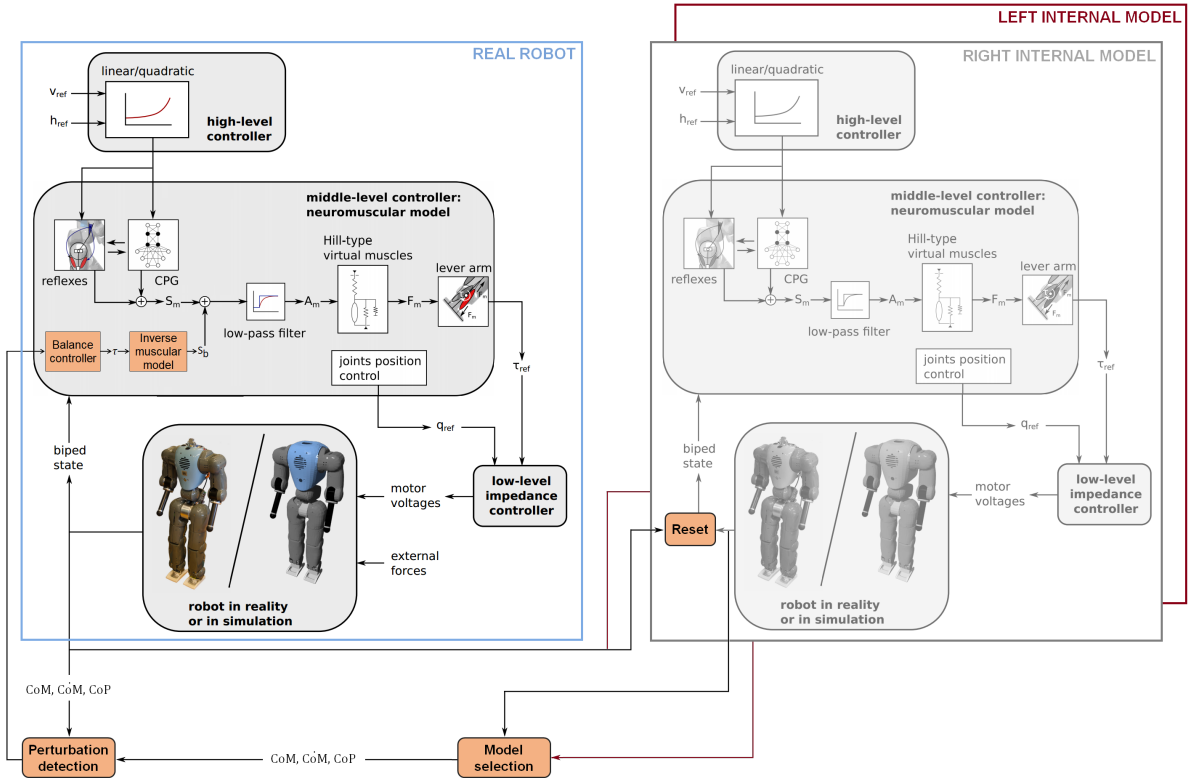


Figure 4.1: General architecture of the controller with the balance controller. An internal model is used to compare data with a walking robot that is not facing external perturbations. If a perturbation is detected, the balance controller is triggered and the resulting stimulations are added to those corresponding to steady-state walking. Adapted from [Van der Noot, 2017b].

4.2 Internal model

An internal model is used for comparing specific data of the real robot with data from a reference walking robot that is not subject to disturbances, represented by the internal model. It is a virtual model running in parallel, at the same speed as the real walking robot and simulating a robot facing no external perturbations. As mentioned in Section 4.1, data of the internal model are periodically reset with data from the real robot in order to make the internal model as close as possible to the real robot. The use of internal models relies on the work developed by Nicolas Van der Noot [Van der Noot, 2017b].

State variables

A set of 184 variables are sent at each reset for updating the model that simulates the real robot. It contains the current position and speed of the real robot waist, the position and speed of the motors and joints, the time, the length of the contractile elements (l_{CE}) of the muscles, ... A complete list is provided in Appendix A.

The accelerations of the joints, motors and the waist position are not provided as their accurate measurements on the real robot are unrealistic. Therefore, because of the noise introduced on the torque readings and incomplete given information, some errors appear between the simulated robot and internal model. For this reason, resets at regular intervals are needed.

When the reset occurs

Reset should not occur during double support phase as inaccuracies can come from the ground contact model which is more sensitive to numerical issues when both feet are in contact with the ground [Van der Noot, 2017b]. Internal models are thus reset during single support phases, typically after a safety margin of 0.05 second after one foot enters a swing phase, i.e. after toe-off.

Two instances of the internal models instead of one for a higher perturbation detection effectiveness

In this work, two instances of the internal model are used in parallel to the real robot. This allows to more effectively detect perturbations at any time. Indeed, resetting data presents a risk in the case of a perturbation occurring just before the reset. In that situation, the error between the robot and the internal model, which increases when an instability occurs, will suddenly drop at the reset. It will thus take more time to observe a significant error or maybe, the difference will never be large enough.

To resolve this problem, two instances of the internal model that are reset at different moments are used. In this manner, one instance is considered during the half period after the reset of the other instance while this latter is considered during the second half period.

Resets occur after toe-off. Typically, one instance is reset after that the right foot is in single stance, i.e. after left toe-off, and is called "right internal model" while the other is reset after that the left foot is in single stance phase, i.e. after right toe-off, corresponding to the "left internal model".

Data from these two models are then selected to provide a global internal model. This considers the values from the right internal model from the left reset time until the right reset time while it takes into account the values from the left internal model during the other half of the time.

A schematic illustrating the use of the right and left internal models is provided in Figure 4.2. This represents an example of the evolution of the CoM y-position of the robot and the right and left internal models. When one instance is selected, corresponding values are shown in thick lines. These values are then concatenated to provide a global internal model used to compare data with the real robot in order to detect perturbations.

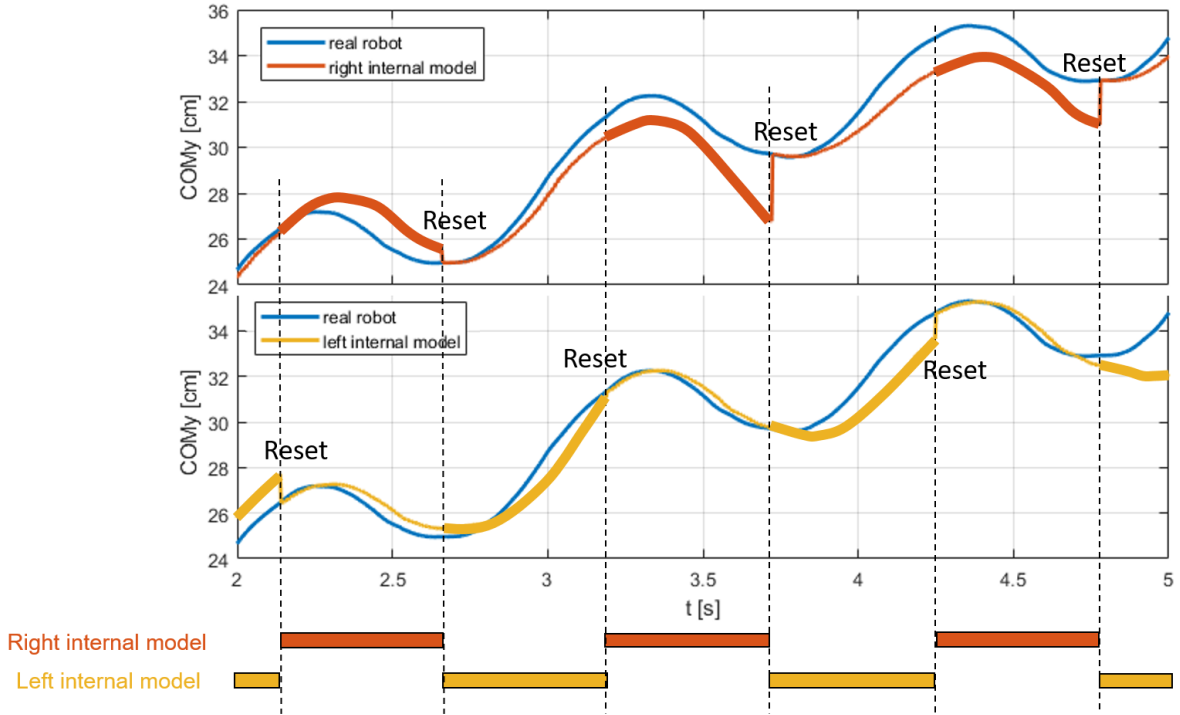


Figure 4.2: Representation of the CoM y-position of the real robot and the right and left internal models. Values of the instances represented in thick line are concatenated to provide a global internal model used to compare data with the real robot.

Concrete example

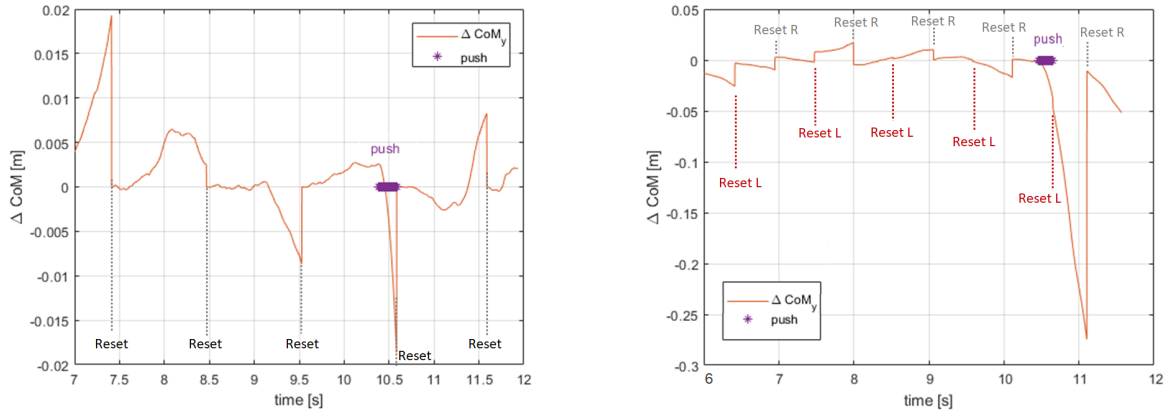
Figures 4.3a and 4.3b illustrate by an example the utility of combining two instances of the internal model. Figure 4.3a compares the CoM y-position of the real robot and one unique internal model while Figure 4.3b considers a global internal model obtained by combining two instances.

A push is applied to the robot in the y-direction around time $t = 10.5$ s. Afterwards, a significant difference in the CoM y-position error should be observable.

However, for the unique internal model in Figure 4.3a, a reset of the data happens at the end of the push, resulting in a drop of the CoM y-position error. No considerable variation in the CoM position error is visible due to a reset occurring too soon after the push. Indeed, the error is kept below 2 cm in absolute values even if the robot fell at the end, at time $t = 12$ s.

When using two instances of the internal model, since the instance that is considered after a reset is the instance that has not been reset, a deviation in the error is observable even if a reset happens at the end of the perturbation. This is illustrated in Figure 4.3b where the error grows up to 20 cm.

To sum up, the use of internal models will constitute a relevant tool for detecting perturbations, which will be detailed in Section 4.4.



(a) considering one unique internal model

(b) using two instances of the internal model that are not reset at the same moment

Figure 4.3: CoM y-position errors between the real robot and the internal model. At the end of the push, data of the internal model is reset.

4.3 Gait cycle phase estimator

In Chapter 2, human analyses highlight that the time at which a push occurs during the gait cycle influences the reaction of the biped. A push applied at heel strike leads more often to a fall than at toe-off. This section reports the development of a gait phase estimator, providing thus a relevant information that will be useful for the detection of a perturbation in the next section.

The considered estimation method relies on the four rhythm generation neurons of the CPG since they have specific values during the gait cycle as mentioned in Section 3.3.2. These data are combined into one value which is then injected into a bijective function to obtain the gait cycle phase.

4.3.1 Rhythm generation neurons

The four rhythm generator neurons of the CPG, described in 3.3.2, provide the main frequency and phasing during the gait. They are thus directly linked with the gait cycle phase.

Although positive firing rates of the neurons are mainly considered in the CPG, positive and negative firing rates will be considered for this gait cycle phase estimator. In this way, more information can be used at each gait cycle phase. Figure 4.4 shows the evolution of the four rhythm generator neurons during a gait cycle.

As already mentioned in Section 3.3.2, the neurons RG_1 and RG_3 are triggered at heel strikes, RG_1 at RHS (0%) and RG_3 at LHS (50%), while RG_2 and RG_4 approximately start during mid-stance [Van der Noot, 2017b].

4.3.2 Estimator method

In the described gait cycle phase estimator, the four signals are combined following several steps leading to a final bijective function determining the gait cycle phase.

First of all, the four neurons are combined to form two signals corresponding to :

$$\begin{aligned} \text{sum} &= RG_1 + RG_3 - RG_2 - RG_4 \\ \text{diff} &= RG_2 + RG_3 - RG_1 - RG_4 \end{aligned}$$

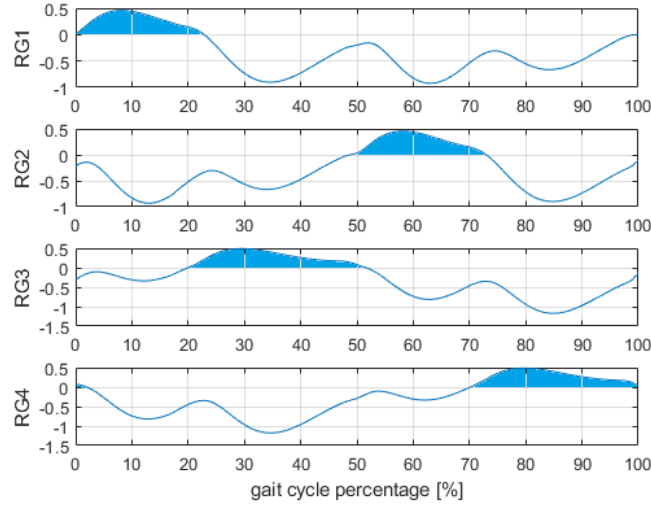


Figure 4.4: Fours rhythm generation neurons of the CPG during one gait cycle. Colored regions correspond to positive values of the signals, representing inhibitory feedback in the Matsuoka oscillators.

These signals are combined such that they form two functions as close as possible to sinusoidal functions phased by $\pi/2$ in order to plot a periodic phase portrait (diff vs sum)¹ and extract the phase angle.

Figure 4.5a represents typical sum and diff functions during a gait cycle while Figure 4.5b shows the corresponding phase portrait of diff vs sum.

The corresponding phase angle ϕ is obtained by taking the arc-tangent of the ratio between diff and sum and adjusted to form a continuous function all along the gait cycle. The following relationships are considered:

$$\phi = \begin{cases} \arctan\left(\frac{\text{diff}}{\text{sum}}\right) - \frac{\pi}{2} & \text{if sum} > 0 \\ \arctan\left(\frac{\text{diff}}{\text{sum}}\right) + \frac{\pi}{2} & \text{otherwise} \end{cases} \quad (4.1)$$

The phase angle ϕ corresponding to the phase portrait is plotted in Figure 4.5c on the x-axis while the gait cycle phase is represented on the y-axis.

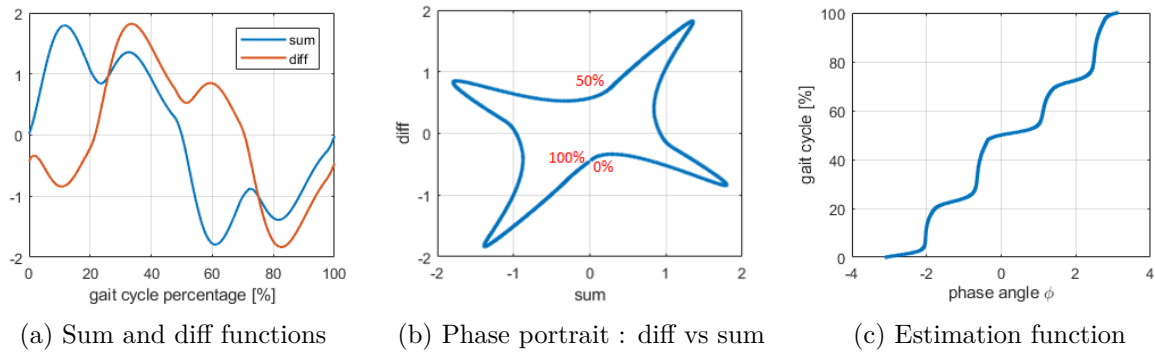


Figure 4.5: Signals composing the gait cycle phase estimator

¹in the case of two perfect sinusoidal functions phased by $\pi/2$, the phase portrait would be a circle.

Through the bijective function from Figure 4.5c, the gait cycle phase can be deduced by the phase angle obtained from the rhythm generation neurons. This function has the shape of four successive sigmoid functions. Therefore, mathematical relationships are determined based on sigmoid functions with a linear adjustment such as followed :

$$\text{gait cycle phase} = \underbrace{A \cdot (\phi - x_1)}_{\text{linear adjustment}} + \underbrace{\frac{B}{1 + \exp(M \cdot (\phi - x_2))}}_{\text{sigmoid function}} + y_1 \quad (4.2)$$

with A and B being amplitudes, M impacting the slope of the sigmoid function, x_1 and x_2 being horizontal shifts and y_1 corresponding to a vertical shift. The mathematical expressions with numerical values can be found in Appendix B.1.

4.3.3 Validation

Tests for validating the described gait cycle phase estimator are based on 40 gait cycles where the robot is walking, out of the walking initialization and facing no perturbations.

Figure 4.6a and 4.6b represent respectively the sum and diff functions of the 40 gait cycles while Figure 4.6c shows the corresponding phase portraits of diff vs sum and Figure 4.6d illustrates the corresponding estimation functions.

