Control-enabled gait training with self-balanced lower-limb exoskeletons

Maxime Brunet

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris - PSL

Algorithmes de contrôle permettant l'entraînement à la marche avec des exosquelettes auto-équilibrés pour les membres inférieurs

Control-enabled gait training with self-balanced lower-limb exoskeletons

Soutenue par

Maxime Brunet

Le 29 juin 2023

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Énergétique

Spécialité

Mathématique et Automatique

Composition du jury :

Katja MOMBAUR Full Professor, University of Waterloo	*Rapporteur*
Aaron YOUNG Associate Professor, Georgia Tech	*Rapporteur*
Heike VALLERY Full Professor, TU Delft	*Président, examinateur*
Pierre-Brice WIEBER Chargé de recherche, INRIA Grenoble	*Examinateur*
Florent DI MEGLIO Maître-assistant HDR, Mines Paris	*Examinateur*
Nicolas PETIT Professeur, Mines Paris	*Directeur de thèse*

MINES PARIS | PSL

Abstract

Self-balanced exoskeletons offer individuals with disabilities recovered ambulation capabilities thanks to their embedded control algorithms which realize autonomous and stable walking. This thesis is concerned with enabling patients to use these systems for rehabilitation purposes according to the concept of active gait rehabilitation. In our interpretation, this concept entails giving back some freedom of motion to the patients. This objective is conflicting with ensuring safety, which implies taking control over the motion. The main topic addressed in the thesis is the challenge of reconciling these two objectives.

Our solution is twofold: empower the patient and monitor safety. Firstly, we introduce a split in the control architecture, allowing patients to actively participate in the swing leg motion. Concurrently, the control algorithm of the support leg regulates the unilateral contact forces. Secondly, we solve online optimal control problems in free final time on a reduced model to generate stabilizable trajectories. The duration of these trajectories serves as a determinant for when the control system should take back control over the execution of the walking pattern.

The manuscript presents the necessary control architecture updates along with the development of the safety management algorithm. Experimental results are provided to assess the effectiveness of our approach in both flat-foot and foot-rolling walking exercises.

Résumé

Les exosquelettes auto-équilibrés, grâce à leurs algorithmes de contrôle qui permettent une marche autonome et stable, offrent aux personnes handicapées des capacités de marche retrouvées. Cette thèse vise à permettre aux patients d'utiliser ces systèmes à des fins de rééducation selon le concept de rééducation active de la marche. Selon notre interprétation, ce concept implique de redonner une certaine liberté de mouvement aux patients. Cet objectif est en conflit avec la garantie de la sécurité, qui implique de prendre le contrôle du mouvement. Le principal sujet abordé dans cette thèse est le défi que représente la conciliation de ces deux objectifs.

Notre solution est double : donner de la liberté au patient et en garantissant l'équilibre. Tout d'abord, nous introduisons une division dans l'architecture de contrôle existante, permettant aux patients de participer activement au mouvement de la jambe de vol. Parallèlement, l'algorithme de contrôle de la jambe de support régule les forces de contact. Deuxièmement, nous résolvons de manière embarquée des problèmes de contrôle optimal en temps final libre sur un modèle réduit afin de générer des trajectoires stabilisables. La durée de ces trajectoires sert à déterminer le moment où le système doit reprendre le contrôle de l'exécution de la marche.

Le manuscrit présente les mises à jour nécessaires de l'architecture de contrôle ainsi que le développement de l'algorithme de gestion de la sécurité. Des résultats expérimentaux sont fournis pour évaluer l'efficacité de notre approche lors d'exercices de rééducation de la marche en utilisant une trajectoire sans et une trajectoire avec déroulé du pied.

Remerciements

Je tiens à remercier tous les membres du jury d'avoir pris le temps de relire ce manuscrit et d'assister à ma soutenance. En particulier, merci à Katja Mombaur, pour tous vos commentaires constructifs sur mon manuscrit, merci à Aaron Young, pour toutes vos questions et d'avoir fait le voyage spécialement pour ma soutenance, à Heike Vallery, pour nos échanges et votre aide à ICRA, ainsi que d'avoir présidée ce jury, et merci à Pierre-Brice Wieber pour la clarté de vos travaux qui m'ont grandement éclairci les idées, ainsi que vos questions très intéressantes lors de ma soutenance.

Merci également à toi Marine pour ton soutient constant, ta patience et l'accueil challengeant mais toujours positif que tu réserves à mes idées toujours surprenantes voir extravagantes depuis plus de quatre années. Merci Florent pour la patience que tu fais preuves depuis mes premiers TD d'automatique avec toi et mes mille questions évidentes posées sans vergogne, mais aussi pour tous nos échanges informels, intéressants et tout autant instructifs. Merci Nicolas pour la confiance que vous m'accordez depuis nos premiers échanges lorsque je cherchais un stage de césure, pour votre disponibilité et votre soutien, tant technique que moral, ainsi que tous nos café-débats sans filtre enrichissants.

Je tiens évidemment à remercier mes parents, pour tout ce qu'ils ont fait pour moi, mais en particulier pour m'avoir toujours permis de travailler dans de très bonnes conditions et de m'avoir toujours poussé à donner le meilleur de moi-même. Je remercie également Fanny, pour m'avoir soutenu depuis le début de cette thèse, autant dans les bons moments que dans ceux plus difficiles, pour toujours avoir été de mon côté quand je me plaignais, peu importe ce que tu en pensais vraiment, pour ta patience quand tu écoutais mes anecdotes et élucubrations pas si (peu) intéressantes, et enfin d'avoir supporté quand je te réveillais avec mes insomnies. Je remercie également tous mes amis d'avoir supporté mon indisponibilité chronique et d'avoir pourtant été là pour me changer les idées à chaque fois que j'en avais besoin.

Enfin, je tiens à remercier tous mes collègues, tant ceux du CAS que

ceux de Wandercraft, pour toutes nos discussions enrichissantes et votre aide, avec une mention spéciale pour mes collègues de Wandercraft qui supportent avec patience d'écouter mes idées farfelues sans s'en plaindre, même si je ne choisissais pas toujours le meilleur moment pour vous embêter.

Contents

Acronyms

CBF	Control Barrier Functions
CoM	Center of Mass
CoP	Center of Pressure
DCM	Divergent Component of Motion
DoF	Degrees of Freedom
$\mathcal{FK}$	Forward Kinematics
HZD	Hybrid Zero Dynamics
$\mathcal{ID}$	Inverse Dynamics
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
LIP	Linear Inverted Pendulum
MPC	Model Predictive Control
NLP	Non Linear Program
OCP	Optimal Control Problem
OP	Optimal Planning
PD	Proportional Derivative
QP	Quadratic Program
SoT	Stack-of-Tasks
TR	Time Rescaling
VG	Virtual Guides

Glossary

SoT_1	SoT used by the original passive walking controller of Atalante
SoT_2	SoT used by the proposed active walking controller during single support phase
$\mathcal{Q}$	nominal trajectory mapping time in $\mathbb{R}^+$ to the space of generalized positions in $\mathbb{R}^{18}$
$\mathcal{T}$	nominal joint trajectory mapping time in $\mathbb{R}^+$ to joint space in $\mathbb{R}^{12}$ [rad]
$\mathcal{T}^{sw}$	nominal swing leg joint trajectory mapping time in $\mathbb{R}^+$ to joint space in $\mathbb{R}^6$ [rad]
P	parametric curve mapping curvilinear abscissa in $\mathbb{R}^+$ to joint space in $\mathbb{R}^6$ [rad]
T	Frenet-Serret unit tangent vector to the curve P mapping curvilinear abscissa in $\mathbb{R}^+$ to joint space in $\mathbb{R}^6$ [rad]
s	curvilinear abscissa mapping time in $\mathbb{R}^+$ to arc length, in joint space, in $\mathbb{R}^+$ [rad]
σ	control variable in $\mathbb{R}^+$ [rad]
$\dot{\sigma}$	control variable derivative in $\mathbb{R}$ [rad.s^{-1}]
$\dot{\sigma}^u$	simulated user velocity in $\mathbb{R}$ [rad.s^{-1}]
η^u	simulated user phase variable in $\mathbb{R}$ [s]
$\dot{\eta}^u$	simulated user phase variable velocity in $\mathbb{R}$
m	Total mass of the patient-exoskeleton system [kg]
L	angular momentum about the CoM in $\mathbb{R}^3$ [rad.s^{-1}]
c	CoM position *w.r.t.* the inertial frame, in $\mathbb{R}^3$ [m]
c^*	horizontal reference CoM position in $\mathbb{R}^2$ [m]

$\dot{c}$	CoM velocity the inertial frame, in $\mathbb{R}^3$ [m]
$\dot{c}^*$	horizontal reference CoM velocity in $\mathbb{R}^2$ [m.s^{-1}]
$\ddot{c}^*$	horizontal reference CoM acceleration in $\mathbb{R}^2$ [m.s^{-1}]
$\ddot{c}^d$	desired CoM acceleration in $\mathbb{R}^3$ [m.s^{-2}]
ω	LIP natural frequency in $\mathbb{R}^+$ [s$^-$1]
ξ	DCM in $\mathbb{R}^2$ [m]
ξ^*	reference DCM in $\mathbb{R}^2$ [m]
p^*	reference CoP command in $\mathbb{R}^2$ [m]
p^d	desired CoP position in $\mathbb{R}^2$ [m]
p^{TR}	time-rescaled CoP trajectory mapping curvilinear abscissa $\sigma \in \mathbb{R}^+$ to CoP in $\mathbb{R}^2$ [m]
p^{OP}	optimal CoP trajectory mapping curvilinear abscissa $\sigma \in \mathbb{R}^+$ to CoP in $\mathbb{R}^2$ [m]
x^{TR}	time-rescaled CoM state trajectory mapping curvilinear abscissa $\sigma \in \mathbb{R}^+$ to the CoM state in $\mathbb{R}^4$ [m, m.s^{-1}]
x^{OP}	optimal CoM state trajectory mapping curvilinear abscissa $\sigma \in \mathbb{R}^+$ to the CoM state in $\mathbb{R}^4$ [m, m.s^{-1}]
q	vector of generalized positions in $\mathbb{R}^3 \times \mathbb{R}^{15}$ [m x rad]
q^{sw}	vector of swing leg joint positions in $\mathbb{R}^6$ [rad]
q^{sp}	vector of support leg joint positions in $\mathbb{R}^6$ [rad]
q^{un}	vector of unactuated DoF in $\mathbb{R}^3 \times \mathbb{R}^3$ [m x rad]
q_t	vector of target joint positions in $\mathbb{R}^{12}$ [rad]
q_t^{sw}	vector of swing leg target joint positions in $\mathbb{R}^6$ [rad]
q_t^{sp}	vector of support leg target joint positions in $\mathbb{R}^6$ [rad]
q^{TR}	time-rescaled joint trajectory, computed from the nominal joint trajectory $\mathcal{T}$, mapping time in $\mathbb{R}^+$ to joint space in $\mathbb{R}^{12}$ [rad]
$\dot{q}$	vector of generalized velocities in $\mathbb{R}^3 \times \mathbb{R}^{15}$ [m x rad]$^{-1}$
$\dot{q}^{sw}$	vector of swing leg joint velocities in $\mathbb{R}^6$ [rad.s^{-1}]
$\dot{q}^{sp}$	vector of support leg joint velocities in $\mathbb{R}^6$ [rad.s^{-1}]
$\dot{q}^{un}$	vector of generalized velocities in $\mathbb{R}^3 \times \mathbb{R}^{15}$ [m x rad]$^{-1}$
$\dot{q}_t$	vector of target joint velocities in $\mathbb{R}^{12}$ [rad.s^{-1}]
$\dot{q}_t^{sw}$	vector of swing leg target joint velocities in $\mathbb{R}^6$ [rad.s^{-1}]

$\dot{q}_t^{sp}$	vector of support leg target joint velocities in $\mathbb{R}^6$ [rad.s^{-1}]
$\ddot{q}$	vector of generalized accelerations in $\mathbb{R}^3 \times \mathbb{R}^{15}$ [m x rad]$^{-2}$
$\ddot{q}^{sw}$	vector of swing leg joint accelerations in $\mathbb{R}^6$ [rad.s^{-2}]
$\ddot{q}^{sp}$	vector of support leg joint accelerations in $\mathbb{R}^6$ [rad.s^{-2}]
$\ddot{q}^{un}$	vector of generalized accelerations in $\mathbb{R}^3 \times \mathbb{R}^{15}$ [m x rad]$^{-2}$
$\ddot{q}_t$	vector of target joint accelerations in $\mathbb{R}^{12}$ [rad.s^{-2}]
$\ddot{q}_t^{sp}$	vector of support leg target joint accelerations in $\mathbb{R}^6$ [rad.s^{-2}]
τ	vector of exoskeleton joint torques in $\mathbb{R}^12$ [N.m]
τ^{sw}	vector of exoskeleton swing leg joint torques in $\mathbb{R}^6$ [N.m]
τ^{sp}	vector of exoskeleton support leg joint torques in $\mathbb{R}^6$ [N.m]
τ_u	vector of patient's joint torques in $\mathbb{R}^{12}$ [N.m]
τ_u^{sw}	vector of patient's swing leg joint torques in $\mathbb{R}^6$ [N.m]
τ_u^{sp}	vector of patient's support leg joint torques in $\mathbb{R}^6$ [N.m]
τ_{PD}^{sw}	vector of exoskeleton swing leg high-gain PD joint torques in $\mathbb{R}^6$ [N.m]
τ_{GC}^{sw}	vector of exoskeleton swing leg gravity compensation torques mapping joint position in $\mathbb{R}^6$ to joint torques in $\mathbb{R}^6$ [N.m]

Chapter 1

Introduction

Chapitre 1- Introduction: *Ce chapitre introductif décrit les exercices de rééducation de la marche, et en particulier s'attarde sur les principales motivations des développements robotiques pour la rééducation. Les différentes familles d'exosquelettes, sous-catégorie des robots de rééducation, sont présentées, ainsi que les différents types d'exercices de rééducation qu'ils permettent. Une caractéristique essentielle de ces exercices est mise en avant : la possibilité pour le patient de modifier le mouvement à l'aide de sa force musculaire. Nous appelons* exercices actifs *les exercices permettant une telle modification du mouvement par le patient, et, par opposition,* exercices passifs *ceux ne permettant pas de modification du mouvement. L'exosquelette auto-équilibré Atalante, cible des travaux présentés dans ce manuscrit, et son mode nominal de fonctionnement, permettant la rééducation passive de la marche, sont décrits. Ensuite, deux catégories de changement permis par les modes de controôle actifs d'exosquelettes sont mis en avant: les changements de la forme des mouvements et les changements de la durée des mouvements. Finalement, la principale contribution de ce manuscrit, un contrôleur offrant une liberté temporelle au patient tout en monitorant l'équilibre, est présentée, ainsi que l'organisation du reste du manuscrit.*

Contents

1.1 Exoskeletons for walking rehabilitation

1.1.1 A concise view of walking rehabilitation

Human walking can be seen as repetitively achieving a gait[1]. Each leg has a stance phase and a swing phase. These define, for the human, single and double support phases. They are illustrated on Figure 1.1. At some instant of the double support phase, the weight is transferred from one leg to the other and the Center of Pressure (CoP)[2] shifts from one foot to the other. At all instants of single support phases, the human body is propelled forward. Each single support phase includes at least one controlled falling phase [Perry2010], *i.e.* a phase during which the Center of Mass (CoM) of the human body is not directly above the base of support, the support foot, and the landing of the other foot is required to prevent an actual fall[3]. Walking requires two primary abilities: transferring the weight during double support phases and moving both the CoM and the swing foot forward during single support phases.

[1] A gait is a cyclic motion pattern that produces locomotion through a sequence of foot contacts with the ground [Haynes2006].

[2] The Center of Pressure is the average of all the contact points weighted by the vertical component of the force pushing on the ground.

[3] This is a particularity of human walking.

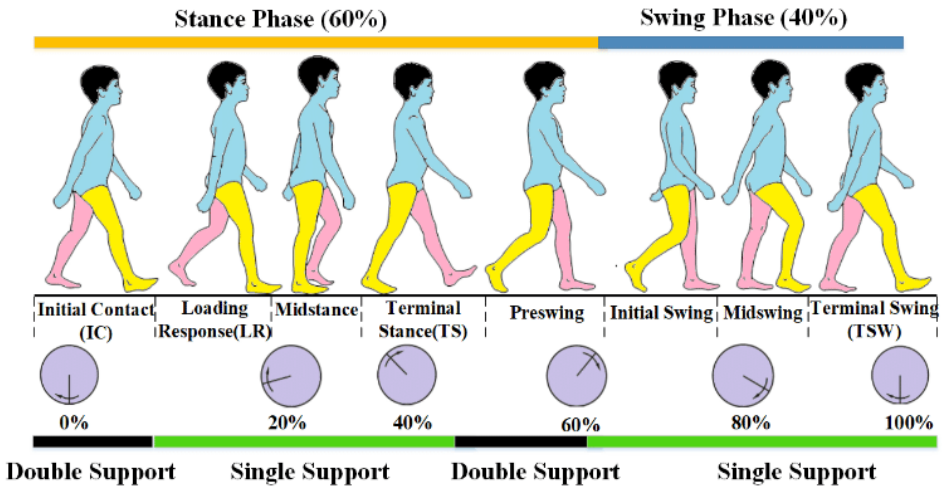


Figure 1.1: Phases of gait cycle (from [Kour2019]).

Impaired patients following gait training programs are trained to regain the ability to walk (see Figure 1.2 for examples). Because of their neurological disorders[4], they have strong difficulties controlling the weight transfer or producing the efforts required to generate healthy single support motion by themselves. Walking rehabilitation is exhausting for both the physiotherapist and the patient because they need to maintain the balance of the patient during the exercise by producing important external forces, in a well-coordinated manner. These efforts often even lead to lower exercise intensity and shorter durations of training sessions than what would be ideal. This is an important matter as strong evidence indicates that higher intensity (in the duration of exercises, number of repetitions and level of effort) leads to faster recovery [Lohse2014; Lang2015; Hornby2019; Moore2020].

With their powerful actuators, robotic devices have the potential to handle safety [Gassert2018], helping both the physiotherapist in their work and the patient in their recovery process. Robotic rehabilitation is especially promising for gait training as the potential load that could be lifted from physiotherapists and patients is heavier. For this reason, exoskeletons are becoming more and more popular for gait training [Doppmann2015; Young2017; Eveld2021; Shahrokhshahi2022; Kang2023].

[4]Neurological disorders encompass numerous disorders related to nervous system disorders such as spinal cord injuries, cerebral palsy, or stroke.

(a) Weight transfer training

(b) Step training

Figure 1.2: Walking rehabilitation examples (from *Helen Hayes Hospital* https://youtu.be/g__BYaS9viw).

1.1.2 Using exoskeletons during training

Exoskeletons are a special class of robots [Gassert2018]. They are meant to be worn by humans *s.t.* they can exert efforts on several body parts simultaneously. They vastly differ from robotic manipulators which interact with humans only at their end-effector (see [Maciejasz2014] and references therein). Exoskeletons are designed to align some of their joints with the morphological joints of the human wearer, *s.t.* they share some Degrees of Freedom (DoF), as opposed to manipulators (which rarely align their joints with humans').

Lower-limb exoskeletons (see examples on Figure 1.3) are a subclass

specifically designed for walking rehabilitation or ambulation[5]. They are the subject of this thesis. From now on, we only discuss lower-limb exoskeletons, and we simply call them "exoskeletons".

Passive vs active gait training

With such systems, two modes of operation must be distinguished. *Passive walking* is the mode in which exoskeletons fully assign the patients' leg motion, irrespective of the patient's efforts. This mode allows *passive gait training* (*i.e.* rehabilitation with no participation required from the patient, see [Quintero2012; Bortole2015]). By contrast, *active walking* is the mode in which exoskeletons allow patients to significantly modify the motion of the patient-exoskeleton system through their efforts (see Section 1.3.1 for examples). Such active walking modes are used for *active gait training* (*i.e.* rehabilitation with participation required from the patient). The control methodology of Atalante described in Section 1.2.2 allows passive gait training.

Passive gait training is reported to be less effective than active gait training, as it leads to the phenomena of "learned helplessness" [Skinner1979; Wool1980]. Intuitively, one can easily understand that rewarding the proper efforts and penalizing the inappropriate ones should encourage the patients toward producing the right efforts. If the patient has no freedom of motion and senses no difference in the robot's motion regardless of their efforts, they will tend to let the robot produce all the efforts. This is a negative and undesired outcome.

Three families of exoskeletons

During gait training the exoskeleton must guarantee the safety of the patient. Chronologically, three types of solutions have been developed for this purpose. They have driven the development of three different types of exoskeletons: *platform-based exoskeletons* (for instance Lokomat pictured in Figure 1.3a), which are virtually fixed to the ground (either by being truly fixed to the ground or by being sufficiently heavy *s.t.* their base remains still), *crutched exoskeletons* (for instance Ekso pictured in Figure 1.3b), designed to be used with crutches or external help for stabilization, and, more recently, *self-balanced exoskeletons* (for instance REX pictured in Figure 1.3c or

[5] Ambulation is the act of moving around by walking. There is no rehabilitation objective during ambulation, only a mobility one.

Atalante Figure 1.3d), designed to balance themselves during walking and not requiring crutches or external help.

These various types of exoskeletons constitute various trade-offs between the realism[6], safety[7] and intensity[8] of the walking exercises. Platform-based exoskeletons are, by definition, the most stable as they ensure the patients will never fall. They also allow high-intensity walking exercises. However, they reduce the realism of the walking exercise. Hence, they favor safety and intensity over the realism of the walking exercise.

The realism of the walking exercises is higher with crutched exoskeletons, as the legs of the patients have to support their weight. However, they leave the patients and/or physiotherapists with the exhausting walking stabilization task, which limits the intensity of the rehabilitation exercises. The probability of falling with these exoskeletons is not small. Hence, they favor realism over the safety and intensity of the walking exercise.

Self-balanced exoskeletons, while not suffering from realism or intensity issues, face a major challenge: ensuring the balance of the walk. Designing a control algorithm to address this issue is not straightforward. In principle, self-balanced exoskeletons could re-employ the vast humanoid and bipedal robots' literature to realize stable walking. However, the presence of a patient in the system implies specific needs, unanswered by the literature[9]. First, some freedom of motion must be given to the patients to boost their rehabilitation, while humanoid robot controllers favor accurate tracking of the planned motion. Second, by producing inappropriate efforts, and because the behavior of the patient is only approximately known and difficult to model, patients may disturb the balance of the system during walking exercises, which implies the design of stabilization algorithms especially robust to disturbances and model errors. We are interested in the design of gait training control methods for self-balanced exoskeletons.

The work presented in this manuscript targets one of these self-balanced exoskeletons, Atalante. We now describe its hardware and the control algorithms designed for passive walking.

[6]We refer to *realism of the walking* for the ratio of patient weight supported by their legs.

[7]We refer to *safety of the walking* for the probability of falling during walking.

[8]We refer to *intensity of the walking exercise* for the duration of the exercise and the level of effort produced by the patient's legs.

[9]Roughly speaking there is a 1-1 ratio between the mass of the human and the mass of the exoskeleton.

(a) Lokomat (https://www.hocoma.com)

(b) Ekso (https://eksobionics.com/)

(c) REX (https://www.rexbionics.com/)

(d) Atalante (https://www.wandercraft.eu/)

Figure 1.3: Exoskeleton examples: (a) platform-based exoskeleton, (b) crutched exoskeleton, (c) self-balanced exoskeleton, (d) self-balanced exoskeleton.

Figure 1.4: Atalante

1.2 Passive walking with a self-balanced exoskeleton

1.2.1 Description of the Atalante platform

The exoskeleton Atalante is pictured on Figure 1.4. It has been certified as a medical device for passive use by paraplegic patients in the European market (in 2019) and in the US market (in 2023). Atalante carries a patient standing on its feet *s.t.* the robot supports its weight. In addition, the patient is strapped to the exoskeleton at his feet, shins, thighs and trunk (see Figure 1.3d). To ease its use in medical care centers it is possible to adjust the length of the shins and thighs to match the patient's body parts length and align their joints with the exoskeleton. In total, the exoskeleton has 13 segments, and 7 of them are attached to the body parts of the patient. The 13 segments constitute a kinematic chain having 12 actuated revolute joints (6 for each leg: two joints at the ankle, one joint at the knee, and three joints at the hip). These joints are mechanically optimized to be best aligned with the morphological joints of the patients.

This mechanical structure is put into motion by 12 independent electrical actuators, each composed of an electrical motor and a gearbox. The whole exoskeleton is powered by two batteries, located on its thighs, which offer an autonomy of two hours under intensive use. Overall, the exoskeleton weighs approximately 80 kg. Except for the ankles, all the gearboxes feature a high reduction factor (approximately 100). The position of each joint of the exoskeleton is measured by encoders located on the motor side. Atalante also possesses 8 single-axis force sensors (4 under each of its feet) used to sense the contact forces. In addition, 5 Inertial Measurement Unit (IMU) are rigidly mounted on its feet, shins and back of the exoskeleton. They serve two purposes. First, they are used to estimate the underactuated position and orientation of the back of the robot, through the sensing of the proper acceleration, which includes the gravity vector defining the local frame, the direct kinematic model of the exoskeleton and the joint angles. Then, they are used for the estimation and control of the mechanical structure's deformations. Indeed, the Atalante mechanical structure undergoes deformations, similar to those of rotational springs located in its hips and ankles [Vigne2021][10].

Atalante's control algorithms are running on the onboard dual-core Intel Core i5-4300U @ 1.90GHz computer. The robot trajectories, tailored to each patient's measurements (their weight, height and length of the different parts of their legs), are generated offline for each patient. Various motion patterns are available: walking in a straight or curved line, walking sideways and backward, turning around, standing up, and sitting down. The modes of control and motions of the exoskeleton are commanded by a two-step procedure: first, the patient uses a remote to select the desired control mode, then the motion is triggered by the user when a specific movement of the patient's trunk is detected, using an IMU fitted within the jacket worn by the patient (visible on Figure 1.3d).

Below, we describe the passive walking controller of Atalante and give a brief overview of the humanoid robots' literature which inspired its development.

1.2.2 Passive walking control architecture: plan then execute

The primary control paradigm of Atalante is to *plan then execute*. It takes the form of a passive walking controller, inspired by the humanoid robots' literature, and is composed of three elements (illustrated on Figure 1.5):

[10]The estimation and control of the deformations are beyond the scope of the presented work, and we refer the interested reader to the work in [Benallegue2014; Benallegue2015; Mifsud2016] for humanoid robots, and in [Vigne2021] for a thorough study on Atalante.

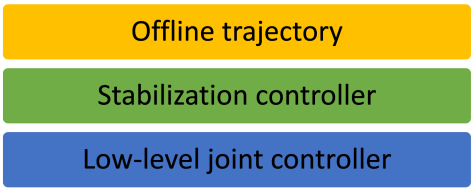


Figure 1.5: Elements of Atalante's passive walking controller.

- a trajectory of the exoskeleton's joints and underactuated DoFs, computed offline;
- a stabilization algorithm, adapting this joint trajectory online to track the underactuated DoFs trajectory;
- a low-level controller, composed of joint-independent high-gain PD controllers, tracking the modified joint trajectory.

To follow the paradigm of humanoid robots, the patient is viewed as a set of rigid bodies, rigidly fixed to the robot, adding mass and inertia to the exoskeleton. The passive walking controller of Atalante is inspired by two broad families of control methodologies used to realize walking motions with humanoid robots[11]. A first version has appeared in [Gurriet2018][12].

Concerning trajectory design, the main sources of inspiration are the works presented below. The *method of virtual constraints and Hybrid Zero Dynamics (HZD)* is exposed in [Westervelt2007; Ames2014; Reher2016; Grizzle2017]. This control methodology is based on the offline design of optimal walking trajectories [Hereid2016], which possess some unstable dynamics that are stabilized online through the feedback of their timescale. The generalization of this methodology is still the subject of some recent work [Castillo2019; Castillo2020; Galliker2022]. A second family of controllers generates the trajectories of the robot online. Because the trajectory is computed online with these methodologies, it is better suited to react to perturbations in a Model Predictive Control (MPC) fashion, execute various motions, and modify the plan online. A line of work relies on some approximation of

[11] The benefit of a third family of control methodologies [Castillo2021; Siekmann2021; Singh2022], leveraging the recent progress of machine learning, is still under investigation for self-balanced exoskeletons [Duburcq2020; Duburcq2022b] and does not constitute the primary control paradigm of Atalante.

[12] The same control methodology can be applied to realize a variety of motions like walking in a straight line, side-ways, turning in place, or climbing stairs (see [Huynh2021] and accompanying video `https://youtu.be/4bmuNlnREaA`).

the dynamics[13] of the robot to generate trajectories of the reduced state of the robot online, either as feedback controllers [Kajita2003] or as the solution to an optimal trajectory design (in the form of a QP) over some horizon [Wieber2006b; Herdt2010; Caron2016; Griffin2017], at a frequency typically lower than 100 Hz. The controller of Atalante uses the same offline design of optimal walking trajectories as [Hereid2016] but does not rely on the HZD stabilization

Concerning stabilization, we employ the whole-body admittance algorithm [Caron2019], which enables position-controlled robots to regulate the unilateral contact forces and track the CoM trajectory. In this paradigm, the state trajectories are converted at high-frequency (typically 1 kHz) into an instantaneous acceleration in the joint-space through Inverse Kinematics (IK)[14]. The next step is to achieve these joint accelerations. In principle, this can be done by torque control or position control. However, mostly due to the patient, the torques are vastly uncertain. This discards the torque control strategies found on Atlas [Johnson2015], TORO [Englsberger2014] or Talos [Villa2022], and favor instead position control ones found on HRP-4 [Kaneko2011].

Finally, low-level controllers are implemented to achieve disturbance rejection and to track the previously defined positions. They take the form of independent high-gain PD controllers (running at 1 kHz). This decentralized approach shows satisfactory results due to the high reduction factor gearboxes (see [Finet2017] for more details).

In this passive walking controller, the patient efforts are not accounted for: they are only seen as unmodeled disturbances to be compensated for by the low-level controllers. In other words, all the patient deviations from the trajectory are hampered by the exoskeleton. Despite its merits for passive gait training, this control architecture is not suited for active gait training, which is the topic of this thesis.

[13] The reduced-state trajectory being generated using an approximate model of the dynamics, some conservatism during planning must be considered, which limits the performance. Some of the most advanced works [Mansard2018; Dantec2022] on humanoid robots control exhibit online motion generation and tracking considering the full dynamics of the robots to alleviate this limitation. However, even by today's computational power standards, these algorithms are still too heavy a computational burden to run online at a high update rate on the embedded computers available in robots and have to run on remote computers. Other recent approaches attempt to ensure a consensus between reduced and complete dynamics, as proposed in [Budhiraja2019].