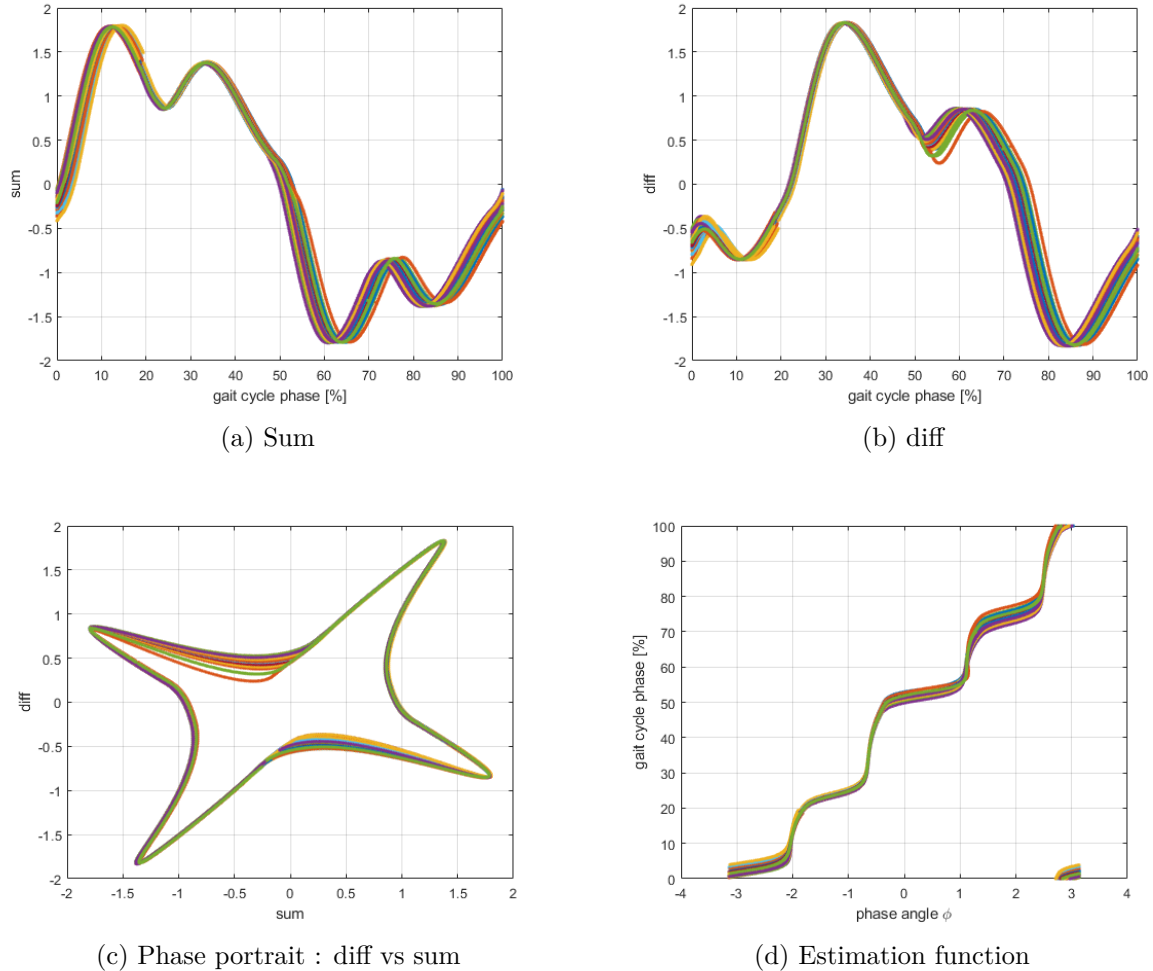


Figure 4.6: Signals used in the gait cycle phase estimator for 40 gait cycles

The gait cycle phase that is used as reference was computed in post-processing of the data. At each RHS, the phase is reset to 0% and each time step corresponds to an increment of $100/T$ %, with T , the time between two consecutive RHS. However, this reference shows inaccuracies since it is only synchronized at RHS and does not consider other information during the gait cycle. A more precise reference could take into consideration the LHS to define the 50% of the gait cycle.

In Figure 4.6a, we can observe that at 0% of the gait cycle phase, the sum is often negative. This results in a phase angle around π instead of $-\pi$ in the estimation function represented by the points in the bottom right corner of Figure 4.6d. However, this does not lead to large errors since the gait cycle phase is cyclic and a gait cycle phase of 100% is reset to 0%.

Moreover, Figure 4.7 shows the difference between the gait cycle phase obtained by the estimator described previously and the reference depending on the gait cycle phase. If the error is higher than 50% in absolute value, which can particularly happen at the beginning or end of a gait cycle, then the error is subtract from $\pm 100\%$.

During each gait cycle, the error is bounded between -4 and 3 %.

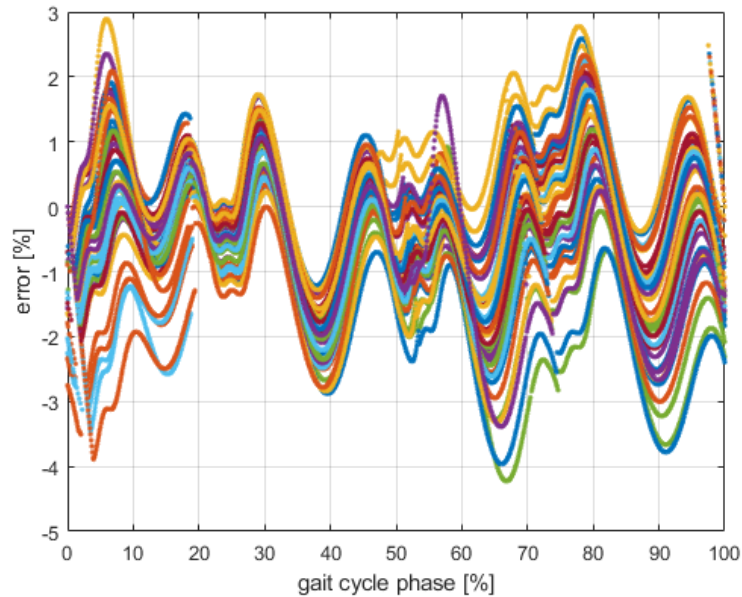


Figure 4.7: Errors between the gait cycle phase estimator and the simplified reference in the absence of perturbation during the walk.

Discussion

The error is thus bounded between -4 and 3%. This is a satisfying result since we only need an estimation of the gait cycle phase. For an improved result, adjustments on the expressions of the sigmoid functions could be achieved in order to reduce slightly the error.

However, inaccuracies in the error computation can also come from the reference phase.

Using the rhythm generation neurons of the CPG thus provides an online estimation regardless of the speed of the robot and takes directly into account information such as foot strikes, which considerably link the estimator to the walking state of the robot.

4.4 Stability criterion based on internal model

The goal of this section is to establish a stability criterion for detecting perturbations on a walking robot that may lead to a fall. In this case, it would trigger a balance controller to recover a stable posture. This controller will be described in the next Chapter 5.

As referred in [Pratt and Tedrake, 2006], it has been recognized that traditional stability margins such as zero moment point (ZMP), extrapolated center of mass (XCoM) described in Section 2.2.2 are not general and insufficient for predicting instabilities under all situations.

Other techniques using the full model of the robot are thus employed to detect falls when walking. As done in [Ruiz-del Solar et al., 2010, Renner and Behnke, 2006], sensor measurements are compared with data extracted from the robot in walking conditions without facing any perturbations. If these differences are too high, a perturbation is then considered.

In this work, balance indication will be based on the center of mass and the center of pressure, and the use of internal model will allow to compare these information with data from normal walking situations. Moreover, the estimation of the gait cycle phase developed in Section 4.3 will be a key information to establish an effective criterion which will depend on the walking state of the robot.

4.4.1 Criteria establishment

Different measurements on the robot may be considered to evaluate its stability. The center of mass (CoM) is a commonly used indicator [Hof et al., 2005, Subburaman et al., 2016], while information involving directly the behaviour of the feet are also used in [Subburaman et al., 2016]. Indeed, the feet constitute a critical element in balance.

As inspired from [Subburaman et al., 2016], data from multiple sensors are used to more efficiently detect a fall than using a single type of sensor. They consider data from CoM, inertial measurement unit (IMU), ZMP and foot pressure sensors and obtain earlier fall prediction and less detection of false perturbations, i.e. detected perturbations whereas no perturbation is applied, mentioned later as *false positives*.

The following criteria compare the CoM position, CoM velocity and CoP position of the robot with the internal model. As explained in Section 4.2, the internal model simulates the walking robot without any perturbations, i.e. under normal conditions. When the errors of the CoM position, CoM velocity or CoP position exceed a certain threshold, a perturbation is then considered.

Figure 4.8 describes the implemented approach for detecting perturbations and establishing the stability criteria.

Perturbations detection From the CoM position, CoM velocity and the CoP position of the robot and the internal model, the norm of these errors are computed as followed :

$$\begin{aligned}
 \text{CoM position error} &= \Delta\text{CoM} = \sqrt{(\Delta\text{CoM}_x)^2 + (\Delta\text{CoM}_y)^2 + (\Delta\text{CoM}_z)^2} \\
 \text{CoM velocity error} &= \Delta\dot{\text{CoM}} = \sqrt{(\Delta\dot{\text{CoM}}_x)^2 + (\Delta\dot{\text{CoM}}_y)^2 + (\Delta\dot{\text{CoM}}_z)^2} \\
 \text{CoP position error} &= \Delta\text{CoP} = \sqrt{(\Delta\text{CoP}_x)^2 + (\Delta\text{CoP}_y)^2}
 \end{aligned} \tag{4.3}$$

These errors are then compared with the values of the stability criteria which vary depending on the gait cycle phase. If the errors are higher than the criteria, then a perturbation is considered.

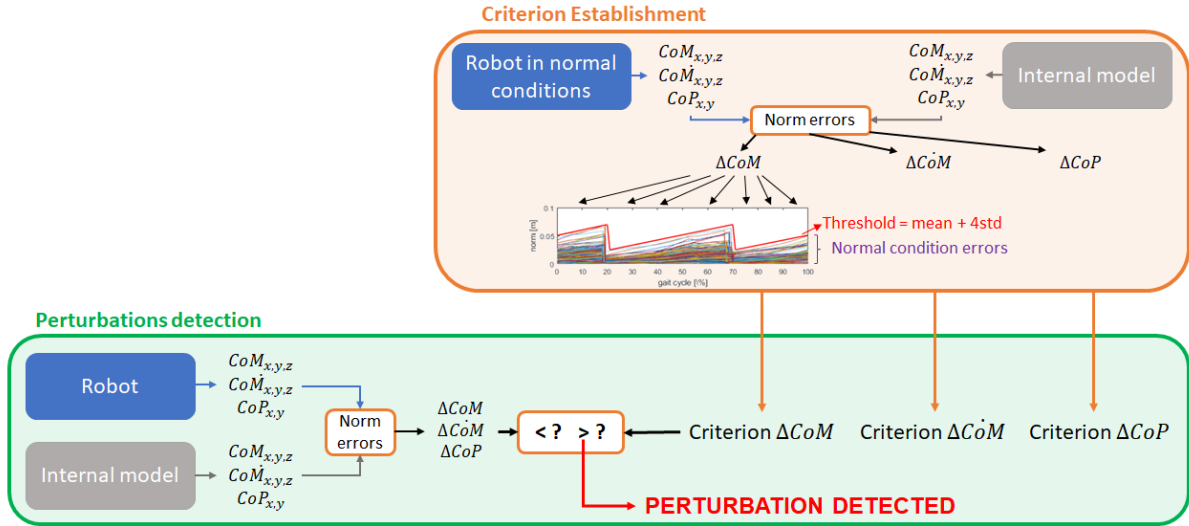


Figure 4.8: Illustration of the implemented approach for establishing the stability criterion and for sensing disturbances.

Criterion establishment When both the robot and the internal model are walking without any disturbances, some errors appear because of torque noise and accelerations that are not provided to the internal model as explained in Section 4.2. These normal conditions errors are not constant during the gait cycle and vary from one gait to another. The thresholds are then placed just above these normal condition errors that are measured.

Several measurements are taken and at each increment of the gait cycle, the mean and the standard deviation of the normal condition errors are computed. Under several statistical assumptions which will be detailed in the next section, the threshold is approximately placed at the value corresponding to the mean + 4 standard deviations. In this way, the threshold is situated above the majority of the errors that are observed in normal conditions.

The determination of the threshold is critical because it must be placed sufficiently high above the normal condition errors in order to avoid detecting a perturbation whereas there is none and sufficiently low such that an instability is sensed as fast as possible.

Finally, considering the threshold as a function of the gait cycle phase, the following constraints should be respected :

- developing a continuous and smooth criterion : since the gait cycle phase estimator presents slightly inaccuracies, a discontinuous or rough criterion would lead to high errors in the detection,
- imposing similar values at 0 and 100 % of gait cycle to ensure continuity,
- allowing a detection as fast as possible,
- ensuring the highest falling detection rate as possible.

Detecting instabilities even if there are none is not the most critical since the balance controller can be triggered even for small perturbations. It will become critical if the controller is applied during a limited time after detecting perturbations. In that respect, assuming that a false perturbation is detected before a real one and that the controller is active only during a short

time after the detection, followed by a dead moment before being active again, the controller will thus be triggered after the false perturbation and inactive during the real instability. This should be avoided.

However, the most important is that all perturbations leading to a fall should be detected.

4.4.2 Determination of the criteria expressions

CoM computation

As defined in Section 1.1, the CoM position (resp. velocity or acceleration) is computed as a weighted sum of the position (resp. velocity or acceleration) of the local centers of mass from each segment of the robot. In Robotran, these computations are symbolically generated.

CoP computation

From the definition in Section 1.1 as well, the CoP is the point location of the vertical ground reaction force vector. It is computed as a weighted sum of the interaction forces created by the contact model between the robot feet and the ground. The ground contact model (GCM) consists in a mesh-based model with 20 points under each foot. This number was selected as a compromise between computational cost and accuracy of contacts with uneven grounds [Van der Noot, 2017b]. It was developed very similarly to the one implemented in [Geyer and Herr, 2010].

Normal condition errors

When the robot is walking without facing any perturbations, errors between the robot and the internal model are observable notably in terms of CoM position, CoM velocity and CoP position. Figure 4.9a depicts the errors in the x-, y-, z- directions as well as the norm of the CoM position in function of the gait cycle phase, Figure 4.9b for the CoM velocity, while Figure 4.9c shows the errors in the x- and y- directions and the corresponding norms of the CoP. 150 gait cycles are represented.

Errors are not constant along the gait cycle. At around 20% and 70% of the gait cycle, the errors drop, as mainly visible on Figure 4.9a. It is due to the switch of the internal model which happens a safety time after toe-off (left TO being around 10-12% and around 60-62% for right TO). After the internal model is switched, the errors generally increase until the next switch.

Concerning the velocity norm error, higher errors are observable at approximately 5-10% and 55-60% of the gait cycle. These moments correspond to the second halves of the double support phases which typically occur from 0 to 10% and from 50 to 60%. Indeed, after a foot has hit the ground, the robot sometimes reacts with a higher body movement, mainly on the torso, which produces higher errors between the robot and the internal model.

Finally, huge errors are observed for the CoP position from roughly 0 to 9 %, from 47 to 59% and from 97 to 100 % of the gait cycle. These intervals are situated around double-support phases and correspond to the errors induced by heel strikes. Indeed, the feet of the robot and the internal model do not touch the ground at the same moment resulting in different positions of the CoP. Indeed, the CoP is situated on the rear foot for one and on the front foot for the other. Moreover, inaccuracies on the CoP can come from the ground contact model during the double-support phases. Higher errors are thus induced during double-support phases than during single-support phases.

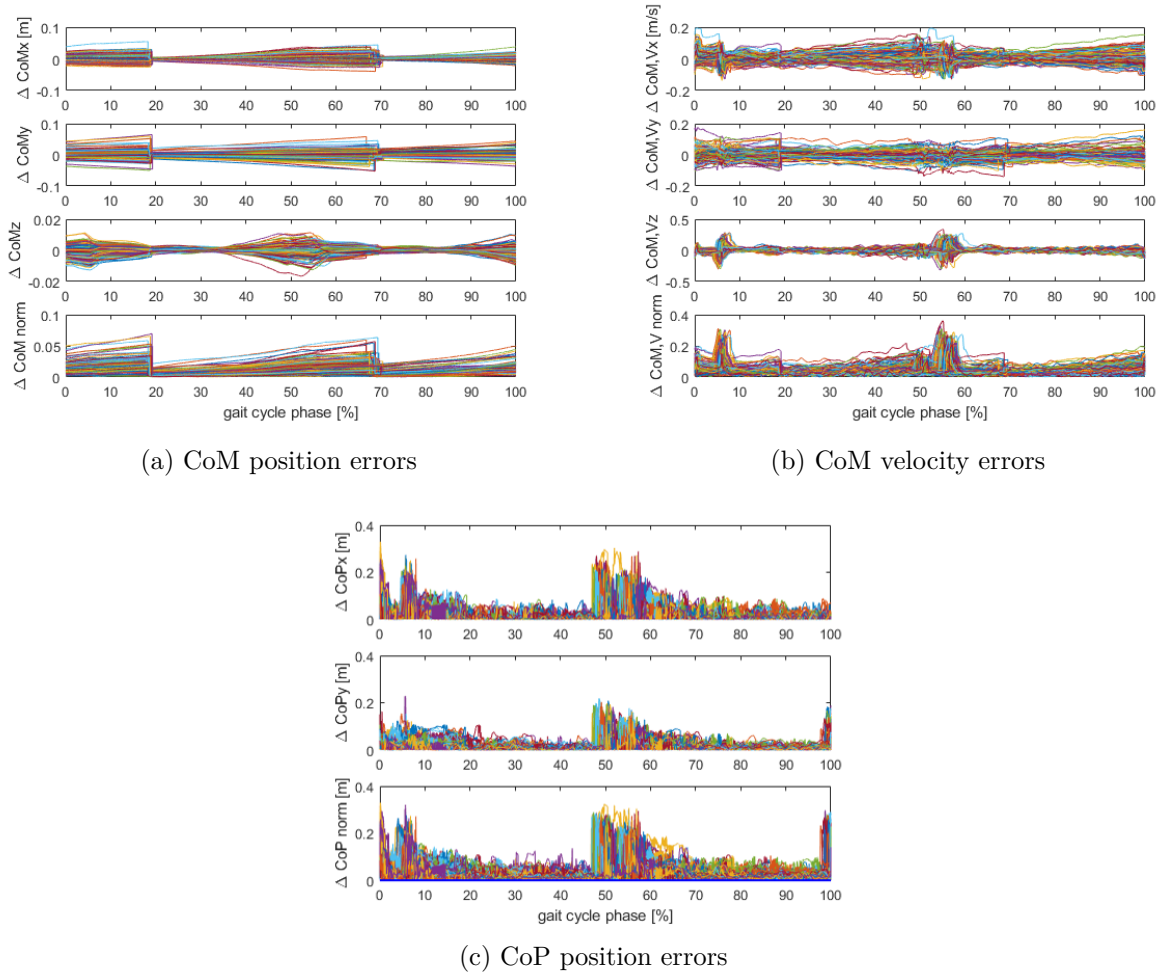


Figure 4.9: Normal condition errors for walking robots facing no perturbation in function of the gait cycle phase. The stability criterion will then be placed just above the norm of the errors.

Data distribution

With all these data, a small statistical analysis is achieved. Taking the errors over all the gait cycles, Figure 4.10 represents the distributions of the errors. For each direction, the CoM position error stays under 6 cm while for the CoM velocity, the error is under 0.3 m/s and under 32 cm for the CoP position.

Regarding the errors in the x-, y-, and z- directions, the distributions have the shape of Gaussian functions centered around zero. Indeed, the errors between the robot and the internal model are sometimes positive, sometimes negative.

However, the norms of the errors do not follow normal distributions. Indeed, according to expressions 4.3, the norm errors correspond to the square root of the sum of squares of normal distributions. Therefore, they roughly follow chi distributions with three degrees of freedom for the CoM position and velocity and with two degrees of freedom for the CoP position. For rightly following chi distributions, the errors in each direction should be normalized to obtain standard normal distributions before computing the norms. Given that statistical analyses do not constitute the main objectives of this master thesis, no normalization will be effectuated and approximated chi distributions will be considered.

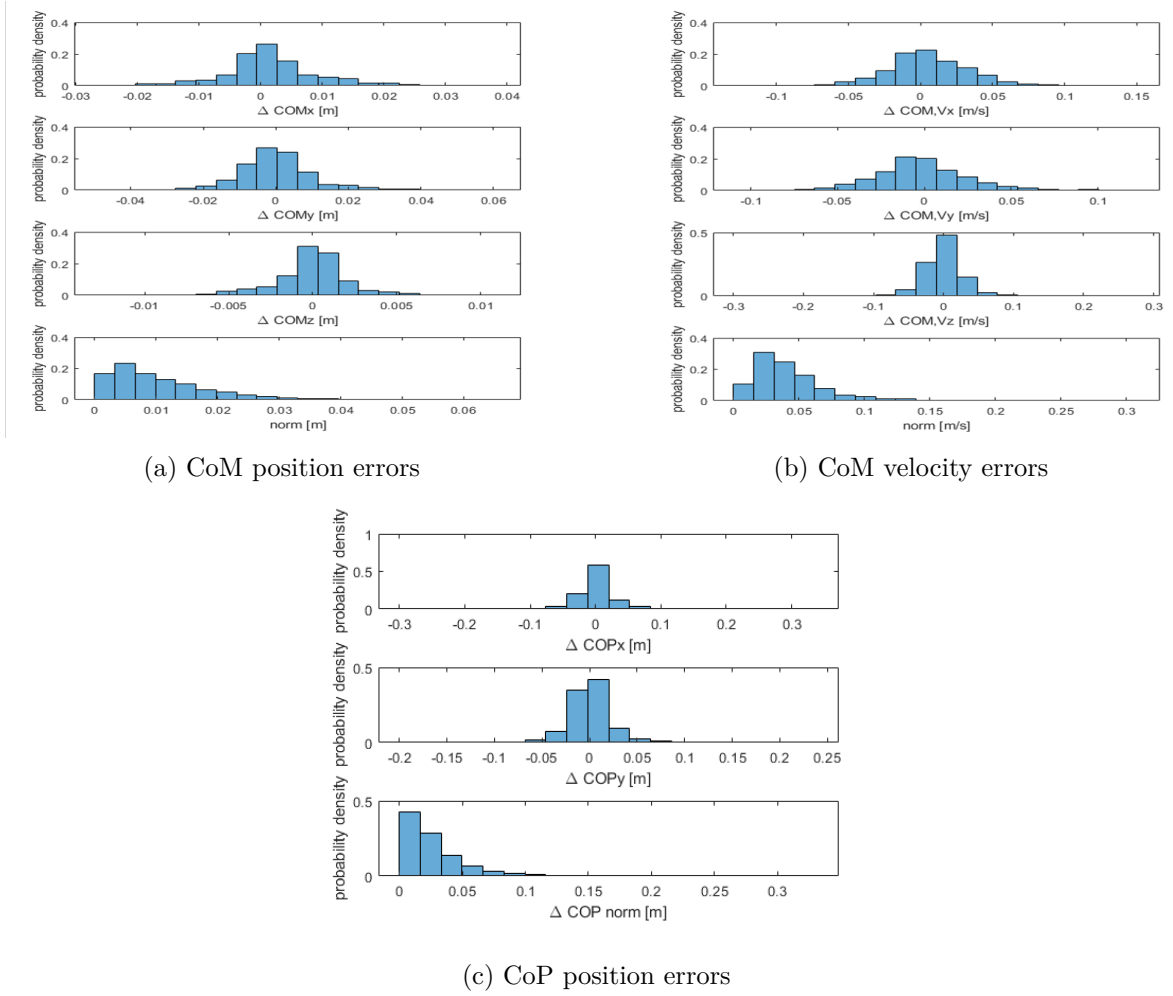


Figure 4.10: Distributions of the walking errors between the real robot and the internal models when no perturbation occurs.

Criterion expressions

As explained in Section 4.4.1, the thresholds for determining a perturbation are placed just above the normal condition norm errors represented by the coloured curves in Figures 4.9.

For that purpose, the gait cycle is divided into several increments. We suppose that the distribution of the errors at each increment follows the same behaviour as the global errors, i.e. a normal distribution for the errors in each direction and a chi distribution for the norms.

To place the threshold above a certain percentage of the data, a normalization of the data in each direction would be required in order to consider normalized values of chi distribution from statistic tables.

Instead, the mean and the standard deviation of the norm errors at each increment are calculated and the threshold is approximately placed at the value corresponding to the mean + 4 standard deviations. These sets of values are then approximated by functions of the first and second orders to from the final criteria.

Figure 4.11 depicts the set of values corresponding to the mean + 4 standard deviations in green while the red curves correspond to the criteria. The mathematical expressions of the criteria can be found in Appendix B.2.

If the distribution was a normal distribution, considering the mean ± 3 standard deviations

would be sufficient to place the threshold beyond 99.7% of the normal condition errors. However, the blue curves in Figure 4.11 show that it is not the case in this situation because of the non-normalized chi distribution. But assuming 4 standard deviations allow to have a threshold situated above roughly 99% of the errors that are observed in normal conditions.

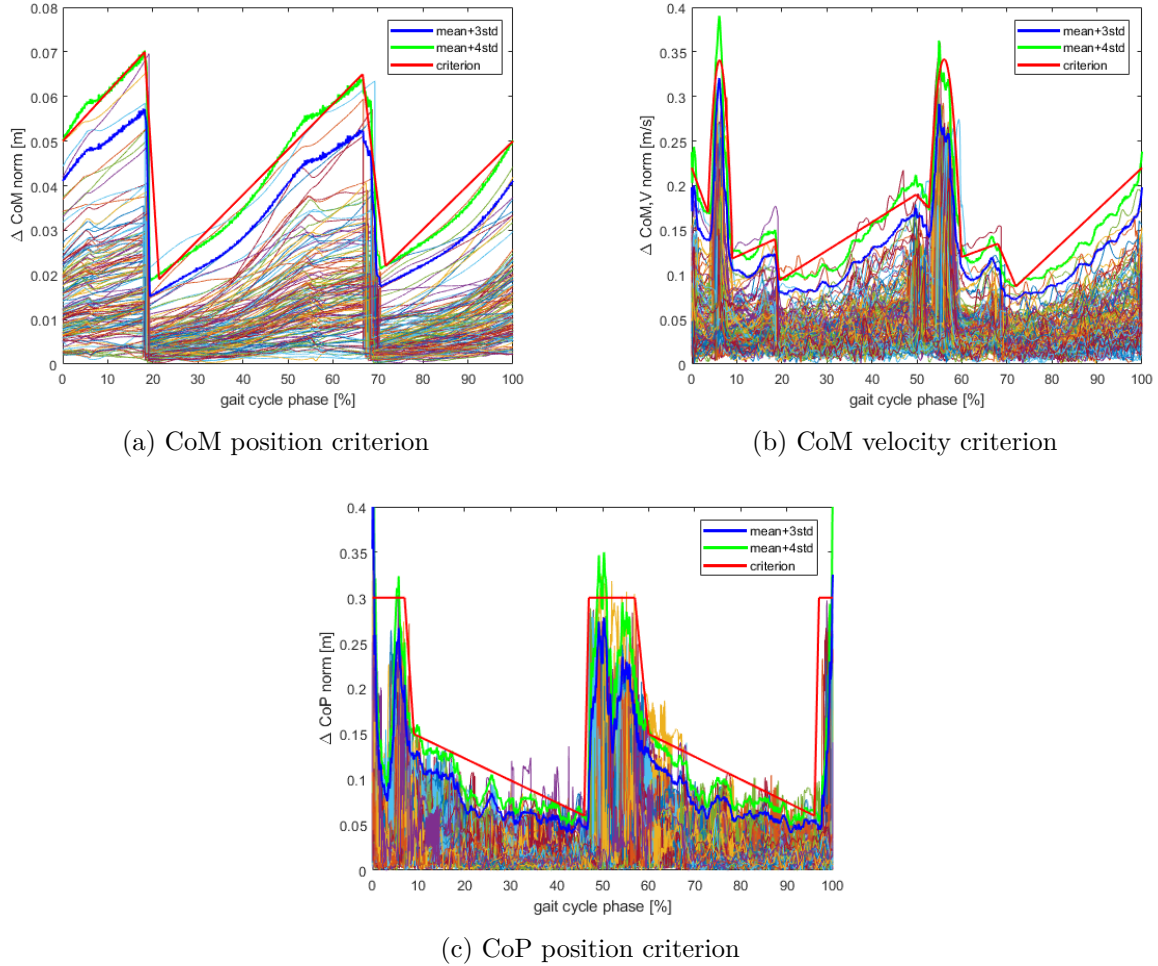


Figure 4.11: Established criteria are represented in red, approximating the set of values corresponding to the mean + 4 standard deviations in green.

In each case, the criteria are continuous as expected and repetitive patterns are observable on the two halves of the gait cycle. Indeed, the two steps composing a gait cycle are similar and the fact that one step happens during right stance and the other one during left stance does not considerably influence the CoM and CoP errors.

An assessment of the performance of these criteria is provided in Section 4.5.

4.4.3 Other tested criteria

Even if the developed stability criteria consider already three different information, other parameters such as the CoM acceleration or the ground reaction force errors were analyzed to constitute eventual criteria.

Criterion based on the CoM acceleration error

Analyses reveal that the CoM acceleration errors are not good indicators of perturbations for several reasons: (i) the joint accelerations are not provided to the internal models at each reset,

(ii) at each strike, the CoM velocity undergoes high variations as shown in Figure 4.9b which leads to high acceleration variations.

[Subburaman et al., 2016] did not consider the acceleration as well, affirming that the acceleration data have a erratic distribution.

Figure 4.12 shows the CoM acceleration for walking robots without any perturbation (in blue), with perturbation and that did not fall (in green), and that fell (in red). For the green and red curves, a push is applied at time $t = 0$ s with the same amplitude of 30N, in the same direction of 180° and at the same moment in the gait cycle, at right heel strike.

However, all curves are strongly overlapping which complicates the extraction of any threshold.

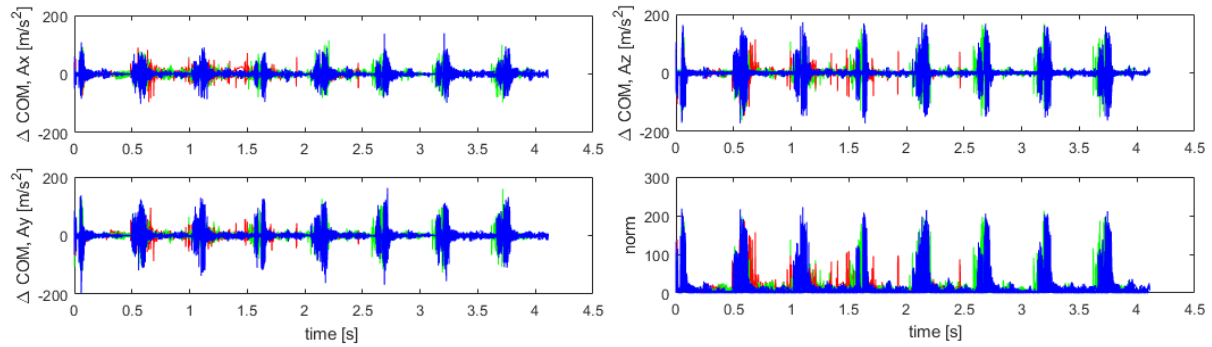


Figure 4.12: CoM acceleration errors. Blue curves correspond to walking robots without any perturbation, green curves to perturbed robots that did not fall and red curves that fell. Pushes occur at relative time $t = 0$ s. The curves are considerably overlapping so no criterion can be made.

Criterion based on the ground reaction force error

As done in [Subburaman et al., 2016] with foot pressure sensors, a criterion based on the ground reaction force errors has been tested.

However, under normal conditions, the ground reaction forces encounter high variations during the gait cycle and especially at heel strikes. Figure 4.13 represents the magnitude of the ground contact force for each foot (first and third graphs) and the corresponding peak values (second and last graphs) of 150 gait cycles.

The average peak force magnitude is situated around 5000 N while during single-support phases (intervals between 13 and 46% and between 63 and 96% of the gait cycle), the force is estimated around 300 N in average, which is more than 10 times smaller than the peak force.

[Van der Noot, 2017b] explains that these force peaks at foot strike come from the developed ground contact model and more precisely from the stiff spring-damper contact model that is used. No such peaks were measured on the real robot during experiments in [Van der Noot and Barrea, 2013].

Due to these highly variable peak forces and inaccuracies from the ground contact model, no criterion has been achieved on the ground reaction force errors.

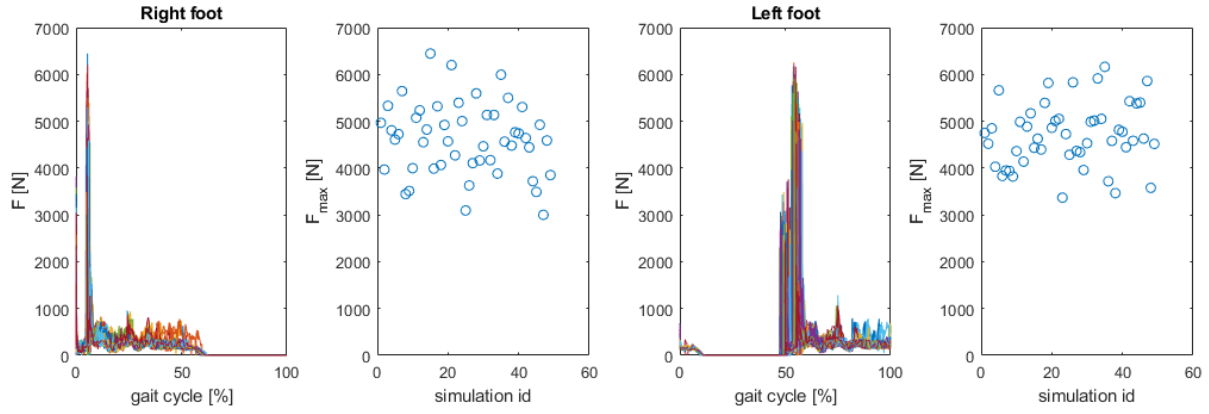


Figure 4.13: Magnitudes of the ground contact force for the right foot (first graph) and left foot (third graph) and corresponding peak values for 150 gait cycles (second and last graphs). The robot is walking without any perturbation.

4.5 Assessment of the performance of the criteria

This section is devoted to the assessment of the stability criteria based on the CoM position, CoM velocity and CoP position errors, developed in Section 4.4 and represented in Figure 4.11. In order to assess the performance of the described criteria, two types of perturbations are employed : (i) pushes applied on the torso of the robot, (ii) a slippery ground.

The following elements are evaluated in both cases : (i) **falling detection rate**, corresponding to the detection percentage of robots that fall, (ii) **perturbation detection rate**, corresponding to the detection percentage of robots that are perturbed, resulting in a fall or not, (iii) **false positive detection rate**, corresponding to the detection percentage of perturbations whereas the robot is not perturbed (see below for more explanations), (iv) **detection delay**, corresponding to the time between the perturbation and the detection.

The false positive detection rate evaluates, in other terms, the probability of occurrence that a criterion is exceeded whereas no perturbation is applied to the robot. These average measurements are evaluated on 250 sequences of gait cycles. As soon as a criterion is exceeded, even during one time step only, the corresponding sequence is considered as a false positive.

4.5.1 Detection of pushes applied on the torso

This section is devoted to the assessment of the developed criteria when the robot faces pushes applied on the torso, as described in Section 3.4.1.

Temporal evolution

A first assessment analyzes the temporal evolution of the CoM position, CoM velocity and CoP position errors with respect to the corresponding criteria.

Figure 4.14 gives an overview of these values in the situation where a 30 N-push is applied at time $t = 0$ s while the robot is already walking and occurring at left heel strike. The push direction is random.

The comparison between the described thresholds plotted in red with the perturbed robots in coloured lines shows that all perturbed robots exceed the criteria.

Red curves corresponding to the criteria do not exactly overlap as some robots reach heel strike earlier or later, which results in shifted gait cycle phases.

For the CoM position errors, most errors exceed the criterion at the 'drop' of the criterion, which corresponds to the switch of the internal model. Indeed, just before the change of internal model, the normal condition errors present higher variations, constraining to place the threshold at higher values. Since the threshold just after the switch is lower, it is thus more easily exceeded.

Concerning the CoP position errors, the detections occur when the threshold is maximum, corresponding to the double-support phases. Indeed, when pushed, the robot adapts the placement of its feet resulting in higher differences particularly at foot-placement. After 1.5s, some CoP errors become higher than 1m. This is due to the fall of the robot and to the fact that no foot touches the ground anymore.

A deeper analysis considering different situations is provided in the next subsections.

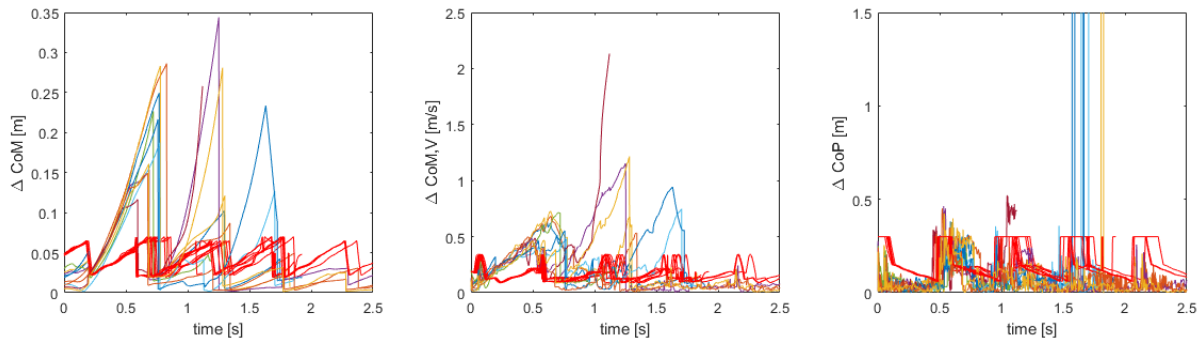


Figure 4.14: Red curves correspond to the criteria, coloured curves to perturbed robots. A push is applied at relative time $t = 0$ s while the robot is already walking. All errors exceed the criteria.