[14] An alternative is to compute the torques directly by Inverse Dynamics ($\mathcal{ID}$) [Herzog2016; Kuindersma2016; Koolen2016]

1.3 Active gait training with self-balanced exoskeletons

1.3.1 Spatial and temporal freedom for active gait training

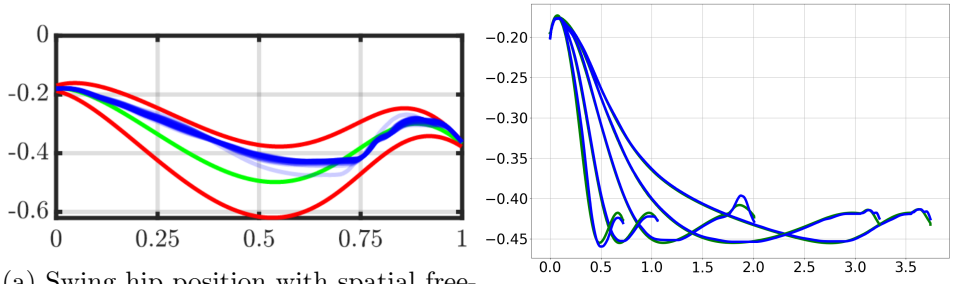
A variety of active walking controllers have been proposed for active gait training with platform-based or crutched exoskeletons. They allow the patient to participate in the motion by relaxing the tracking of a reference trajectory[15].

Reference trajectory tracking can be relaxed in various ways, leading to different types of freedom: either *temporal freedom*, *spatial freedom*, or both at the same time can be offered. We consider that a control law offers *temporal freedom* (respectively *spatial freedom*) if, through their efforts, a patient can substantially change the duration of the steps (respectively the shape of the steps' trajectory *i.e.* the geometric path followed during the steps). These two freedoms are illustrated on Figure 1.6. Most of the active gait training controllers found in the literature offer some form of both temporal and spatial freedom. For instance, in [Aoyagi2007; Banala2009; Vallery2009; Duschau-Wicke2010], a nearest-neighbor algorithm is used to offer temporal freedom by finding the closest point, within a reference trajectory, to the current position of the legs of the patient (in joint-space)[16]. This methodology is extended in [Martínez2018] to provide step-to-step variation of the step length, through online modification of the reference trajectory between each step. Another control law is proposed in [Martínez2019], based on a flow field rather than a potential field, in order to combine in a more native way the freedom provided by the controller from [Duschau-Wicke2010] and some form of assistance to the user.

Altering the trajectories of the system is troublesome as it may jeopardize the balance of the exercises. This problem is not critical in the works cited above because the exoskeletons under consideration are platform-based exoskeletons, where the safety of the exercise is never at risk, or crutched exoskeletons, which leave the physiotherapist or patient in charge of the balance of the exercise. With self-balanced exoskeletons, maintaining balance while allowing freedom is a seldom studied problem. This specific problem lies at the intersection of the literature on humanoid walking robots (see Section 1.2.2) and the one on robotic gait training (see above). While these two fields of research have received numerous contributions over the past

[15] Some control laws do not use any reference trajectory. This is the case, for instance, in [Ekkelenkamp2005; Vallery2008; Murray2015]. However, these paradigms deviate further from Atalante's passive walking paradigm, therefore we do not rely on them for inspiration.

[16] A PD controller with deadband is employed to help the patient to stay in the vicinity of the reference trajectory while offering spatial freedom around the reference trajectory.



(a) Swing hip position with spatial freedom from [Gurriet2020]

(b) Swing hip position with simulated temporal freedom

Figure 1.6: Comparison between spatial and temporal freedom on one joint trajectory. The larger envelope on the right makes temporal freedom easier to perceive for patients.

20 years, their intersection has only drawn little attention. This is certainly caused by the small number of available self-balanced exoskeletons[17].

1.3.2 Exploiting spatial freedom with self-balanced exoskeletons

To our knowledge, only one active walking controller for self-balanced exoskeletons, demonstrated with Atalante, has been proposed in [Gurriet2020]. This controller, illustrated on Figure 1.6a, offers spatial freedom to the patient by relaxing the tracking of the trajectory on the swing hip and knee, and uses Control Barrier Functions (CBF) to limit the deviation of these joints from a nominal walk pattern. Specifically, safe regions are designed in joint-space *s.t.* the freedom of the patient is saturated at the beginning and the end of each step, and, therefore, the exoskeleton feet lift and land correctly.

This controller constitutes the first demonstration of an active walking controller for self-balanced exoskeletons. It offers spatial freedom around the offline trajectory, but no temporal freedom. In this thesis, we propose a rehabilitation controller for Atalante which offers temporal freedom around the offline trajectory.

[17]To this day, REX (Rex Bionics) and Atalante [Gurriet2018] are the only self-balanced exoskeletons on the market, while the XoMotion (Human in motion Robotics) [Shahrokhshahi2022] is still under development

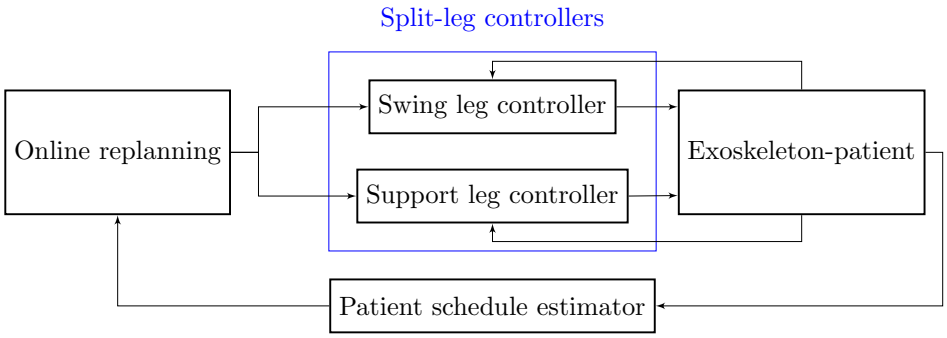


Figure 1.7: Proposed control architecture for active gait training: split-leg controllers carefully synchronized with patient schedule through online replanning.

1.3.3 Contribution of the thesis: exploit temporal freedom and maintain balance

Temporal freedom has the potential to make the patient participate more significantly. This point is illustrated on Figure 1.6 which reports the swing hip position as a function of time for various degrees of spatial (left) and temporal (right) freedom, respectively. The envelope of Figure 1.6b is larger than Figure 1.6a, hinting that temporal freedom would be easier to perceive for patients. Our objective in this thesis is to develop a new feature for Atalante: a rehabilitation controller giving substantial temporal freedom to the patient without requiring the physiotherapist's assistance.

We propose a control system update. This update has a dual purpose: it *empowers the patient and monitors safety.*

Overview of the solution The modifications are schematically depicted on Figure 1.7. We choose to let the patient physically contribute to the motion of the swing leg and, consequently, allow modulation of the velocity at which a predefined gait is traveled. Because this directly impacts safety, a new reference trajectory is computed for the low-level controllers of the whole system. This is done online by an optimization-based trajectory planning algorithm.

Detailed view The two legs of Atalante are the subjects of very distinct changes. On the swing leg, we proceed as follows. The nominal gait for the two-legged system, generated using the same offline optimization methodology

as for the passive walking, serves to define a geometric path for the swing leg. The exoskeleton efforts in the longitudinal direction of the path are nullified and left for the patient to produce, while the robot motion is strictly controlled in the hyperplane orthogonal to the path. To this end, we rely on the Virtual Guides (VG) methodology which maps the high-dimensional user efforts to a one-dimensional quantity: the velocity at which the swing leg's geometric path is followed. Implicitly, this defines a new schedule for the nominal trajectory, the *patient schedule*, which is followed as long as safety is not threatened.

On the support leg, the control structure is also updated. Because the swing leg DoFs have been updated (see above), only the support leg can be used for the admittance controller. This admittance controller successfully tracks reference trajectories that satisfy the *unilateral contact constraints*.

To define a reference trajectory synchronous with the patient schedule, a simple time rescaling of the nominal gait would be possible but will most likely violate the aforementioned constraints. Instead, we chose to adopt an online planning strategy.

The online planning strategy we consider solves an Optimal Control Problem (OCP) over an unspecified horizon for a Linear Inverted Pendulum (LIP) model, which represents the overall balance dynamics of the system (patient+exoskeleton). The patient schedule defined above is treated as a penalty on the final time. The unilateral contact constraints is a constraint on the CoP. To ensure the feasibility of the next step, it is sufficient to require that the trajectory endpoint satisfies some geometric constraints. These are included in the OCP. The simplicity of the LIP model allows us to use a fast resolution method for this nonlinear OCP, relying on the theoretical study about the nature of the set of feasible times for the LIP dynamics (*i.e.* the set of final times for which a solution to the OCP exists).

Note that, although it is known the Atalante platform undergoes deformations, we do not try to compensate them in the swing or support leg control algorithms, nor the replanning. This choice is made in order to alleviate some complexity to the already complex problem at hand. However, these deformations might affect the performance of our algorithms when testing them on the actual platform Atalante.

Thesis outline This manuscript presents the development of the solution described above and experimental results stressing its performance.

Chapter 2: Atalante: models and passive control laws. This chapter presents in detail the experimental platform Atalante and its control methodology during passive walking.

Chapter 3: Empower the patient for gait training. This chapter presents the controller enabling temporal gait training. In particular, the split of the exoskeleton's DoF in half is detailed. The poor performance of a simple time rescaling strategy is exposed.

Chapter 4: Monitor safety. This chapter presents the replanning methodology developed to ensure the balance of the system. The method for fast solving of the OCP, compatible with real-time requirements on the Atalante's embedded controller, is provided. A theoretical study of the nature of the solution space is provided in Appendix A. Extensive experimental results are presented.

Chapter 5: Conclusion and perspectives. This chapter stresses the benefits and limitations of the proposed active gait training controller. Possible extensions are presented.

Publications in this thesis

The work done in this thesis led to the following published results:

- Maxime Brunet, Marine Pétriaux, Florent Di Meglio, and Nicolas Petit. "Fast Replanning of a Lower-Limb Exoskeleton Trajectories for Rehabilitation". In: 2022 IEEE 61st Conference on Decision and Control (CDC). Dec. 2022
- Maxime Brunet, Marine Pétriaux, Florent Di Meglio, and Nicolas Petit. *Enabling Safe Walking Rehabilitation on the Exoskeleton Atalante: Experimental Results*. Apr. 17, 2023. preprint to appear in 2023 IEEE International Conference on Robotics and Automation (ICRA)

A video describing the main experiments done in this thesis is available at https://youtu.be/_1A-2nLy5ZE.

Chapter 2

Atalante: models and passive walking controller

Chapitre 2 - Atalante: modèles et contrôleur de marche passif: *Dans ce chapitre, nous décrivons le robot Atalante et les principaux modèles utilisés pour décrire le système patient-exosquelette. Ensuite, nous décrivons le contrôleur nominal d'Atalante qui permet de réaliser des exercices passifs de rééducation de la marche, composé de deux éléments principaux. Le premier est un problème de commande optimale résolu hors-ligne pour obtenir des trajectoires du système. Nous décrivons en particulier deux trajectoires de marches qui seront utilisées comme cas d'usage du travail présenté dans ce manuscrit : une trajectoire de marche simple, sans déroulé du pied, et une trajectoire de marche plus complexe, avec un déroulé du pied. Le deuxième élément composant ce contrôleur est une loi de contrôle utilisée en ligne pour stabiliser ces trajectoires lors de leur exécution.*

Contents

In this chapter, we describe the hardware and the passive walking control law of the exoskeleton Atalante. Contrary to walking rehabilitation, during which the patient efforts on the exoskeleton are substantially impacting the motion of the robot, in *passive walking* the controllers completely assign the motion of the robot, no matter the level of patient efforts.

We start in Section 2.1 by describing the mechanical structure, actuators and sensors of Atalante. Then, we derive the equation of motions under the assumption that the patient motion is completely assigned by the exoskeleton. The governing equations are similar to those of a humanoid robot. Next, in Section 2.2, we describe how trajectory generation for the patient-exoskeleton system can be cast into a non-linear Optimal Control Problem (OCP). A direct collocation transcription of the OCP is described. It is used offline. We detail two specific instances of OCPs:

- a *flat-foot* trajectory, for which the supporting leg switches instantaneously from one leg to the other. There is no double support phase in between two single support phases, and the support foot is always resting flat on the ground.

- a *foot-rolling* trajectory, for which there exist double support phases. During these, the supporting feet are not resting still on the ground

but are rolling on their front (for the trailing foot) or back edges (for the leading foot). The support foot is resting flat on the ground during single support phases.

The flat-foot trajectory is simpler to study but has a lower average translational velocity and is less anthropomorphic than the foot-rolling trajectory. It will serve as our use-case trajectory throughout Chapter 3 and Chapter 4. The foot-rolling trajectory will be used in the experiments discussed in Chapter 5. Finally, in Section 2.3, we expose how these offline trajectories are tracked by a stabilization control law based on a linearized model of the system, relying on position-based impedance control [Hogan1984; Heinrichs1996], also called *admittance control*.

2.1 Atalante: description and dynamical models

2.1.1 Definitions of the frames of reference

We consider three axes defined *w.r.t.* the patient, as depicted on Figure 2.1. The X-axis points forward, the Y-axis to the left, and the Z-axis is aligned with the local gravity vector, pointing upward. The local frame of reference is chosen lying on the ground, under one of the exoskeleton feet[1]. The sagittal, frontal and transverse planes are the three orthogonal planes intersecting at the patient's CoM as depicted on Figure 2.1.

2.1.2 Description of Atalante

The Atalante exoskeleton is depicted on Figure 2.2. It is composed of 13 rigid bodies linked together by twelve actuated revolute joints (in blue). Except for the two Henke ankle axes, all of them are orthogonal to one of the planes defined on Figure 2.1 when the exoskeleton is standing still. The axis of each knee joint is orthogonal to the sagittal plane. The three axes of the three hip actuators of each hip are orthogonal to one of the sagittal, frontal or transverse planes. Each ankle possesses two joints: one orthogonal to the sagittal plane, and one aligned with the morphological Henke axis[2]. The rigid body connecting the two legs of the exoskeleton is called the *back* body. The upper-leg part (between the sagittal hip joint and the knee joint) is called the *thigh* body. The lower-leg part (between the knee joint and the

[1] It is updated during the walk.

[2] This Henke axis is contained in the sagittal plane, and is equivalent to the X axis depicted on Figure 2.1, but tilted by 37 degrees upward around the Y axis.

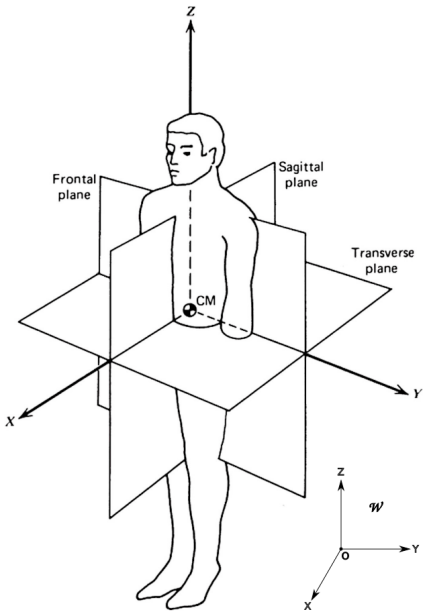


Figure 2.1: Definition of the sagittal, frontal and transverse planes (adapted from [Winter2009].

sagittal ankle joint) is called the *shin* body. The metal plate in contact with the ground which supports the patient's foot is called the *foot* body.

Atalante is equipped with several types of sensors (see on Figure 2.2). In details, encoders are sensing the angular positions[3] of each joint of Atalante. Single-axis force sensors are located under each foot of the robot (4 sensors under each foot) and sense the unilateral contact forces with the ground. Several Inertial Measurement Unit (IMU), rigidly fixed to the exoskeleton, are sensing the proper acceleration and angular velocity of several bodies of the exoskeleton.

The patient is fastened using a special vest and straps for the feet, shins and thighs, visible on Figure 2.2.

2.1.3 Kinematic model

The fastening system of Atalante completely assigns the positions of the lower limbs of the patient *w.r.t.* the robot. The patient's torso is also fastened to the exoskeleton, but some freedom in the upper-body motion is allowed (in

[3]The velocity and acceleration of each joint are obtained through numerical differentiation of their position.

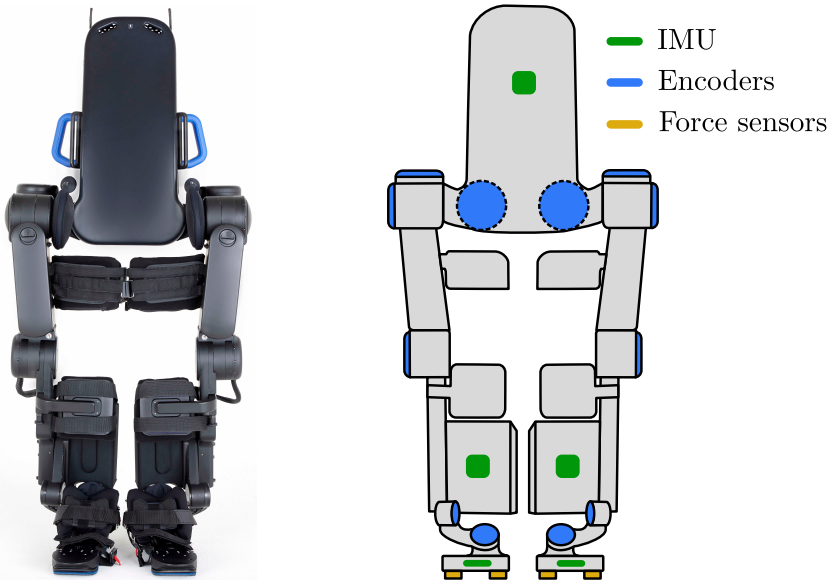


Figure 2.2: Atalante picture (Left). Atalante schematic hardware (Right).

particular in their head and arms motion). We neglect the impact of these disturbing motions on the dynamics of the patient-exoskeleton system and simply consider that the patient is an additional rigid body attached to the exoskeleton.

According to the description above, we model the patient-exoskeleton system as an articulated rigid-body system of mass m (the patient mass plus the exoskeleton's) with 12 actuated Degrees of Freedom (DoF), corresponding to the joints of the exoskeleton, aligned with the patient's, and 6 unactuated DoF, defining the position and orientation of the back of the exoskeleton, *w.r.t.* the inertial frame $\mathcal{W}$.

In details, following [Featherstone2008], we denote $q \in \mathbb{R}^{18}$ the vector of generalized positions, $(q^{sw}, q^{sp}) \in \mathbb{R}^6 \times \mathbb{R}^6$ the vector of actuated positions for the swing leg and the support leg, and $q^{un} \in \mathbb{R}^6$ the vector of unactuated positions, *s.t.*

$$q = \begin{pmatrix} q^{sw} \\ q^{sp} \\ q^{un} \end{pmatrix} \tag{2.1}$$

Likewise, we denote $\dot{q} \in \mathbb{R}^{18}$ (resp. $\ddot{q} \in \mathbb{R}^{18}$) the vector of generalized velocities (resp. acceleration), $(\dot{q}^{sw}, \dot{q}^{sp}) \in \mathbb{R}^6 \times \mathbb{R}^6$ (resp. $(\ddot{q}^{sw}, \ddot{q}^{sp}) \in \mathbb{R}^6 \times \mathbb{R}^6$) the vector of actuated velocities (resp. acceleration) for the swing leg and the

support leg, and $\dot{q}^{un} \in \mathbb{R}^6$ (resp. $\ddot{q}^{un} \in \mathbb{R}^6$) the vector of unactuated velocities (resp. acceleration), *s.t.*

$$\dot{q} = \begin{pmatrix} \dot{q}^{sw} \\ \dot{q}^{sp} \\ \dot{q}^{un} \end{pmatrix} \quad \ddot{q} = \begin{pmatrix} \ddot{q}^{sw} \\ \ddot{q}^{sp} \\ \ddot{q}^{un} \end{pmatrix} \tag{2.2}$$

We employ the general model of the displacement of a rigid body B *w.r.t.* to a frame W, living in the Special Euclidean Group $SE(3)$. It is defined by the position $O_B \in \mathbb{R}^3$ of the origin of B, and the rotation matrix ${}^W R_B \in SO(3)$. This displacement is represented by a homogenous transformation matrix, denoted ${}^W X_B \in SE(3)$, which writes

$${}^W X_B = \begin{bmatrix} {}^W R_B & O_B \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2.3}$$

In the rest of the thesis, ${}^W X_B$ is referred to as a *transform*. The Special Euclidean Group $SE(3)$ is equipped with the composition operator $\times_\circ$ *s.t.*

$$\times_\circ : {}^W X_L \times {}^L X_B \in SE(3)\times SE(3) \mapsto {}^W X_L \times_\circ {}^L X_B \triangleq {}^W X_B \in SE(3) \tag{2.4}$$

For simplicity of notations, we omit this composition operator for the rest of the thesis.

From the vector of generalized positions, and a kinematic model defining the kinematic chain of the robot, we define the Forward Kinematics ($\mathcal{FK}$) function as the map

$$\mathcal{FK} : q \in \mathbb{R}^{18} \mapsto {}^W X_B(q) \in SE(3) \tag{2.5}$$

We denote $\tau \in \mathbb{R}^{12}$ the vector of exoskeleton joint torques to be chosen by the controllers, $(\tau^{sw}, \tau^{sp}) \in \mathbb{R}^6 \times \mathbb{R}^6$ the vectors of exoskeleton swing and support joint torques, and $(\tau_u^{sw}, \tau_u^{sp}) \in \mathbb{R}^6 \times \mathbb{R}^6$ the vectors of swing and support joint torques created by the patient, *s.t.*

$$\tau = \begin{pmatrix} \tau^{sw} \\ \tau^{sp} \end{pmatrix} \quad \tau_u = \begin{pmatrix} \tau_u^{sw} \\ \tau_u^{sp} \end{pmatrix} \tag{2.6}$$

The Center of Mass (CoM) of the patient-exoskeleton system is denoted $c \in \mathbf{R}^3$, and the total angular momentum of the system *w.r.t.* the CoM, in the inertial frame $\mathcal{W}$, is denoted L *s.t.*

$$c = \frac{\sum_k m_k O_k}{\sum_k m_k} \qquad L = \sum_k (O_k - c) \times m_k \dot{O}_k + I_k \omega_k \tag{2.7}$$

with $O_k \in \mathbb{R}^3$ (resp. $\dot{O}_k \in \mathbb{R}^3$) the position (resp. velocity) of the origin of the frame F_k attached to the part $k \in [0 \dots 13]$ of the robot, *w.r.t.* the inertial frame $\mathcal{W}$, $\omega_k \in \mathbb{R}^3$ the rotation velocity of the frame F_k *w.r.t.* the inertial frame $\mathcal{W}$, $m_k \in \mathbb{R}^+$ the mass of the part k (including the patient mass), and $I_k \in M_{(3,3)}$ the inertia matrix of the part k of the system (including the patient inertia matrix), and $\times$ the cross product of $\mathbb{R}^3$.

We denote $.^{x,y}$ the 2D vector representing the position of 3D points projected onto the horizontal plane of reference. Analogously, we denote $.^z$ the height of 3D points, or the vertical component of 3D vectors.

Finally, we denote ξ the Divergent Component of Motion (DCM), defined as follows

$$\xi = c^{x,y} + \frac{\dot{c}^{x,y}}{\omega} \in \mathbb{R}^2 \tag{2.8}$$

with $\omega \triangleq \sqrt{\frac{g}{c^z}}$, and p the Center of Pressure (CoP), defined as

$$p \triangleq \frac{\sum p_i^{x,y} f_i^z}{\sum f_i^z} \in \mathbb{R}^2 \tag{2.9}$$

with $f_i \in \mathbb{R}^3$ the external force vectors acting on the patient-exoskeleton system, in the inertial frame $\mathcal{W}$, and $p_i \in \mathbb{R}^3$ the location of application of the external forces, in the inertial frame $\mathcal{W}$. Note that, by definition, the CoP belongs to the convex hull of all the contact points, also called the support polygon $\mathcal{SP}$. These quantities are of particular importance for the design of the stabilization control law detailed in Section 2.3.

2.1.4 Lagrangian dynamics

With the above notations, the Lagrangian dynamics write, following [Featherstone2008][4],

$$M(q)\ddot{q} + C(q,\dot{q}) = H(\tau + \tau_u) + \sum_i J_i(p_i,q)^T f_i \tag{2.10}$$

with $M(q)$ the generalized inertia matrix of the system, $C(q,\dot{q})$ the combined gravity and inertia effects vector, $H \triangleq \begin{bmatrix} I_{12} & O_6 \end{bmatrix}^\top$ the matrix mapping the joint torques to the actuated generalized coordinates, $J_i(p_i,q)$ the associated Jacobian matrices at each contact point p_i.

The Lagrangian dynamics (2.10) are identical to the one of walking robots, except for the vectors of patient joint torques τ_u. These dynamics

[4]The patient efforts are internal to the patient-exoskeleton system. Hence, their only effect appear in the joint position and can be expressed as additional joint torques.

are instrumental in the offline trajectory generation framework described in Section 2.2, where patient torques are set to zero to model a paraplegic patient[5].

2.1.5 Newton-Euler dynamics

The Newton and Euler equations [Wieber2006a], in the inertial frame $\mathcal{W}$, are, in accordance with (2.10),

$$m(\ddot{c} + g) = \sum_i f_i \ , \ \dot{L} = \sum_i (p_i - c) \times f_i \tag{2.11}$$

2.2 Offline trajectory generation

We now describe the Optimal Control Problem (OCP) formulation describing the walking of Atalante, the transcription of this OCP into a discrete optimization problem, and a numerical method to solve it. Then, two illustrative OCP examples are discussed: a simple *flat-foot* OCP and a more advanced *foot-rolling* OCP.

2.2.1 Generic walking motion optimal control problem

More details on the content of this section can be found in [Hereid2016; Gurriet2018].

OCP formulation

The problem of multi-contact trajectory generation is an OCP for hybrid dynamic. Each contact phase, defined by a different set of contact bodies (*i.e.* one or two feet resting flat or on one edge on the ground), defines distinct dynamics. These different contact phases, also called *domains*, are separated using several time instants t_i. The OCP allows to optimize the vector τ of joint torques, the timing of each step t_i, the contact forces f_i and application point p_i to minimize a general cost function subjected to constraints of various nature. The general OCP defining walking motion of humanoid robots is of the following form [Goswami2019, Section VII], which is also considered for Atalante,

[5] Paraplegic patients cannot voluntarily produce efforts with their legs. Patients suffering from spinal cord injuries may produce erratic involuntary efforts (this syndrome is called spasticity). The efforts are unpredictable by nature and are therefore unmodeled.

Problem 2.1 (Generic multi-step walking motion OCP)**.**

$$\min_{\tau,t_j,f_i,p_i} \sum_{j=1}^{n} \left(\int_{t_{j-1}}^{t_j} \Phi_j(q(t),\dot{q}(t),\tau(t))dt + \sum_{k=0}^{n_j} \Phi_{j,k}(t_j, q(t_{j,k}), \dot{q}(t_{j,k}), \tau(t_{j,k})) \right)$$

s.t. *for* $j = 1 \dots n,\ \forall t \in [t_{j-1}, t_j]$

$$\ddot{q}(t) = f(q(t), \tau(t), p_i, f_i(t)) \tag{2.12a}$$

$$g_j(t, q(t), \dot{q}(t), \tau(t), p_i, f_i(t)) \geq 0 \tag{2.12b}$$

for $j = 1 \dots n$,

$$q(t_j^-) = q(t_j^+) + k_q(t_j^-, q(t_j^-), \dot{q}(t_j^-)) \tag{2.12c}$$

$$\dot{q}(t_j^-) = \dot{q}(t_j^+) + k_{\dot{q}}(t_j^-, q(t_j^-), \dot{q}(t_j^-)) \tag{2.12d}$$

$$r_{cyc_j}(q(t_{j-1}), \dot{q}(t_{j-1}), q(t_j), \dot{q}(t_j)) = 0 \tag{2.12e}$$

for $j = 1 \dots n,\ $ *for* $k = 1 \dots n_j$,

$$r_{eq_{j,k}}(t_j, q(t_{j,k}), \dot{q}(t_{j,k}), \tau(t_{j,k}), p_i, f_i(t_{j,k})) = 0 \tag{2.12f}$$

$$r_{in_{j,k}}(t_j, q(t_{j,k}), \dot{q}(t_{j,k}), \tau(t_{j,k}), p_i, f_i(t_{j,k})) \geq 0 \tag{2.12g}$$

with

$$f(q, \tau, p_i, f_i) = M^{-1}(q) \left(-C(q, \dot{q}) + H\tau + \sum_i J_i(p_i, q)^T f_i \right) \tag{2.13}$$

where (2.13) is obtained from (2.10), expressed in the forward sense (*i.e.* with torques as inputs), where the patient torques τ_u are identically null.

The cost functions Φ_i are *running costs* and are used, for instance, to guide the OCP toward solutions with the best energy efficiency, desired walking velocity, and those with the highest margins (for instance in the sense of penalizing solutions requiring contact forces generating large moments along the vertical axis of the inertial frame $\mathcal{W}$). The cost functions $\Phi_{i,j}$ favor solutions passing through waypoints at specific times[6] and are used to produce trajectories with their feet rolling on edges between taking off, rising their feet high enough, or avoid known obstacles like stairs.

The first type of constraints included in Problem 2.1 are called *running constraints* and must be satisfied at all times. The running constraints (2.12a,2.12b) are used to express the physical constraints acting on the system, like the dynamics of the system, the friction cone constraints (which include the unilateral contact constraints), the joint position limits, or

[6]The duration of each step being free to choose by the optimizer, the waypoints are defined *w.r.t.* the total duration of each step, and not at absolute times.

the actuation torque limits. They can also be used to incorporate constraints like minimal foot clearance, bounded CoM height, or bounded torso orientation. The second type of constraints included in Problem 2.1 are *waypoints constraints* and hold only at specific times[6]. Among them, constraints (2.12c) and (2.12d) can be incorporated to reflect the impact dynamics, discrete by nature. However, no impact map $(k_q, k_{\dot{q}})$ has been implemented to generate trajectories for Atalante so far, and, simply,

$$\begin{cases} k_q \triangleq 0 \\ k_{\dot{q}} \triangleq 0 \end{cases} \tag{2.14}$$

A special constraint can be included to enforce the periodicity through (2.12e). It is used to generate trajectories that can be infinitely looped onto themselves by symmetrization. Finally, equality and inequality waypoint constraints (2.12f) and (2.12g) are used to shape trajectories [6] by placing feet or CoM at specific locations, constraining the total footprint of trajectories, or bounding the velocity of some bodies of the system.

OCP transcription

The OCP defined in continuous time by Problem 2.1 is transcribed into an optimization problem using a finite number of variables using the *Direct Collocation* method [Hargraves1987; vStryk1993]. Direct multiple shooting methods [Bock1984; Mombaur2001; Hereid2015] have first been used to solve non-linear OCPs of the type of Problem 2.1. However, they have proved hard to scale with the increasing number of variables (arising for robots with a high number of DoF and/or with numerous domains). Instead, Direct Collocation has been found to be robust and scalable. The results reported in [Huynh2021] constitute an example of successful optimization of trajectories for Atalante involving more than 10 independent domains.