General performance

Table 4.1 gathers the results of the assessment based on pushes of four different amplitudes. Between 60 (for 50 N) and 100 (for 20 N) pushes are applied in random directions and random moments in the gait cycle to obtain a global performance of the criteria.

Figure 4.15 graphically shows the detection delays in box plots².

	CoM position criterion				CoM velocity criterion				CoP position criterion			
push amplitude [N]	20	30	40	50	20	30	40	50	20	30	40	50
falling detection rate [%]	100	100	100	100	100	100	100	100	100	100	100	100
perturbation detection rate [%]	100	100	100	100	100	100	100	100	100	100	100	100
false positive detection rate [%]	7.5				6.1				21.5			
average detection delay [s]	0.34	0.26	0.25	0.20	0.20	0.12	0.10	0.08	0.52	0.43	0.44	0.40
av. detection delay [% g. cycle]	32.11	25.47	24.95	19.01	19.30	11.45	9.80	7.42	51.12	42.67	42.93	39.14

Table 4.1: Assessment of the CoM position and velocity and CoP difference criteria

As desired, all perturbations and thus all falls are detected by each criterion. The falling detection rates and perturbation detection rates are maximum.

Concerning the false positive detection rates, the rate of the CoP criterion is considerably higher compared to the other criteria but a compromise between a small false positive detection rate

²On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol. Points are drawn as outliers if they are greater than $q_3 + w \cdot (q_3 - q_1)$ or less than $q_1 - w \cdot (q_3 - q_1)$, where w is the maximum whisker length, and q_1 and q_3 are the 25th and 75th percentiles of the sample data, respectively [Matlab, 2017].

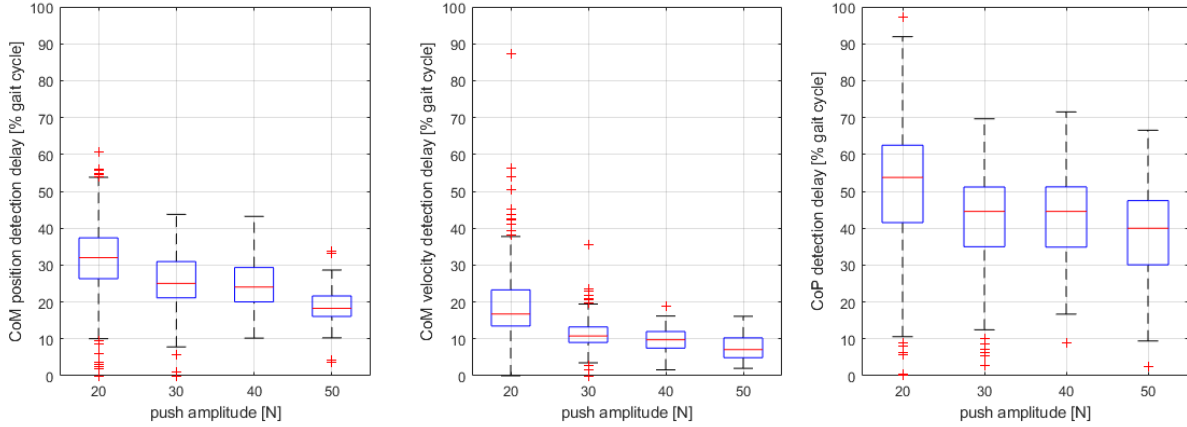


Figure 4.15: Box plots of CoM position and velocity and CoP difference detection delays in percentage of gait cycle for 20, 30, 40 and 50N push amplitudes.

and a short detection delay made difficult the establishment of the CoP criterion.

As observed from the values in Table 4.1 and in Figure 4.15, the higher the push amplitude, the shorter the detection delay. Indeed, the robot reacts more against large perturbations, thus differences between the robot and the internal model are more quickly measurable.

The CoM velocity criterion is the fastest criterion. For pushes greater of 30N, it takes less than 20% of the gait cycle for detecting which is, in average, less than 13% compared to the CoM position criterion and is less than 32% compared to the CoP position criterion.

Finally, the detection delays for 20N-pushes contain many outliers, but since most of these outliers correspond to robots that did not fall, this is not significant.

Discussion These results show that the CoM position and velocity criteria allow a fast detection with a low number of false positives. Indeed, it takes less than around 20% of a gait cycle for detecting a push. Although different assumptions were taken to establish these criteria, considering the mean + 4 standard deviations of the data provides a good compromise between a low false positive rate and a fast detection. Moreover, the CoM velocity criterion is more reactive than the CoM position, which suits with human data from [Hof et al., 2010], observing that after a push, the CoM position is only slightly changed while the CoM velocity is altered.

However, the CoP criterion presents a lower performance. Indeed, the detection takes on average more than twice the delay of the CoM position and CoM velocity criteria and the CoP criterion is three times more likely to detect false positives. This could come from the fact that the CoP errors between the robot and the internal model are less significant after a push, but considering lower thresholds would imply more false positives due to the variation of the errors in normal conditions. Moreover, the CoP is highly depending on the foot placement which occurs at heel strike. The robot must thus sometimes wait until the next double-support phase to see a significant error in the CoP position. Under other types of perturbations, the CoP error may give other results as shown in Section 4.5.2 with a slippery ground.

As this analysis considers random push directions and random moments in the gait cycle, a deeper analysis evaluating the influence of these parameters is provided in the next subsection.

Influence of the push direction and the moment in the gait cycle

In Section 3.4, an analysis on the initial robustness of the robot was carried out. Results showed that the falling probabilities notably depend on the push amplitude, push direction and on the moment in the gait cycle at which the perturbation is applied.

This subsection analyzes the influence of the push direction and the gait cycle moment on the detection delays. All pushes have an amplitude of 30 N. 4 push directions, -180° , -90° , 0° and 90° , and 4 moments in the gait cycle, right heel strike, left toe-off, left heel strike and right toe-off are considered.

Figure 4.16 shows the detection delays for each case, with a varying number of pushes between 25 and 100 pushes. Each line in the figure corresponds to a criterion.

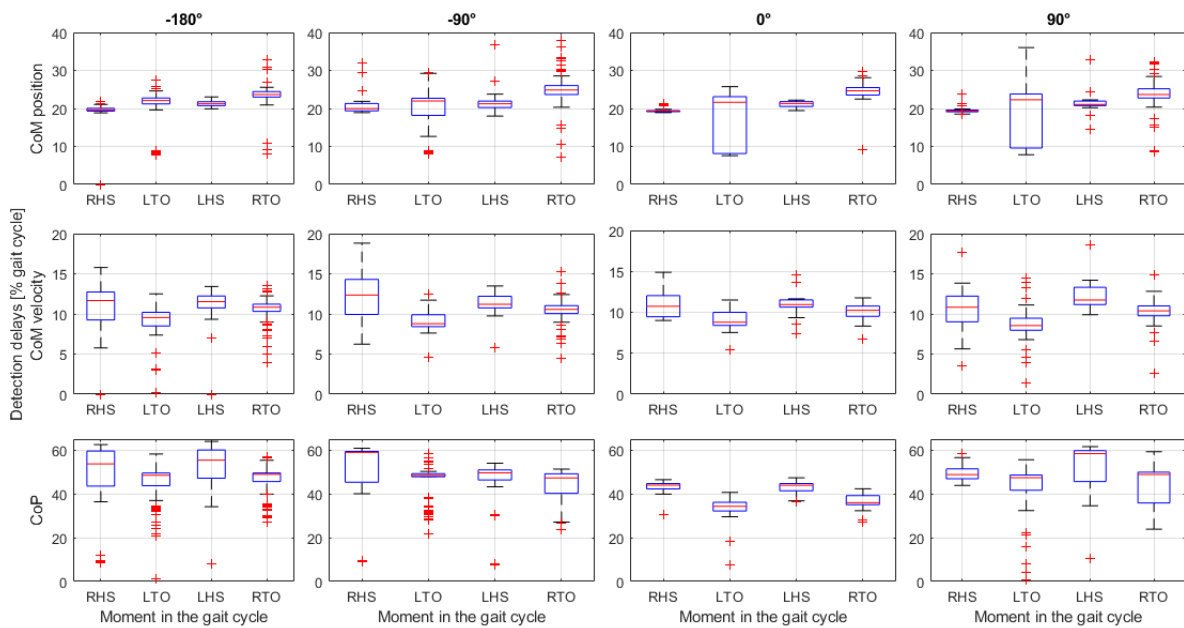


Figure 4.16: Influence of the push direction and the gait cycle moment on the detection delays. 4 push directions (-180° , -90° , 0° and 90°) and 4 moments (right heel strike, left toe-off, left heel strike and right toe-off) are considered.

First of all, for each criterion, the median values of the delays for similar moments in the gait cycle but different push directions are roughly identical. This general trend thus shows that the detection delay does not significantly depend on the push direction.

At toe-off (LTO and RTO), detection delays are globally smaller and outliers are also more frequent compared to heel strikes (RHS and LHS). This can be explained by the fact that, as shown in Figure 3.9, the robot recovers more often balance when the push occurs at toe-off. Indeed, at toe-off, the robot begins a single-support phase. It is thus more able to react and make higher movements than during double-support phases. Its better performance results thus in smaller detection delays and in more varied delays due its greater reactivity.

Concerning the CoM position criterion, delays are highly regrouped at heel strikes (RHS and LHS). It takes 20% of the gait cycle for detecting, which corresponds to the moment when the criteria 'drops' due to the switch of the internal model. It is in accordance with the observations from Figure 4.14.

Finally, box plots of the CoP criterion show a different behaviour. In general, box plots are more extended and unlike the rest, pushes in the 0^{circ} direction are faster detected. Indeed, in the case of forward pushes, the robot places its foot earlier resulting in higher CoP position errors and thus faster delays.

Discussion Several observations were made from Figure 4.16. The following points sum up the observations :

- The detection delay does not significantly depend on the push direction (except for the CoP criterion with forward pushes).
- The robot reacts better to pushes occurring at toe-off, leading to faster detections.
- The detection delays depend on the moment when the push is applied, due to the reaction of the robot and to the phase-depending criteria. Indeed, the normal condition errors present higher variations at some moments, i.e. before the switch of internal model for the CoM position criteria, constraining to place the threshold at higher errors.

Finally, in any cases, the CoM velocity presents the fastest detection.

4.5.2 Detection of slippery grounds

Other kinds of perturbations besides pushes may occur. In this subsection, the criteria are evaluated in the case of slippery ground.

For this purpose, while the robot is walking, the static and kinetic friction coefficients of the ground contact are suddenly changed to correspond to a slippery ground. The friction coefficients approximately correspond to rubber on ice : $\mu_{kinetic} = 0.1, 0.12, 0.15$ and $\mu_{static} = 0.3$ ³[Physics & Astronomy Online, 2012, Wikibooks, 2017].

Table 4.2 and Figure 4.17 gather the results of the assessment with a slippery ground. Generally, the robot walks 1 or 2 steps before the slippery ground affects its gait. That is why the average detection delay is more than 100% in some cases, corresponding to a detection after more than two steps (or a complete stride). Then, the robot falls all the time.

	CoM position criterion			CoM velocity criterion			CoP criterion		
	$\mu_{kinetic}$	μ_{static}		$\mu_{kinetic}$	μ_{static}		$\mu_{kinetic}$	μ_{static}	
	0.1	0.12	0.15	0.1	0.12	0.15	0.1	0.12	0.15
	0.3			0.3			0.3		
falling detection rate [%]	100	100	100	100	100	100	100	100	100
perturbation detection rate [%]	100	100	100	100	100	100	100	100	100
false positive detection rate [%]	7.5			6.1			21.5		
average detection delay [s]	0.80	1.02	1.52	0.68	0.71	1.08	0.77	0.94	1.50
av. detection delay [% g. cycle]	78.45	100.01	146.98	65.37	68.02	103.51	75.32	90.93	142.99

Table 4.2: Assessment of the criteria with slippery ground

The smaller the kinetic friction coefficient, the more slippery the ground and the faster the perturbation detection.

The CoM velocity criterion is again the fastest criterion. The CoP criterion is, in this case, faster than the CoM position criterion, which shows the utility of a criterion directly based on the behaviour of the feet.

³In normal condition, the friction coefficients of the ground are : $\mu_{kinetic} = 0.8$ and $\mu_{static} = 0.9$, roughly corresponding to rubber on concrete.

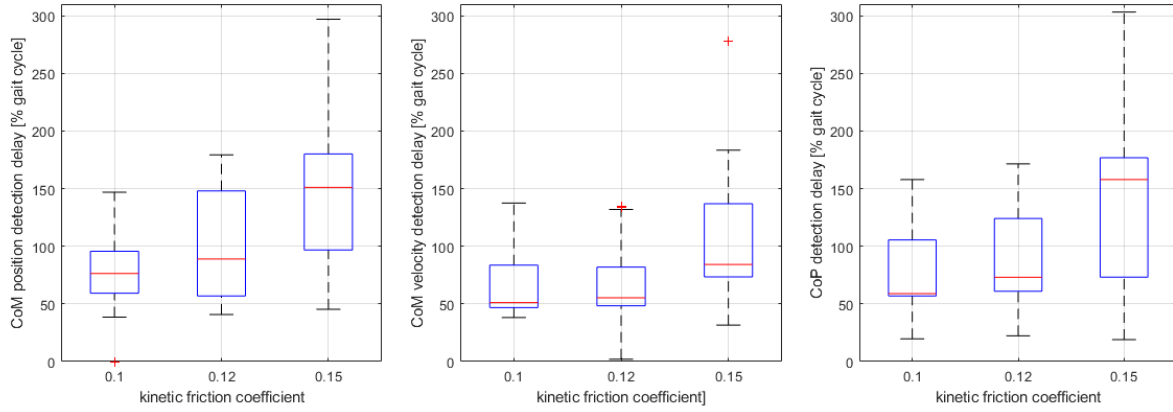


Figure 4.17: Box plots of CoM position and velocity and CoP detection delays in percentage of gait cycle for 0.1, 0.12 and 0.15 kinetic friction coefficient.

Finally, the detection delays in the case of a slippery ground are larger than for pushes applied on the robot. They also present widely various values, as shown in Figure 4.17, with the wide widths of the box plots. This is mainly due to the fact that the robot takes more time before being affected when the ground has changed and that it adopts various movements to tackle the slippery ground. Nevertheless, all falls are detected.

This assessment with a slippery ground shows the utility of having a criterion directly based on the behaviour of the feet, although its performance is lower in the case of pushes. However, under both types of perturbations, the CoM velocity criterion shows the best results. Due to the various behaviours of the robot when facing a slippery ground, no analysis of the balance controller under this perturbation will be carried out in the next chapter.

4.6 Discussion

In this chapter, three criteria have been developed to detect perturbations by comparing specific values of the robot and internal model : (i) CoM position, (ii) CoM velocity, (iii) CoP position. As soon as one criterion is exceeded, the balance controller developed in the next chapter will be active. More precisions about the activation and deactivation of the balance controller are given in Section 5.1.

As stated in [Subburaman et al., 2016], using multiple data provides better detection than using a single type of sensor. Indeed, it allows earlier detection and less instances of false positives. This is what has been achieved in this chapter with the use of three different information. The criterion based on the CoM velocity gives better detection compared to the other criteria. However, in the case of other types of perturbations, the performance of the criteria may be different as illustrated with slippery ground where the CoP criterion was more efficient than the CoM position criterion. This is why all criteria will be considered.

Finally, the establishment of the criteria would not have been possible without the use of internal model and the gait cycle phase estimator described earlier in the chapter.

Chapter 5

Balance controller

In the previous chapter, the development of stability criteria depending on the gait cycle phase was developed. Using internal model was a key element for comparing specific data of the robot with data from a reference walking robot that is not subject to disturbances.

As mentioned, an overrun of the stability criteria will trigger the use of the following balance controller which will be running in parallel to the CPG-reflexes rules needed for walking. The stimulations deduced from both modules will then be summed, then injected in the virtual Hill muscles and corresponding torques will be applied to the robot through adequate motor voltages.

This chapter is devoted to the balance controller itself which constitutes the key feature for improving the stability of the robot. Implemented methods are first presented, then results and discussion are provided to analyze the impact of this additional balance controller.

5.1 Triggering of the balance controller

Previously, three criteria have been developed to detect perturbations by comparing specific values between the robot and the internal model : (i) CoM position, (ii) CoM velocity, (iii) CoP position. However, the precise activation and deactivation of the balance controller have not been specified yet.

By facility and efficiency, as soon as one criterion is exceeded, the balance controller becomes active during the period where the criterion is surpassed. The activation will thus be instantaneous and no smooth transition will be employed since the balance controller only provides slight adjustments to the reference joint torques. This allows also to make the balance controller completely active as soon as possible after the detection of a perturbation.

However, the activation of the balance controller will be of a limited duration, imposed to a maximum of 0.35s, less than a half a gait cycle. When activated, the two instances of the internal model are not switched so that the CoM of the robot is more attracted by the active instance of the internal model and so that the CoM error is not suddenly decreased.

Once deactivated, a dead time will be imposed during which the balance controller can not be triggered in order to let the robot regain a 'normal' gait, i.e. controlled only by the reflexes-rules and CPG.

A condition must nevertheless be satisfied to allow the dead time : the criterion must be exceeded during a minimal time of 0.015s and by 50% of its value. This limits the impact of false positives. Indeed, false positives occur during a short time (in average 0.007s) and generally exceed slightly the criterion, unlike real perturbations which exceed more than twice the value of the criterion as illustrated in Figure 4.14. Making the balance controller inactive because of a false positive

should be avoided.

On several simulations with a total of 60 perturbations, around 40 false positives have been detected. But only two have triggered a dead time on the balance controller.

5.2 Implemented methods

In Chapter 2, different types of controllers for recovering balance were introduced, notably position-based controllers with trajectory planning and force-based controllers. In this work, a force-controlled balancing system is used. It provides several advantages since (i) it can quickly adapt to sudden external changes unlike position-based controllers; (ii) force control enables amongst others to realize velocity-based and position-based control; and finally, (iii) no force measurements are needed at the contacts with the environment [Hyon et al., 2007].

This section develops the implemented force-based controller. Two methods with a similar general principle are presented but results are only carried out on the second method.