The general idea behind Direct Collocation is to approximate the trajectory at a finite number of points x_i, called *collocation points*, replacing the explicit forward integration of the dynamical systems with a series of integration constraints. For the trajectory represented by these points x_i to be an approximation of the continuous solution of OCPs, a piece-wise polynomial interpolation is used to estimate the continuous solution between collocation points. Direct Collocation is generally used with defect constraints on defect variables δ. Defect variables δ are additional optimization variables that relax equality constraints at collocation points. Defect constraints are bounds on the defect variables δ (typically a small value). Introducing defect variables

and constraints, instead of closed-form constraints (*i.e.* directly expressed as a function of the trajectory's optimization variable), has several advantages, like being faster to compute, decoupling the optimization into many independent subproblems, and allowing the solver to violate the equality constraints during the first iterations of the optimization. We refer the interested reader to [Hargraves1987; vStryk1993; Duburcq2022a] for more details about Direct Collocation and its implementation.

OCP resolution

The resulting discrete optimization problem is a relatively large Non Linear Program (NLP). Typical setup includes up to 5000 variables in [Huynh2021]. However, it is very sparse [Hereid2016], which allows it to be solved efficiently by solvers such as IPOPT [Wächter2006].

The output of the trajectory generation framework is a trajectory $\mathcal{Q}$

$$\mathcal{Q} : t \in [0, T_f] \mapsto \mathcal{Q}(t) \in \mathbb{R}^{18} \tag{2.15}$$

2.2.2 Flat-foot trajectory optimal control problem

The general multi-step OCP defined by Problem 2.1 can be further specialized to generate trajectories of interest. A first example of such specialization gives the *flat-foot* trajectory. It is composed of 5 different contact phases:

- **STARTD:** a starting double-support weight transfer phase, during which both feet are constrained to rest flat and still on the ground, the weight of the system is gradually shifted toward the starting stance foot, and the initial joint position is constrained (corresponding to a standing posture).

- **STARTS:** a starting single-support phase, during which only the starting stance foot is constrained to rest flat and still on the ground, and the CoM and flying foot are both moved forward.

- **CS:** a cyclic single-support phase, during which only the left foot is constrained to rest flat and still on the ground, the CoM and right flying foot are both moved forward and the initial and final states are constrained to be symmetric. Gathering this phase and its symmetric constitute a cyclic trajectory, which can be repeated as much as requested at run-time.

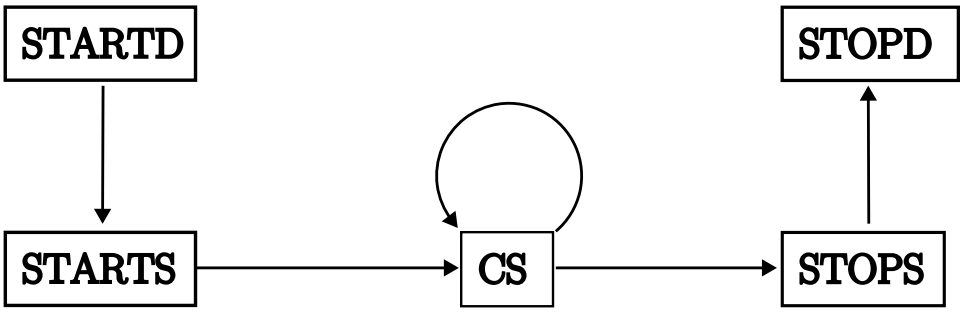


Figure 2.3: Transitions graph of flat-foot trajectory.

- **STOPS:** a stopping single-support phase, during which only the starting stance foot is constrained to rest flat and still on the ground, and the CoM and flying foot are both moved forward.
- **STOPD:** a stopping double-support weight transfer phase, during which both feet are constrained to rest flat and still on the ground, the weight of the system is gradually shifted from the stopping stance foot to the middle of the feet, and the final posture is constrained (corresponding to a standing posture).

In addition, continuity constraints are added so that the whole motion is continuous, as illustrated on Figure 2.3.

The particularities of this OCP are that the CoP is always constrained to lie within a box smaller than the actual support polygon $\mathcal{SP}$ (to add safety margins), the step duration is arbitrarily fixed to 1.0 s for single support phases and 0.5 s for double support ones, the step length is fixed (approximately to 15 cm), and the feet are constrained to take off and land horizontally to the ground. Thus, the average translational velocity is approximately 15 cm/s.

The CoM, CoP and DCM of the trajectory solution of the flat-foot OCP are reported on Figure 2.4, and the swing foot clearance during the (CS) phase is given on Figure 2.6 (Left).

2.2.3 Foot-rolling trajectory optimal control problem

A second example of specialization of Problem 2.1 gives the foot-rolling trajectory generation OCP, generating a more complex but more anthropomorphic walking trajectory. In detail, most of the OCP is similar to the one described above for the flat-foot trajectory. The major difference is

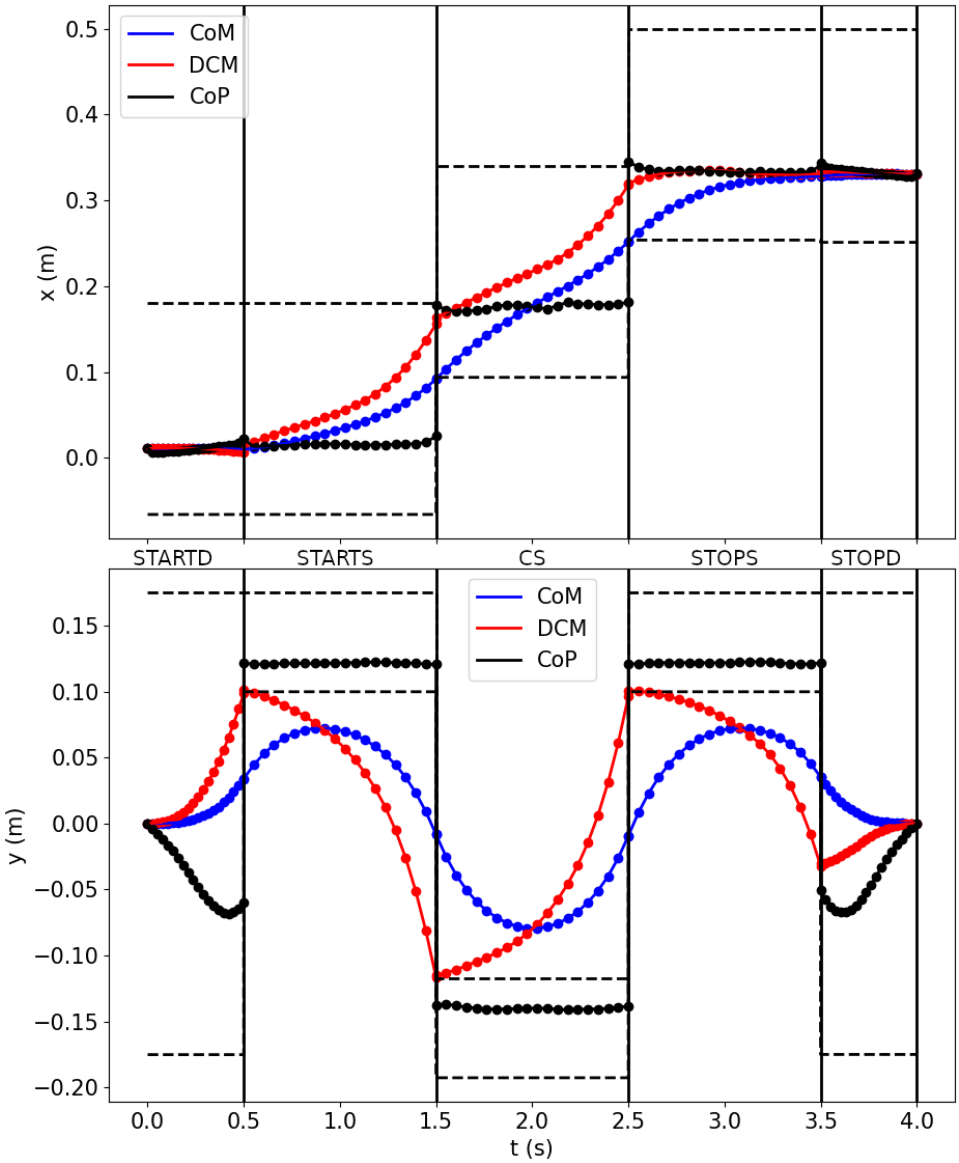


Figure 2.4: CoM, CoP and DCM of the trajectory generated by the flat-foot OCP (no double support phase between single support ones).

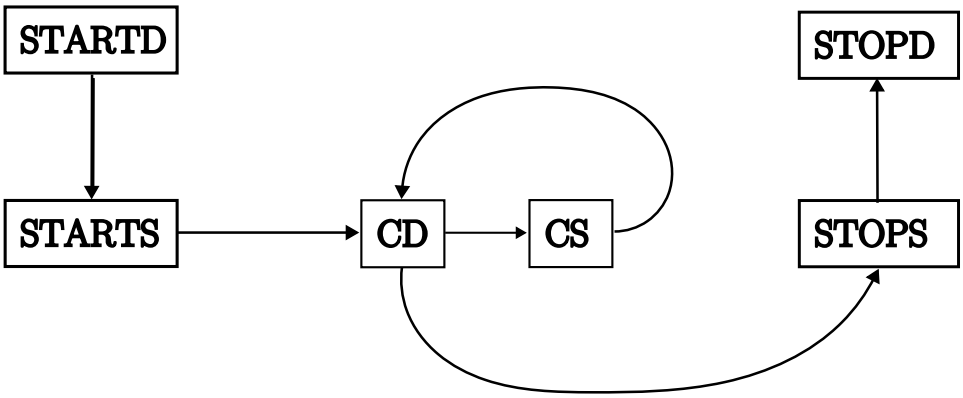


Figure 2.5: Transitions graph of foot-rolling trajectory.

the addition of a cyclic double support domain (CD), described below (and depicted on Figure 2.5):

- **CD:** a cyclic double-support weight transfer phase, during which the heel of the leading foot and the toe of the trailing foot are constrained to rest still on the ground, but their pitch angle is not constrained, *s.t.* the feet of the robot are rolling on their edges while the weight of the system is gradually shifted from the trailing foot to the leading foot, and the initial and final states are constrained to be symmetric.

For all single support domains defined in Section 2.2.2, the swing foot is no longer constrained to land flat on the ground at the end of the domain, but rather on its heel. The step timing is arbitrarily fixed to 0.81 s, for single support phases, to 0.23 s, for the (CD) phase, and to 0.4 s for the (STARTD) and (STOPD) phases. The step length is fixed approximately to 40 cm. The CoM, CoP and DCM of the trajectory solution of the foot-rolling OCP are displayed on Figure 2.7, and the swing foot clearance during the (CS) phase is illustrated on Figure 2.6 (Right). The comparison of the swing foot clearance given on Figure 2.6 shows the improvements offered by the foot-rolling OCP over the flat-foot OCP, namely: longer step-length, higher foot clearance and more anthropomorphic toe-off/heel-strike phases. The average translational velocity is approximately 40 cm/s.

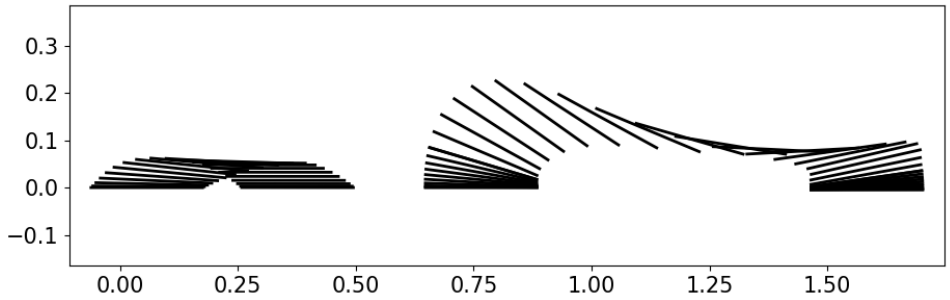


Figure 2.6: Swing foot clearance comparison between the trajectory generated by the flat-foot (Left) and the foot-rolling (Right) OCP. Lines representing the sole of the swing foot at various instants of the (CS) phase, for the flat-foot OCP, and the lumped (CD)-(CS)-(CD) phase, for the foot-rolling OCP in the XZ-plane. The foot-rolling steps are longer and more anthropomorphic.

2.3 Center of Mass stabilization and admittance controller

The two trajectories presented in the previous section satisfy the Lagrangian dynamics (2.10) and all the constraints acting on the system. However, due to the unstable nature of the dynamics (see the eigenvalues of (2.17) for instance), they can not be directly executed on the robot as pure open-loop joint torque trajectories, nor can they be naively followed by independent joint controllers, without leading to the divergence of the generalized coordinates q and, eventually, falling. For this reason, a stabilization algorithm, adapted from the humanoid robots' literature, ensures the correct execution of the trajectories presented above on Atalante[7].

In details, the stabilization algorithm tracks the horizontal CoM state trajectory. This stabilization algorithm is a feedback law designed using a linearized model of the patient-exoskeleton system, which performs a particular feedback on the CoM through the contact forces. The contact forces are in turn regulated by the so-called *admittance control* method.

[7]The stabilization algorithm also helps to mitigate all the uncertainties like the unknown patient model, the unmodeled effects like the flexibilities of the structure [Vigne2021], the actuator dynamics and the dynamics of their low-level current control laws.

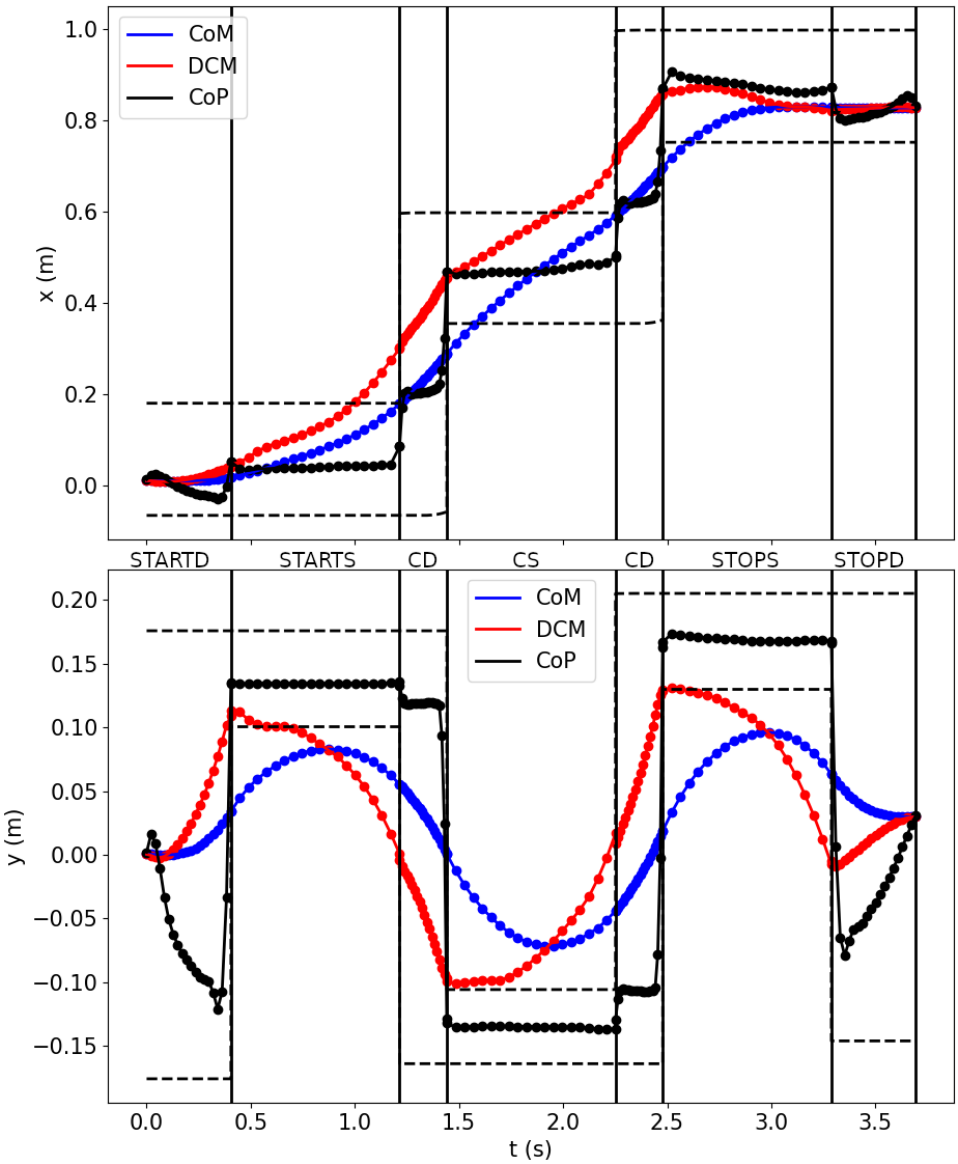


Figure 2.7: CoM, CoP and DCM of the trajectory generated by the foot-rolling OCP (with double support phase between single support ones).

2.3.1 DCM control using admittance

The linearized model of the system from which the feedback law stems is derived for the Newton-Euler dynamics (2.11). Following [Wieber2016], dynamics (2.11) can be simplified into the Linear Inverted Pendulum (LIP) model. First, notice that (2.11) yields

$$\frac{mc \times (\ddot{c} + g) + \dot{L}}{m(\ddot{c}^z + g^z)} = \frac{\sum p_i f_i^z}{\sum f_i^z} \tag{2.16}$$

Along gait patterns of moderate velocity, the angular momentum variations are small and can be neglected, *i.e.* $\dot{L} = 0$. Further, assuming that the robot walks on horizontal ground and that the CoM remains at a constant height c^z, the X and Y dynamics of (2.16) simplify to the following uncoupled LIP dynamics

$$\ddot{c}^{x,y} = \omega^2 (c^{x,y} - p) \tag{2.17}$$

where $\omega = \sqrt{\frac{g}{c^z}}$ is the natural frequency, and p is the CoP defined by (2.9). The CoP acts as bounded virtual[8] actuator in (2.17) (recall that, by definition, the CoP is bounded inside the support polygon $\mathcal{SP}$).

The dynamics (2.17) reproduce the unstable nature of the system. For stabilization, a stabilizing state feedback law on the CoM, such as the one detailed in [Caron2019], is used to drive $c^* - c$ to 0. First, one defines

$$p^d = p^* - (1 + \frac{k_p}{\omega})(\xi^* - \xi) - \frac{k_i}{\omega} \int (\xi^* - \xi) + k_d(\dot{\xi}^* - \dot{\xi}) \tag{2.18}$$

with ξ the DCM, readily computed from the horizontal LIP state ($c^{x,y}$,$\dot{c}^{x,y}$) (see (2.8)), ξ^* the DCM reference trajectory, computed from the horizontal state trajectory (c^*,$\dot{c}^*$), p^* the associated CoP trajectory, and $k_p, k_d, k_i \succ 0$ three diagonal matrices. Note that the actual CoP being bounded, regulating the DCM instead of the CoM gives the "Best CoM-ZMP[9] regulator" in a sense defined in [Sugihara2009]. Formula (2.18) encodes a feedback which, for any $k_p > 0$, $k_i = 0$, $k_d = 0$, drives any initial condition of the LIP in the controllable region to 0. [Caron2019] extends the formula to $k_i > 0$, $k_d > 0$.

Then, because there is no physical actuator directly controlling the CoP p, *admittance control* [Caron2019] is used to drive it to the desired value p^d (2.18). The admittance control methodology encompasses a general class

[8]The CoP is only indirectly commanded by the joint actuators through the (unknown) contact equation, hence we call it a *virtual* actuator.

[9]On planar ground, the Zero Moment Point (ZMP) and the CoP coincide at all times. Hence, we do not distinguish them in this thesis.

of control laws having a force regulation objective to be satisfied through kinematic actuators. This methodology considers one or several kinematic feedback laws, called *admittance tasks*, to regulate the force errors to zero. It is used, for instance, in [Yokoi2003; Englsberger2012; Li2012].

On Atalante, there is only one admittance task, a horizontal CoM acceleration task, which defines the desired horizontal CoM acceleration $\ddot{c}^d$ as follows

$$\ddot{c}^d = \ddot{c}^* + A(p - p^d) \tag{2.19}$$

with p the measured CoP, $\ddot{c}^*$ the horizontal state trajectory acceleration, and $A \succ 0$ a diagonal gain matrix to be tuned. This admittance task is included in a hierarchical Inverse Kinematics (IK) problem called a Stack-of-Tasks (SoT) [Mansard2009] framework. The hierarchy yields that this equation holds at almost all times. We describe this problem and SoT framework in the next section.

2.3.2 Hierarchical Inverse Kinematic problems: the Stack-of-Tasks framework

Inverse Kinematics (IK) problems are ubiquitous in the humanoid robots' literature. They naturally arise because objectives are best formulated in spaces (like the Cartesian inertial frame $\mathcal{W}$) which are different from the actuators' space (the joint space). Hence, the $\mathcal{FK}$ map (2.5) needs to be inverted to generate trajectories to be followed in the joint space.

Mathematically, if the $\mathcal{FK}$ map were invertible, the solution q^* would write

$$q^* = \mathcal{FK}^{-1}(X^d) \tag{2.20}$$

with X^d the desired transform of some rigid body of the robot. However, the $\mathcal{FK}$ map is not injective due to kinematic redundancy. A classical workaround is to define objectives as task errors and to ensure they are asymptotically driven to zero.

In details, following the task-function approach [Samson1991], a *task function* e_i is any twice differentiable function that maps the robot configuration space to $\mathbb{R}^n$. Its image space is called the *task space*. The function e_i may define various objectives. We detail here a few important task functions.

The task function e_q is expressed directly from the robot's generalized coordinates

$$e_q : q \in \mathbb{R}^{18} \mapsto q^d - q \in \mathbb{R}^{18} \tag{2.21}$$

with $q^d \in \mathbb{R}^{18}$ the desired configuration, which is time-varying. The task function $e_x \in \mathbb{R}^3$ in the literature is written *w.r.t.* the position x of a point,

like the CoM, in Cartesian inertial frame $\mathcal{W}$

$$e_x : q \in \mathbb{R}^{18} \mapsto x^d - x(q) \in \mathbb{R}^3 \tag{2.22}$$

with $x^d \in \mathbb{R}^3$ the desired location, which is time-varying. Finally, the last important task function e_X is expressed from the transform of one of the robot's rigid bodies, in the sense of (2.3), and using the matrix log map, which maps the Lie group $SE(3)$ to the Lie algebra $se(3) = \mathbb{R}^6$,

$$e_X : q \in \mathbb{R}^{18} \mapsto \log(X^d \mathcal{FK}(q)) \in \mathbb{R}^6 \tag{2.23}$$

with $X^d \in SE(3)$ the desired transform, which is time-varying.

For every task functions e_i, we define its Jacobian J_i as follows

$$\dot{e}_i = J_i(p_i, q)\dot{q} \tag{2.24}$$

which yields

$$\ddot{e}_i = \dot{J}_i(q)\dot{q} + J_i(p_i, q)\ddot{q} \tag{2.25}$$

For each task function e_i, the Jacobian J_i and its time derivative $\dot{J}_i$, defining the objective w_i, are computed using the efficient library *Pinocchio* [Carpentier2019]. From the task acceleration (2.25), we define the objective w_i for all task functions e_i as follows

$$w_i(\ddot{q}) = ||\ddot{e}_i - \ddot{e}_i^*||^2 = ||J_i(p_i, q)\ddot{q} - (\ddot{e}_i^* - \dot{J}_i(q)\dot{q})||^2 \tag{2.26}$$

with a target task acceleration $\ddot{e}_i^*$, illustrated on a Cartesian task, driving the task error e_i exponentially to zero, such as

$$\ddot{e}_i^* = \ddot{x}^d - k_{p_i} e_i - k_{di}\dot{e}_i \tag{2.27}$$

with $k_p, k_d \succ 0$ and $\ddot{x}^d$ denoting the feedforward in the Cartesian space.

The desired motion of the robot is usually defined by several objectives w_i, *s.t.* the total objective w writes

$$w(\ddot{q}) = \sum_i \lambda_i w_i(\ddot{q}) \tag{2.28}$$

with $\lambda_i \in \mathbb{R}^+$ weights expressing the relative importance of the objectives w_i. The IK problem is to find optimal joint accelerations $\ddot{q}$ which minimize w. In addition to these objectives, the physical limitations of the robot, such as the joint position, velocity and acceleration bounds, must be embedded into the IK problem. Finally, it writes

$$\begin{aligned} \ddot{q}^* = & \arg\min \sum_i \lambda_i w_i(\ddot{q}) \\ & s.t. \\ & A\ddot{q} \leq B \end{aligned} \tag{2.29}$$

with A (resp. B) a matrix (resp. vector) to represent the aforementioned bounds. While the IK problem (2.29) has been successfully used to realize stable walking of humanoid robots [Caron2019], we favor a different approach. Indeed, the tuning of the weights λ_i can be cumbersome, and some objectives often have a strict priority over others (for instance, the robot's feet should always be on the ground, while the posture tasks are used only for regularization). An alternate formulation has been proposed, the Stack-of-Tasks (SoT) framework [Mansard2009], enabling one to solve IK problems composed of objectives listed in strict hierarchical order.

A Stack-of-Tasks (SoT) with $n \geq 0$ levels is an ordered list of objectives w_i^j, grouped in several levels j, with w^j their cost defined as follows

$$w^j(\ddot{q}) = \sum_i \lambda_i^j w_i^j(\ddot{q}) \tag{2.30}$$

The levels are sorted in strict decreasing priority order, *s.t.* the cost w^j can only be decreased if the value w_k, of all the levels $k < j \leq n$, remains constant (*i.e.* equal to the optimal cost w^{k*}). Mathematically, a Hierarchical IK problem with constraints writes

$$\begin{aligned} \forall j = 0 \dots n, \quad \ddot{q}^* = & \arg\min \sum_i w_i^j(\ddot{q}) \\ & s.t. \\ & A\ddot{q} \leq B \\ & \forall k < j, \; w^k(\ddot{q}) = w^{k*} \triangleq \min_{\ddot{q}} \sum_i w_i^k(\ddot{q}) \end{aligned} \tag{2.31}$$

On Atalante, the SoTs are solved in real-time using state-of-the-art hierarchical optimization routines, such as the one proposed in [Escande2014]. The key idea is to solve the Hierarchical IK problems (2.31) iteratively, starting with the highest priority objectives, and then minimizing the objectives of lower priority in the level set of the higher priority objectives already minimized. We briefly illustrate below this iterative process.

For ease of notations, let us define a simplified Hierarchical IK problem, with only two levels ($n = 1$), one task at each level, and no constraint. For example, one has

$$\begin{aligned} \ddot{q}^* = & \arg\min \; w_0^1(\ddot{q}) \\ & s.t. \\ & w^0(\ddot{q}) = w^{0*} \end{aligned} \tag{2.32}$$

with

$$\begin{cases} w^0(\ddot{q}) = w_0^0(\ddot{q}) = ||J_0^0(q)\ddot{q} - b^0||^2 \\ b^0 = \ddot{e}_0^{0^*} - \dot{J}_0^0(q)\dot{q} \end{cases} \tag{2.33}$$

and

$$\begin{cases} w^1(\ddot{q}) = w_0^1(\ddot{q}) = ||J_0^1(q)\ddot{q} - b^1||^2 \\ b^1 = \ddot{e}_0 1^* - \dot{J}_0^1(q)\dot{q} \end{cases} \tag{2.34}$$

As there is no constraint, in this simple case, all the solutions $\ddot{q}^{0^*}$ minimizing $w^0(\ddot{q})$ write

$$\ddot{q}^{0^*} = J_0^{0\dagger}(q)b^0 + P^0\ddot{q}^1 \tag{2.35}$$

with $.^\dagger$ the matrix pseudo-inverse, $P^0 \triangleq (I - J_0^{0\dagger}(q)J_0^0(q))$ a projector on the null-space $\ker(J_0^0(\ddot{q}))$ of $J_0^0(\ddot{q})$, and $\ddot{q}^1$ an arbitrary vector. If the kernel $\ker(J_0^0(\ddot{q}))$ is not null, then one can attempt to minimize the cost of the level 1 problem, of lower priority.

The level 1 problem (2.34) is solved in level 0's kernel by selecting $\ddot{q}^1$ only. A direct resolution of this simple case yields

$$\ddot{q}^{1^*} = \left(J_0^1(q)P^0\right)^\dagger \left(b^1 - J_0^1(q)J_0^{0\dagger}(q)b^0\right) \tag{2.36}$$

and the solution of problem (2.34) write

$$\ddot{q}^* = J_0^{0\dagger}(q)b^0 + P^0\ddot{q}^{1^*} \tag{2.37}$$

Equation (2.37) and (2.36) show how one can iteratively solve a simple Hierarchical IK problem such as (2.32). For more complex problems, we refer the interested reader to [Escande2014] to get efficient solvers addressing them.

2.3.3 The nominal SoT_1 of Atalante

The goal of Atalante's nominal SoT, denoted SoT_1, is to stabilize the system around the nominal trajectory, computed offline as described in Section 2.2. To this end, most objectives included in SoT_1 are the tracking of quantities defined by the nominal trajectory, except for the task at level 1, which is defined by the admittance horizontal acceleration (2.19).

Namely, the level 0 of SoT_1, hence the highest priority, includes two tasks on the robot's feet,

$$w_i^0(\ddot{q}) = ||J_i^0(q)\ddot{q} - (\ddot{e}_i^{*^0} - \dot{J}_i^0(q)\dot{q})||^2 \tag{2.38}$$

defined by the transform task function (see Eq. (2.23))

$$e_i^0 = \log({X^0}^{nom}{}_i X^0{}_i(q)) \tag{2.39}$$

where $i = 0$ (resp. $i = 1$) denotes the left foot (resp. the right foot), ${X^0}_i$ is the left (resp. right) foot transform *w.r.t.* the inertial frame $\mathcal{W}$, $J_i^0(q)$ is the left (resp. right) foot transform Jacobian, and $K_p^0, K_d^0 \succ 0$ are two matrices[10]. With these notations, level 0 writes

$$w^0(\ddot{q}) = w_0^0(\ddot{q}) + w_1^0(\ddot{q}) \tag{2.40}$$

The level 1 of SoT_1 has only one task, defined by (2.19)

$$w^1(\ddot{q}) = ||J^1_{c^{x,y}}(q)\ddot{q} - (\ddot{c}^d - \dot{J}^1_{c^{x,y}}(q)\dot{q})||^2 \tag{2.41}$$

where $J^1_{c^{x,y}}(q)$ is the horizontal CoM transform Jacobian. Recall, that

$$\ddot{c}^d = \ddot{c}^* + A(p - p^d) \tag{2.42}$$

with $\ddot{c}^*$ is horizontal CoM acceleration of the nominal trajectory.