5.2.1 General principle

The balance controller is implemented such that, to recover balance, it aims to (i) compute a force attracting the CoM of the robot to the CoM of the internal model and (ii) determine the joint torques of the lower limbs to achieve it.

To attract the CoM of the robot to the one of the internal model, a virtual force is computed according to a feedback proportional-derivative (PD) control law. An illustration of this force f_v is given in Figure 5.1.

Then, the joint torques required to applied this virtual force are derived from a force-torque relationship through the Jacobian. This allows to avoid calculating inverse dynamics.

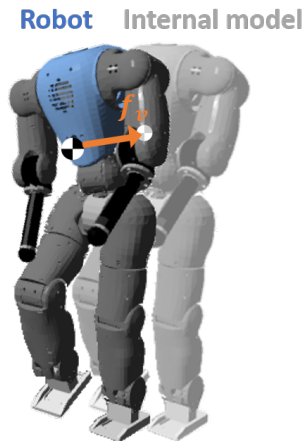


Figure 5.1: Representation of the virtual force f_v attracting the CoM of the robot to the one of the internal model.

5.2.2 Virtual control force

The virtual force $\mathbf{f}_v$ applied to the CoM of the robot corresponds to a feedback proportional-derivative control law as followed :

$$\mathbf{f}_v = \mathbf{K}_P(\mathbf{r}_{\text{CoM, ref}} - \mathbf{r}_{\text{CoM}}) + \mathbf{K}_D(\dot{\mathbf{r}}_{\text{CoM, ref}} - \dot{\mathbf{r}}_{\text{CoM}}) \quad (5.1)$$

with $\mathbf{K}_P$, $\mathbf{K}_D$ the PD gain vectors, $\mathbf{r}_{\text{CoM}}$ and $\dot{\mathbf{r}}_{\text{CoM}}$ the CoM position and CoM velocity vectors, and $\mathbf{r}_{\text{CoM, ref}}$ and $\dot{\mathbf{r}}_{\text{CoM, ref}}$ the reference CoM position and velocity vectors obtained from the kinematics of the internal model. This law makes the CoM of the robot attracted by the CoM of the internal model like a spring-damping system.

5.2.3 Quasi-static force-torque relationship

In the case of a manipulator robot with N actuators, as represented in Figure 5.2, the joint torques $\boldsymbol{\tau}$ required to exert a desired force $\mathbf{F}$ at its end-effector are given by :

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \quad (5.2)$$

where $\mathbf{J}$ is the Jacobian matrix associated to the differential relationships between the joint displacements $d\mathbf{Q}$ and the end-effector displacements $d\mathbf{X}$ such that $d\mathbf{X} = \mathbf{J}d\mathbf{Q}$. Friction in the joint mechanisms and gravity are neglected.

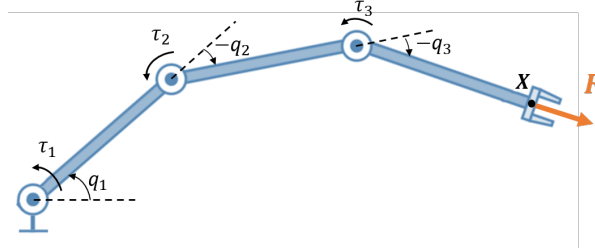


Figure 5.2: Mechanism with 3 actuators applying a force F at its end-effector.

Indeed, following the proof from [Spong et al., 2006] let $\delta\mathbf{Q}$ and $\delta\mathbf{X}$ be virtual displacements of the joints and end-effector respectively. The principle of virtual work states that the system is in equilibrium if, and only if, the virtual work δWork vanishes for arbitrary virtual displacements that satisfy geometric constraints. The virtual work is given by :

$$\delta\text{Work} = \boldsymbol{\tau}^T \delta\mathbf{Q} - \mathbf{F}^T \delta\mathbf{X} \quad (5.3)$$

Using the relationship on the Jacobian, $\delta\mathbf{X} = \mathbf{J}\delta\mathbf{Q}$, we obtain :

$$\begin{aligned} \delta\text{Work} &= \boldsymbol{\tau}^T \delta\mathbf{Q} - \mathbf{F}^T \mathbf{J} \delta\mathbf{Q} \\ &= (\boldsymbol{\tau}^T - \mathbf{F}^T \mathbf{J}) \delta\mathbf{Q} = 0 \end{aligned}$$

For any arbitrary virtual displacements that satisfy geometric constraints, we have thus :

$$\boldsymbol{\tau}^T - \mathbf{F}^T \mathbf{J} = 0 \Rightarrow \boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}$$

Analogy with the robot

The relationship (5.2) can be applied to the robot by comparing the manipulator to the robot in one-support phase and considering the center of the stance foot as the base, the leg as the linkage system and the CoM as the end-effector.

During double-support phases, the base is placed on the front foot and the back leg is not considered until the next heel strike. At each instant, the joint torques of the stance leg are thus calculated as if it is a manipulator system from the stance foot to the CoM.

To sum up, the first method consists in :

1st method :

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f}_v \quad (5.4)$$

with τ the joint torques, $\mathbf{J}$ the Jacobian of the CoM with respect to the stance foot and $\mathbf{f}_v$ the virtual force from (5.1).

Limitations

However, [Park, 2011] highlights limitations of this force-torque relationship. He explains that for floating base robots, or for underactuated systems, the relationship (5.2) does not hold. Indeed, in the case of a floating base, motions imply the generation of kinetic energy and the system can therefore not be considered in static equilibrium. The virtual work would correspond to the kinetic energy and thus $\delta \text{Work} \neq 0$.

Concerning the underactuation, a system is considered as *underactuated* if "it is not able to command an arbitrary instantaneous acceleration " [Tedrake, 2018] and therefore, it cannot be commanded to follow arbitrary trajectories. [Park, 2011] uses a counter-example to demonstrate that the force-torque relationship is incorrect but explains also that this inaccurate mapping has no critical effect on the performance of the robot control.

In the case of the robot, at each double-support phase, the base is suddenly changed from the back foot to the front foot. Moreover, during one-support phase, the center of the stance foot where the base is fixed moves since the foot slightly rolls on the ground.

Due to these limitations and poor obtained outcomes, results on this 1st method will not be presented in Section 5.4. This first method allows nevertheless to introduce the second method.

5.2.4 Force-torque relationship based on ground forces

The second method to obtain the joint torques to apply to the lower limbs has been implemented, inspired from [Hyon et al., 2007] and from a master thesis [Heremans, 2015].

A relationship between control forces and joint torques is still employed through the Jacobian but the problem is now considered in another direction. Figure 5.3 gives an illustration of the situation. The base of the linkage system is placed at the CoM while the forces at the end-effector correspond to contact forces applied on different points of the foot in stance with the ground. The resultant of the contact forces is called the *Ground Applied Force* (GAF) and is opposite to the ground reaction force (GRF). The contact forces thus represent the forces that the robot applies to the environment through the contact points.

Again, in double-support phases, the foot in stance is assimilated to the front foot.

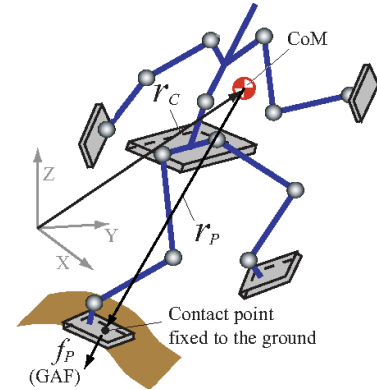


Figure 5.3: Coordinate reference with one contact point on the ground. Adapted from [Hyon et al., 2007].

Transformation of the virtual force into ground force.

As a first step, let consider one contact point on the foot where the GAF is applied, as shown in Figure 5.3. This point corresponds to the center of pressure (CoP). The first objective is to determine the GAF f_P in order to apply a given force f_v to the CoM.

Gravity is neglected, assuming that it is already taken into consideration in the stimulations needed for walking. Muscles stimulations that will be computed from this balance controller are supposed to be only slight adjustments needed to apply a force f_v to the CoM.

[Hyon et al., 2007] suggest to define the origin of the frame (x, y, z) at the center point between the feet for simplicity. The origin is thus moving according to the state of the robot. Assume

$\mathbf{r}_C = [x_C, y_C, z_C]^T$ the position vector of the CoM in the frame (x, y, z) and $\mathbf{r}_P = [x_P, y_P, z_P]^T$ the position vector from the CoM to the contact point.

From the Lagrangian formulation of the dynamics of the robot and assuming a quasi-static motion, the relationship between the GAF and the virtual control force is given by :

$$f_P \approx f_v \quad (5.5)$$

The detailed proof is provided in [Hyon et al., 2007] where gravity must be neglected.

Optimal distribution of the ground force to multiple contact points

Instead of using one contact point which coincides with the CoP, multiple contact points are considered for several reasons : (i) the contact points are situated at fixed locations on the feet, unlike using one contact point whose location moves. The contact forces at each contact point vary depending on the distribution of the GAF. (ii) An exact calculation of the CoP location is avoided such that this method can be employed without force sensors on the feet.

Multiple contact points are mainly used in [Hyon et al., 2007] for considering points on both feet and on the hands, in order to notably handle double-support phases and human-humanoid interaction situations through the hand.

The objective is now to determine how to wisely distribute the GAF to individual contact forces at each contact point.

Let consider a total of α contact points defined as $\mathbf{r}_S = [\mathbf{r}_{S1}, \mathbf{r}_{S2}, \dots, \mathbf{r}_{S\alpha}]^T \in \mathcal{R}^{3\alpha}$ where each vector component is $\mathbf{r}_{Si} = [x_{Si}, y_{Si}, z_{Si}]^T \in \mathcal{R}^3$, corresponding to the position vector of the contact point from the CoM. The associated applied contact forces at the contact points are defined as $\mathbf{f}_S = [\mathbf{f}_{S1}, \mathbf{f}_{S2}, \dots, \mathbf{f}_{S\alpha}]^T \in \mathcal{R}^{3\alpha}$ where each vector component is $\mathbf{f}_{Si} = [f_{xSi}, f_{ySi}, f_{zSi}]^T \in \mathcal{R}^3$, the force applied by the robot at the contact point i . Figure 5.4 illustrates these coordinate definitions.

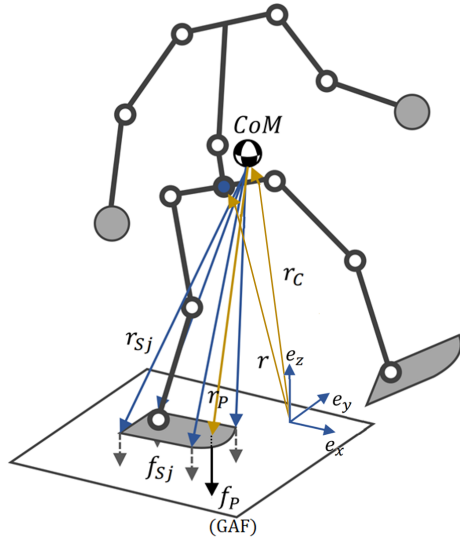


Figure 5.4: Coordinate definitions. Adapted from [Heremans, 2015].

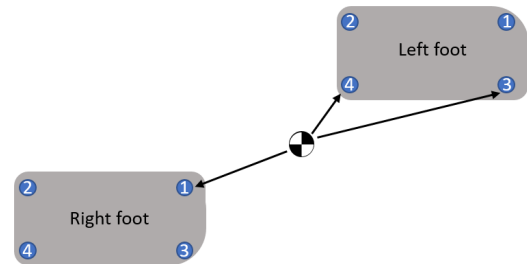


Figure 5.5: Position of the contact points on both feet.

The number and location of the contact points can be chosen arbitrary as long as the CoP can be defined in the supporting convex hull formed by the contact points. In the considered 3D

situation, a minimum of three non-collinear contact points on each foot is needed for defining the CoP during the walk. In order to cover the largest possible surface, four contact points are placed on the extremity of each foot. as illustrated in Figure 5.5.

The distribution of the GAF to each ground contact point is achieved through the definition of the CoP. Indeed, the CoP corresponds to a weighted sum of the interaction forces between the feet and the ground and can be expressed as :

$$x_p = \frac{\sum_{j=1}^{\alpha} x_{Sj} f_{zSj}}{\sum_{j=1}^{\alpha} f_{zSj}} , \quad y_p = \frac{\sum_{j=1}^{\alpha} y_{Sj} f_{zSj}}{\sum_{j=1}^{\alpha} f_{zSj}} \quad (5.6)$$

with the GAF being the sum of the respective contact forces :

$$\mathbf{f}_P = \sum_{j=1}^{\alpha} \mathbf{f}_{Sj} \quad (5.7)$$

The previous equations (5.6) and (5.7) can be rewritten in matrix notation :

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} f_{zP} = \underbrace{\begin{bmatrix} x_{S1} & x_{S2} & \dots & x_{S\alpha} \\ y_{S1} & y_{S2} & \dots & y_{S\alpha} \\ 1 & 1 & \dots & 1 \end{bmatrix}}_{\mathbf{A}_z} \begin{bmatrix} f_{zS1} \\ f_{zS2} \\ \dots \\ f_{zS\alpha} \end{bmatrix} \quad (5.8)$$

where $\mathbf{A}_z$ represents a contact-force distribution matrix.

With this relationship, the normal contact forces f_{zSi} can be calculated for a desired normal GAF f_{zP} and CoP location $\mathbf{r}_P$. However, inverting this system leads to an indefinite result since the rank of the matrix $\mathbf{A}_z$ is equal to three whereas there are more than three contact points. To solve this issue, [Hyon et al., 2007] propose to use the pseudo-inverse, $\mathbf{A}_z^\# = \mathbf{A}_z^T (\mathbf{A}_z \mathbf{A}_z^T)^{-1}$, to obtain an optimal contact force distribution that minimizes the L_2 norm of the total contact forces.

$$\begin{bmatrix} f_{zS1} \\ f_{zS2} \\ \dots \\ f_{zS\alpha} \end{bmatrix} = \mathbf{A}_z^\# \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} f_{zP} \quad (5.9)$$

A similar procedure is used in the x and y directions. The distribution of the horizontal forces is realized in proportion to the normal contact forces so that the contact points situated closer to the CoP, thus with larger normal contact force receive more ground friction.

$$\begin{bmatrix} x_p \\ 1 \end{bmatrix} f_{xP} = \underbrace{\begin{bmatrix} x_{S1} & x_{S2} & \dots & x_{S\alpha} \\ 1 & 1 & \dots & 1 \end{bmatrix}}_{\mathbf{A}_x} \begin{bmatrix} f_{xS1} \\ f_{xS2} \\ \dots \\ f_{xS\alpha} \end{bmatrix}, \quad \begin{bmatrix} y_p \\ 1 \end{bmatrix} f_{yP} = \underbrace{\begin{bmatrix} y_{S1} & y_{S2} & \dots & y_{S\alpha} \\ 1 & 1 & \dots & 1 \end{bmatrix}}_{\mathbf{A}_y} \begin{bmatrix} f_{yS1} \\ f_{yS2} \\ \dots \\ f_{yS\alpha} \end{bmatrix} \quad (5.10)$$

and using the pseudo-inverses $\mathbf{A}_x^\# = \mathbf{A}_x^T (\mathbf{A}_x \mathbf{A}_x^T)^{-1}$ and $\mathbf{A}_y^\# = \mathbf{A}_y^T (\mathbf{A}_y \mathbf{A}_y^T)^{-1}$:

$$\begin{bmatrix} f_{xS1} \\ f_{xS2} \\ \dots \\ f_{xS\alpha} \end{bmatrix} = \mathbf{A}_x^\# \begin{bmatrix} x_p \\ 1 \end{bmatrix} f_{xP}, \quad \begin{bmatrix} f_{yS1} \\ f_{yS2} \\ \dots \\ f_{yS\alpha} \end{bmatrix} = \mathbf{A}_y^\# \begin{bmatrix} y_p \\ 1 \end{bmatrix} f_{yP} \quad (5.11)$$

Simplified CoP position

In the previous relationships expressing the individual contact forces, 5.9 and 5.11, the position of the CoP $\mathbf{r}_P = [x_P, y_P, z_P]$ must be known. As already mentioned, the exact calculation of the CoP location is avoided so that no force sensors are needed on the feet.

An approximation of the CoP position is thus achieved.

As mentioned previously in Section 2.2.2, the CoP is equivalent to the ZMP when the ZMP lies inside the base of support. The CoP position can thus be obtained by computing the ZMP which is the point on the ground at which the moments about the x- and y-axes equal zero.

A simplifying model, represented in Figure 5.6, is used, assuming that the rate change of the angular momentum around the CoM is negligible and neglecting the walking dynamics of the robot. Since the moment in $\hat{x}$ and $\hat{y}$ are null at the ZMP, the desired CoP position can be written as follows :

$$x_p = \frac{z_P f_{u,x}}{mg + f_{u,z}}, \quad y_p = \frac{z_P f_{u,y}}{mg + f_{u,z}} \quad (5.12)$$

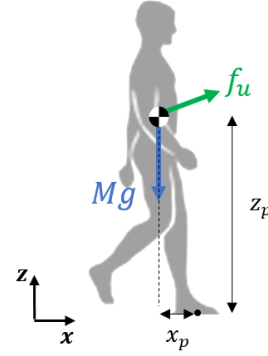


Figure 5.6: Simplified model to approximate the CoP position.

Joint torques

Finally, the joint torques are obtained following the force-torque relationship :

$$\boldsymbol{\tau} = \mathbf{J}_S(q)^T \mathbf{f}_S \quad (5.13)$$

with $\mathbf{J}_S(q) \in \mathcal{R}^{3\alpha \times n}$ the Jacobian from the CoM to the supporting contact points, n the number of joints and $\mathbf{f}_S$ the contact forces.

Summary

The overall procedure, highly inspired from [Hyon et al., 2007], can thus be summarized as follows :

1. Calculate the forward kinematics (position and Jacobians of all contacts and CoM).
2. Determine the virtual force $\mathbf{f}_v$ following the PD feedback rule by (5.1).
3. Deduce the desired GAF $\mathbf{f}_P \approx \mathbf{f}_v$.
4. Determine the position of the contact points $\mathbf{r}_{S_i}$ depending on the foot in stance.
5. Calculate the desired position of the CoP by (5.12), the pseudo-inverse matrices and the individual contact forces $\mathbf{f}_{S_i}$ by (5.9) and (5.11).
6. Obtain the joint torques from the contact forces by (5.13).

2nd method :

$$\boldsymbol{\tau} = \mathbf{J}_S(q)^T \mathbf{f}_S \quad (5.14)$$

with $\mathbf{J}_S(q) \in \mathcal{R}^{3\alpha \times n}$, the Jacobian from the CoM to the supporting contact points, n the number of joints and $\mathbf{f}_S$ the contact forces.