The level 2 of SoT_1 has only one task, concerning the robot's back orientation,

$$w^2(\ddot{q}) = ||J^2_{back}(q)\ddot{q} - (\ddot{e}^*_{back} - \dot{J}^2_{back}(q)\dot{q})||^2 \tag{2.43}$$

defined by the rotation task function

$$e_{back} = \log_3({R^2}^{nom}{}_{back} {R^2}_{back}(q)) \tag{2.44}$$

where ${R^2}_{back}$ is the orientation of the robot's back *w.r.t.* the inertial frame $\mathcal{W}$, $J^2_{back}(q)$ is the Jacobian of the orientation of the robot's back, and $K_p^2, K_d^2 \succ 0$ are two high gain matrices.

Finally, the level 3 of SoT_1 is composed of one task, expressed on the robot's generalized position q, defined as follows

$$w^3(\ddot{q}) = ||\ddot{q} - (\ddot{q}^{nom} + K_p^2(q^{nom} - q) + K_d^2(\dot{q}^{nom} - \dot{q}))||^2 \tag{2.45}$$

where $(q^{nom}, \dot{q}^{nom}, \ddot{q}^{nom})$ are the generalized position, velocity and acceleration of the nominal trajectory.

From all the objectives of SoT_1 described above, only the task at level 1 (see (2.42)) contains a term not only expressed using the nominal trajectory. This suggests that the stabilization will be obtained by updating the horizontal position of the CoM, as can be visible in experiments on the hip lateral motion. SoT_1 can be summarized as follows, with levels listed in decreasing order of priority

[10] Chosen as high gain.

Stack of Tasks SoT_1

Level 0: tracking of swing and support feet pose by (2.40);

Level 1: tracking of CoM acceleration by (2.42), realizing (2.19);

Level 2: tracking of back orientation by (2.43);

Level 3: tracking of joint by (2.45).

On Atalante, the Hierarchical IK problem SoT_1 is solved online by a solver inspired from [Escande2014]. Its output is the optimal acceleration $\ddot{q}_t$. This signal is integrated twice to get joint position and velocity targets (q_t, $\dot{q}_t$). Finally, these joint targets are tracked using a high-gain PD controller

$$\begin{bmatrix} \tau^{sw} \\ \tau^{sp} \end{bmatrix} = K_p(q_t - \begin{bmatrix} q^{sw} \\ q^{sp} \end{bmatrix}) + K_d(\dot{q}_t - \begin{bmatrix} \dot{q}^{sw} \\ \dot{q}^{sp} \end{bmatrix}) \tag{2.46}$$

From the definition of SoT_1 and (2.46), the nominal controller is now fully described. It is schematically depicted on Figure 2.8.

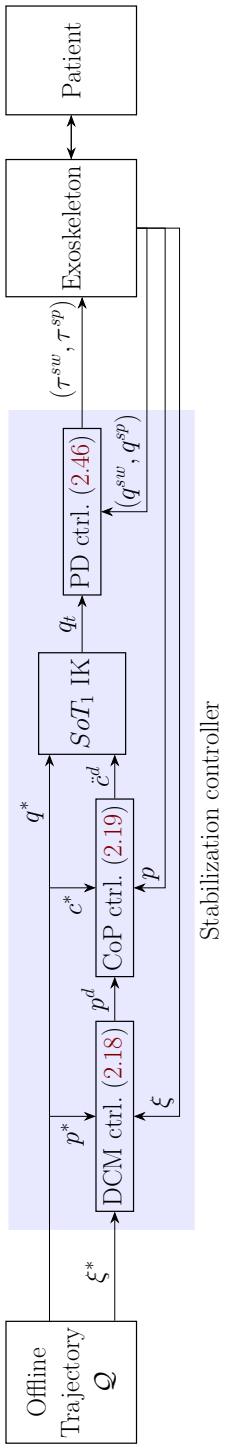


Figure 2.8: Atalante's passive walking controller: offline trajectories stabilized with admittance control.

Chapter 3

Empowering the patient for gait training

Chapitre 3 - Donner de la liberté au patient: *Dans ce chapitre, nous proposons un nouveau contrôleur pour l'exosquelette Atalante qui donne au patient la possibilité de participer au mouvement. Plus précisément, nous laissons le patient modifier la vitesse de la jambe de vol en temps réel tout en stabilisant, grâce à la jambe de support, le centre de masse du système autour de la trajectoire de référence synchronisée sur la jambe de vol. Nous utilisons le principe des guides virtuels sur la jambe de vol pour libérer le mouvement du patient le long de la trajectoire de référence. Nous modifions la technique de stabilisation utilisée par le contrôleur passif d'Atalante pour n'utiliser que la jambe de support. Une simple dilatation temporelle de la trajectoire nominale est utilisée pour synchroniser la trajectoire de référence du centre de masse avec la jambe de vol. Les performances de stabilisation de ce contrôleur sont évaluées en simulation et à l'aide d'expériences avec un mannequin.*

Contents

In this chapter, we propose a new controller for Atalante which enables the user to participate in the motion. More precisely, we adapt the existing controllers to

1. let the patient modify the swing leg's schedule
2. stabilize the CoM dynamics around its reference trajectory using the support leg.

We modify the existing controllers only during single support phases, and keep them unchanged during the double support phases[1].

We propose to use the Virtual Guides (VG) framework [Joly1995; Sanchez Restrepo2018], originally developed for teleoperation and co-manipulation, on the swing leg Degrees of Freedom (DoF). This requires some adaptations and yields a controller akin to a path control feedback [Banala2007; Duschau-Wicke2010] that aims at stabilizing a parametric curve in the articular space. Because the velocity along the curve is left free to choose by this controller, the patient is free to express an intent via the displacement of the system along this path. This displacement defines the so-called *patient*

[1]We explore the possibility to modify the controllers during the double support phases in Appendix B.

schedule. On the other hand, we propose to use an adaptation of whole-body admittance, originally developed for the stabilization of humanoid robots during walking, for the support leg only. The purpose of the admittance task is to stabilize the CoM dynamics around a reference trajectory. To make this reference consistent with the swing leg motion, a natural idea is to apply a Time Rescaling (TR) of the nominal CoM trajectory following the patient schedule. The stabilizing properties of admittance around the nominal trajectory are not diminished when using it on the support leg only, as shown in Section 3.2.4. More precisely, its stabilization properties are preserved within a neighborhood of the nominal trajectory, when rescaled by the patient schedule, as shown in Section 3.3.

The chapter is organized as follows. First, we describe the VG control law of the swing leg. Second, we describe our adaptation of the whole-body admittance control law, for its use on the support leg only, along with a natural synchronization strategy with the swing leg. Then, we prove that, provided the support leg has sufficient control authority (a sufficient number of DoF), and the motion of the swing leg is known, the use of the sole support leg is sufficient for the admittance task to succeed. Finally, we study, in simulation and experimentally, the stabilization property offered by our controller when the patient schedule differs from the nominal schedule.

3.1 Satisfying the patient intent with the swing leg

The VG methodology [Joly1995] allows a parametric curve P to be followed at a velocity prescribed by efforts not produced by the robot. This control methodology is classically used during teleoperation and co-manipulation in order to constrain a slave robot onto a desired manifold (*e.g.* in Cartesian space) and/or, more generally, help an operator guide the robot during a specific task.

The VG framework has two ingredients

1. a control law providing a contraction property in the direction orthogonal to the path. Usually, this control law is implemented as torques at the joint level.

2. a control law accounting for the patient efforts. This controller is implemented as a real-time re-scheduling of the nominal trajectory.

In a way, the VG methodology is a technological solution to realize a path controller, which can be compared to several alternative formulations [Banala2007; Aoyagi2007; Vallery2009; Duschau-Wicke2010; Martínez2018;

Martínez2019]. All these path controllers are stabilizing around the closest point in the trajectory from the current state of the robot. We choose the VG methodology as it possesses the advantage of explicitly minimizing the robot forces acting on the patient, as will be detailed next. Also, its computation load is light, which is handy for embedded implementation.

Virtual Guides (VG) allows the patient's efforts onto the exoskeleton to drive the velocity of his swing leg in real-time within a predefined path (computed from the nominal swing leg trajectory). This approach enables us to satisfy the user intent without actually measuring the patient's efforts, which is convenient since Atalante does not possess torque or force sensors at the interfaces with the patient nor at the joint level.

3.1.1 Choice of the Virtual Guides definition space

Virtual Guides (VG) are defined in a certain task-space[2], in which both internal efforts (robot efforts) and external efforts (in our case, patient efforts), are defined. In many occurrences, *e.g.* manipulators interacting with their environment at their end-effector, VG are defined in Cartesian task-space. This is not the case here, because exoskeletons interact with their users at every link. Indeed, the patient does not exert efforts only at the end-effector (the swing foot). This appears in the Lagrangian dynamics (2.10) of the patient-exoskeleton system, where the patient efforts appear at every joint. Besides, defining the control law of the exoskeleton in Cartesian space would not enforce inter-joint coordination (but joint space does) due to the non-injectivity of the joint to Cartesian position mapping. For these two reasons, we consider the VG approach in joint-space, similar to most of the path-control laws for exoskeletons [Aoyagi2007; Vallery2009; Duschau-Wicke2010; Martínez2018; Martínez2019].

3.1.2 Construction of a parametric curve of reference

As already mentioned, Virtual Guides (VG) aim at constraining the motion of the robot along a parametric curve or manifold. The parametric curve used by our controller is built from a nominal joint trajectory $\mathcal{T}$

$$\mathcal{T} : t \in [0, T_f] \mapsto \mathcal{T}(t) \in \mathbb{R}^{12} \tag{3.1}$$

[2]Here, any normed space enabling one to define error metrics (*e.g.* the Cartesian space $\mathbb{R}^3$, the Special Orthogonal group of rotation matrices $SO(3)$, or the Special Euclidean group $SE(3)$).

readily obtained from a trajectory $\mathcal{Q}$ computed by the offline trajectory generation pipeline presented in Section 2.2. The swing leg trajectory $\mathcal{T}^{sw}$ is readily extracted from the complete trajectory $\mathcal{T}$

$$\mathcal{T}^{sw} : t \in [0, T_f] \mapsto M_i \mathcal{T}^{sw}(t) \in \mathbb{R}^6 \tag{3.2}$$

with $M_i \in \mathbb{R}^{6\times 12}, i \in \{l, r\}$ a masking rectangle matrix depending on the stance side (left or right). To obtain a parametric curve from the swing leg trajectory $\mathcal{T}^{sw}$, we reparametrize it *w.r.t.* its curvilinear abscissa s

$$\mathrm{s} : \tau \in [0, T_f] \mapsto \int_0^{\tau} ||\dot{\mathcal{T}}^{sw}(t)||_2 \, dt \in [0, L_{max}] \tag{3.3}$$

with L_{max} the total length (*in joint space*) of the swing leg trajectory. By this formula, the curvilinear abscissa s is monotonous, therefore, assuming further that the Euclidean norm $||\dot{\mathcal{T}}^{sw}(t)||_2$ is non-zero for all t, s can be inverted. Then, we define the parametric curve as

$$P \triangleq s \in [0, L_{max}] \mapsto \mathcal{T}^{sw} \circ \mathrm{s}^{-1}(s) \in \mathbb{R}^6 \tag{3.4}$$

To process this equation, one can numerically integrate the differential equation

$$\dot{s}(t) = ||\dot{\mathcal{T}}^{sw}(t)||_2 \tag{3.5}$$

Then, we gather the two grid vectors $\boldsymbol{t} = \{\frac{i}{N} T_f, \ \forall i \in [0, N]\}$ and $\boldsymbol{s} = \{s(\boldsymbol{t}_i), \ \forall i \in [0, N]\}$, and we fit 3^{rd} order Bézier curves (Splines) allowing us to evaluate $\mathrm{s}(t)$ for arbitrary values of t. The same fitting method is used to evaluate $\mathrm{s}^{-1}(\sigma)$ for arbitrary values of σ. An example of the result representing the nominal trajectory is depicted on Figure 3.1, which stresses that the mapping $t \mapsto \mathrm{s}$ is monotonous, invertible but non-linear.

3.1.3 Virtual Guides controller

In the following

$$\sigma : t \mapsto \sigma(t) \in [0, L_{max}] \tag{3.6}$$

is now used as a control variable. It defines the current set-point $P(\sigma(t))$ along the swing leg path. Schematically, the VG methodology constrains the position x of the robot onto the parametric path P, as pictured on Figure 3.2. At the current point $P(\sigma(t))$, the Frenet-Serret unit tangent vector to the curve P, pointing in the direction of motion is

$$T(\sigma) \triangleq \frac{dP}{d\sigma}(\sigma) \tag{3.7}$$

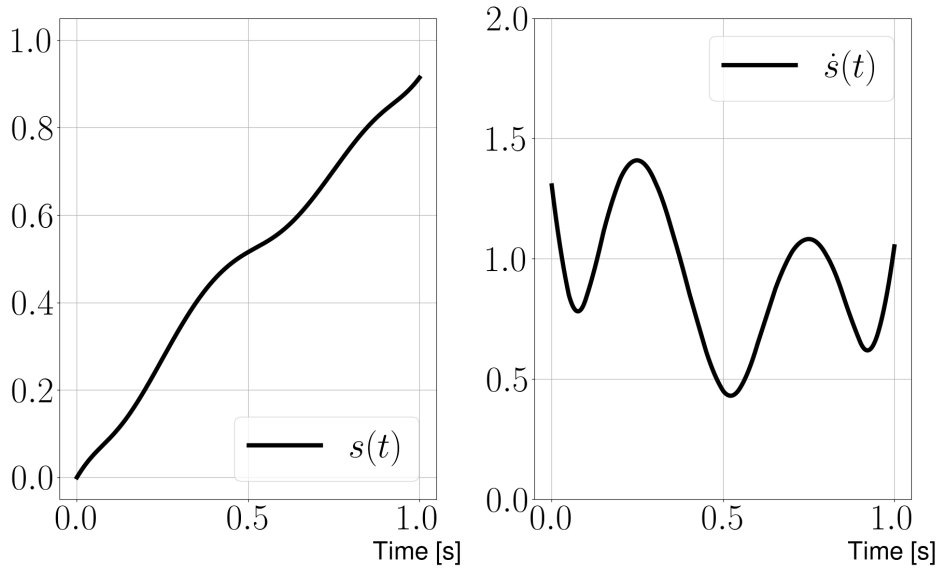


Figure 3.1: Curvilinear abscissa s (Left) and curvilinear velocity $\dot{s}$ (Right) over one step, computed from the swing leg trajectory $\mathcal{T}^{sw}$ obtained from the flat-foot OCP (notice that s is strictly monotonous).

The exoskeleton joint torques of the swing leg τ^{sw} are computed as the sum of two torque vectors. First, a (vector-valued) torque τ^{sw}_{PD} aiming at providing a contraction property in the direction orthogonal to T, computed as a high-gain PD controller

$$\tau^{sw}_{PD}(\sigma, \dot{\sigma}, q^{sw}, \dot{q}^{sw}) = K^{sw}_p(P(\sigma) - q^{sw}) + K^{sw}_d(T(\sigma)\dot{\sigma} - \dot{q}^{sw}) \tag{3.8}$$

with $K^{sw}_p, K^{sw}_d \succ 0$ constant diagonal gain matrices. Second, a gravity compensation torque τ^{sw}_{GC}, aiming at compensating the gravity effects on the robot, computed by inverting the dynamics (2.10) of the exoskeleton (using a holonomic fixed support-foot constraint [Featherstone2008]), *s.t.*

$$\tau^{sw}_{GC}(q) = H^{sw}\left(H_M\left[I - J_M\right]H^{\top}\right)^{\dagger} H_M\left[I - J_M\right]C_R(q, 0) \tag{3.9}$$

$$J_M = J^{\top}(q)(J(q)M_R^{-1}(q)J^{\top}(q))^{-1}J(q)M_R^{-1}(q) \tag{3.10}$$

$$H_M = HM_R^{-1}(q) \tag{3.11}$$

with M_R the generalized inertia matrix of the robot only and $C_R(q, 0)$ the gravity effects vector of the robot only. Finally, the swing leg τ^{sw} exoskeleton

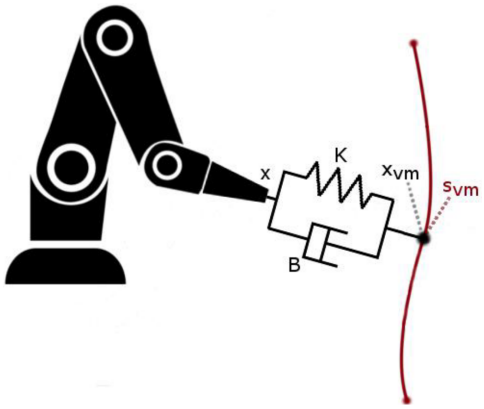


Figure 3.2: Schematic description of the VG approach.

torques are computed as

$$\tau^{sw}(\sigma, \dot{\sigma}) = \tau^{sw}_{PD}(\sigma, \dot{\sigma}, q^{sw}, \dot{q}^{sw}) + \tau^{sw}_{GC}(q) \tag{3.12}$$

This PD plus gravity compensation controller is very similar to the path-control controllers in [Banala2007; Duschau-Wicke2010], the main difference with these lies in the choice of the control variable.

Following our VG approach, we define the control variable σ such that the projection of the efforts τ^{sw}_{PD} along the path P is nullified, which reads

$$T(\sigma)^{\top} \tau^{sw}_{PD}(\sigma, \dot{\sigma}, q^{sw}, \dot{q}^{sw}) = 0 \tag{3.13}$$

Notice that we only nullify the efforts of the PD controller, as we do not aim at canceling the gravity compensation efforts in any direction. This yields, by injecting (3.8) and solving for $\dot{\sigma}$ [Sanchez Restrepo2018],

$$\dot{\sigma}(\sigma, q^{sw}, \dot{q}^{sw}) \triangleq \frac{T(\sigma)^{\top} \left[K^{sw}_{p}(q^{sw} - P(\sigma)) + K^{sw}_{d} \dot{q}^{sw} \right]}{T(\sigma)^{T} K^{sw}_{d} T(\sigma)} \tag{3.14}$$

In summary, our VG controller boils down to implementing the joint torques (3.12) and integrating (3.14) to get σ from $\dot{\sigma}$.

3.1.4 Behavior of Virtual Guides

In this section, we illustrate the difference between the above-defined VG controller and the path-control nearest-neighbor approach. This approach, detailed in [Duschau-Wicke2010], consists in solving the following minimization

problem

$$\sigma_{nn} = \arg\min_{\sigma} ||P(\sigma) - q_m^{sw}||^2 \tag{3.15}$$

where q_m^{sw} is the vector of measured positions of the swing leg's joints. We call "nearest neighbor" the position $q_{nn}^{sw} = P(\sigma_{nn})$ corresponding to σ_{nn} defined by (3.15).

Almost a closest point algorithm: toy examples

In this section, we illustrate the asymptotic behavior of the VG dynamics (3.14). To this end, we study the behavior of σ defined by dynamics (3.14) with arbitrarily fixed values of q^{sw} and $\dot{q}^{sw}$. We denote these fictitious joint positions and velocities as follows

$$\forall t \ , \ q^{sw}(t) = q_{fi}^{sw} \ , \ \dot{q}^{sw}(t) = \dot{q}_{fi}^{sw} \tag{3.16}$$

In addition, we choose $K_p^{sw}, K_d^{sw} \succ 0$ as constant gain matrices (with K_d the critically damped matrix)

$$K_p = \alpha I_6 \ , \ K_d = 2\sqrt{K_p} \tag{3.17}$$

for some $\alpha \in \mathbb{R}^+$. We study the convergence of (3.14) in the case where the measured position lies perfectly on the trajectory, and the measured velocity is null, *i.e.*

$$q_{fi}^{sw} = \mathcal{T}^{sw}(t_{fi}) \ , \ \dot{q}_{fi}^{sw} = 0_6 \tag{3.18}$$

with $t_{fi} \in [0, T_f]$, the fictitious time. The solutions of (3.14), integrated over 200 ms from several initial conditions σ_0 (*s.t.* $t_0 \triangleq \mathrm{s}^{-1}(\sigma_0)$), and for several t_{fi}, are reported on Figure 3.3. The final value of the solution σ is denoted σ_f (*i.e.* $t_f \triangleq \mathrm{s}^{-1}(\sigma_f)$). These solutions illustrate that the convergence of σ to $\sigma_{fi} \triangleq \mathrm{s}(t_{fi})$ is not guaranteed, which implies the existence of several equilibrium points to (3.14). By definition, the equilibrium points of the dynamical system defined by (3.14) are *s.t.*

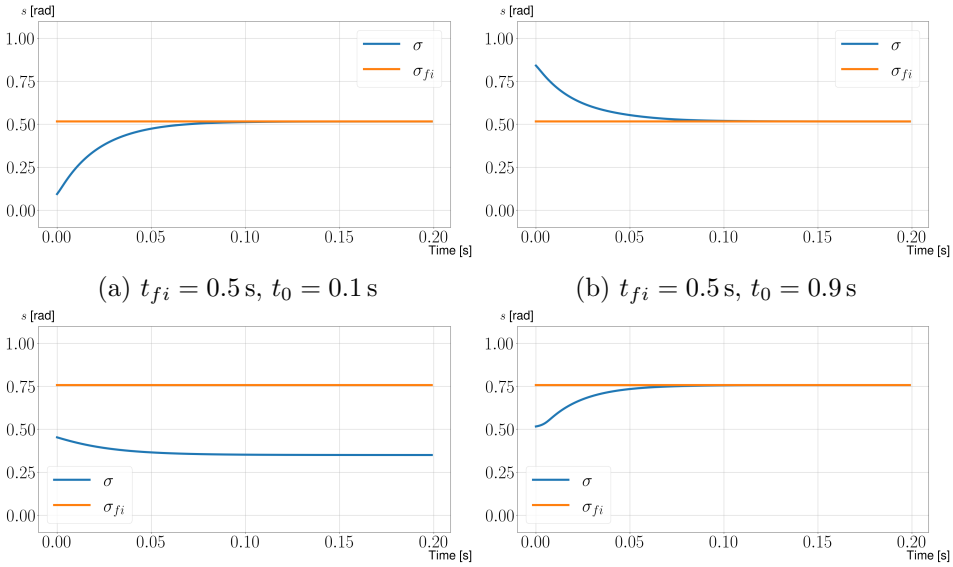
$$T(\sigma)^\top \left[K_p^{sw}(q^{sw} - P(\sigma)) + K_d^{sw}\dot{q}^{sw} \right] = 0 \tag{3.19}$$

Interestingly, by using the weighted norm

$$f \triangleq ||K_p^{sw}(q^{sw} - P(\sigma)) + K_d^{sw}\dot{q}^{sw}||_N^2 \tag{3.20}$$

with

$$N = K_p^{-\frac{1}{2}} \tag{3.21}$$



(a) $t_{fi} = 0.5\,\text{s}$, $t_0 = 0.1\,\text{s}$

(b) $t_{fi} = 0.5\,\text{s}$, $t_0 = 0.9\,\text{s}$

(c) $t_{fi} = 0.8\,\text{s}$, $t_0 = 0.4\,\text{s}$

(d) $t_{fi} = 0.8\,\text{s}$, $t_0 = 0.5\,\text{s}$

Figure 3.3: Curvilinear abscissa convergence may vary with initialization.

we note that

$$\frac{df}{d\sigma} = -2T(\sigma)^\top \left[K_p^{sw}(q^{sw} - P(\sigma)) + K_d^{sw}\dot{q}^{sw}\right] \tag{3.22}$$

Note that f is proportional to the numerator in (3.14). Hence,

$$\dot{\sigma} = 0 \iff \frac{df}{d\sigma} = 0 \tag{3.23}$$

Therefore, the local minima of f are the equilibrium points of the dynamics (3.14).

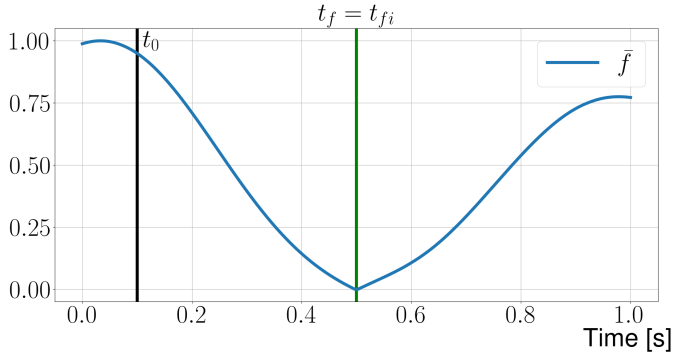
For an arbitrary function h defined over $[0, L_{max}]$, we denote $\bar{h}$ its normalized version,

$$\bar{h} \triangleq \frac{h - \min_{[0,L_{max}]} h}{\max_{[0,L_{max}]} h - \min_{[0,L_{max}]} h} \tag{3.24}$$

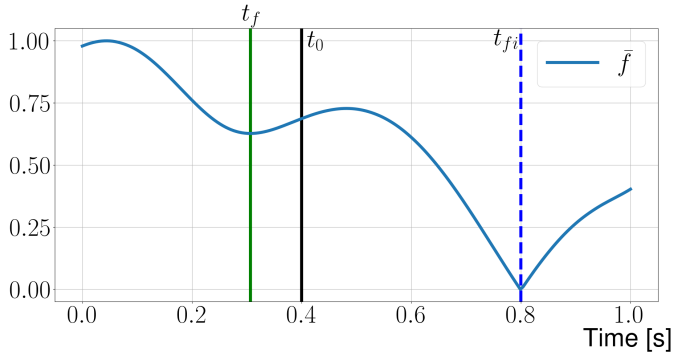
We denote $g(\sigma) \triangleq ||P(\sigma) - q_{fi}||_2^2$ the Euclidean distance from q_{fi}.

On Figure 3.4, which depicts $\bar{f}$, we verify that the local minima of f are indeed the equilibrium points of (3.14). In this example, with $\dot{q}_{fi}^{sw} = 0$ and K_p a constant diagonal matrix,

$$f(\sigma) = ||K_p^{sw}(q^{sw} - P(\sigma))||_N^2 = \alpha^2 g \tag{3.25}$$



(a) $t_{fi} = 0.5\,\mathrm{s}$, $t_0 = 0.1\,\mathrm{s}$



(b) $t_{fi} = 0.8\,\mathrm{s}$, $t_0 = 0.4\,\mathrm{s}$

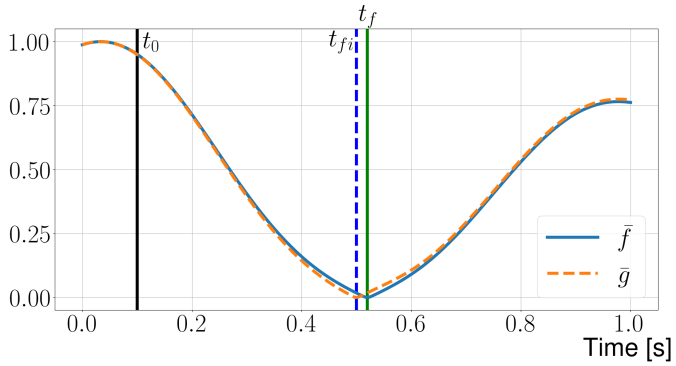
Figure 3.4: Normalized f (defined by (3.20)) with $\dot{q}^{sw} = 0$: existence of local minima depends on the fictitious time t_{fi}.

Hence, $\bar{g}$ is identical to $\bar{f}$. Therefore, Figure 3.4 also reveals that not all the equilibrium points of (3.14) minimize the Euclidean distance, as the path-control methodology does. The dynamics (3.14) converge toward a local minimum, closer to the initial state.

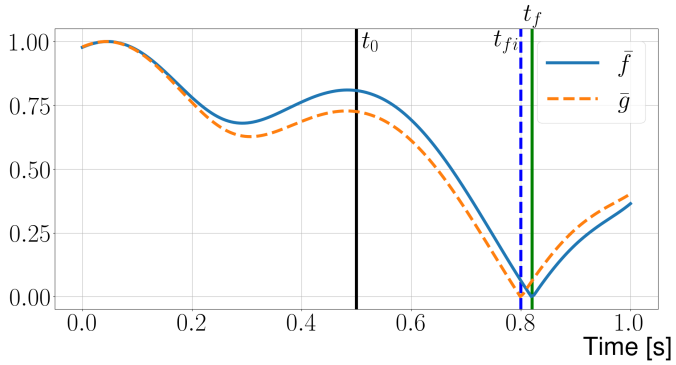
The VG methodology also differs from the path-control approach because it is impacted by the measured velocity. In particular, a non-zero measured velocity $\dot{q}^{sw}_{fi}$ can shift its equilibrium points. This point is illustrated on Figure 3.5, where the VG equilibrium is slightly different from the closest point when the measured velocity $\dot{q}^{sw}_{fi}$ is not null, for instance when the velocity is nominal

$$\dot{q}^{sw}_{fi} = \dot{\mathcal{T}}^{sw}(t_{fi}) \tag{3.26}$$

Further, in practice, the measured position is never precisely on the



(a) $t_{fi} = 0.5\,\text{s}$, $t_0 = 0.1\,\text{s}$, $\dot{q}^{sw}_{fi} \neq 0$



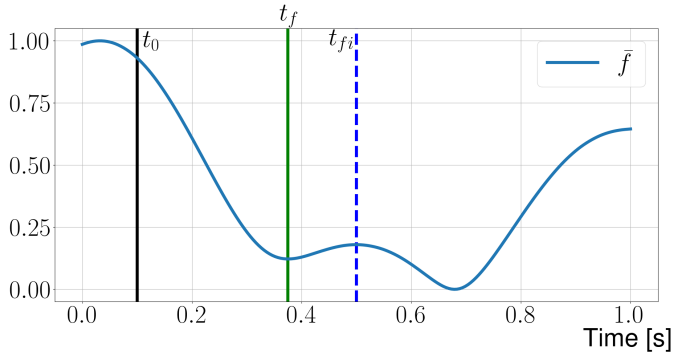
(b) $t_{fi} = 0.8\,\text{s}$, $t_0 = 0.5\,\text{s}$, $\dot{q}^{sw}_{fi} \neq 0$

Figure 3.5: Normalized f and Euclidean distance with $\dot{q}^{sw} \neq 0$.