5.2.5 From torques to stimulations

As explained in 4.1 and illustrated in Figure 4.1, the desired joint torques are then transformed into muscle stimulations owing to a inverse muscular model, as globally explained in Section 3.3.4. These stimulations are then added to the stimulations required for walking and computed by CPG-reflexes rules.

5.2.6 Parameters determination

Some parameters have been mentioned without explaining how their values have been obtained, notably (i-ii) the PD gains of the virtual control force from (5.1) K_P and K_D , (iii) the maximum duration of the balance controller.

An estimation of these parameters was first obtained by manual testing. Then, their values were determined by comparing different sets of values with numerous simulations.

K_{Px} and K_{Py} are set to 100 [N/m], K_{Pz} to 40 [N/m] and K_D to 2[Ns/m] in all directions. Moreover, the balance controller is active during a maximum of 0.35s. The same parameters are used for both methods.

The proportional gains K_P are dominant with respect to the derivative gains K_D . Indeed, since the CoM velocity is highly impacted by the dynamics of the robot when walking, which results in numerous oscillations in the CoM velocity error, small derivative gains allow to reduce this influence on the control force.

5.3 Validation method

All perturbations that are applied to the robot are horizontal pushes applied on the torso of the robot. As characterized in Section 3.4.1, pushes occur during a period of 0.2 second and during two pushes, a resting time between 5 to 6 seconds is imposed. The robot is considered as having survived to a push when it has not fallen in the interval of five seconds after the push. The push amplitude is fixed to 30N while the push direction and the moment in the gait cycle may vary.

All simulations are run a minimum of 5 times under the same conditions in order to ensure the validity of the results.

5.4 Results

In order to analyze the performance of the second method, different analyses have been carried out.

First of all, results over one of the most important improvement are presented. This concerns situations where the push is applied laterally at heel strike.

Then, the influence of the gait cycle phase on the hip muscles activations and CoP displacement is analyzed for lateral pushes. For some analyses, a comparison with human data is provided.

Finally, the global performance of the balance controller under various situations is provided by comparing the falling probabilities with the initial data without balance controller.

5.4.1 Reaction to inward lateral pushes at heel strike

The balance controller has first been designed to tackle a specific type of perturbations, i.e. lateral pushes occurring at heel strike and oriented from the stance leg to the swing leg, i.e. inward push. Without the balance controller in these situations, the robot showed a falling

probability of 1, as represented in Figure 3.9, meaning that the robot fell at each time. Using the balance controller allows to reduce the falling probability to 0.6.

The following results present the outcomes obtained after a push occurring at RHS and oriented at 90° .

Reaction of the robot

Figure 5.7 provides an overview of the typical behaviour of the robot after a push. The detection of the perturbation takes around 0.1s, after which the balance controller is active during 0.35s. Torque adjustments are thus provided for the joints of the stance right leg. The next foot placement, after $t + 0.5$, is impacted by the push since the foot is placed more to the left compared to a normal gait represented by the internal model in transparent.

By attracting the CoM of the robot to the one of the internal model, the virtual force allows to reduce the displacement of the robot induced by the push. But this force does not exclusively counteract the perturbation since the robot places its foot more to the left at the next step.

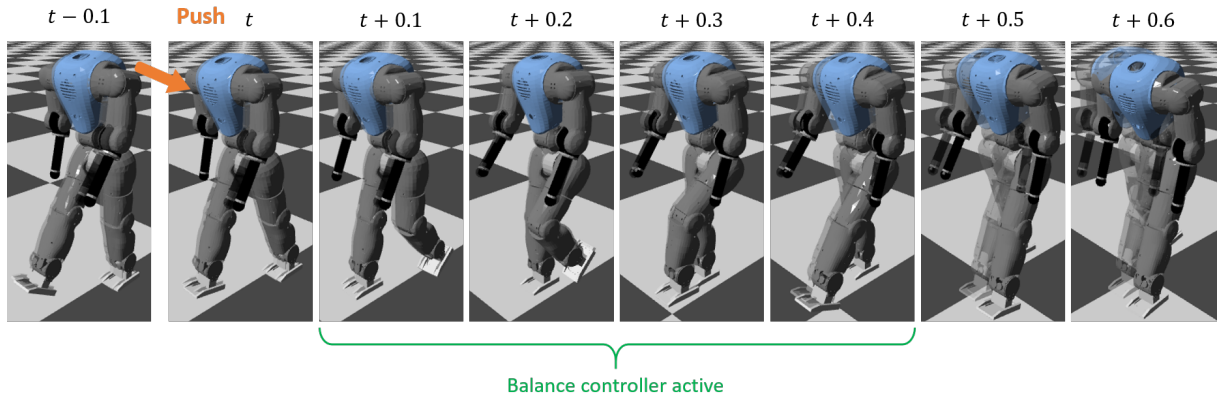


Figure 5.7: Reaction of the robot during the 0.6 seconds following a push (time intervals of 0.1s).

Nevertheless, the reaction of the robot may change from one simulation to another since the robot is never perfectly in the same state. This reflects reality since humans present also different responses depending on the situation, on their experience, intention and expectations [Horak, 2006].

Intermediate results of the second method

Detailed results on the second method are given in order to validate the different steps. Figure 5.8a shows the evolution of the virtual force $\mathbf{f}_v$ from (5.1) and the desired CoP position from (5.12), Figure 5.8b the CoM position with respect to the middle of the waist and Figure 5.8c the individual contact forces in the y-direction from (5.11).

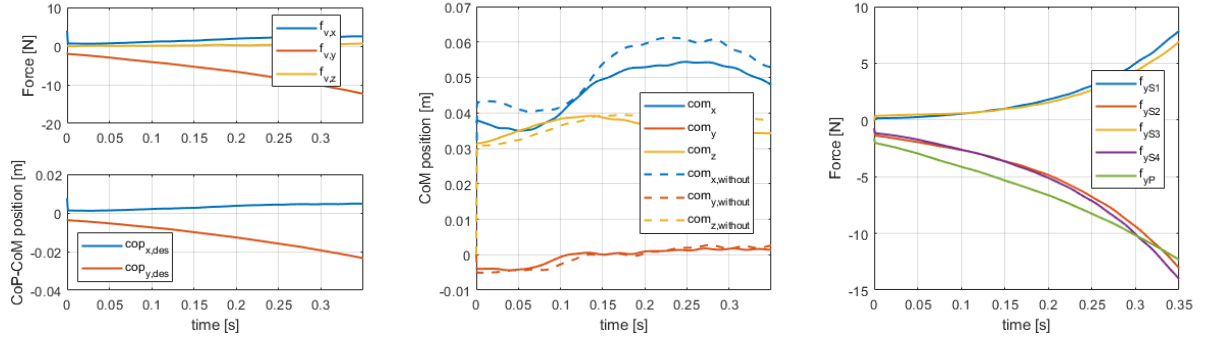
As illustrated, the virtual force is, in absolute value, larger in the y-direction since the push is applied in the lateral direction.

Moreover, the magnitude of the forces is low compared to the push amplitude (30 N). This shows that a small recovery force is sufficient to improve balance. The y-force decreases almost linearly from -1 to -12 N during the activation of the controller due to the increasing CoM position error between the robot and the internal model.

The desired position of the CoP with respect to the CoM position, computed from (5.12), follows the direction of the virtual force, as shown in Figure 5.8a. Moreover, in Figure 5.8b, the CoM position in solid lines is less deviated compared to a push without balance controller, represented

in dotted lines. This is especially observable for the x-position. This reflects that the controller applies contact forces so that the CoM is less deviated, reducing the displacement of the robot and helping to regain balance.

Finally, the distribution of the y-ground applied force (GAF) on the contact points is represented in Figure 5.8c. Although the total y-GAF is negative, the points situated at the front of the foot (1, 3) have a positive force while the points situated the the back of the foot (2, 4) have a negative value. The distribution of the contact forces follows also the shape of the GAF since the absolute value of the force at each contact point increases with the time.



(a) virtual force f_v and desired CoP position (b) CoM position with and without balance controller (c) contact forces in the y-direction

Figure 5.8: Evolution of controller data when the controller is active after the push detection.

Right joint torques

The joint torques deduced from the torque-force relationship (5.14) are depicted in Figure 5.9 for seven pushes where the robot regained balance. These graphs show the general patterns of the computed joint torques.

The torques are only applied to the joints of the right leg as the robot is in right stance during the activation of the controller. A small exception concerns the magenta curve where the robot reaches left heel strike slightly before the end of the activation of the balance controller.

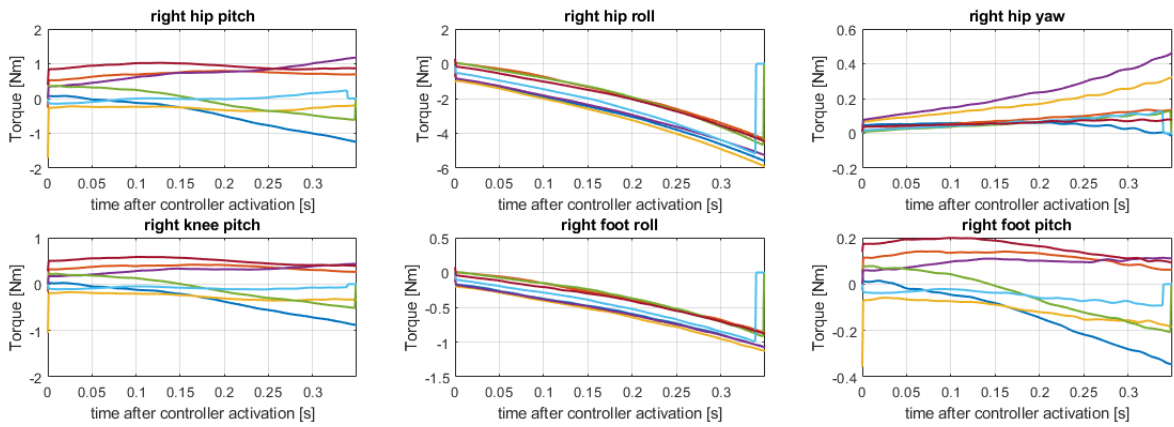


Figure 5.9: Joint torques computed from the torque-force relationships of both methods.

These graphs show that the most requested joint torque is the hip roll. Its value is negative, such as for the foot roll, representing a lateral outward movement as the robot is in right stance and

the CoM is attracted to the right. Other joint torques present more various but constant values. This is due to the force in the x-direction which is, in some simulations positive and in others negative but globally constant.

Thus, the virtual force in the y-direction impacts the roll joints while the virtual forces in the x- and z- directions influence the pitch and yaw joints.

Finally, this method allows to adopt both ankle and hip strategies to recover balance.

Right muscles stimulations

As explained in 5.2.5, the desired joint torques are then transformed into muscles stimulations by the inverse muscular model. These stimulations will then be summed with the stimulations required for walking. The muscles stimulations are bounded between 0 and 1.

Figure 5.10 shows the corresponding muscles stimulations deduced from the torques of Figure 5.9. The muscles are depicted in Figure 3.3. The thirteen right leg muscles are represented in order to identify the muscles that are the most activated.

First of all, the transverse muscles HER and HIR are slightly activated or not at all, showing their negligible contribution. Concerning the lateral muscles, HAB and EVE present an increasing stimulation with the time. These muscles allow to produce the outward roll at the hip and the foot joints. Regarding the sagittal muscles, the SOL, GAS, HAM and GLU have insignificant values. The muscles TA, VAS and HRL are more activated, especially at the end to provide more torques at the pitch joints.

This figure shows thus the muscles stimulations that are more solicited by using the balance controller when a push occurs at RHS in the lateral direction. Two muscles in the lateral plane are significantly activated to provide hip and ankle strategies : HAB and EVE.

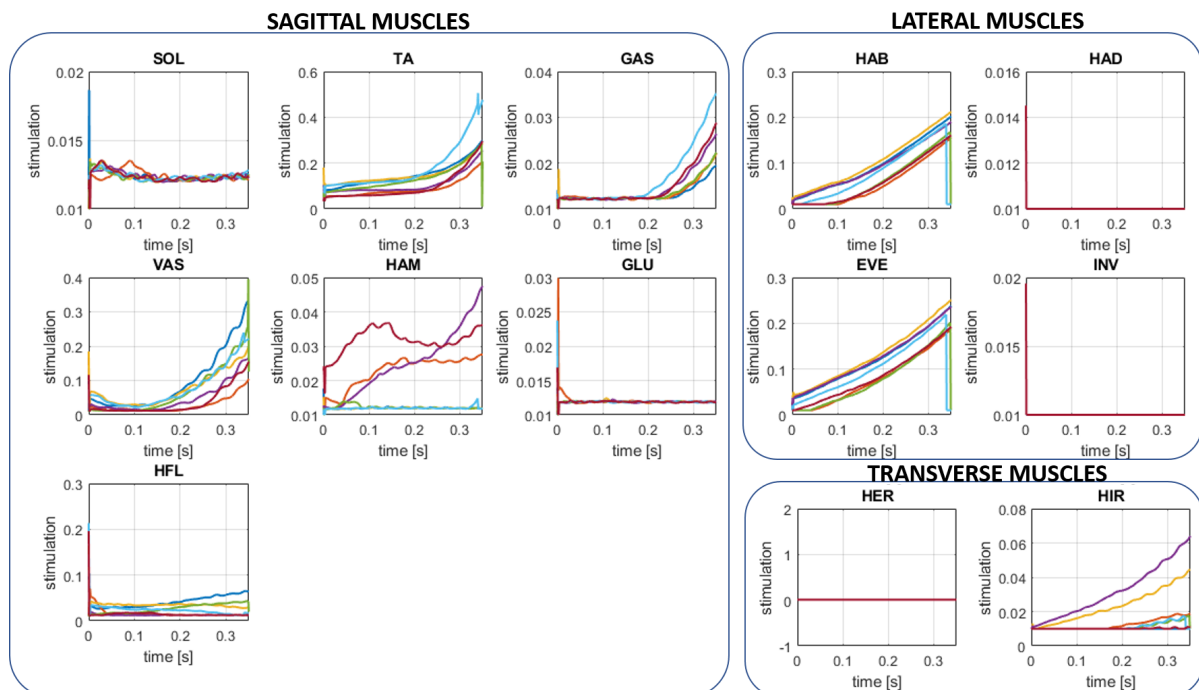


Figure 5.10: Muscles stimulations of the right leg computed from the inverse muscular model.

5.4.2 Influence of the gait cycle phase with lateral pushes

In the previous subsection, all pushes occur at right heel strike. As already mentioned, the robot reacts differently depending on the phase at which the push occurs. The following results aim to analyze the influence of the gait cycle phase on the robot reaction and notably on the hip muscles activations and the CoP displacement during the seconds that follow a push.

The gait cycle phase is divided into six moments : 0, 10, 30, 50, 60, 80% at which the push occurs. These moments are not equally spaced but allow to analyze specific moments such as the beginning of the double-support phase (0 and 50%), around the beginning of a single-support phase (10 and 60%) and the middle of a swing phase (30 and 80 %).

The push is applied in the positive y-direction, i.e. 90° during all the gait phases, assuming that a push oriented at -90° would give similar results on the opposite leg.

In the following figures, the time $t = 0$ corresponds to the beginning of the activation of the balance controller. The push is applied around 0.1 s before and is represented by a * on the x-axis.

Robots pushed at 50% of the gait cycle always fell. Results of the CoP displacements are still provided for this phase but not for the hip stimulations.

CoP lateral displacement

Figure 5.11 shows the y-displacement of the CoP of the robot with respect to the middle waist position while the push is applied at different moments of the gait cycle.

A positive CoP y-position corresponds to a foot placed on the left side of the robot while a negative value corresponds to the right side. Transitions from positive to negative values or inversely correspond to double-support phases. In average, a gait cycle lasts around one second, so around four steps are represented in the figures.

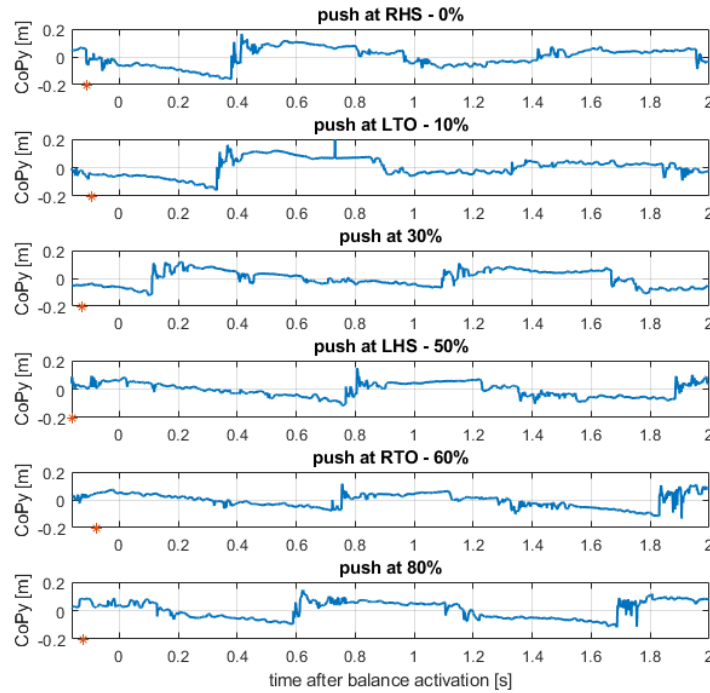


Figure 5.11: Lateral CoP displacement with respect to the middle of the waist after a push represented by the *.

When the push arises at 0, 10 or 30%, the CoP value decreases to the right from $t = 0$ until the double-support phase. This reflects the activation of the balance controller as explained previously with Figure 5.8a.

For pushes at 50 or 60%, the robot has in general more difficulties to recover balance. The CoP is again displaced to the right during the balance activation. However, no visible step around 0.4s is visible. Indeed, the right leg which is in swing makes almost a cross-over step and places its foot just in front of the left foot. So no double-support phase transition is visible.

The wider CoP displacements occur at RHS and LTO due to the push which is oriented to the free leg in swing. The left foot is placed more outward.

Except for the pushes at 50 and 60%, after 2 steps (around $t = 1s$) the CoP becomes more constant during the stance phases and is closer to 0. This reflects that the robot has recovered balance.

A study from [Hof and Duysens, 2013] analyzed CoP human responses to lateral balance perturbations during walking. A comparison of the CoP data with human responses is provided in Figure 5.12. The subjects walked at a speed scaled according to their leg length. The moments at which the pushes occur are not exactly the same but the differences are slight. Human data are only provided from 4 to 56% of the gait cycle. Moreover, to conform to their reference, data are inverted and shifted. In their data, thick lines correspond to CoP recordings while thin blue lines correspond to CoP curves for unperturbed strides ($\pm SD$ in dotted lines).

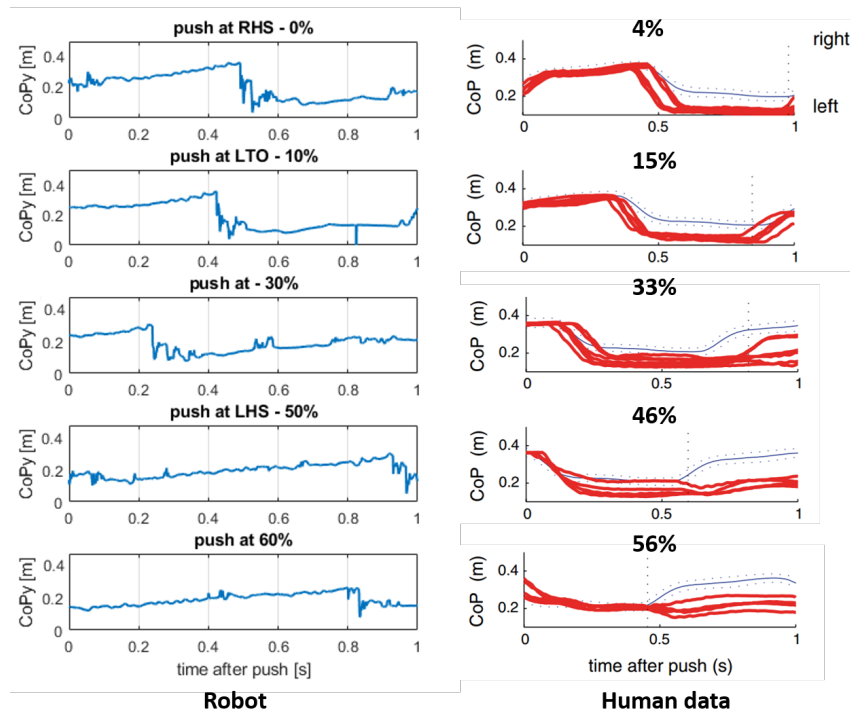


Figure 5.12: Lateral CoP displacement of the robot (left) and from human data [Hof and Duysens, 2013] (right).

The general patterns show similarities between the lateral displacements of the CoP obtained from the robot and from human data.

Larger stride width After pushes occurring between 0 and 33%, the left foot is placed more laterally, enlarging the stride width. The lateral displacement is larger for pushes close to RHS than around 30% of the gait cycle.

Cross stepping For pushes close to 46 and 56% of the gait cycle, the right leg is placed more to the left side in the step following the push. This is also observed in simulation where no clear transition marking the switch between the left and right foot in stance was observable. Cross stepping happened in some cases when the push occurs at 56%. That was not the case in the measurements of the robot.

Ankle and hip strategies Moreover, at 4 and 15% of the gait cycle, after being pushed, the CoP slightly increases, which could reflect the use of roll ankle and hip strategies in humans. The time taken to make the next step after the push are also comparable for the three first moments.

Double-support phases During double-support phases, the transition from one foot to another is raw in the simulations contrary to human data. This is due to inaccuracies of the ground contact model that is used in the simulations.

Finally, a scaling would be required to compare more precisely the data because of the smaller size of the robot. This will not be done in this report.

The global comparison thus highlights important similarities of the lateral CoP displacement between the robot and human data.

Hip muscles activations

Since the pushes are applied in the lateral direction, some leg muscles in the lateral plane are significantly activated by the balance controller. This was shown in Figure 5.10 with the right HAB and EVE muscles.

This subsection aims to analyze the influence of the gait cycle phase on the right HAB muscle and left HAD muscle after a push in the 90° direction. Unlike the right leg, the left HAD muscle is activated to provide inward movements. Plots are only provided during the first half of the gait cycle since robots always feel at 50% of the gait cycle, therefore it is not considered as reference.

The stimulations obtained from the balance controller are plotted in Figure 5.13a for the right HAB muscle and in Figure 5.13c for the left HAD muscle. The total activations obtained after summing the simulations required for walking and the ones from the balance controller and after being filtered are represented in Figures 5.13b for the right HAB and 5.13d for the left HAD. These activations are the inputs of the Hill virtual muscles.

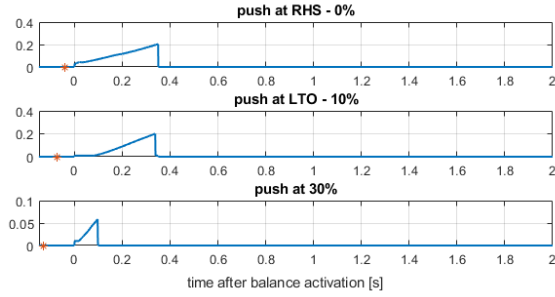
When the perturbation is detected, the simulations of the right HAB or left HAD muscles increase with the time due to the growing virtual force. The values of the stimulations are similar with respect to the time after balance activation. However, the stimulations are either injected to the right HAB muscle if the right foot is in stance or to the left HAD muscle if the left foot is in stance.

Once added with the stimulations for walking and filtered, the activations show very similar patterns but shifted due to the different timings. No significant modification is observable except for the left HAD muscle when the push occurs at 30% of the gait cycle. After a peak at around 0.15s, the activations should drop to 0 if we compare with the shifted patterns in the upper figures. This is not the case but allows the robot to regain balance.

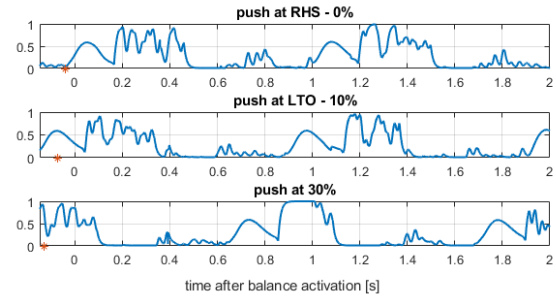
In general the adjustments given by the balance controller are thus low compared to the one required for walking but are sufficient for the robot to recover balance.

Moreover, although the balance controller is instantaneously activated, this does not induce discontinuous activations for the muscles. Indeed, the balance controller provides only slight

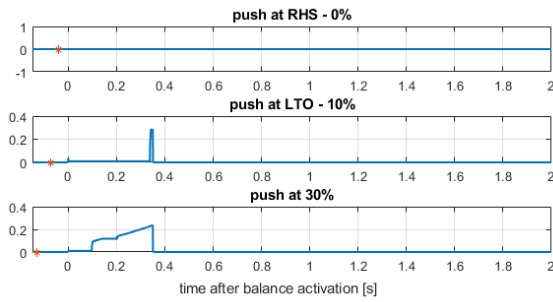
additional muscles stimulations and the filtering allows to smooth the values. Transition during the activating and deactivating of the balance controller is thus not needed.



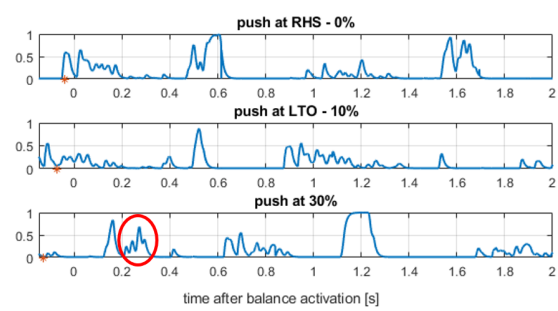
(a) Additional stimulations of the right HAB muscle from the balance controller.



(b) Total activations of the right HAB muscle.



(c) Additional stimulations of the left HAD muscle from the balance controller.



(d) Total activations of the left HAD muscle.

Figure 5.13: Right HAB and left HAD stimulations from the balance controller and total activations for different gait cycle phases.

5.4.3 General performance compared to the initial robustness

In order to evaluate the global performance of the balance controller, a comparison of the falling probabilities with the initial robustness from Figure 3.9 is provided in Figure 5.14. The pushes occur at four different moments in the gait cycle (right heel strike, left toe-off, left heel strike or right toe-off) and in eight different directions (-135° , -90° , -45° , 0° , 45° , 90° , 135° and 180°). The initial falling probabilities are depicted in blue while there are in green for the balance controller. Significant improvements are represented under the red dotted rectangles.

Results show that the falling probabilities have significantly decreased in the lateral directions. At right heel strike for pushes oriented at 90° , the falling probability has dropped from 1 to 0.6, at left toe-off for 90° pushes, the falling probability has dropped from 0.6 to less than 0.1 and finally, at left toe-off for pushes oriented at 135° , it has dropped from 0.9 to 0.4. Similar but opposite results are obtained for the second half of the gait cycle.

On the other hand, the performance of the robot against forward pushes has decreased. Indeed, with the balance controller, due to torques adjustments, the robot places its foot earlier on the ground, makes a smaller step and falls forward.

By deactivating the balance controller only for forward pushes, this reduction of performance can be avoided. To this end, the push direction should be estimated, which can be achieved by measuring the direction the CoM velocity or the direction difference between the real robot and

the internal model.

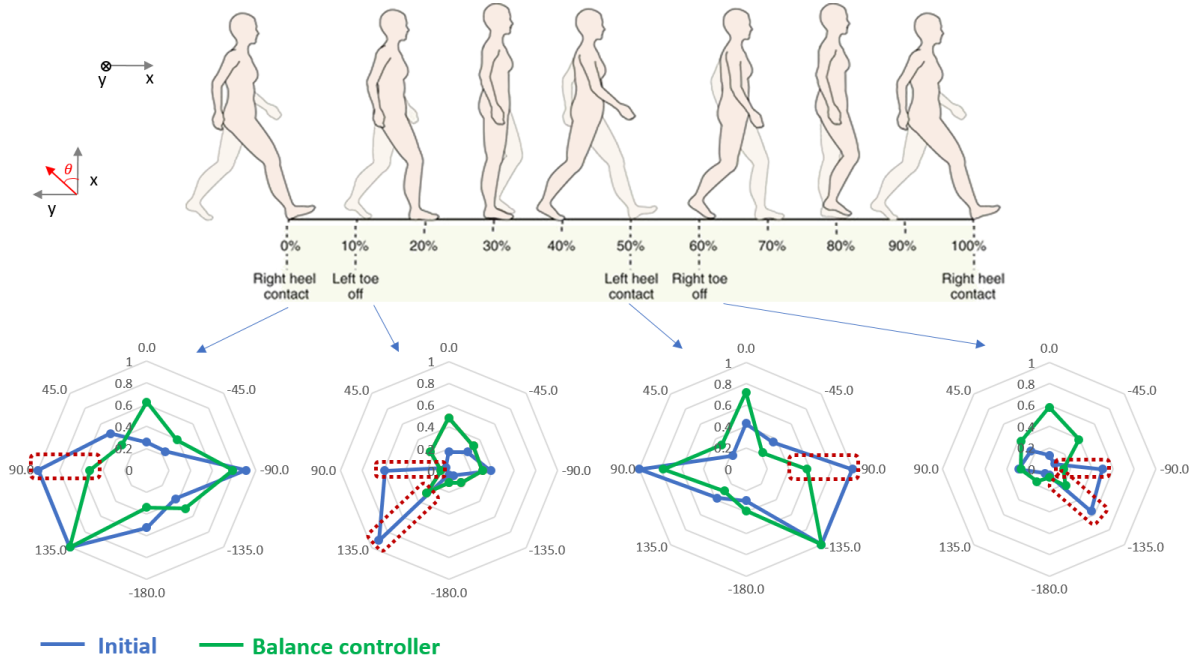


Figure 5.14: Comparison of the falling probabilities at four moments in the gait cycles and for push directions with and without the balance controller.

Higher push amplitudes

The performance of the balance controller for lateral pushes oriented at 90° and occurring at RHS has been evaluated with higher push amplitudes. Without balance controller, the falling probability in this situation was 1 for 40N- and 50N-pushes.

Using the balance controller, the falling probabilities are 0.9 for 40N-pushes and 0.97 for 50N-pushes. This balance controller is thus inappropriate against large perturbations.

5.5 Discussion

In this chapter devoted to the implementation of an additional balance controller, stability improvements were observable notably for lateral pushes. However, some assumptions or developed points may be further discussed.

Dynamics of the robot

A heavy assumption was taken during the development of the equations in Section 5.2. The system was considered as quasi-static whereas it is constantly moving and some moments in the gait cycle are considered as unstable, notably at toe-off. Nevertheless, the developed method allows an enhancement of the stability against lateral pushes but it was not successful for all simulations. Inaccuracies due to the taken assumptions may probably influence the robustness of the method.

Moreover, the virtual force from (5.1) follows a proportional-derivative feedback law to control the CoM position of the robot with respect to the CoM of the internal model. However, this force is considerably affected by the dynamics of the robot. When perturbed, the CoM velocity

difference between the internal model and the robot is high, this is why a small derivative gain was considered. A derivative gain was still used to benefit from the fast difference errors when the robot is perturbed, making the force higher at the beginning of the balance activation before the CoM position errors grow and become the dominant term.

One might think about dynamics compensation in the computation of the virtual force. This would probably be difficult since the CoM acceleration presents raw values notably due to the impacts with the ground.

Situations leading to a fall

From one simulation to another, the balance controller does not always perform well even if the push direction and moment in the gait cycle are similar.

An analysis aimed to identify the internal parameters of the robot that would predict a fall. This would allow to establish margins on specific values for which the controller can successfully be apply.

However, too many potential parameters were identified that could influence a fall, such as the x- and y- CoM position, the x- and y- CoM velocity, the x- and y- CoM position with respect to the CoP position, the timing (at foot placement, during the balance controller activation), ...

Due to the high number of potential parameters, this analysis has not been conducted and is let for a future work.

Human comparison

In the literature, very few studies were devoted to phase-dependent responses to mechanical perturbations during gait [Hof and Duysens, 2013]. This latter study allowed to compare lateral CoP displacement with the robot, as illustrated in Figure 5.12.

They also showed that recovery is fast, within one stride cycle, which is in line with other studies [Oddsson et al., 2004]. It justifies the activation of the balance controller during a short time limited to 0.35s.

Finally, they also mentioned that after a perturbation, voluntary activity typically starts after 150ms. Using a balance controller that is not triggered directly after a perturbation but after a small time needed for the detection, allows to be faithful to this human feature. With the developed stability criterion, the delay is situated around 0.12s for 30N-pushes which almost corresponds to a bio-inspired reaction.

Thus, with respect to a timing point of view, the developed stability criterion and balance controller present similarities with human reactions.

Other types of controllers

The developed controller is unfortunately not appropriated to face large perturbations. Indeed, it provides ankle and hip strategies on the stance leg, which is particularly used when standing or for small to medium perturbations.

This controller has the particularity of requiring only a small number of information and a low computational cost. Information on the direction of the push, inverse kinematics model or forces at the contact point are for example not necessary.

When facing larger perturbations, a stepping strategy is often used to recover balance. Indeed, when pushed, humans generally let themselves fall but places their foot with an adequate foot placement and balance can be recovered. A further work could be the implementation of a stepping controller.

Chapter 6

Future perspectives & Conclusion

This master thesis aimed to improve the stability of the bio-inspired robot COMAN while walking. Two main objectives have been followed during this work : (i) detecting perturbations and (ii) improving balance when perturbations have been sensed.

Although potential sources of improvement have already been proposed at the end of Chapter 5, a selection of the propositions that could bring major benefits to this work will be suggested in this chapter. Then, a conclusion will end this work.

6.1 Future perspectives

In this work, balance improvements were mainly observed in situations where the robot is laterally pushed at heel strike. However, further improvements could be achieved to enlarge the performance of the robot after being perturbed.

Adaptive gains of the balance controller

Up to now, the gains of the balance controller are similar for all kinds of perturbations. Adjusted gains could be employed for adapting the contribution of the balance controller with respect to the initial performance of the robot. Typically, for pushes occurring at heel strike, the robot needs more help to recover balance than at toe-off. Moreover, the performance of the robot was worsened for forward pushes. The gains could thus be adapted according to the push direction or the moment in the gait cycle. Determining the push direction could be achieved based on the falling direction through the CoM velocity or the direction difference between the real robot and the internal model.

Controlling the swing leg

In this work, when the robot is perturbed, adjustments are only made on the joints of the stance leg, using hip and ankle strategies. However, for dealing with larger perturbations, a stepping strategy is generally used. Although the robot performs already slight natural foot displacements when perturbed, a control of the swing leg or of the foot placement would probably further enhance the performance of the robot against perturbations.

Impact of the balance controller

Using a balance controller makes the robot able to counteract perturbations and to operate in various working environments where unpredictable disturbances may occur. Nevertheless, one may also analyze whether making the robot more robust to perturbations might also extend the range of motion of the robot, and notably for low speeds.

Falling movements

Finally, when a fall is unavoidable, specific movements can be triggered to limit the impact with the ground. Indeed, falling can harm the robot such as for humans. In order to lower the damage of the fall, a sequence of movements allowing a wider contact area and a longer lapse of time could be implemented as in [Ruiz-del Solar et al., 2010]. On the other hand, developments can also be carried out to make the robot able to get back up after a fall so that it becomes more independent from humans and could interact in more various environments.

6.2 Conclusion

Ensuring balance while the biped faces perturbations during walking is a complex challenge. Indeed, each human reacts differently depending on the situation and its experience, making it difficult to establish rules that would help the robot to recover balance against all kinds of perturbations. Although humans have high abilities against perturbations, humanoid robots in general are still far from reaching them.

In this work, motivation for improving the stability of humanoid robots was presented, as well as a review of the current literature related to human balance, fall detection for humanoid robots and different types of balance controllers in Chapter 2.

All the implemented methods were tested with the simulator Robotran on a model of the COMAN robot, in the continuity of the work achieved mainly by Dr. N. Van der Noot [Van der Noot, 2017b]. The existing bio-inspired controller was presented in Chapter 3.

The use of internal model was a key feature for improving the stability. This model simulates a robot facing no perturbations and allowed to have a reference robot for detecting perturbations and have a reference position of the CoM of a unperturbed robot.

For triggering the use of the implemented balance controller, a stability criterion must be exceeded, as developed in Chapter 4. Three different data were compared between the robot and the internal model in order to sense perturbations. A detection rate of 100% of perturbations was achieved. Among the CoM position, CoM velocity and the CoP position, the criterion based on the CoM velocity allowed the fastest detection. This was the case for pushes applied on the torso of the robot but also for slippery grounds.

Since the stability criteria depend on the gait cycle phase, an gait cycle phase estimator was developed, allowing to obtain an estimation of the gait phase through the RG neurons of the CPG. The phase error was lower than 4%.