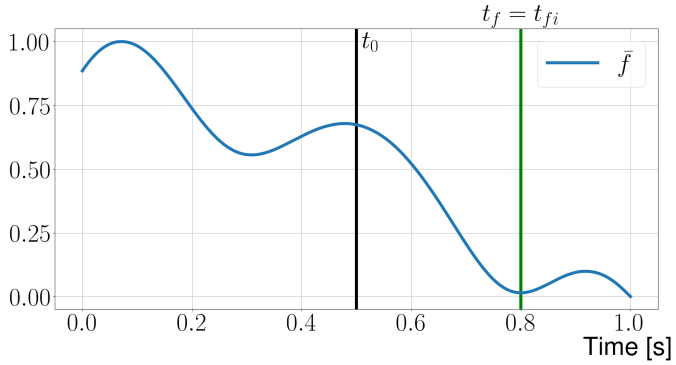
trajectory (*i.e.* $\nexists\, t$ *s.t.* $q^{sw}_{fi} = \mathcal{T}^{sw}(t)$). We study the impact of this discrepancy by adding a component to q^{sw}_{fi} belonging to the hyperplane orthogonal to $T(\sigma_{fi})$, in the direction of the nominal acceleration projected in this hyperplane, *i.e.*

$$q^{sw}_{fi} = \mathcal{T}^{sw}(t_{fi}) + a_{fi} \frac{\ddot{\mathcal{T}}^{sw}(t_{fi}) - T(\sigma_{fi})^\top \ddot{\mathcal{T}}^{sw}(t_{fi}) T(\sigma_{fi})}{||\ddot{\mathcal{T}}^{sw}(t_{fi}) - T(\sigma_{fi})^\top \ddot{\mathcal{T}}^{sw}(t_{fi}) T(\sigma_{fi})||_2} \tag{3.27}$$

for some $a_{fi} \in \mathbb{R}$. Choosing $a_{fi} \neq 0$ modifies both the VG equilibrium point and the nearest-neighbor, as depicted on Figure 3.6. Note that the impact on the equilibrium point is sensitive to the values of t_{fi}, t_0 and a_{fi}. On Figure 3.6a, the equilibrium is greatly modified, differing from both the Euclidean minimum distance and σ_{fi}. However, on Figure 3.6b the impact of this change seems negligible, and the VG equilibrium is still equal to the nearest-neighbor, which is also equal to σ_{fi}.



(a) $t_{fi} = 0.5\,\text{s}$, $t_0 = 0.1\,\text{s}$, $a_{fi} = 0.1$



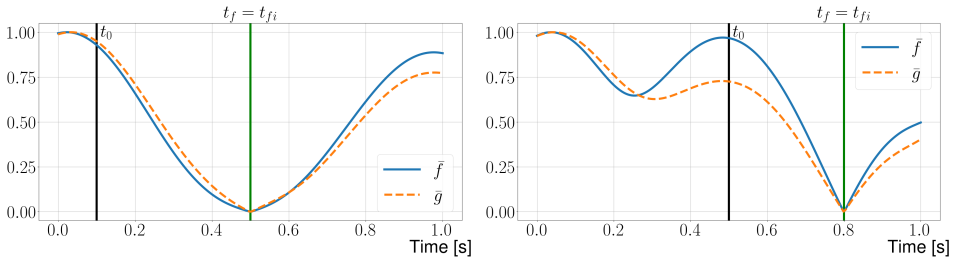
(b) $t_{fi} = 0.8\,\text{s}$, $t_0 = 0.5\,\text{s}$, $a_{fi} = 0.1$

Figure 3.6: Normalized f with $\dot{q}^{sw} = 0$ and $a_{fi} \neq 0$: orthogonal components create and move local minima.

Finally, all the previous analyses were conducted with isotropic matrices K_p and K_d (proportional to identity), which is never the case on the robot. We study the impact of anisotropic K_p and K_d with the example matrices below

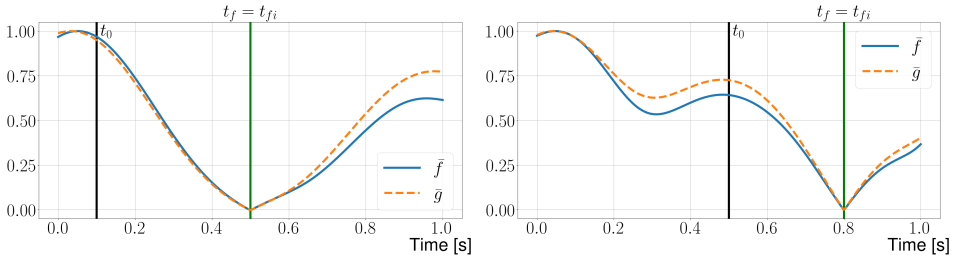
$$K'_p = \text{Diag}(\begin{bmatrix}1e4 & 1e4 & 1e4 & 2e4 & 2e4 & 2e4\end{bmatrix}) \ , \ K'_d = 2\sqrt{K'_p} \tag{3.28}$$

The results for this choice of K'_p and K'_d are pictured on Figure 3.7. Interestingly, this modification changes the profile of f, but the nearest-neighbor is still an equilibrium of the VG, and the other VG equilibrium can be changed, as revealed on Figure 3.7b. Nonetheless, with the gains tuned for the exoskeleton K_p^{exo} and K_d^{exo}, all the VG equilibria are unchanged *w.r.t.* the isotropic matrices case, as shown on Figures 3.7c and 3.7d.



(a) $t_{fi} = 0.5\,\text{s}$, $t_0 = 0.1\,\text{s}$, K'_p, K'_d

(b) $t_{fi} = 0.8\,\text{s}$, $t_0 = 0.5\,\text{s}$, K'_p, K'_d



(c) $t_{fi} = 0.5\,\text{s}$, $t_0 = 0.1\,\text{s}$, K_p^{exo}, K_d^{exo}

(d) $t_{fi} = 0.8\,\text{s}$, $t_0 = 0.5\,\text{s}$, K_p^{exo}, K_d^{exo}

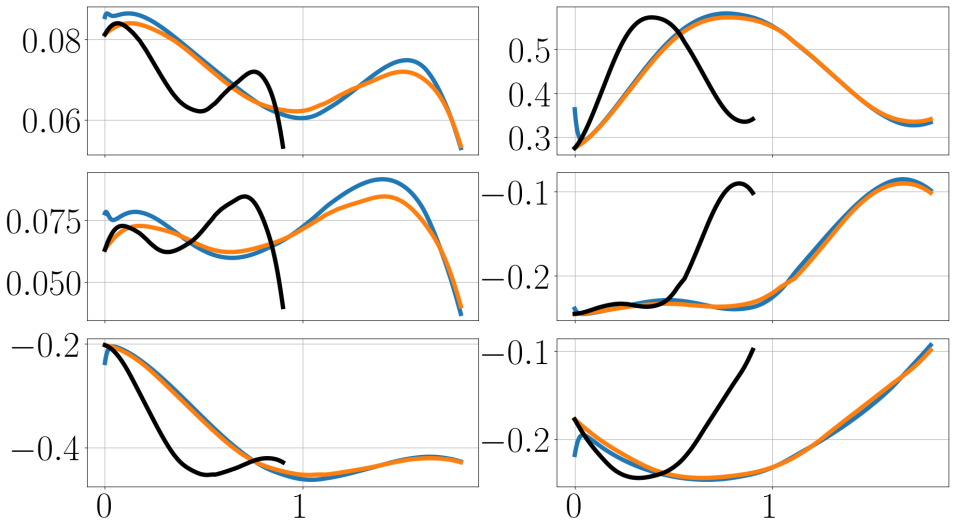
Figure 3.7: Torque norm and Euclidean distance (null measured velocity and $a_{fi} = 0$) with anisotropic gain matrices.

In summary, the VG controller locally behaves similarly to a path control algorithm, when initialized close enough to the nearest neighbor. Additionally, it cancels efforts tangential to the path. Hence, as expected, Figure 3.8 shows that the reference set-points $(P(\sigma), T(\sigma)\dot{\sigma})$ of the PD controller Eq. (3.8) closely follow the fictitious joint positions and velocities, even in the case when these positions and velocities are time-varying.

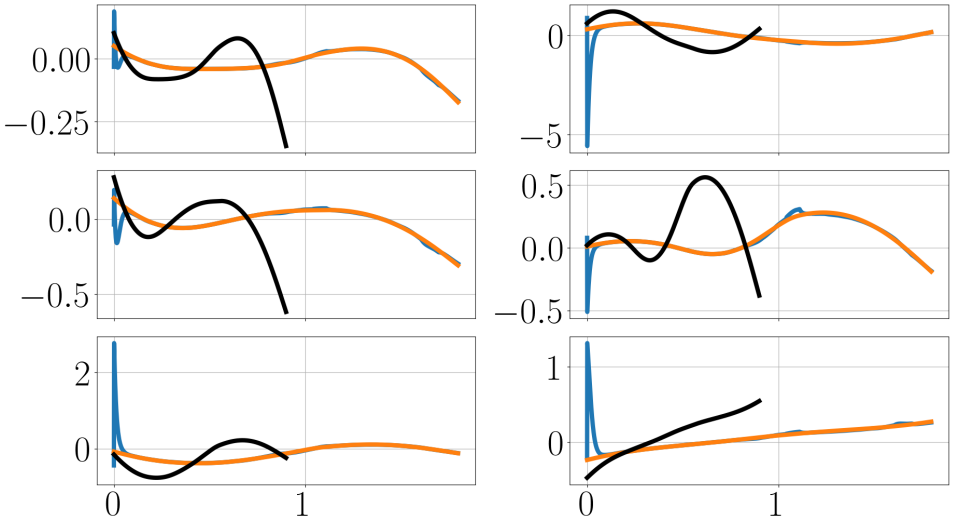
Virtual Guides behavior: experiment with a user

The VG behavior during an experiment with an able-bodied user inside the exoskeleton is depicted on Figure 3.9 and Figure 3.10. This experiment has been conducted using the complete algorithmic solution proposed in this chapter. An experimented user of the exoskeleton provides external help, holding the handles of the exoskeleton on its sides, to ensure the balance during the whole experiment and focus on the VG behavior. The support leg torques τ^{sp} used during this experiment are yet to be specified, but we focus here on the behavior of the swing leg VG controller, which is already fully described by (3.12) and (3.14).

On Figure 3.9a, the nominal position trajectory is depicted in black

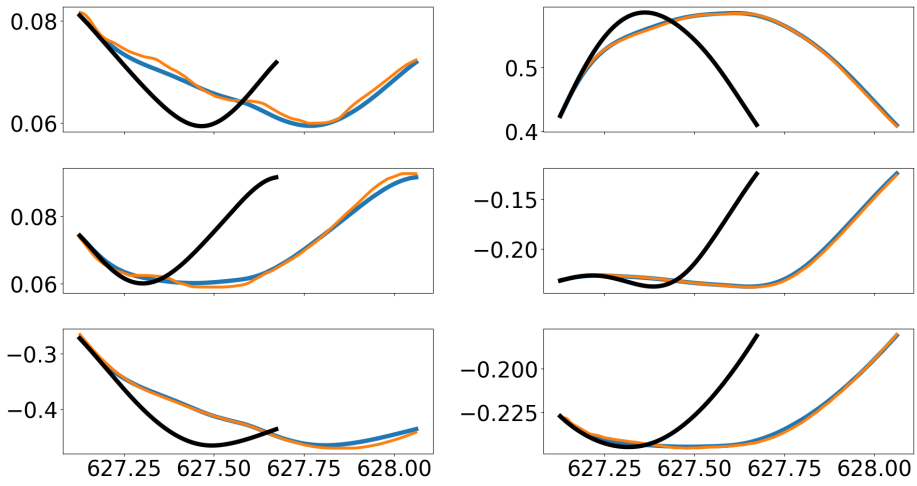


(a) Joint position

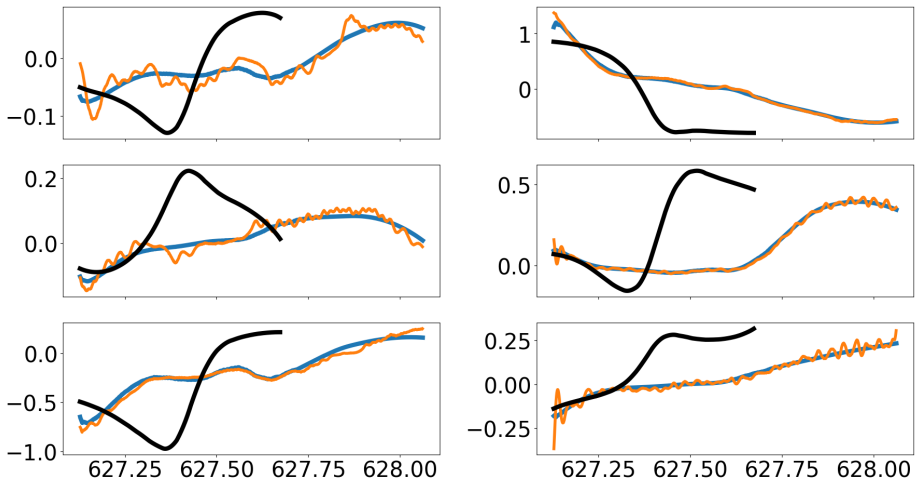


(b) Joint velocity

Figure 3.8: Simulated joint position and velocity (Black: nominal state $(\mathcal{T}^{sw}, \dot{\mathcal{T}}^{sw})$, blue: time rescaled nominal state $(P(\sigma), T(\sigma)\dot{\sigma})$, orange: fictitious state $(q_{fi}^{sw}, \dot{q}_{fi}^{sw})$).



(a) Joint position



(b) Joint velocity

Figure 3.9: Experimental joint position and velocity (Black: nominal state $(\mathcal{T}^{sw}, \dot{\mathcal{T}}^{sw})$, blue: time rescaled nominal state $(P(\sigma), T(\sigma)\dot{\sigma})$, orange: measured state $(q_m^{sw}, \dot{q}_m^{sw})$).

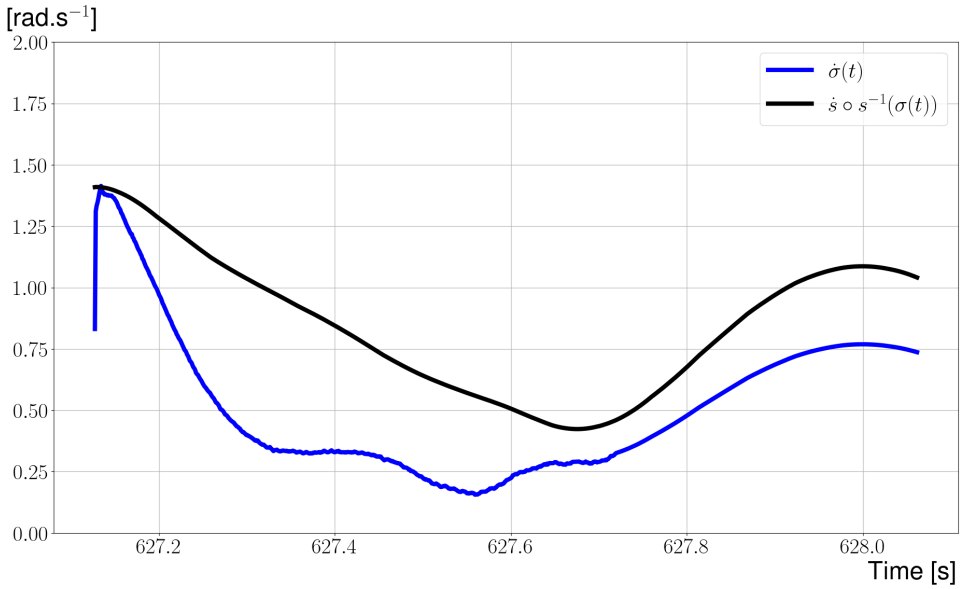


Figure 3.10: Experimental curvilinear velocity varying more than -50% of the nominal curvilinear velocity (Black: nominal curvilinear velocity $\dot{s}$ evaluated at control variable $s^{-1}(\sigma)$, blue: Virtual Guides $\dot{\sigma}$).

and the position rescaled in time by the VG controller according to (3.14) is depicted in blue. A clear consequence of the tangential VG controller is that the measured position (in orange) is closely followed by the target position. Figure 3.9b depicts the nominal, target, and measured velocities. As expected, rescaling $\mathcal{T}^{sw}$ in time amplifies the corresponding velocity by a factor $\dot{\sigma}$, as per (3.8).

3.2 Stabilizing the Center of Mass with the support leg

In this section, we detail our approach to stabilize the Center of Mass (CoM) dynamics around a reference CoM trajectory. We adapt the existing admittance controller. Whole-body admittance control is used to stabilize the CoM around the trajectory during passive walking, as detailed in Section 2.3. However, during single support phases, the swing leg torques are entirely defined by the VG controller. Therefore, we propose below (in Section 3.2.1) an adaptation of the passive walking admittance SoT from Section 2.3 to compute the support leg joint targets taking into account the motion of the swing leg. In Section 3.2.4, we expose the rationale of our admittance SoT

modification and provide a numerical illustration of its performance for the admittance task.

3.2.1 Modification of admittance's SoT

As discussed earlier, the VG methodology implies that the swing leg target position q_t^{sw} follows a fixed path P. To prescribe the motion of the swing leg, we design a single support Stack-of-Tasks (SoT), denoted SoT_2, with a task on the swing leg joints having higher priority than the CoM acceleration tracking task, as follows

Stack of Tasks SoT_2

Level 0: *swing leg joint* position and support foot pose tracking;

Level 1: CoM acceleration tracking (2.19);

Level 2: back orientation tracking;

Level 3: regularizing joint tracking.

The swing leg task is to follow the swing leg target joint position. It writes

$$e_0^0 = P(\sigma) - q^{sw} \tag{3.29}$$

We design the reference acceleration of this task *s.t.* it yields fast convergence to the same targets $(P(\sigma), T(\sigma)\dot{\sigma})$ as the VG controller (3.8)

$$\ddot{q}^{sw^*} = K_p^{sw}(P(\sigma) - q_t^{sw}) + K_d^{sw}(T(\sigma)\dot{\sigma} - \dot{q}_t^{sw}) \tag{3.30}$$

with (K_p^{sw}, K_d^{sw}) high gains and $(q_t^{sw}, \dot{q}_t^{sw})$ the swing leg joint target computed during the previous control loop. This task is the one with the highest priority, therefore its associated cost $w_0^0 = ||\ddot{q}^{sw^*} - \ddot{q}^{sw}||_2^2$ is always identically zero, yielding

$$\ddot{q}_t^{sw} = \ddot{q}^{sw^*} \tag{3.31}$$

Thus, fast convergence of $(q_t^{sw}, \dot{q}_t^{sw})$ to $(P(\sigma), T(\sigma)\dot{\sigma})$ is ensured (by the choice of high gains matrices (K_p^{sw}, K_d^{sw})), which enables the IK solver to compensate for the bias introduced by the swing leg motion in the CoM acceleration task. This property is further derived in Section 3.2.4.

As in Section 2.3, the optimal acceleration of the support leg $\ddot{q}_t^{sp}$, resulting from tasks of level $1-3$, is integrated twice to define the target support leg position and velocity $(q_t^{sp}, \dot{q}_t^{sp})$. Finally, the support leg joint torques τ^{sp} are computed using a high-gain PD controller

$$\tau^{sp} = K_p^{sp}(q_t^{sp} - q^{sp}) + K_d^{sp}(\dot{q}_t^{sp} - \dot{q}^{sp}) \tag{3.32}$$

with $K_p^{sw}, K_d^{sw} \succ 0$ constant diagonal gain matrices. As the swing and support legs are only defined during single support phases, we keep the SoT_1 during double support phases[3].

3.2.2 Synchronization of legs by Time Rescaling (TR)

The SoT_2 described in the previous section incorporates the same CoM acceleration task as SoT_1 at level 1. The desired acceleration for this task is computed by (2.19) which, in turn, aims at tracking the desired CoP computed from (2.18). These two desired quantities are computed to stabilize the CoM around a state trajectory x, with corresponding input trajectory p. We now define these reference trajectories synchronized with the patient schedule σ.

A natural way to define two synchronized trajectories x and p from the nominal trajectory $\mathcal{Q}$ is to first time-rescale the nominal trajectory $\mathcal{Q}$ with the patient schedule σ, defining the *time-rescaled trajectory* q^{TR}

$$q^{TR}(\sigma) \triangleq \mathcal{Q} \circ \mathrm{s}^{-1}(\sigma) \tag{3.33}$$

and then, to compute the *time-rescaled state trajectory* x^{TR} and corresponding time-rescaled input trajectory p^{TR} from it

$$\begin{cases} x = x^{TR}(\sigma) \triangleq \mathcal{FK} \circ q^{TR}(\sigma) \\ p = p^{TR} \triangleq \mathcal{ID} \circ q^{TR}(\sigma) \end{cases} \tag{3.34}$$

readily obtained using Forward Kinematics ($\mathcal{FK}$) (see *e.g.* [Carpentier2019] for implementation) and Inverse Dynamics ($\mathcal{ID}$).

The kinematic reference trajectories required for the other tasks are readily computed from the time-rescaled trajectory q^{TR} using $\mathcal{FK}$. The derivation of the controller that we propose in this Chapter is now complete. It is schematically depicted on Figure 3.11.

3.2.3 Trajectory smoothing

Task changes produce discontinuities of target trajectories (q_t,$\dot{q}_t$). These occur either when lifting the foot off the ground (switching from SoT_1 to SoT_2), or when landing the foot (switching from SoT_2 to SoT_1).

[3]Keeping during the double support phases is the safest approach, using all DoF to stabilize the CoM dynamics. An opposite approach, using all DoF to follow the patient schedule, is sketched in Appendix B.

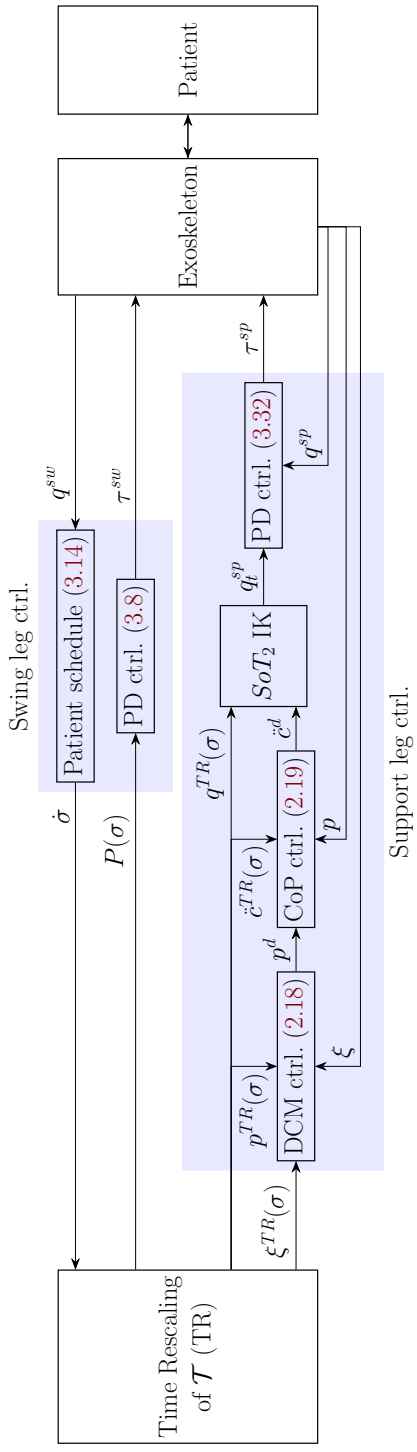


Figure 3.11: Split-leg rehabilitation controller using the naive TR synchronization strategy.

The discontinuities at foot lift are smoothed when switching from SoT_1 to SoT_2 by interpolating between the nominal trajectory $\mathcal{T}^{sw}$ and the last swing joint target position q_t^{sw}. This interpolation is performed using well-designed 5^{th} order Splines over the remaining duration of the step. The smoothing is performed once per step when the swing foot lifts off, hence the modified swing leg trajectory is entirely known at the beginning of the swing phase, and so the swing leg path P can be computed as explained in Section 3.1.2.

Besides, there are also target discontinuities when switching from SoT_2 to SoT_1 as, in the general case, the foot landing position in Cartesian space is not exactly on the trajectory due to CoM tracking errors. When the stabilization effect of admittance control is efficient, the CoM stays close to the nominal one. In this case, when tasks change occurs at impact, the swing foot lands almost at the expected place, and the targets are sufficiently close to make the discontinuity imperceptible. When the stabilization effect of admittance control is less efficient, the CoM can get noticeably far from the nominal one, which results in larger discontinuities due to the tasks change and the lack of target reconciliation strategy. We tackle the loss of balance from which they result in Chapter 4.

3.2.4 Impact of the modification of the SoT

We now expose the rationale of our modification of admittance control. For this purpose, we propose a formal analysis of the changes. We consider the exoskeleton in single support. It constitutes an open kinematic chain. We split this kinematic chain into a support sub-chain (the support leg), in contact with the ground, on which the admittance task bears, and a swing sub-chain (the swing leg), with no contact with the ground. We only study the impact of the SoT modification on the target CoM acceleration, assuming perfect tracking of the target joint acceleration q_t. To lighten the notation, we do not distinguish target and actual positions for the rest of this section, *i.e.* $q = q_t$.

We advocate that, provided a sufficient number of DoFs are included in the support sub-chain, the admittance task (2.19) can be satisfied with performance similar to the original SoT_1, exposed in Section 2.3. This is suggested by computations that we now detail.

Consider the schematic representation of the exoskeleton depicted on Figure 3.12 during the single support phase. The kinematic chain is opened, and it consists of 13 bodies B_j linked together by 12 revolute articulations J_i. Each body B_j has an associated frame F_j with an origin O_j. The articulation J_i links the body B_i to B_{i+1}, with $i \in [0, 11]$. The first chain

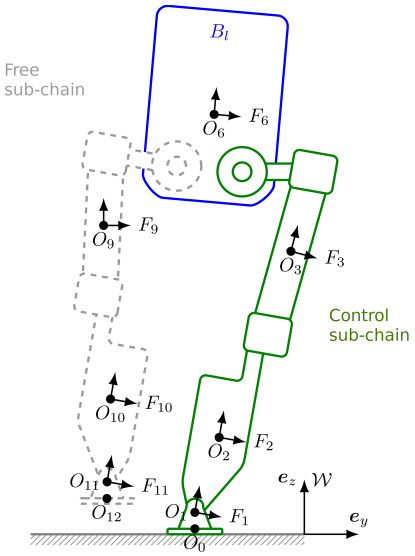


Figure 3.12: Atalante's kinematic chain during single support phases.

goes from the support foot B_0 to the back B_6, and the second chain from B_6 to the end of the robot's kinematic chain B_{12}. Both sub-chains possess 6 DoF.

Let us write, *w.r.t.* the inertial frame $\mathcal{W}$, c^{sw} (resp. c^{sp}) the CoM position of the support sub-chain (resp. the swing sub-chain) with mass m^{sp} (resp. m^{sw}). One gets

$$\ddot{c} = \frac{m^{sp}}{m^{sp}+m^{sw}}\ddot{c}^{sp} + \frac{m^{sw}}{m^{sp}+m^{sw}}\ddot{c}^{sw} \tag{3.35}$$

The subscript $\cdot|_k$ is used for quantities defined *w.r.t.* the reference frame F_k, expressed in the inertial frame $\mathcal{W}$. Ω_j is the angular velocity vector of frame F_j *w.r.t.* the inertial frame $\mathcal{W}$, and a_k the acceleration of O_k. The acceleration due to the swing leg motion is readily computed as

$$\ddot{c}^{sw} = \ddot{a}_6 + \dot{\Omega}_6 \wedge O_6O_{c^{sw}} + \ddot{c}^{sw}|_6 + \Omega_6 \wedge (\Omega_6 \wedge O_6O_{c^{sw}}) + 2\Omega_6 \wedge \dot{c}^{sw} \tag{3.36}$$

where all the quantities in the second line are independent of the acceleration $\ddot{q}^{sp}$ of the 6 DoF of the support sub-chain. To highlight the dependence of

the various factors *w.r.t.* the variables of both legs, we express $\ddot{c}$ as

$$\begin{aligned}\ddot{c} = & \frac{m^{sp}}{m^{sp}+m^{sw}}\ddot{c}^{sp}(\ddot{q}^{sp}, q^{sp}) \\ & + \frac{m^{sw}}{m^{sp}+m^{sw}}(\ddot{a}_6(\ddot{q}^{sp}, q^{sp}) + \dot{\Omega}_6(\ddot{q}^{sp}, \dot{q}^{sp}, q^{sp}) \wedge O_6O_{c^{sp}}(q^{sp}, q^{sw})) \\ & + \frac{m^{sw}}{m^{sp}+m^{sw}}b(\ddot{q}^{sw}, \dot{q}^{sp}, \dot{q}^{sw}, q^{sp}, q^{sw}) \\ & \triangleq f(\ddot{q}^{sp}, \dot{q}^{sp}, q^{sp}, q^{sw}) + \frac{m^{sw}}{m^{sp}+m^{sw}}b(\ddot{q}^{sw}, \dot{q}^{sp}, \dot{q}^{sw}, q^{sp}, q^{sw})\end{aligned} \tag{3.37}$$

with

$$b(\ddot{q}^{sw}, \dot{q}^{sp}, \dot{q}^{sw}, q^{sp}, q^{sw}) = \ddot{c}^{sw}|_6 + \Omega_6 \wedge (\Omega_6 \wedge O_6O_{c^{sw}}) + 2\Omega_6 \wedge \dot{c}^{sw} \tag{3.38}$$

This term b, entirely determined by the known acceleration $\ddot{q}^{sw}$ of the swing sub-chain, can be seen as a bias in the equation (3.37) governing $\ddot{c}$.

As appears above, the first and second rows in equation (3.37), gathered into a single function $f(\ddot{q}^{sp}, \dot{q}^{sp}, q^{sp}, q^{sw})$, are controlled by the acceleration $\ddot{q}^{sp}$ of the 6 DoF of the support sub-chain. Provided a sufficient number of DoFs are used, the support sub-chain should be able to compensate for the bias b created by the swing sub-chain and generate the desired total CoM acceleration. Mathematically, if b belongs to the image of the mapping f, then it can be compensated for. The support leg possesses 6 DoF, while the CoM acceleration is a 3D quantity, therefore it seems possible to realize any given CoM acceleration.

To further validate this observation, the proposed split of DoF, and subsequent changes in the SoT, we perform comparative simulations at nominal velocity. The nominal curvilinear abscissa is our control variable for the swing leg

$$\sigma(t) = \mathrm{s}(t) \tag{3.39}$$

which yields

$$\begin{cases} q_t^{sw} = P \circ \mathrm{s}(t) = \mathcal{T}^{sw}(t) \\ \dot{q}_t^{sw} = T \circ \mathrm{s}(t)\dot{\mathrm{s}}(t) = \dot{\mathcal{T}}^{sw}(t) \end{cases} \tag{3.40}$$

In this case, the reference CoM (resp. CoP) trajectory is simply the nominal CoM (resp. CoP) trajectory

$$\begin{cases} x = \mathcal{FK} \circ \mathcal{T}(t) \\ p = \mathcal{ID} \circ \mathcal{T}(t) \end{cases} \tag{3.41}$$

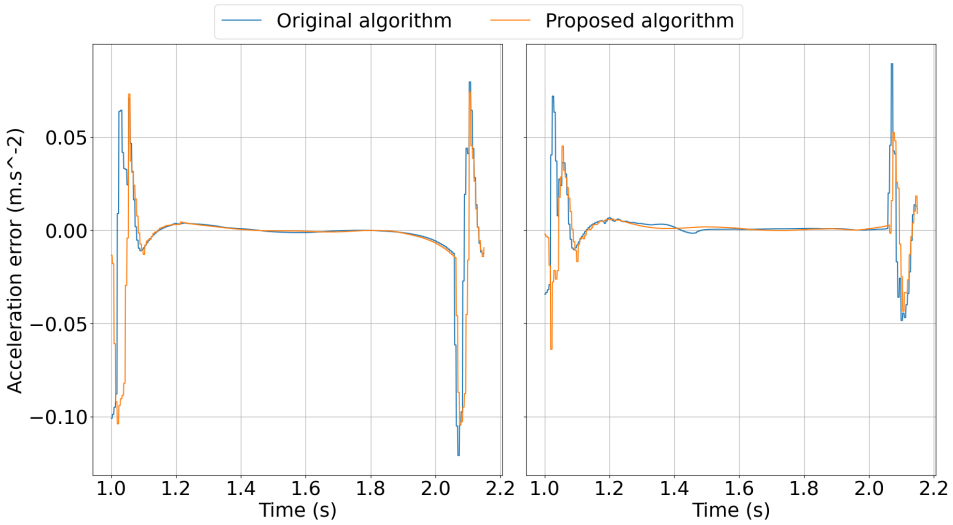


Figure 3.13: CoM acceleration IK task error in simulation (same error magnitude of the original and proposed algorithm). Original algorithm: SoT_1. Proposed algorithm: SoT_2. Left: forward direction. Right: lateral direction.

readily computed using Forward Kinematics ($\mathcal{FK}$) and Inverse Dynamics ($\mathcal{ID}$), as in (3.34).

Figure 3.13 depicts the error of the CoM acceleration task during one simulated step both for the original algorithm, using SoT_1 during the complete step (equivalent to the passive walking controller of Atalante described in Chapter 2), and for the proposed algorithm, using SoT_2 during single support phases and SoT_1 during double support phases. The error of the CoM acceleration task is the difference between the desired CoM acceleration $\ddot{c}^d$, as defined by (2.19), and the target CoM acceleration, as defined by the target joint acceleration $\ddot{q}^t$ solution of the IK solution to SoT_1 or SoT_2. As shown on Figure 3.13, the error magnitude of the CoM acceleration task during the single support phase (from 1.1 s to 2.05 s) is identical for the original algorithm and the proposed algorithm. In addition, Figure 3.13 includes leading and tailing double support phases (from 1.0 s to 1.1 s and from 2.05 s to 2.15 s), *s.t.* one can check that the error magnitude is not increased due to the sudden tasks changes at the single-double support phase changes. Thus, the comparison confirms that the proposed split has a sufficient number of actuation DoF to track the desired acceleration $\ddot{c}^d$ as well as the original admittance control scheme during single support phases.

3.3 Impact on safety of patient empowerment

In this section, we first study the stabilization performance of the proposed rehabilitation controller, both in simulation and during experiments. All the simulation results presented in this section have been obtained using the open-source simulator Jiminy [Duburcq2019]. Then, we provide a numerical illustration of the limits of the proposed synchronization strategy, which motivates the next Chapter.

3.3.1 Stability analysis

The safety of exoskeletons is directly linked to the stability of the walking exercises. Hence, the stability must be rigorously evaluated, as recommended in [Pinto-Fernandez2020]. To do so, we evaluate in this chapter the stability of the walking exercises enabled by this controller over 10-second simulations and experiments. We use a dummy instead of a patient during all experiments, as shown on Figure 3.14, because able-bodied users tend to involuntarily stabilize the exoskeleton using their upper bodies and/or legs (while disabled patients might not be able to act similarly).

In order to ensure the repeatability of the experiments, one must generate virtual patient efforts τ_u^{sw} and τ_u^{sp}, or the consequences of these efforts expressed through the variations of the patient velocity $\dot{\sigma}^u$. Because Atalante does not possess any force sensor at the interfaces with the patient nor at the joint level, we do not have any experimental data to reproduce the patient efforts (τ_u^{sw},τ_u^{sp}). Therefore, we generate virtual control variable velocity profiles $\dot{\sigma}^u$ to simulate the consequences of variations of the patient efforts, and not virtual efforts directly. We generate these profiles $\dot{\sigma}^u$ for arbitrarily tailored functions $\dot{\eta}^u$ as follows

$$\sigma^u(t) = \mathrm{s} \circ \eta^u(t) \tag{3.42}$$

We design the $\dot{\eta}^u$ functions with one non-nominal phase variable velocity change. These functions are parametrized with three variables: the non-nominal phase variable velocity $\bar{\dot{\eta}}$, the starting time of the velocity change t_η, and the duration of the modification $\Delta\tau$. To generate a periodic signal, we make these functions start and end at nominal velocity. The η^u phase variable function is created from a piecewise affine function, with a discontinuous phase variable velocity change, smoothed by a Gaussian filter. An example of an η^u function is pictured on Figure 3.15. Note that these functions are not, a priori, exactly representative of patient efforts. However, they allow

Figure 3.14: Dummy inside Atalante during passive walking (ensures that no user stabilizes the system).

for an intuitive and systematic exploration of a class of velocity profiles with various levels of difficulty.

Note that, as we constrain the η^u function to end at a specific final value η_f, only a limited set of combinations of non-nominal velocity, duration and start time can be explored.

3.3.2 Simulation results

The stability performance evaluation of the controller proposed in this chapter consists of 10-second rigid-body walking simulations for each set of parameters. A simulation is deemed successful if the robot has not fallen at the end of the simulation. The aggregated results are shown on Figure 3.16 where, for each set of parameters (custom velocity duration, custom velocity), the

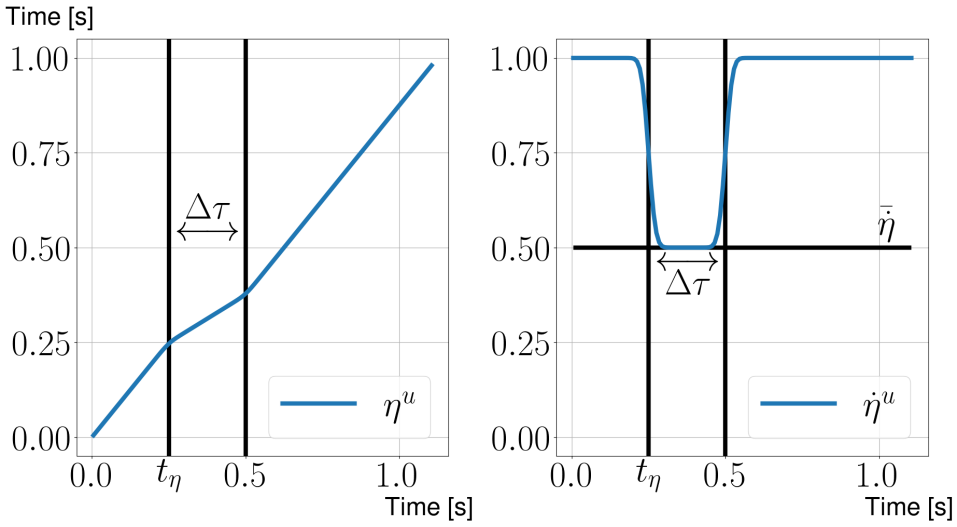


Figure 3.15: η^u (Left) and $\dot{\eta}^u$ (Right) functions, along with the three parameters (emulation of patient-induced slow-downs).

rates of success are lumped in three bins: above 90%, between 35% and 90%, and below 35%. The custom velocity parameter $\bar{\dot{\eta}}$ is varying along the Y axis of Figure 3.16, while the custom velocity duration parameter $\Delta\tau$ is varying along the X axis. The heatmap on Figure 3.16 shows the stabilization algorithm has good performance only around the nominal velocity $\dot{\eta}^n \triangleq 1.0$, for custom velocities $\bar{\dot{\eta}}$ between 0.8 and 1.5. With lower or higher velocities custom velocities $\bar{\dot{\eta}}$, the algorithm fails to stabilize the exoskeleton, especially when the velocity change is too long.

Surprisingly, the stabilization algorithm manages to stabilize some time-rescaled trajectories which do not respect the constraints acting on the system. Indeed, while the nominal trajectory respects both the direct (the torque limits) and indirect (the unilateral force constraint) actuation limits, as detailed in Section 2.2, the time-rescaled nominal trajectory, defined by (3.33), violates the indirect actuation limit even for small variations around the nominal velocity. This is illustrated on Figure 3.17, where the rate of time-rescaled trajectories respecting the unilateral force constraint for each set of parameters (custom velocity duration $\Delta\tau$, custom velocity $\bar{\dot{\eta}}$) is reported. These rates, aggregated along the starting time of the velocity change t_η dimension, are lumped in three bins: above 90%, between 35% and 90%, and below 35%. Comparing Figure 3.17 with Figure 3.16 shows that the stabilization algorithm manages to stabilize even some time-rescaled

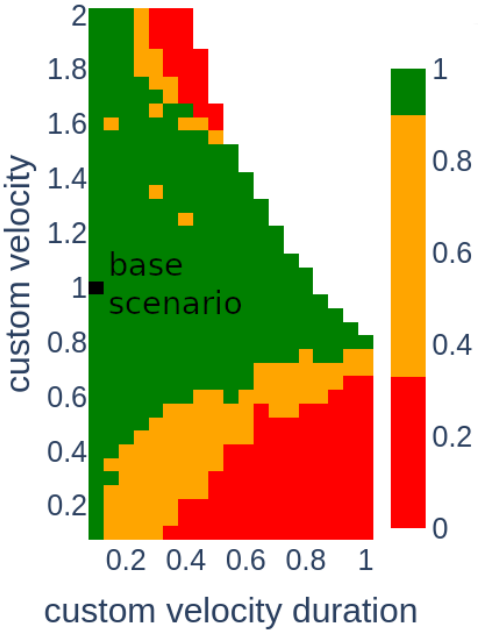


Figure 3.16: Heatmap of 10 s walking simulation success (good stabilization performance around the base scenario, poor ones in the low-velocity range). Success rate computed for each (custom velocity duration, custom velocity) couple, aggregating the (custom velocity start) dimension.

trajectories which do not respect the unilateral force constraint. While this performance is surprising at first, they only reflect that the stabilization algorithm manages to dynamically find trajectories, close to the time-rescaled trajectories, which respect all the actuation constraints. In other words, these results reveal that there exist trajectories respecting all the actuation constraints in a neighborhood of trajectories that violate constraints, and the stabilization algorithm acts as a dynamic generation algorithm of these trajectories. The distance of these trajectories to the tracked trajectory, which does not respect the constraints, is reflected in the tracking error of the stabilization algorithm.

3.3.3 Experimental observations

The custom functions depicted in Section 3.3.1 are used again for experimental stability evaluation. As opposed to the simulations, the initial state of the exoskeleton-dummy system is standing still. Hence, a starting step, different

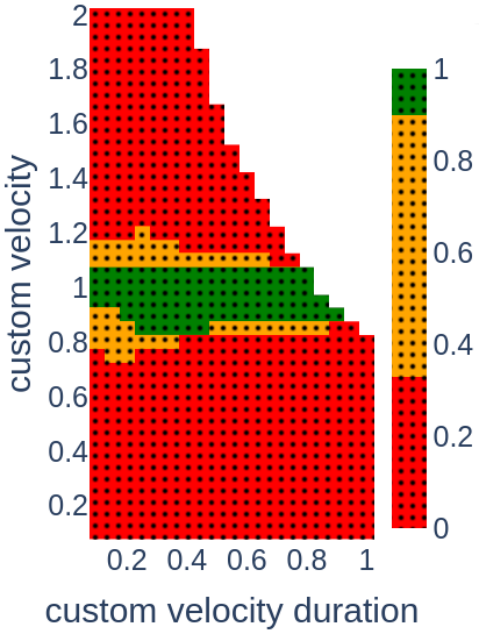


Figure 3.17: Heatmap of time-rescaled nominal trajectories violating the unilateral force constraint. Rate of time-rescaled trajectories respecting the constraint computed for each (custom velocity duration, custom velocity) couple, aggregating the (custom velocity start) dimension.

from the next cyclic steps, is required. In order to get closer to the simulation setup, we manually stabilize this starting step and, if required, the two first cyclic steps by gently holding the exoskeleton side handles.

A reduced number of parameter combinations is tested to keep the total duration of the experiments at a reasonable level (*i.e.* being able to complete the experiments in one day). The custom velocity $(t_\eta, \delta\tau, \bar{\dot{\eta}})$ triplets are chosen to span as much as possible the simulations test range, but they are also adapted during the test sessions *w.r.t.* the first results. For instance, for each such triplet, we stop decreasing the custom velocity as soon as we find an unstable velocity profile. This decision is guided by the results depicted on Figure 3.16. For this reason, we only test velocities higher than $\dot{\eta}_{min} = 0.6$.

The stabilization results are shown on Figure 3.18. We could not try velocities higher than $\dot{\eta}_{max} = 1.5$ due to the electrical power limits of the exoskeleton. We see that the walking experiment is successfully stabilized for all the velocities in the $[1.0, 1.5]$ velocity range, and for all velocity change

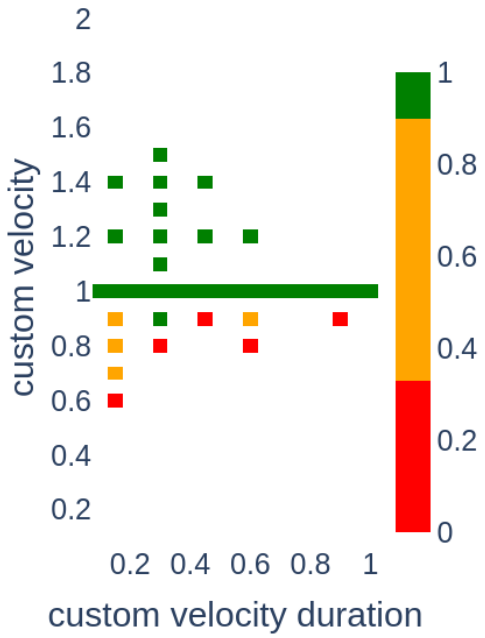


Figure 3.18: Heatmap of 10 s walking experiments success. Success rate computed for each (custom velocity duration, custom velocity) couple, aggregating the (custom velocity start) dimension. The results are in accordance with simulation, with lower performance in the low-velocity range.

durations. However, walking with our rehabilitation controller at velocities lower than the nominal one often leads to falling, even for short-duration changes. These results are mostly in accordance with the simulation results of Figure 3.16, but with lower performance in the low-velocity range.

The observed performance reduction from simulation to experiments can be explained by the model discrepancies with reality. Indeed, there are multiple effects unaccounted for in simulations, among which the most important one is the presence of flexibilities which result in several centimeters end-effector position errors [Vigne2021; Vigne2022]. Other factors which could also explain the performance reduction are, but not limited to, the unknown interaction of the dummy with the exoskeleton and model uncertainty (its total mass is precisely known, as with patients, but the limb masses and inertias are approximated), or sensor errors.

These experimental performances are not sufficient for the rehabilitation of deficient patients yet, as they will most likely have issues providing enough effort to go as fast as the nominal velocity.

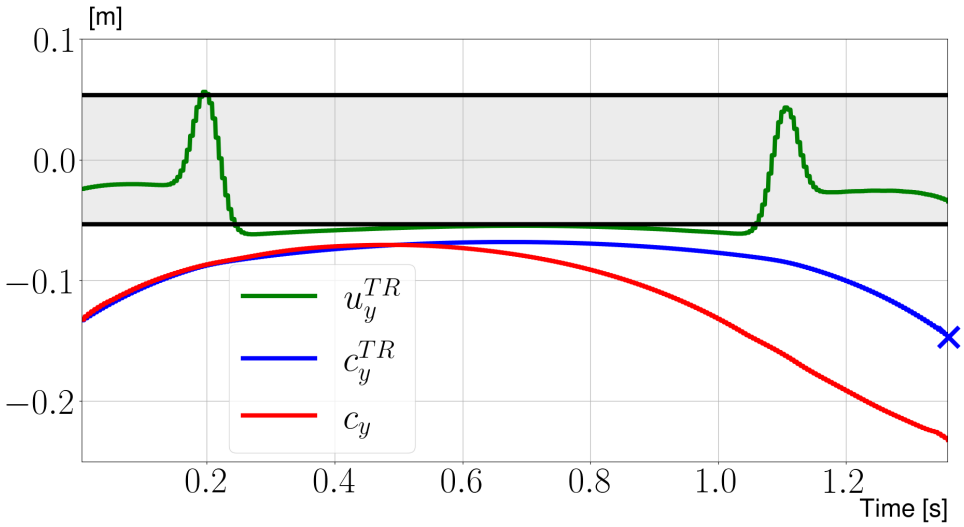


Figure 3.19: Reference CoP u_y^{TR} (violating the support polygon limits in black), CoM c_y^{TR} and measured CoM c_y (diverging toward the end of the step) positions along the Y axis of the inertial frame $\mathcal{W}$. Reference quantities computed using the TR strategy and a simulated patient velocity $\dot{\sigma}^u$ as low as 60% ($t_\eta = 0.2\,\text{s}$, $\Delta\eta = 0.9\,\text{s}$ and $\bar{\dot{\eta}} = 0.6$). Black horizontal lines represent the limits of the support polygon $\mathcal{SP}$ on the Y axis.

3.3.4 Performance analysis during a single step simulation

The shortcomings of our rehabilitation controller in terms of balance, especially in the low-velocity range, come from the choice of the Time Rescaling (TR) strategy described in Section 3.2.2. Indeed, this TR strategy does not take the unilateral contact constraints into account, as already exposed in Section 3.3.2. As a result, the CoP trajectory p^{TR} is not confined to the support polygon $\mathcal{SP}$, *i.e.* the state trajectory x^{TR} does not respect the input-constrained dynamics, and the stabilization algorithm is unable to generate a CoP command which stabilizes the trajectory. This is illustrated on Figure 3.19 which reports the result of a simulation with a custom velocity $\bar{\dot{\eta}} = 0.6$ for $\delta\tau = 0.9\,\text{s}$. The reference CoP, in green, is not contained in the support polygon $\mathcal{SP}$ when modulating the trajectory at 60% of the nominal velocity. The final state (the endpoint of the red line) is different from the nominal final state $x_f \triangleq \mathcal{FK} \circ \mathcal{T}(T_f)$ (the endpoint of the blue line). The 8 cm resulting error is sufficient to make the robot fall at the end of the step.

This simulation reveals that the choice of the reference trajectory according to the TR strategy is too naive to ensure the safety of the walking exercise, and calls for a more careful choice of legs synchronization strategy. This is the topic of the next chapter.

Chapter 4

Monitoring safety

Chapitre 4 - Gestion de la sécurité: *Dans ce chapitre, nous proposons une nouvelle stratégie de synchronisation des jambes et détaillons son implémentation dans le contrôleur développé au chapitre précédent. L'objectif de cette modification est d'augmenter l'équilibre lors des exercices de rééducations actifs avec Atalante. Cette nouvelle stratégie s'appuie sur un problème de contrôle optimal en temps final libre pour un modèle réduit du système qui est résolu en ligne à haute fréquence. Parce que résoudre un tel problème à une fréquence suffisamment élevée pour le confort du patient est difficile, nous avons développé une méthode de résolution originale réalisant une dichotomie sur la fonction de faisabilité d'un problème quadratique. La solution de ce problème fournit une nouvelle trajectoire pour le centre de masse du système, d'une durée cohérente avec la vitesse de la jambe de vol du patient, ou aussi proche que possible de cette vitesse, tout en respectant les contraintes dynamiques du système. Les modifications apportées au contrôleur sont donc doubles : la trajectoire de référence est remplacée par cette trajectoire optimale, et, lorsque nécessaire, la vitesse du patient est limitée pour préserver l'équilibre. Les performances de stabilisation de ce contrôleur sont évaluées en simulation et à l'aide d'expériences avec un mannequin. Des expériences illustrant le comportement général de ce contrôleur avec un patient sont également fournies.*

Contents

The experiments of Section 3.3 have shown that the Time Rescaling (TR) strategy fails to generate trajectories that can be stabilized by admittance. In this chapter, we derive a novel Optimal Planning (OP) strategy and modifications to the controller from the previous chapter to solve this issue and increase performance.

The OP strategy leverages an Optimal Control Problem (OCP) to generate a CoM trajectory, meant to be tracked by admittance control, while taking the unilateral force constraint into account. The literature about generating CoM trajectories online is broad and dates back to the early 2000s, first as intricate *preview regulators* [Kagami2002; Kajita2003; Kagami2011], which require a CoP reference planner to provide the reference CoP to track, then as plain OCPs [Wieber2006b], which only require the contact sequence, and can even optimize their location [Herdt2010]. Since then, the literature about online walking trajectory generation has flourished, leveraging reduced [Feng2013; Caron2016] or complete [Erez2013; Koenemann2015; Dantec2022] dynamical models of the robots, to optimize first a reduced set of parameters (*e.g.* the horizontal motion of the CoM only [Wieber2006b]), and then gradually increase the number of free parameters (*e.g.* optimizing a combination the 2D CoM motion [Khadiv2016; Caron2020; Smaldone2021], CoM height [Caron2020], stepping location [Khadiv2016; Smaldone2021], and switching time [Khadiv2016; Smaldone2021]).

The OCP we formulate is specific for two reasons:

1. Our primary objective is to maximize patient expressiveness. Hence, our OCP has a free final time.

2. The optimal solution acts as a safety filter on the patient schedule. Hence, it has to respect hard real-time constraints (typically running at least at $1kHz$).

Among all the online CoM trajectory generation algorithms for biped robots published over the past 2 decades, we found only 5 explicit occurrences of algorithms optimizing the switching time (note that including quadrupeds adds only 2 occurrences). Namely, [Khadiv2016; Griffin2017; Smaldone2021; Ahn2021; Katayama2022], among which two demonstrated their algorithm in simulation only. In view of application, we did not consider these 2 algorithms with no experimental evidence of their performance. Among the three remaining algorithms, only one [Griffin2017] satisfies the hard 1kHz real-time constraint. However, this algorithm relies on the definition of the CoP trajectory as a constant piecewise function with specified durations of each piece, therefore the duration adjustment is not a free variable in their optimization, but rather comes as an input from a stabilizing feedback on the measured CoM state. Thus, this algorithm is not suitable for our use case, in which we require the OCP to maximize a cost formulated on the final time of the trajectory.

In details, we formulate an OCP as follows. We model the system as subjected to the LIP dynamics with the unilateral force constraint being the only actuation constraint. In addition, we limit the horizon to the end of the current step, and we fix the footprints to be the same footprints as in the nominal trajectory. The final time of the trajectory is free, and we set a penalty on its difference with some patient-dependent target time. This target time is computed by assuming the patient will produce efforts that keep the instantaneous swing leg velocity constant up until the end of the current step. To reflect that patient expressiveness has the highest priority over all other objectives, we formulate the OCP as a bi-level optimization problem, with the cost on the final time being the sole cost in the upper-level problem. The sublevel cost is only a control input regularizing cost. The importance of the sublevel comes from its constraints: the input-constrained LIP dynamics, plus some well-chosen final state equality constraint ensuring the recursive feasibility of the steps. Because of the sublevel constraints, the final time penalty can not always be nullified. Thus, our OP strategy reveals when the patient-driven swing leg velocity cannot safely be satisfied. In this case, the final time of the optimal solution provides us with the optimal safe velocity of the swing leg (*i.e.* which maximizes the patient desire by finding the closest final time to the patient target time).

The controller from Chapter 4 is adapted to this online CoM trajectory

generation strategy in two different ways. First, the admittance reference trajectory, used to control the support leg, is replaced by the solution to the OCP. Then, the patient swing leg velocity is replaced by the optimal velocity from the OCP. When the OCP manages to nullify the final time penalty, the optimal velocity is identical to the patient velocity and this change has no impact. When the final time penalty is not nullified, the patient velocity is filtered. Hence, the OCP also acts as a safety filter of the patient velocity. Therefore, it has to run at the same frequency as the VG controller. It is known from expert knowledge at Wandercraft that increasing the VG update frequency (3.14) directly increases the user's ability to drive the motion along the swing leg path P. It has been found that running VG at $1kHz$ yields a good trade-off between computation load and user efforts.

Solving an OCP with free final time at such a high-frequency replanning is challenging, especially with the computational power limitations of the Atalante onboard computer. As we found no readily applicable solver in the literature, we have developed an original solving method of our bi-level OCP based on bisection on the feasibility function of Quadratic Program (QP)s. We show that this approach is optimal in almost all situations[1] encountered for the LIP dynamics in one dimension. We provide numerical hints that this is probably always true in cases of practical interest. Further, even in cases where our method does not find the optimal trajectory, *i.e.* when the upper-level cost is not minimized, the constraints of the sublevel are always satisfied. Therefore, even suboptimal trajectories are safe (*i.e.* respecting the dynamics and boundary conditions).

The chapter is organized as follows. First, we motivate and derive our OP strategy, along with the modifications it induces in the rehabilitation controller from Chapter 3. Then, we study the optimality of the bisection method used to determine the solution to the OCP. Finally, we quantify the stabilization performance of this controller with extensive simulations and during experiments with an able-bodied user.

4.1 Online Planning and velocity filter

In this section, we first describe the low-level fixed final time OCP with the high-frequency requirement in mind. Then, we compute a target final time from the patient-driven swing leg velocity, and we formulate the complete bi-level OCP with a cost on the final time at the upper level. Finally, we

[1] *i.e.* for a large class of initial conditions and target final times

describe how we modify the controller from the previous chapter for its use with the bi-level OCP.

4.1.1 Trajectory generation with fixed final time

The target frequency of the OCP solving is the same as the VG controller frequency, *i.e.* 1 kHz. For this reason, we use a reduced model of the system dynamics, namely the LIP dynamical model (2.17). Recall that it writes, for each dimension $i \in \{x, y\}$ of the horizontal plane,

$$\dot{\chi}_i = A\chi_i + Bu_i \tag{4.1}$$

with the one dimensional state $\chi_i \triangleq (c_i, \dot{c}_i)$, and the CoP command $u_i = p_i$. This yields, in two dimensions,

$$\dot{\chi} = \mathcal{A}\chi + \mathcal{B}u \tag{4.2}$$

with $\chi \triangleq (\chi_x, \chi_y)$, $\mathcal{A} = \begin{pmatrix} A & O_{2x2} \\ O_{2x2} & A \end{pmatrix}$, $\mathcal{B} = \begin{pmatrix} B & O_{2x1} \\ O_{2x1} & B \end{pmatrix}$, and $u \triangleq \begin{pmatrix} u_x \\ u_y \end{pmatrix} \in \mathcal{SP}$. For some duration $T \in \mathbb{R}^+$, we define the set of admissible controls $U_{ad}(T)$

$$U_{ad}(T) \triangleq \{u \text{ s.t. } \forall t \in [0, T],\ u(t) \in \mathcal{SP}(t)\} \tag{4.3}$$

where $\mathcal{SP}$ is the support polygon and, from some initial state χ^0 and some input $u \in U_{ad}(T)$, we denote χ^u the forward integration of (4.2) from χ^0, *s.t.*

$$\forall t \in [0, T],\ \chi^u(t) = e^{At}\chi^0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \tag{4.4}$$

To get to a simple yet insightful OCP, we only optimize for the state trajectory and control input up until the end of the current step. For this reason, the support polygon $\mathcal{SP}$, which varies with each contact phase, is constant over the optimization domain considered.

The VG controller restricts the freedom of the patient in such a way that the path of the patient-exoskeleton system's swing foot, in the back referential, is identical to the nominal path. Hence, with this framework, the swing foot location in the world frame $\mathcal{W}$ could theoretically be left free for the optimization to choose, by changing the CoM final location[2]. However,

[2]Choosing the CoM location is not equivalent to choosing the position of the back, which defines the landing location of the swing leg. However, the latter cannot directly be included in a LIP-based OCP. Choosing the CoM position at the end of the step is a surrogate, which is reasonable as the back frame is roughly at the center of the mass distribution of the patient-exoskeleton system.

the landing location of the swing foot is constrained by the kinematic limits of the exoskeleton joints, but this constraint can not be incorporated in LIP-based optimization and would require a non-trivial approximate surrogate. For this reason, we do not include the landing location of the swing foot in the optimization variables. Instead, we consider it fixed, at the location defined by the nominal trajectory $\mathcal{T}$, which respects the kinematic limits of the exoskeleton according to Section 2.2.

It is critical, for our application, to guarantee the feasibility of the next step *i.e.* the *recursive feasibility* of the OCP. Indeed, if not enforced, it would jeopardize the safety of the walking exercise, hence the safety of the patient. Classically, the recursive feasibility of LIP-based OCPs is either guaranteed with a terminal constraint on the state, or only softly guaranteed with an integral cost on some diverging quantities[3], guiding the planner toward recursively feasible trajectories [Wieber2016]. The soft guarantee is not enough in our case for safety reasons, hence we follow the terminal constraint approach. One type of such constraint is *capturability constraints*, *i.e.* a constraint ensuring that the robot will be driven to a complete rest in N steps. There is strong numerical evidence [Ciocca2017] that including a capturability constraint does not jeopardize the recursive feasibility, even when this constraint is recursively postponed into the future because the objective is to keep walking, and not to stop. However, this is not guaranteed, and we prefer not to take any risk for the safety of the patient. Thus, we include a different constraint, ensuring strict recursive feasibility of our OCP.

Designing a constraint that ensures the recursive feasibility of LIP-based OCPs requires some knowledge or assumptions about the desired motion after the optimization horizon, as studied in [Scianca2020]. In their work, they do not possess any offline trajectory to inform them about the desired plan after the horizon, hence they study several ways to estimate this plan. While they show that they manage to ensure the recursive feasibility of their OCP, their solution entails the design of a sophisticated terminal constraint. Luckily, we have an offline trajectory at our disposal to guide us, enabling us to design a simpler terminal constraint.

Assuming the desired plan is to keep walking, we design our terminal constraint as an equality constraint on the final state of the trajectory, with the nominal final state as value, *i.e.* the final state of the nominal trajectory. Indeed, the nominal trajectory satisfies the input-constrained full dynamics from its initial state to its final state in a certain time T^{nom}. If there exist

[3]According to the LIP model, minimizing the integral of any derivative of the CoM over a sufficiently long enough horizon is enough [Wieber2019]

also a trajectory satisfying the input-constrained LIP dynamics joining the same initial and final conditions over the same duration T^{nom}, then we can use the nominal final state as a constraint for our OCP. This LIP feasibility check is performed numerically by solving a QP problem at the beginning of each exercise. The nominal walking trajectory being cyclic, it is sufficient to check it over a single step.

Finally, our terminal constraint writes

$$\chi^u(T) = \chi^f \triangleq \mathcal{FK} \circ \mathcal{T}(T^{nom}) \tag{4.5}$$

with $T^{nom} \in \mathbb{R}^+$ the duration of the nominal trajectory $\mathcal{T}$. We denote Ω the set of feasible commands respecting the initial and final conditions (χ^0, χ^f)

$$\Omega(\chi^0, \chi^f, T) \triangleq \left\{ u \in U_{ad}(T),\ \chi^u(0) = \chi^0,\ \chi^u(T) = \chi^f \right\} \tag{4.6}$$

and the fixed final time OCP writes[4]

Problem 4.1. *Given χ^0, χ^f and T, find u^{opt}* s.t.

$$u^{opt} = \underset{u \in \Omega(\chi^0, \chi^f, T)}{\arg\min} \int_0^T u^2 dt$$

This OCP constitutes the lower level of our bi-level strategy.