In Chapter 5, the additional balance controller was developed in order to improve the stability of the robot. The balance controller provides muscle stimulations of the stance leg so that the CoM of the robot was attracted by the CoM of the internal model. These stimulations were then summed to the one required for walking. Hip and ankle strategies were employed.

A comparison of the performance of the robot with and without balance controller showed that the balance controller significantly reduces the falling probabilities in the case of inward lateral pushes occurring at heel strike.

With respect to a timing point of view, the developed stability criterion and balance controller present similarities with human reactions.

Finally, perspectives were proposed in this chapter to further enhance the balance controller and therefore the performance of the robot against larger perturbations.

Bibliography

- [Boston Dynamics, 2018] Boston Dynamics (2018). Atlas | Boston Dynamics. Available at <https://www.bostondynamics.com/atlas>.
- [Chen and Chou, 2010] Chen, C.-J. and Chou, L.-S. (2010). Center of mass position relative to the ankle during walking: A clinically feasible detection method for gait imbalance. *Gait & Posture*, 31(3):391–393.
- [Darpa, 2015] Darpa (2015). Darpa Robotics Challenge Finals 2015. Available at <http://archive.darpa.mil/roboticschallenge/>.
- [Dekker, 2009] Dekker, M. H. P. (2009). Zero-moment point method for stable biped walking. *University of Technology, Eindhoven*.
- [Dzeladini et al., 2014] Dzeladini, F., van den Kieboom, J., and Ijspeert, A. (2014). The contribution of a central pattern generator in a reflex-based neuromuscular model. *Frontiers in Human Neuroscience*, 8.
- [Falconer, 2013] Falconer, J. (2013). This Humanoid Robot Gets Pushed Around But Stays on Its Feet. Available at <https://spectrum.ieee.org/automaton/robotics/humanoids/iit-coman-humanoid-robot>.
- [Faraji et al., 2018] Faraji, S., Razavi, H., and Ijspeert, A. (2018). Push recovery with stepping strategy based on time-projection control.
- [Geyer and Herr, 2010] Geyer, H. and Herr, H. (2010). 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.
- [Greiner, 2017] Greiner, P. (2017). Gait modulation of a humanoid robot, using bio-inspired mechanisms. Master’s thesis, Université Catholique de Louvain.
- [Hak et al., 2012] Hak, L., Houdijk, H., Steenbrink, F., Mert, A., Wurff, P. v. d., Beek, P. J., and Dieën, J. H. v. (2012). Speeding up or slowing down?: Gait adaptations to preserve gait stability in response to balance perturbations. *Gait & Posture*, 36(2):260–264.
- [Heremans, 2015] Heremans, F. (2015). Controlling the balance of a humanoid robot subject to large perturbations. Master’s thesis, Université Catholique de Louvain.
- [Hof and Duysens, 2013] Hof, A. L. and Duysens, J. (2013). Responses of human hip abductor muscles to lateral balance perturbations during walking. *Experimental Brain Research*, 230(3):301–310.
- [Hof et al., 2005] Hof, A. L., Gazendam, M. G. J., and Sinke, W. E. (2005). The condition for dynamic stability. *Journal of Biomechanics*, 38(1):1–8.

- [Hof et al., 2010] Hof, A. L., Vermerris, S. M., and Gjaltema, W. A. (2010). Balance responses to lateral perturbations in human treadmill walking. *Journal of Experimental Biology*, 213(15):2655–2664.
- [Horak, 2006] Horak, F. B. (2006). Postural orientation and equilibrium: what do we need to know about neural control of balance to prevent falls? *Age and Ageing*, 35(suppl_2):ii7–ii11.
- [Hyon et al., 2007] Hyon, S. H., Hale, J. G., and Cheng, G. (2007). Full-Body Compliant Human-Humanoid Interaction: Balancing in the Presence of Unknown External Forces. *IEEE Transactions on Robotics*, 23(5):884–898.
- [IEEE Spectrum, 2015] IEEE Spectrum (2015). A Compilation of Robots Falling Down at the DARPA Robotics Challenge. YouTube, available at <https://www.youtube.com/watch?v=g0TaYhjp0fo>.
- [Ijspeert, 2008] Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653.
- [Kalyanakrishnan and Goswami, 2011] Kalyanakrishnan, S. and Goswami, A. (2011). Learning to predict humanoid fall. *International Journal of Humanoid Robotics*, 08(02):245–273.
- [Koolen et al., 2012] Koolen, T., Boer, T. d., Rebula, J., Goswami, A., and Pratt, J. (2012). Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113.
- [Kudoh et al., 2006] Kudoh, S., Komura, T., and Ikeuchi, K. (2006). Stepping motion for a human-like character to maintain balance against large perturbations. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2661–2666.
- [Lateral Edge, 2017] Lateral Edge (2017). The Necessity of Exercising in the Frontal Plane. Available at <http://www.lateraledgeonline.com/blog/2017/3/1/the-necessity-of-exercising-in-the-frontal-plane>.
- [Li et al., 2015] Li, Z., Zhou, C., Castano, J., Wang, X., Negrello, F., Tsagarakis, N. G., and Caldwell, D. G. (2015). Fall Prediction of legged robots based on energy state and its implication of balance augmentation: A study on the humanoid. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5094–5100.
- [Matlab, 2017] Matlab (2017). Box plot. Available at <https://ch.mathworks.com/help/stats/boxplot.html>.
- [O. Höhn and W. Gerth, 2009] O. Höhn and W. Gerth (2009). Probabilistic Balance Monitoring for Bipedal Robots. *The International Journal of Robotics Research*, 28(2):245–256.
- [Oddsson et al., 2004] Oddsson, L. I. E., Wall, C., McPartland, M. D., Krebs, D. E., and Tucker, C. A. (2004). Recovery from perturbations during paced walking. *Gait & Posture*, 19(1):24–34.
- [Pai and Patton, 1997] Pai, Y.-C. and Patton, J. (1997). Center of mass velocity-position predictions for balance control. *Journal of Biomechanics*, 30(4):347–354.
- [Park, 2011] Park, J. (2011). The relationship between controlled joint torque and end-effector force in underactuated robotic systems. *Robotica*, 29(4):581–584.
- [Physics & Astronomy Online, 2012] Physics & Astronomy Online (2012). Coefficients of Friction. Available at <http://www.physlink.com/Reference/FrictionCoefficients.cfm>.

- [Pratt et al., 2006] Pratt, J., Carff, J., Drakunov, S., and Goswami, A. (2006). Capture Point: A Step toward Humanoid Push Recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207.
- [Pratt and Tedrake, 2006] Pratt, J. E. and Tedrake, R. (2006). Velocity-Based Stability Margins for Fast Bipedal Walking. In *Fast Motions in Biomechanics and Robotics*, Lecture Notes in Control and Information Sciences, pages 299–324. Springer, Berlin, Heidelberg.
- [Rebula et al., 2007] Rebula, J., Canas, F., Pratt, J., and Goswami, A. (2007). Learning Capture Points for humanoid push recovery. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 65–72.
- [Renner and Behnke, 2006] Renner, R. and Behnke, S. (2006). Instability Detection and Fall Avoidance for a Humanoid using Attitude Sensors and Reflexes. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2967–2973.
- [Rietdyk et al., 1999] Rietdyk, S., Patla, A. E., Winter, D. A., Ishac, M. G., and Little, C. E. (1999). Balance recovery from medio-lateral perturbations of the upper body during standing. *Journal of Biomechanics*, 32(11):1149–1158.
- [Ruiz-del Solar et al., 2010] Ruiz-del Solar, J., Moya, J., and Parra-Tsunekawa, I. (2010). Fall detection and management in biped humanoid robots. In *2010 IEEE International Conference on Robotics and Automation*, pages 3323–3328.
- [Spong et al., 2006] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). Force Control. In *Robot Modeling and Control*, pages 281–296. John Wiley & Sons.
- [Stephens and Atkeson, 2010] Stephens, B. J. and Atkeson, C. G. (2010). Push Recovery by stepping for humanoid robots with force controlled joints. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 52–59.
- [Subburaman et al., 2016] Subburaman, R., Lee, J., Caldwell, D. G., and Tsagarakis, N. G. (2016). Multi-sensor based fall prediction method for humanoid robots. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 102–108.
- [Tedrake, 2018] Tedrake, R. (2018). Fully-actuated vs Underactuated Systems. Available at <http://underactuated.mit.edu/underactuated.html?chapter=intro>.
- [Themes, 2016] Themes, U. F. O. (2016). Fundamentals of Human Gait. Available at <https://musculoskeletalkey.com/fundamentals-of-human-gait/>.
- [Tinetti et al., 1988] Tinetti, M. E., Speechley, M., and Ginter, S. F. (1988). Risk factors for falls among elderly persons living in the community. *The New England Journal of Medicine*, 319(26):1701–1707.
- [Tsagarakis et al., 2013] Tsagarakis, N. G., Morfey, S., Cerda, G. M., Zhibin, L., and Caldwell, D. G. (2013). COMpliant huMANoid COMAN: Optimal joint stiffness tuning for modal frequency control. In *2013 IEEE International Conference on Robotics and Automation*, pages 673–678.
- [Van der Noot, 2017a] Van der Noot, N. (2017a). Model-based robot monitoring interface. *Whole-body Adaptive Locomotion and Manipulation, WALK-MAN*. 7th Framework Programme.
- [Van der Noot, 2017b] Van der Noot, N. (2017b). *Rich and Robust Bio-Inspired Locomotion Control for Humanoid Robots*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Université Catholique de Louvain.

- [Van der Noot and Barrea, 2013] Van der Noot, N. and Barrea, A. (2013). Real-time control of a humanoid robot: Dynamic walking using a bio-inspired approach. Master’s thesis, Université Catholique de Louvain.
- [Van der Noot et al., 2018] Van der Noot, N., Ijspeert, A. J., and Ronsse, R. (2018). Bio-inspired controller achieving forward speed modulation with a 3d bipedal walker. *The International Journal of Robotics Research*, 37(1):168–196.
- [Vukobratović et al., 2007] Vukobratović, M., Borovac, B., and Potkonjak, V. (2007). Towards a unified understanding of basic notions and terms in humanoid robotics. *Robotica*, 25(1):87–101.
- [Wikibooks, 2017] Wikibooks (2017). Physics Study Guide/Frictional coefficients. Available at https://en.wikibooks.org/wiki/Physics_Study_Guide/Frictional_coefficients.
- [Winter, 1995] Winter, D. (1995). Human balance and posture control during standing and walking. *Gait & Posture*, 3(4):193–214.
- [Xinjilefu et al., 2015] Xinjilefu, X., Feng, S., and Atkeson, C. G. (2015). Center of mass estimator for humanoids and its application in modelling error compensation, fall detection and prevention. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 67–73.

Appendix A

State variables sent to the internal models

A set of 184 variables is sent at each reset for rebuilding a close model that simulate the real robot. It contains 12 kinematic variables of the waist including the current position and speed of the real robot waist and 172 controller state variables [Van der Noot, 2017a] (the number of parameters is indicated between brackets):

- position and velocity of the 23 motors (i.e. before the SEA) (46)
- position and velocity of the 23 joints (i.e. after the SEA) (46)
- time in seconds (1)
- state of each leg : swing or stance (2)
- supporting leg state (1)
- the length of the contractile element (l_{CE}) of the 48 muscles (48)
- the firing rate (x_i) and the self-inhibition (v_i) of the 12 Matsuoka oscillators (24)
- the state of the 2 turning neurons (x_i) (2)
- the current speed (v_{ref}) and heading (h_{ref}) commands (2)

Appendix B

Stability criteria

B.1 Gait cycle phase estimator equations

The following equations system estimates the gait cycle phase depending on the phase angle ϕ as detailed in Section 4.3 and as represented in Figure B.1.

$$\text{gait cycle phase} = \begin{cases} 3.9 \cdot (\phi + \pi) + \frac{15.7}{1 + \exp(-16 \cdot (\phi + 2.0))} & \text{if } -\pi \leq \phi \leq -1.74 \\ 4.4 \cdot (\phi + 1.74) + \frac{22}{1 + \exp(-15 \cdot (\phi + 0.6))} + 21 & \text{if } -1.74 < \phi \leq 0.21 \\ 3 \cdot (\phi - 0.1) + \frac{16.5}{1 + \exp(-15 \cdot (\phi - 1.16))} + 51 & \text{if } 0.21 < \phi \leq 1.57 \\ 5 \cdot (\phi - 1.57) + \frac{20.5}{1 + \exp(-16 \cdot (\phi - 2.54))} + 72.2 & \text{if } 1.57 < \phi \leq 3.0 \\ 100 & \text{if } \phi \geq 3.0 \end{cases} \quad (\text{B.1})$$

In Figure B.1, the blue curve/dots are obtained by computing the mean of 40 estimation functions for each increment of the gait cycle phase. The 40 estimation functions are represented in Figure 4.6d.

The four colored curves correspond to the four sigmoid functions whose analytical expressions are given in B.1.

Some individual blue points are visible for low values of the gait cycle phase. This is due to the negative value of the sum signal at the beginning of several gait cycles. They are not taken into consideration when building the bijective estimation function.

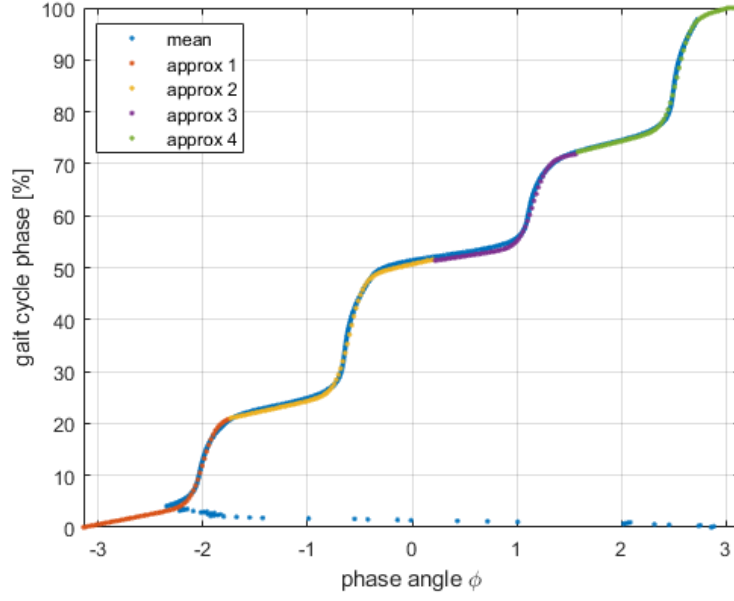


Figure B.1: Estimation function to express analytic relationships between the phase angle ϕ and the gait cycle phase

B.2 Stability criteria equations

Stability criteria are based on three specific data. When their errors between the robot and the internal model exceed the stability criteria defined by the following equations, then a perturbation is considered. The equations correspond to the graphs in Figure 4.11

CoM Position criterion

$$\Delta \text{CoM pos, threshold} = \begin{cases} \frac{0.07 - 0.05}{18.2} \% \text{ gait cycle} + 0.05 & \text{between 0 and 18.2\%} \\ \frac{0.019 - 0.07}{22.4 - 18.2} + 0.07 & \text{between 18.2 and 22.4\%} \\ \frac{0.062 - 0.019}{66.6 - 22.4} (\% \text{ gait cycle} - 22.4) + 0.019 & \text{between 22.4 and 66.6\%} \\ \frac{0.022 - 0.065}{71.7 - 66.6} (\% \text{ gait cycle} - 66.6) + 0.065 & \text{between 66.6 and 71.7\%} \\ \frac{0.05 - 0.022}{100 - 71.7} (\% \text{ gait cycle} - 71.7) + 0.022 & \text{between 71.7 and 100\%} \\ 0.07 & \text{otherwise} \end{cases} \quad (\text{B.2})$$

CoM Velocity criterion

$$\Delta\text{CoM vel, threshold} = \begin{cases} \frac{0.17 - 0.22}{3.6} \% \text{ gait cycle} + 0.22 & \text{between 0 and 3.6\%} \\ -0.0258(\% \text{ gait cycle} - 3.6)^2 + 0.1327(\% \text{ gait cycle} - 3.6) + 0.17 & \text{between 3.6 and 9\%} \\ \frac{0.14 - 0.1183}{18.5 - 9} (\% \text{ gait cycle} - 9) + 0.1183 & \text{between 9 and 18.5\%} \\ \frac{0.094 - 0.14}{19.5 - 18.5} (\% \text{ gait cycle} - 18.5) + 0.14 & \text{between 18.5 and 19.5\%} \\ \frac{0.19 - 0.094}{49.8 - 19.5} (\% \text{ gait cycle} - 19.5) + 0.094 & \text{between 19.5 and 49.8\%} \\ \frac{0.175 - 0.19}{52.7 - 49.8} (\% \text{ gait cycle} - 49.8) + 0.19 & \text{between 49.5 and 52.7\%} \\ -0.0149(\% \text{ gait cycle} - 52.7)^2 + 0.1(\% \text{ gait cycle} - 52.7) + 0.175 & \text{between 52.7 and 60\%} \\ \frac{0.135 - 0.12}{68 - 60} (\% \text{ gait cycle} - 60) + 0.12 & \text{between 60 and 68\%} \\ \frac{0.087 - 0.135}{72 - 68} (\% \text{ gait cycle} - 68) + 0.135 & \text{between 68 and 72\%} \\ \frac{0.22 - 0.087}{100 - 72} (\% \text{ gait cycle} - 72) + 0.087 & \text{between 72 and 100\%} \\ 0.34 & \text{otherwise} \end{cases} \quad (\text{B.3})$$

CoP Position criterion

$$\Delta\text{CoP}_{\text{pos, threshold}} = \begin{cases} 0.30 & \text{between 0 and 8\%} \\ \frac{0.15 - 0.3}{12 - 8} \% \text{ gait cycle} + 0.30 & \text{between 8 and 12\%} \\ \frac{0.06 - 0.15}{43 - 12} \% \text{ gait cycle} + 0.15 & \text{between 12 and 43\%} \\ \frac{0.3 - 0.06}{16 - 43} \% \text{ gait cycle} + 0.06 & \text{between 43 and 46\%} \\ 0.30 & \text{between 46 and 57\%} \\ \frac{0.015 - 0.30}{62 - 57} (\% \text{ gait cycle} - 57) + 0.3 & \text{between 57 and 62\%} \\ \frac{0.06 - 0.015}{93 - 62} (\% \text{ gait cycle} - 62) + 0.015 & \text{between 62 and 93\%} \\ \frac{0.30 - 0.06}{96 - 96} (\% \text{ gait cycle} - 96) + 0.06 & \text{between 93 and 96\%} \\ 0.30 & \text{between 96 and 100\%} \\ 0.03 & \text{otherwise} \end{cases} \quad (\text{B.4})$$