4.1.2 Bi-level trajectory generation with free final time

The patient's efforts directly affect the instantaneous swing leg velocity through (3.14). At each time, the swing leg velocity equivalently defines a forecasted step duration T as follows

$$T = \frac{L_{max} - \sigma}{\dot{\sigma}} \tag{4.7}$$

Imposing (4.7) as the final time of Problem 4.1 would synchronize the support and swing leg trajectories. However, Problem 4.1 may be unfeasible for some values of T. Therefore, we search for the closest step duration for which Problem 4.1 has a solution.

The patient desired velocity is thus only a target velocity from now on and denoted $\dot{\sigma}^t$. It is directly computed from (3.14), saturated to be strictly positive, *s.t.*

$$\dot{\sigma}^t = \max \left(\epsilon,\ \frac{T(\sigma)^\top \left[K_p^{sw}(q^{sw} - P(\sigma)) + K_d^{sw} \dot{q}^{sw} \right]}{T(\sigma)^T K_d^{sw} T(\sigma)} \right) \tag{4.8}$$

[4]The presented quadratic cost function can easily be changed to incorporate extra tuning parameters to affect performance, without loss of generality.

with $\epsilon \in \mathbb{R}^{+*}$ a small parameter. The target velocity is converted into a target time T^t until the end of the current step as follows

$$T^t = \frac{L_{max} - \sigma}{\dot{\sigma}^t} \tag{4.9}$$

Problem 4.1 has solutions only for durations in $\mathbb{T}(\chi^0, \chi^f) \subset \mathbb{R}^+$, the set of feasible times for which Ω is not empty

$$\mathbb{T}(\chi^0, \chi^f) \triangleq \{T > 0,\ \Omega(\chi^0, \chi^f, T) \neq \emptyset\} \tag{4.10}$$

However, there is no guarantee that a given T^t is in $\mathbb{T}(\chi^0, \chi^f)$. For this reason, we include the target time T^t in the upper-level cost of a bi-level OCP as follows

Problem 4.2. *Given χ^0, χ^f and T^t, find u^{opt} and T^{opt} as*

$$T^{opt} = \underset{T \in \mathbb{T}(\chi^0, \chi^f)}{\arg\min}\ ||T - T^t||^2$$
$$s.t.\ u^{opt} = \underset{u \in \Omega(\chi^0, \chi^f, T)}{\arg\min} \int_0^T u^2 dt$$

Problem 4.2 is a minimal time problem for input-constrained linear dynamics of dimension 4. The solutions T^{opt} and u^{opt} are determined using a bisection method on the final time T, granting high-numerical efficiency. We implement this algorithm at 1 kHz.

Bisection resolution on the feasibility: In details, at time t_k, given an initial condition χ^0, a target time $T_k^t > 0$, and an initial guess $T^0 \in \mathbb{T}(\chi^0, \chi^f)$, we solve Problem 4.2 using bisection between T_k^t and T^0 on the feasibility function[5] of Problem 4.1 (the lower-level of Problem 4.2). Classically, the search interval is reduced by a factor 2^N using bisection, where N is the maximum number of iterations (typically 10). We obtain the solution u_k^{opt}, defined over $T_k^{opt} \in \mathbb{T}(\chi^0, \chi^f)$. Then, we update the initial condition χ^0 and initial guess T^0 using the optimal solution (u_k^{opt}, T_k^{opt}) as follows

$$\chi^0 = \chi^{u_k^{opt}}(dt)\ , \quad T^0 = T_k^{opt} - dt \tag{4.11}$$

where $dt = t_{k+1} - t_k > 0$ is the duration between two control loop iterations. We repeat this procedure until $T_0 \leq 0$.

[5]We treat the feasibility function as a boolean.

For a given $T \in \mathbb{T}(\chi^0, \chi^f)$, determining u^{opt} solution of Problem 4.1 is a fixed horizon input constrained LTI problem, which can be readily solved numerically because it is convex. A more challenging point is the description of the set $\mathbb{T}$ constraining Problem 4.2. It is the subject of Section 4.2, where we leverage our main result to characterize the nature of $\mathbb{T}$ and conclude on the optimality of the solution found using bisection.

Before detailing in Section 4.2 how we solve Problem 4.2, we describe how we use T^{opt}, to filter the patient target velocity $\dot{\sigma}^t$, and u^{opt}, to define the support leg targets.

4.1.3 Modification of the rehabilitation controller

Two modifications of the rehabilitation described in Chapter 3 are necessary for using the solution of Problem 4.2.

First, the reference trajectory (ξ^*, p^*) tracked by the admittance-based stabilizer, previously computed by the time rescaling of the nominal trajectory $\mathcal{T}$, must be replaced by the optimal reference trajectory (ξ^{opt}, p^{OP}) $s.t.$

$$\xi^* = \xi^{opt}, \quad p^* = p^{OP} \tag{4.12}$$

with $p^{OP} = u^{opt}$ and ξ^{opt} readily computed from $x^{OP} = \chi^{u^{opt}}$.

Then, the swing leg schedule must be replaced by the optimal schedule σ^{opt} computed from the optimal time T^{opt} as follows

$$\dot{\sigma}^{opt} \triangleq \frac{L_{max} - \sigma^{opt}(\tau)}{T^{opt}(\tau)} \tag{4.13}$$

with σ^{opt} recursively computed from the optimal velocity $\dot{\sigma}^{opt}$ by numerically integration. The derivation of the controller that we propose in this Chapter is now complete. It is schematically depicted on Figure 4.1, with the Optimal Planning (OP) block detailed in Algorithm 1.

This OP strategy induces two different changes. First, the reference trajectory (ξ^*, p^*) is guaranteed to respect the input-constrained LIP dynamics. This is illustrated on Figure 4.2a, where, as opposed to Figure 3.19, the CoP reference trajectory p_y^{OP} is entirely contained in the support polygon (black horizontal lines), and the CoM trajectory c_y^{opt} is offsetted toward the $y = 0.0$ line. As a result, the endpoint of the forward integration of the full-state dynamics (2.10) (the endpoint of the red line) is close to the nominal final state (the blue cross): the state is successfully driven to the final state using the stabilization controlled Section 3.2. This recursively ensures the success of the walk. Second, the simulated target velocity $\dot{\sigma}^t$ is filtered by the OP

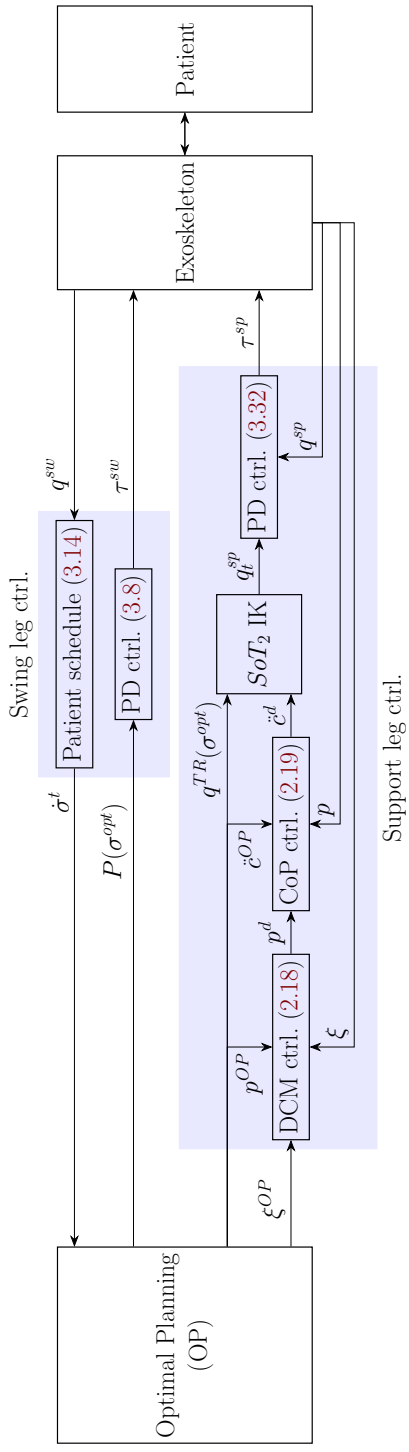


Figure 4.1: Split-leg rehabilitation controller using the careful OP synchronization strategy.

Algorithm 1 Optimal Planning (OP)

Require: χ^0
Require: $T_0 \in \mathbb{T}(\chi^0, \chi^f)$
  while $T_0 > 0$ **do**
    $T_k^t \leftarrow \frac{L_{max} - \sigma_{k-1}^{opt}}{\dot{\sigma}_k^t}$ ▷ Convert target velocity to target time
    $T_k^{opt} = \underset{T \in \mathbb{T}(\chi^0, \chi^f)}{\arg\min} \; ||T - T_k^t||^2$ ▷ Bisection on QP feasibility function
      s.t. $u_k^{opt} = \underset{u \in \Omega(\chi^0, \chi^f, T)}{\arg\min} \int_0^T u^2 dt$ ▷ Solved as a QP
    $\dot{\sigma}_k^{opt} \leftarrow \frac{L_{max} - \sigma_{k-1}^{opt}}{T_k^{opt}}$ ▷ Convert optimal time to optimal velocity
    $\sigma_k^{opt} \leftarrow \sigma_{k-1}^{opt} + \dot{\sigma}_k^{opt} dt$ ▷ Integrate optimal velocity over dt
    $\chi^0 \leftarrow \chi^{u_k^{opt}}(dt)$ ▷ Update current state
    $T^0 \leftarrow T_k^{opt} - dt$ ▷ Update current step duration
    output $(\sigma_k^{opt}, u_k^{opt}(dt), \chi^0)$ ▷ Output toward VG and admittance
  end while

strategy. This is illustrated on Figure 4.2b, where the optimal velocity $\dot{\sigma}^{opt}$ is identical to the target velocity $\dot{\sigma}^t$ until 1.1 s, *i.e.* the target time T^t is in the set of feasible times $\mathbb{T}(\chi^0, \chi^f)$. After 1.1 s, the optimal velocity $\dot{\sigma}^{opt}$ is different from $\dot{\sigma}^t$, and the optimization finds the duration $T^{opt} \in \mathbb{T}(\chi^0, \chi^f)$ closest to T^t. In the next section, we show our bisection almost always finds the optimal feasible final time.

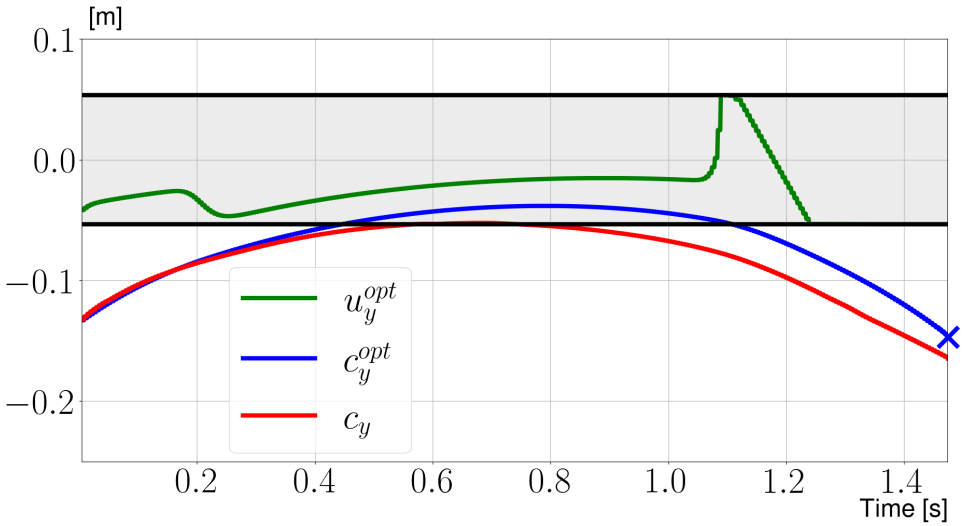
4.2 Bisection on trajectory final time

The support polygon $\mathcal{SP}$ is constant over the optimization horizon and equal to the footprint of the support foot, which is assumed to be rectangular, *s.t.*

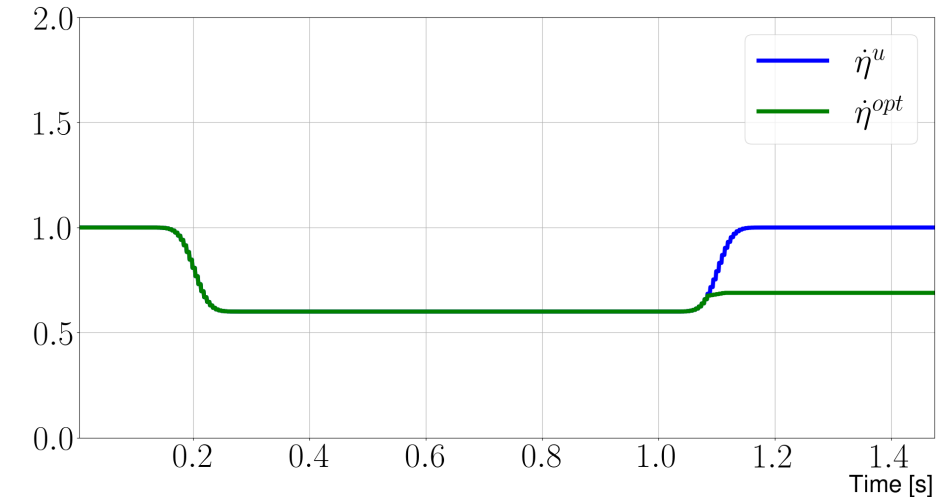
$$u \in \mathcal{SP} \triangleq [u_{mx}, u_{Mx}] \times [u_{my}, u_{My}] \tag{4.14}$$

with $u_{mx} < u_{Mx}$ and $u_{my} < u_{My}$. Since the local frame is defined by the support foot, this yields that the x and y dynamics of the LIP model (2.17) are *decoupled*. Therefore, with the trajectory duration T fixed, the set of feasible commands is also decoupled, *i.e.*

$$\Omega(\chi^0, \chi^f, T) = \Omega_x(\chi_x^0, \chi_x^f, T) \times \Omega_y(\chi_y^0, \chi_y^f, T) \tag{4.15}$$



(a) Reference CoP u_y^{OP}, CoM c_y^{OP} and measured CoM c_y positions along the Y axis of the inertial frame $\mathcal{W}$. Black horizontal lines represent the support polygon limits on the Y axis.



(b) Simulated target velocity $\dot{\sigma}^t$ (with $t_\eta = 0.2\,\text{s}$, $\Delta\eta = 0.9\,\text{s}$ and $\bar{\dot{\eta}} = 0.6$) and optimal velocity $\dot{\sigma}^{opt}$.

Figure 4.2: Reference quantities computed using the OP strategy and a simulated target velocity. The support polygon limits are not violated using the OP strategy, the CoM reference is successfully tracked, and the simulated target velocity is followed as much as possible.

with

$$\Omega_i(\chi_i^0, \chi_i^f, T) \triangleq \left\{ u_i \in U_{adi}(T),\ \chi_i^{u_i}(0) = \chi_i^0,\ \chi_i^{u_i}(T) = \chi_i^f \right\} \tag{4.16}$$

and

$$U_{adi}(T) \triangleq \{u_i \text{ s.t. } \forall t \in [0,T],\ u_i(t) \in [u_{mi}, u_{Mi}]\} \tag{4.17}$$

for $i \in \{x, y\}$.

Hence, the sole link between the sets of feasible commands in x and y dimensions is the trajectory duration T, which has to belong to both sets of feasible times for trajectories to exist with this duration, *i.e.*

$$\mathbb{T}(\chi^0, \chi^f) = \mathbb{T}_x(\chi_x^0, \chi_x^f) \cap \mathbb{T}_y(\chi_y^0, \chi_y^f) \tag{4.18}$$

In the next section, we characterize the set $\mathbb{T}$.

4.2.1 Set of admissible final times

In this section, we characterize the sets $\mathbb{T}_x$ and $\mathbb{T}_y$. Since the analysis is independent of the direction, we use the notation $\chi_\bullet$ (resp. $u_\bullet$) to denote χ_x or χ_y (resp. u_x or u_y).

Preliminary results on minimum and maximum time solutions

Lemma 1. For all $(\chi_\bullet^0, \chi_\bullet^f) \in \mathbb{R}^4$ and $T > 0$, if there exists a solution $u_\bullet \in \Omega(\chi_\bullet^0, \chi_\bullet^f, T)$, then a minimum time solution (noted $u_{\min}$) always exists and, when the set $\mathbb{T}$ is upper-bounded, a maximum time solution (noted $u_{\max}$) exists. They are global optima.

Proof. Equation (A.2) is linear, and U_{ad} is compact and convex, hence, when a solution $u_\bullet \in \Omega(\chi_\bullet^0, \chi_\bullet^f, T)$ exists, a minimum time solution $u_{\min}$ exists from [Liberzon2012, Theorem 4.3].

When the set $\mathbb{T}$ is upper-bounded, we note $\overline{T}$ its supremum. Given a sequence $(T_k, u_{\bullet k})$ s.t. $\lim_{k\to\infty} T_k = \overline{T}$, consider the sequence $(\overline{T}, \tilde{u}_{\bullet k})$ of prolonged $u_{\bullet k}$ on $[T_k, \overline{T}]$ by the null function, then the proof provided in [Liberzon2012, Theorem 4.3] is straightforwardly extended to the $\tilde{u}_{\bullet k}$ sequence, yielding the existence of $u_{\max}$. Hence, $\overline{T}$ is maximum. □

We denote

$$\begin{aligned} (T_{\min}, u_{\min}) &\triangleq \arg\min_{u \in \Omega(\chi_\bullet^0, \chi_\bullet^f, T>0)} T, \qquad (T_{\max}, u_{\max}) \triangleq \arg\min_{u \in \Omega(\chi_\bullet^0, \chi_\bullet^f, T>0)} -T \end{aligned} \tag{4.19}$$

The following Theorem 1 fully characterizes the sets $\mathbb{T}_{x,y}$ and is instrumental to efficiently solve the bi-level OCP.

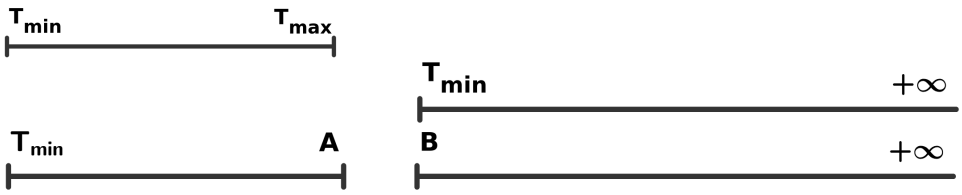


Figure 4.3: Illustration of the main result, Theorem 1.

Theorem 1 (Description of $\mathbb{T}$)**.** The set of feasible times $\mathbb{T}(\chi^0_\cdot, \chi^f_\cdot)$ is either empty, or of the form $[T_{\min}, T_{\max}]$, or of the form $[T_{\min}, +\infty[$, or of the form $[T_{\min}, A] \cup [B, +\infty[$, $A < B$.

Proof. The proof of Theorem 1 is provided in Appendix A. □

Theorem 1 reveals that $\mathbb{T}_\cdot$ is composed of one or two intervals in each of x and y dimensions. This result also holds for the intersection of the two sets, hence it holds for $\mathbb{T} = \mathbb{T}_x \cap \mathbb{T}_y$.

This is instrumental for numerically solving Problem 4.2. Knowing that $\mathbb{T}$ is composed of one or two intervals, the solution is simply the projection of T^t onto them. As detailed in Section 4.2.2, numerical evidence hints toward $\mathbb{T}$ being composed of a single interval in our practical case, therefore the projection is readily obtained by a bisection method applied to the feasibility function of a quadratic program.

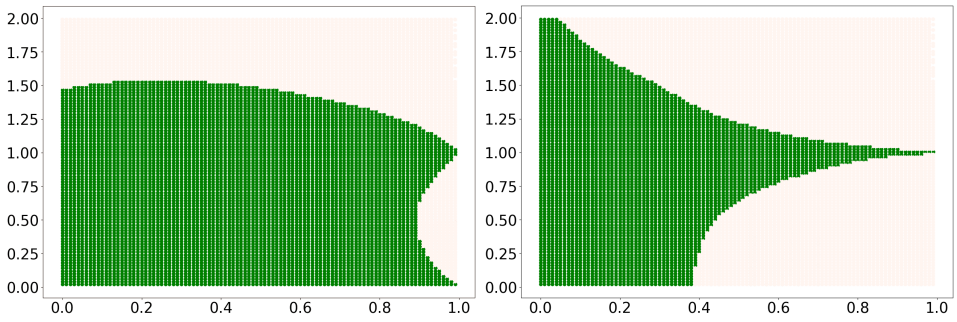
4.2.2 Numerical method

Theorem 1 describes $\mathbb{T}$. We notice, numerically, that $\mathbb{T}$ is most often a single interval for the sets of initial conditions and target time typically encountered in walking rehabilitation, which enables us to use bisection to solve Problem 4.2. To compute $\mathbb{T}$, we use the nominal terminal condition χ^f, which is the actual terminal condition set in Problem 4.2, and a time-varying initial condition χ^t, computed from the nominal trajectory $\mathcal{T}$ as follows

$$\chi^t = \mathcal{FK} \circ \mathcal{T}(t) \tag{4.20}$$

In order to ease the visualization of $\mathbb{T}$, we rather compute the set of feasible velocities $\mathbb{V}$ defined according to (4.9) as follows

$$\mathbb{V}(\chi^t, \chi^f) = \left\{ \frac{L_{max} - \mathrm{s}(t)}{T}, \quad T \in \mathbb{T}(\chi^t, \chi^f) \right\} \tag{4.21}$$



(a) x dimension (b) y dimension

Figure 4.4: Feasible velocities set $\mathbb{V}(\chi_{\bullet}^{t}, \chi_{\bullet}^{f})$ along the nominal CoM trajectory (computed from $\mathcal{T}$). A hole appears at the end of the step for the x dimension.

Figure 4.4 displays the set of feasible velocities $\mathbb{V}$ for each dimension x and y, over time, estimated by exhaustive numerical search. Figure 4.4b shows that $\mathbb{V}_y(\chi_y^t, \chi_y^f)$ is an interval for all initial conditions χ^t (varying with time). However, Figure 4.4b reveals that $\mathbb{V}_x(\chi_x^t, \chi_x^f)$ is the union of two intervals for initial conditions χ^t evaluated at times near the end of the nominal trajectory duration. This numerically illustrates our main Theorem 1, as it states that $\mathbb{T}$ can be the union of two intervals.

Nonetheless, Figure 4.5, which depicts the intersection of $\mathbb{V}_x$ and $\mathbb{V}_y$, *i.e.* $\mathbb{V}(\chi^t, \chi^0)$ according to (4.18), reveals that $\mathbb{V}$ is actually always a single interval along the nominal CoM trajectory. This does not guarantee that $\mathbb{V}$ is always a single interval along all possible CoM trajectories, but of all the numerical exhaustive searches that we have performed along various CoM trajectories (recursively defined through the optimal solutions of Problem 4.2 with various target velocity profiles $\dot{\eta}^t$), none revealed $\mathbb{V}$ being the union of two intervals.

However, if, for any set of initial and final condition (χ^0, χ^f), the set of feasible times $\mathbb{T}$ becomes the union of two intervals, then only the optimality of the upper-level of Problem 4.2 would be lost (*i.e.* the bisection method would return a feasible duration $T \neq T^{opt}$). The solution u^{opt}, by-product of the bisection method, would still minimize Problem 4.1, and therefore, respect the input-constrained LIP dynamics and the two boundary conditions (χ^0, χ^f). Thus, the trajectory would still be safe for the robot to stabilize around, and only the optimality of the rehabilitation objective would be lost.

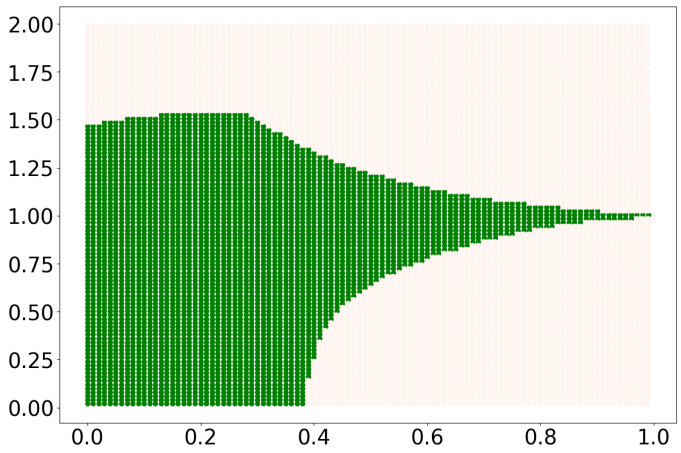


Figure 4.5: Feasible velocities set $\mathbb{V}(\chi^t, \chi^f)$ along the nominal CoM trajectory (computed from $\mathcal{T}$). The hole of the x dimension is filtered out by the intersection with the y dimension.

QP resolution and feasibility check

Problem 4.1 defines a fixed-time OCP that can be addressed using a direct numerical method. Conveniently, the input signal is represented by a piecewise $\mathcal{C}^1$ function in between non-uniform nodes. The dynamics and the value of the integral cost are exactly represented using the first-order hold quadrature. This allows the expression of boundary conditions and input constraints under an affine form in a finite number of variables, and the cost as a quadratic function of these variables. The same discretization procedure is employed in the $x - y$ directions, resulting in a QP with $4P + 4$ variables and $4P + 4$ affine constraints. The outcome of the QP resolution is a feasibility boolean, and, when it is feasible, a solution to Problem 4.1.

Towards real-time implementation

Typical numerical setups consider $2P + 2 = 10$ variables, with $N = 10$ maximum number of iterations. The employed software is a streamlined implementation of the positive definite QP dual algorithm from [Goldfarb1983] specifically coded in `C++` for this application to minimize any overheads. The problem is treated as dense. Mean CPU times on a laptop with turboboost disabled (*s.t.* the CPU frequency is similar to the embedded computer of Atalante) is 0.2 ms, varying between 0.07 ms and 0.75 ms, which are lower than the 10 ms reported in [Caron2017] and the 100 ms reported in [Pon-

ton2021] where similar online planning problems are addressed. They are also lower or equal to those reported in [Fernbach2020; Caron2016] where fixed-time online planning problems are solved. These computation times are in line with the target 1 kHz frequency of the OCP solving.

4.2.3 Simulation results

In this section, we first illustrate the performance of our bisection search of the optimal trajectory duration with a highly varying simulated patient input. Then, we study the increase of stability offered by the use of the OP strategy instead of the TR strategy using extensive full-body simulations of the patient-exoskeleton system under a variety of simulated patient inputs.

Results for highly varying target velocity profile

We first perform a simulation with a strongly varying target velocity profile $\dot{\sigma}^t$. Similarly to Section 3.3.1, we generate this target velocity profile $\dot{\sigma}^t$ from an arbitrarily tailored function η^t as follows

$$\sigma^t(t) = \mathrm{s} \circ \eta^t(t) \tag{4.22}$$

A strongly oscillating velocity $\dot{\eta}^t$ along the geometric path is considered to simulate the behavior of a highly demanding patient. The nominal velocity is 1 and the variations are $\pm 50\%$. This defines a signal $t \mapsto T^t(t)$ from (4.9) and the curvilinear abscissa s. For reference, the set of feasible velocities $\mathbb{V}(\chi^{u^{opt}}, \chi^f)$ is estimated by exhaustive numerical search at each instant. Figure 4.6 depicts the corresponding target velocity $\dot{\eta}^t$ and optimal velocity $\dot{\eta}^{opt} \triangleq \frac{d}{dt}\, \mathrm{s}^{-1} \circ \sigma^{opt}$. As is visible in Figure 4.6, the resolution of Problem 4.2 satisfies the simulated patient's intent at the beginning of the simulation. Gradually the feasible set gets more stringent and at some point, near $t = 0.4\,\mathrm{s}$, the proposed algorithm has to intervene. The desired time T^t is no longer feasible on many occasions. The situation worsens until the end of the simulation. Notably, in the end, the walking velocity has to be sped up significantly. This comes from the numerical discretization of the horizon of QP problem solving Problem 4.1.

Results on full-body simulations

Extensive closed-loop rigid-body simulations of the patient-exoskeleton system have been performed to evaluate the stability increase of the walking exercises offered by the use of the OP strategy (in particular *w.r.t.* the TR

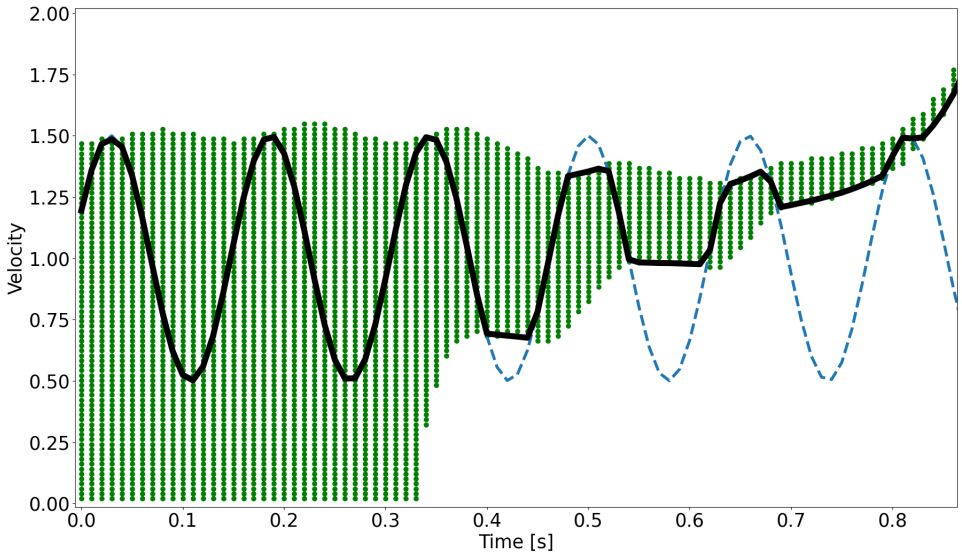


Figure 4.6: Velocity of the trajectory (1 is the nominal velocity). (Dotted blue): target velocity $\dot{\eta}^t$. (Green dots): set $\mathbb{T}$ determined by an exhaustive search, for reference. (Solid black): optimal velocity $\dot{\eta}^{opt}$ from the OP strategy.

strategy). To simulate the behavior of the patient, we consider the same smoothed piecewise velocity signals as the ones described in Section 3.3.1. Figure 4.7 reports the results (for each duration $\Delta\tau$ and magnitude $\bar{\dot{\eta}}^t$, a vast list of possible starting times t_η for the square disturbance is considered, and we report the success rate). Figure 4.7 (left) reports simulation results obtained with the controller derived in Chapter 3, hence using the TR strategy, while Figure 4.7 (right) reports the results obtained using the same controller with the modifications presented in this chapter, hence using the OP strategy. Again, a simulation is considered stable if the simulated patient-exoskeleton system walks for at least 10 s without falling.

These results show the substantial improvement in the stability of the walking exercises provided by the use of OP strategy over the TR one. In details, a total of 2917 simulations have been conducted. In total, less than 8% of considered velocity profiles are failing with the OP strategy, while more than 30% were with the TR strategy. Therefore, safety is ensured for almost all considered cases, except for some very low-velocity cases with

[7]The white spaces in this figure correspond to unfeasible values of the parameters violating the constraint $\sigma < L_{max}$.

long durations. A careful examination of simulation results reveals that fall occurs mostly when the slow-down takes place at the late stages of the step.

4.3 Experimental results

Two types of experiments are conducted. First, we compare the stability performance of the two strategies and conclude on the vast superiority of the OP strategy. Then, we explore the performance of the OP strategy in terms of compliance with the patient schedule.

4.3.1 Stability comparison

We perform extensive stability comparisons as in Section 3.3.3 by replacing the patient with a dummy and simulating the behavior of the patient as described in Section 3.3.2. An experiment consists of walking 10 steps in a straight line. A practitioner keeps hold of the two lateral exoskeleton handles and is allowed to create an effort with one finger on each hand only. This creates a very low upper-bound on the external forces[8].

The reported results on Figure 4.8 show a great safety improvement offered by the OP strategy in the low-velocity range, below 90% of nominal velocity, compared to the TR strategy. They stress that, using the OP strategy, the proposed controller is completely preserving the balance of the system for velocities about as low as 70% of the nominal velocity, and velocities as low as 50% of the nominal velocity provided the change duration is lower or equal to 300 ms. These results are in complete alignment with the stability results obtained in simulation and reported in Section 4.2.3.

In addition, the mean computation time during these experiments is 0.039 ms, and varies between 0.22 ms and 0.11 ms, which validates the efficiency of our bisection-based solver (running on a i7-1185G7E CPU at fixed 1.8 GHz frequency).

[8]Stabilization of patients walking with Atalante is still requiring some very low external efforts, probably due to the inaccuracy of the patient model and/or their loose fastening to the exoskeleton for safety reasons. Able-bodied users doing walking rehabilitation can walk fully autonomously by moving their upper body to help maintain balance. The proposed setup somewhat emulates this.

[9]The white spaces in this figure corresponds to unfeasible values of the parameters violating the constraint $\sigma^* \leq L_{max}$.

4.3.2 Rehabilitation: experiments with an able-bodied user

We report below the results of two walking experiments with an able-bodied user using the proposed controller with the OP strategy:

- a 10-steps walking experiment, during which the user drives the walking velocity between 25% and 150% of the nominal walking velocity;
- a single-step walking experiment, during which the user brings the walking velocity to 0 for 2 seconds, then resumes walking.

10-steps walking experiment

We first report a single step velocity (on Figure 4.9) and CoM trajectory (on Figure 4.10). Figure 4.9 reveals how the OP strategy accounts for the unilateral contact constraint by projecting the target velocity on the set of feasible velocities. In detail, during the first $450ms$ of the single support phase, the OP strategy leaves the target velocity unchanged because the solution of Problem 4.2 is $T^{opt} = T^t$ ($\dot{\sigma}^{opt}$ in green completely overlaps $\dot{\sigma}^t$ in blue). Hence, the patient's desire is fully respected. During the remaining 500 ms, the OP strategy starts filtering the target velocity to preserve the balance of the system, $\sigma^{opt} \neq \dot{\sigma}^t$. Gradually, σ^{opt} is constrained around 71% normalized velocity.

The OP strategy also wisely chooses the CoM reference trajectory and satisfies the final state constraint. Figure 4.10 shows the final state constraint is satisfied by both the replanned (OP) and time-rescaled (TR) CoM trajectories' endpoints. However, the CoM trajectory computed with the OP strategy is very different from the one with TR strategy, and, in particular, exhibits a minimum on the Y axis 2.5 cm closer to the support foot (centered at 0.0 cm) than the nominal trajectory: the exoskeleton sways its hip toward the support foot to accommodate for the user's low-velocity desire while remaining stable with the OP strategy, while the TR strategy only accommodates for the user desire without taking care of the balance of the system. The OP strategy's behavior is consistent with human behavior.

Finally, Figure 4.11 shows $\dot{\sigma}^{opt}$ over the whole experiment, with double support areas in grey. During this experiment, the user varies the level of effort produced by their legs during the single support phases. Note that (3.14) is only used during these phases while the user's desire is ignored during double support phases[10].

[10] More precisely, the reference trajectory used during double support is computed once, at the beginning of the step. For this, we use the OP strategy and the mean velocity of the previous step, for the sake of continuity.

Mid-step complete stop experiment

We report a single step velocity (on Figure 4.12) and CoM trajectory (on Figure 4.13). Figure 4.12 reveals how the OP strategy accounts for the unilateral contact constraint by letting the user drive the walking velocity to almost 0% of the nominal walking velocity for approximately 2 seconds, when possible, then projects the target velocity on the set of feasible velocities by limiting the target slow-down during the late part of the step to roughly 40% of the nominal walking velocity.

The CoM trajectory computed with OP strategy, reported on Figure 4.13, differs even more from the one with TR strategy than reported previously. In particular, it exhibits a minimum on the Y axis 5 cm closer to the support foot (centered at 0.0 cm) than the nominal trajectory: the exoskeleton sways its hip toward the support foot twice more than during the previous experiment to accommodate for the user desire to stop complete walking in the middle of the step. This illustrative experiment reveals the great adaptability to the user desire offered by the OP strategy.

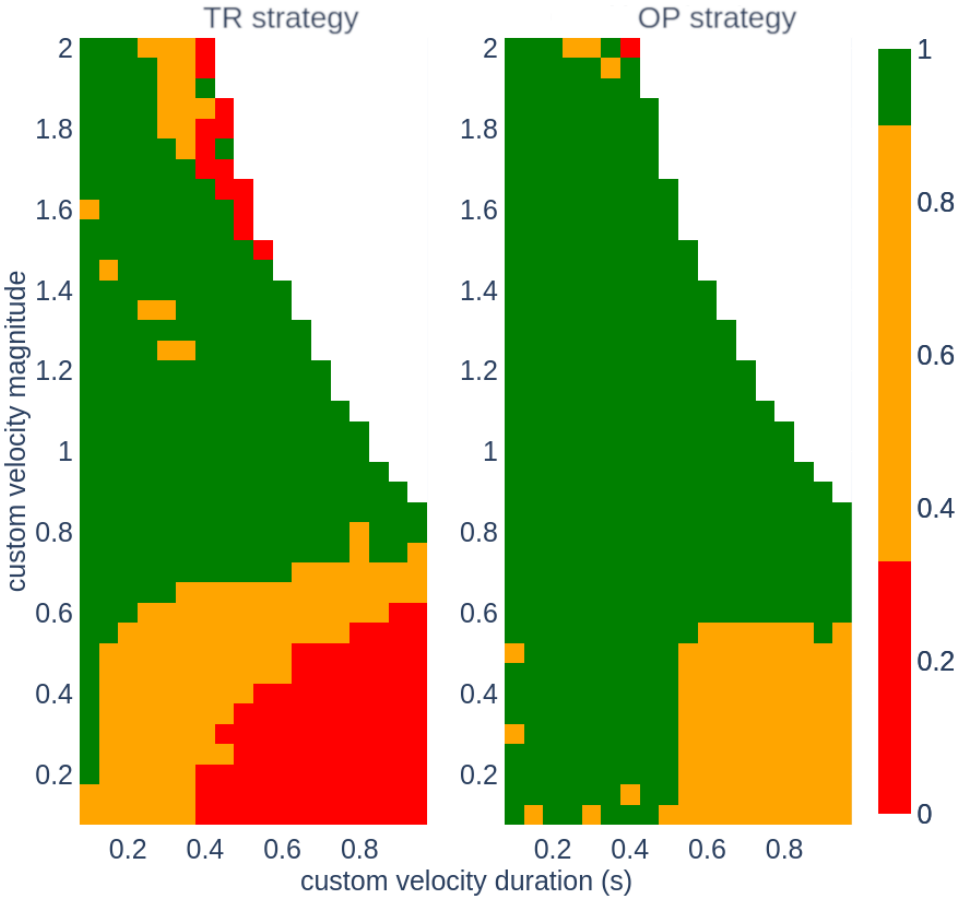


Figure 4.7: Comparison of rate of success heatmaps for velocity variations having various durations and magnitudes[7]. The OP strategy increases the stabilization performance in the low-velocity range (*w.r.t.* the TR strategy). Left: TR strategy. Right: OP strategy.

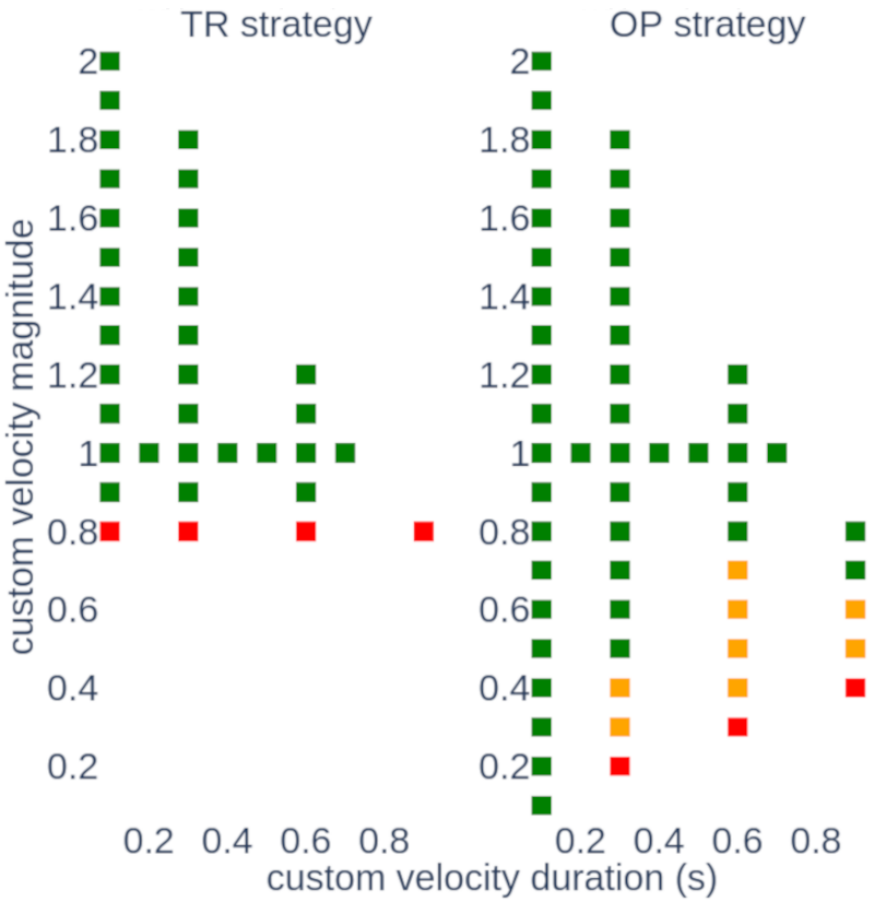


Figure 4.8: Comparison of experimental stability for velocity variations having various durations and magnitudes[9]. Experimental results are in accordance with the simulations, with lower performance in the low-velocity range. Green: stable without external help. Orange: stable with practitioner help. Red: unstable. Left: TR strategy. Right: OP strategy.

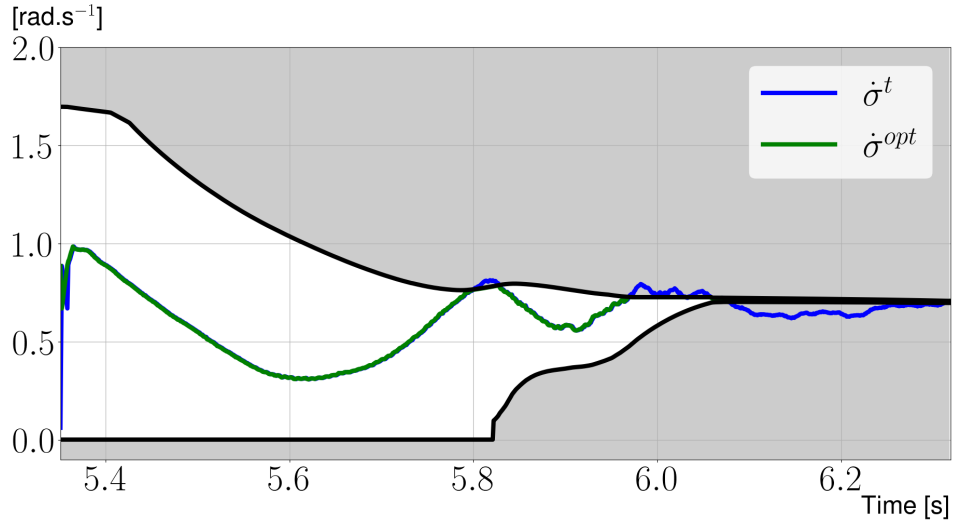


Figure 4.9: Effective velocity $\dot{\sigma}^{opt}$ and target velocity $\dot{\sigma}^t$ over a step with an able-bodied user (step 6 of Figure 4.11). Black curves: lower and upper limits of the set of feasible velocities.

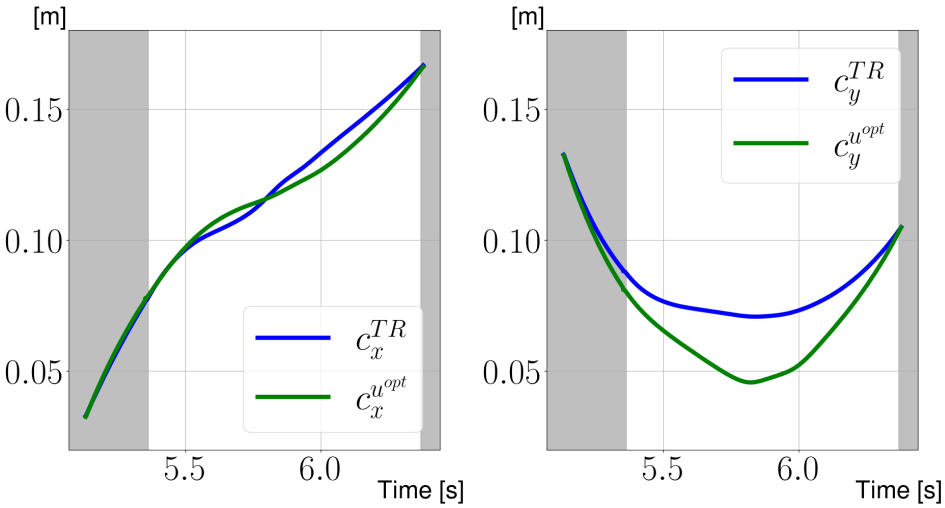


Figure 4.10: CoM from TR and OP strategies over a step with an able-bodied user (step 6 of Figure 4.11). Grey areas: double support phases. White areas: single support phases. Left: X axis. Right: Y axis.

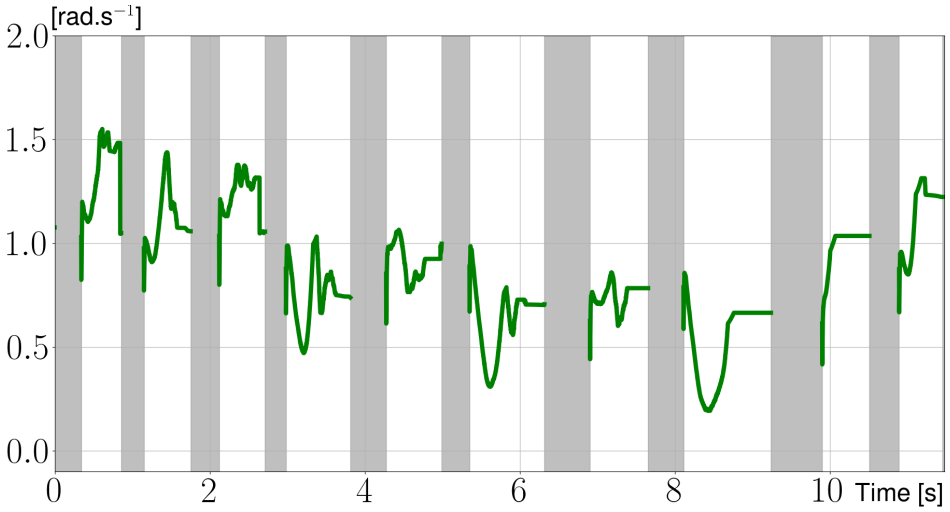


Figure 4.11: Effective velocity $\dot{\sigma}^{opt}$ over a 10-steps experiment with an able-bodied user. Grey areas: double support phases. White areas: single support phases.

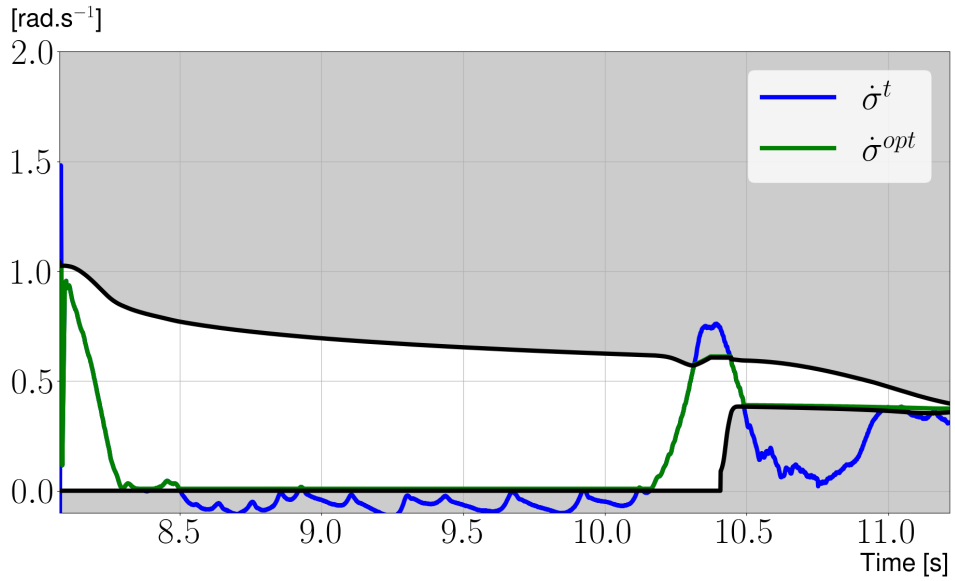


Figure 4.12: Effective velocity $\dot{\sigma}^{opt}$ and target velocity $\dot{\sigma}^t$ over a step comprising a full stop, with an able-bodied user. Black curves: lower and upper limits of the set of feasible velocities.

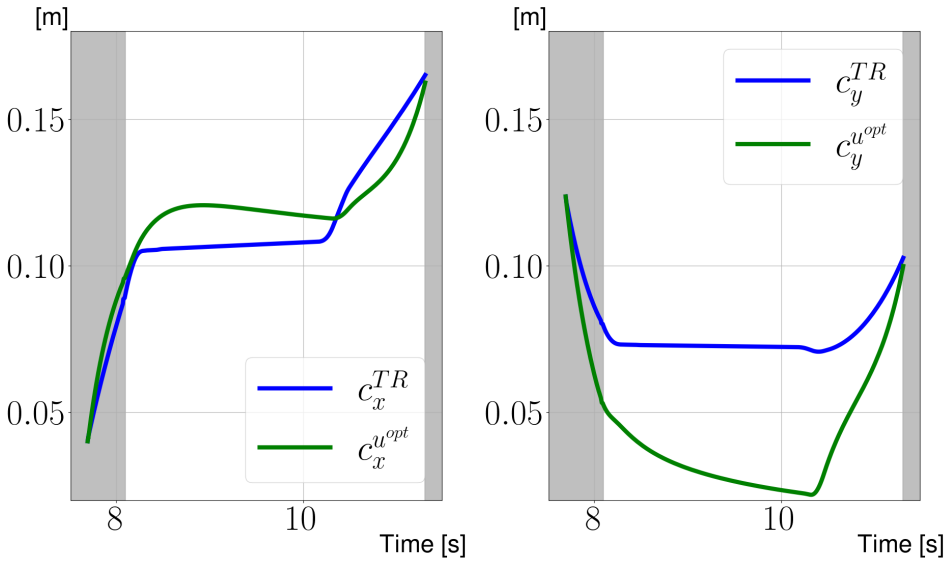


Figure 4.13: CoM from TR and OP strategies over a step comprising a full stop, with an able-bodied user. Grey areas: double support phases. White areas: single support phases. Left: X axis. Right: Y axis.

Chapter 5

Discussion and perspectives

Chapitre 5 - Discussion et perspectives: *La contribution principale présentée dans ce manuscrit est un nouveau contrôleur de rééducation active à la marche qui laisse le patient modifier la vitesse d'exécution du mouvement, tout en limitant son désire pour respecter la dynamique du système et assurer sa sécurité. Dans ce chapitre, les performances quantitatives de ce contrôleur sont rappelées et illustrées dans deux vidéos, tandis que ses limitations et pistes d'amélioration sont discutées.*

The proposed control methodology updates the existing controller of Atalante in two ways: on one hand, the walking velocity is relaxed according to the patient's willingness and capacity to participate in the motion, enabling walking rehabilitation with Atalante. On the other hand, the patient schedule is monitored and followed at best while respecting the dynamics of the system, guaranteeing the safety of the walking exercises. The proposed controller is the first safe walking rehabilitation controller designed for self-balanced exoskeletons and the main contribution of the thesis.

Thesis outcome

Using the presented *empower and monitor* control methodology, we have been able to modulate by $\pm 50\%$ the temporal execution of two straight-line walking trajectories of practical interest: the simple flat-foot trajectory, at an average translational velocity of 15 cm/s, and the more anthropomorphic foot-rolling trajectory, at an average translational velocity of 40 cm/s. Videos obtained with an able-bodied user are given below:

- Experiments with the flat-foot trajectory: https://youtu.be/_1A-2nLy5ZE.
- Experiments with the foot-rolling trajectory: https://youtu.be/ZKOouUXfTvw.

Limitations and possible extensions

We have explored the possibility of offering temporal freedom to the user with the self-balanced exoskeleton Atalante. The obtained results are encouraging. However, the clinical evaluation of the proposed control methodology remains to be done. Among the various metrics of importance, one should consider, without loss of generality, the level of restored functional ambulation ability (for instance evaluated with the Functional Ambulation Categories test), the fatigue level of physiotherapists, and the sense of safety of patients.

On the methodological side, many ways of improvement shall be considered in the future. It is possible to combine temporal freedom with spatial freedom, which has already been considered in the literature. For this purpose, one could introduce constraint relaxation in the spirit of [Gurriet2020], and/or formulate a more general OCP allowing one to include additional variables such as the location of the swing-foot landing. Additionally, a mathematical formulation could also consider the reverse execution of the trajectories as required in some repetitive physical training, or reproduce the behavioral description of the two-thirds power law [Ivanenko2002] to better fit human intent.

Besides, for improved stabilization properties, it could be of interest to consider other tasks than the considered admittance task, like foot damping control (see [Caron2019]), or reduction of the disturbances coming from the flexibilities, see [Vigne2021].

Finally, the class of trajectories under consideration could be extended. Two extensions of dominant interest are turn-in-place and slope/stairway trajectories. Interestingly, these two use cases challenge the representativeness of the LIP model for mobile/changing reference frames along trajectories and some adjustments are necessary. Two distinct paths can be followed to improve the dynamical model in Problem 4.2 and extend the proposed methodology: leverage the recent progress of online whole-body MPC (see [Dantec2022; Mastalli2022]), or compute an efficient oracle offline using machine learning (see [Castillo2021; Siekmann2021; Singh2022]).

Appendix A

Proof of Theorem 1

Contents

As detailed in Section 4.2, the x and y dynamics of the LIP model (2.17) are totally *decoupled*, and the Theorem 1 is about the nature of $\mathbb{T}(\chi_{\cdot}^{0}, \chi_{\cdot}^{f})$ in one dimension. Therefore, we conduct the proof of the theorem in one dimension and $\chi_{\cdot}$ notations in place of x or y dimension will be omitted in this appendix.

The proof of Theorem 1 is organized as follows. First, after some preliminary change of coordinates and definitions in Appendix A.1, we exhibit in Appendix A.2 particular regions of the phase portrait which serve to decompose the proof. Then, we study the boundedness of $\mathbb{T}$ in Appendix A.3. Finally, we assess its convexity properties in Appendix A.4 and Appendix A.5 by differentiating according to the boundedness of $\mathbb{T}$.

A.1 Preliminaries

A.1.1 Decoupling dimensions

To ease the analysis of the set of feasible times $\mathbb{T}(\chi^0, \chi^f)$, we perform the following change of coordinates, which diagonalizes the dynamics (2.17) with

$$x_1 = c + \frac{\dot{c}}{\omega} \qquad x_2 = -c + \frac{\dot{c}}{\omega} \tag{A.1}$$

Then, (2.17) takes the diagonal form $\dot{x} \triangleq Ax + Bu$ with

$$\dot{x} = \begin{pmatrix} \omega & 0 \\ 0 & -\omega \end{pmatrix} x + \begin{pmatrix} -\omega \\ -\omega \end{pmatrix} u \tag{A.2}$$

with

$$x \triangleq (x_1, x_2) \tag{A.3}$$

The solution of (A.2) with input u, from the initial condition $x^0 \in \mathbb{R}^2$ is denoted x^u. For all vector variables, a subscript $_1$ or $_2$ indicates the first or second coordinate. Zero-order hold of u for a duration $d = t_2 - t_1$ yields the solution

$$x^u(t_2) = \begin{pmatrix} e^{\omega d} & 0 \\ 0 & e^{-\omega d} \end{pmatrix} x^u(t_1) + \begin{pmatrix} 1 - e^{\omega d} \\ e^{-\omega d} - 1 \end{pmatrix} u \tag{A.4}$$

The change of coordinate between χ and x is trivial, *s.t.* the set of feasible times $\mathbb{T}(\chi^0, \chi^f)$ can be equivalently denoted $\mathbb{T}(x^0, x^f)$. For brevity purposes, we omit the dependence of the set of feasible times $\mathbb{T}$ on (χ^0, χ^f) when convenient.

A.1.2 Definitions

A piecewise constant control input u having N steps over an interval $[0, t_f]$ is defined using a finite (irreducible) partition

$$0 < d_1 < d_2 + d_1 < ... < d_n + ... + d_1 = t_f \tag{A.5}$$

with u taking values only in $\{u_m, u_M\}$. For convenience, it is described by its first value and the durations, e.g. for 3 steps of respective durations d_1, d_2, d_3 starting with u_m, a sequence

$$(u_m, d_1, d_2, d_3) = \text{seq} \mapsto u$$

gives $u(t) = u_m$ for $0 \leq t < d_1$, $u(t) = u_M$ for $d_1 \leq t < d_2 + d_1$, $u(t) = u_m$ for $d_2 + d_1 \leq t < d_3 + d_2 + d_1$. For any initial condition x^0, and any seq

defining a control u as detailed above, over $\tau \in [0, t_f]$ we note the solution $x^{\text{seq}} \triangleq x^u$ of the differential equation $\dot{x} = Ax + Bu$ which is

$$\phi(x^0, \text{seq}, \tau) \triangleq x^{\text{seq}}(\tau) = e^{A\tau}x^0 + \int_0^\tau e^{A(\tau-s)}Bu(s)ds$$

By extension, we define

$$\phi(x^f, \text{seq}, -\tau) \triangleq e^{-A\tau}x^f - \int_0^\tau e^{-A(\tau-s)}Bu(T-s)ds$$

We define several subsets of $\mathbb{R}^2$ as follows

$$\mathcal{D} \triangleq \{(x_1, x_2),\ x_2 = -x_1\}$$

$$\mathcal{D}^+ \triangleq \{(x_1, x_2),\ x_2 > -x_1\}\ , \quad \mathcal{D}^- \triangleq \{(x_1, x_2),\ x_2 < -x_1\}$$

and

$$\mathcal{U}_m^- \triangleq \{(x_1, x_2),\ x_1 < u_m\}\ , \quad \mathcal{U}_m^+ \triangleq \{(x_1, x_2),\ x_1 > u_m\}$$
$$\mathcal{U}_M^- \triangleq \{(x_1, x_2),\ x_1 < u_M\}\ , \quad \mathcal{U}_M^+ \triangleq \{(x_1, x_2),\ x_1 > u_M\}$$

Finally, we define two open double cones

$$\mathcal{C}_M \triangleq \{\mathcal{D}^+ \cap \mathcal{U}_M^-\} \cup \{\mathcal{D}^- \cap \mathcal{U}_M^+\}\ , \quad \mathcal{C}_m \triangleq \{\mathcal{D}^- \cap \mathcal{U}_m^+\} \cup \{\mathcal{D}^+ \cap \mathcal{U}_m^-\}$$

Next, we study the solutions of minimal and maximal time OCPs (4.19). This study stresses the role of several regions in the phase plane being key in the reachability of a target x^f from an initial condition x^0.

A.2 Regions of interest in the phase portrait

Lemma 2. The solution $u_{\min}$ is bang-bang, i.e. takes only values in $\{u_m, u_M\}$, with a maximum number of one switch. The same property holds for $u_{\max}$ when it exists.

Proof. Consider the Hamiltonian

$$H(t, \lambda^0, \lambda, x, u) = \mu + \lambda(t)(Ax + Bu)$$

Using the PMP, the adjoint equation and solution write

$$\begin{cases} \dot{\lambda} = -\frac{\partial H}{\partial x} = -\lambda(t)A \\ \lambda(t) = \lambda^0 e^{-At} \end{cases}$$

and the switching function is

$$\Gamma(t) = \lambda B = \lambda^0 e^{-At} B = -\omega \lambda^0 \begin{pmatrix} e^{-\omega t} & e^{\omega t} \end{pmatrix}^T$$

If ${\lambda^0}_1 {\lambda^0}_2 < 0$, then there exists a unique switching time $\frac{1}{2\omega} \log(-\frac{\lambda^0_1}{\lambda^0_2})$ for which Γ changes sign. Otherwise, Γ has a constant sign. This concludes the proof for $u_{\min}$. The proof regarding $u_{\max}$ is identical. □

Lemma 2 highlights the importance of the phase portrait in Figure A.1 corresponding to constant control values u_m and u_M. It is split into nine open regions, some of them being open semi-infinite strips, whose boundaries are the trajectories passing through the equilibrium points for u_m and u_M. We denote each region $\mathcal{R}_i,\ i = 1, ..., 9$. Also, we will note $\mathcal{R}_{ijk...} \triangleq \mathcal{R}_i \cup \mathcal{R}_j \cup \mathcal{R}_k \cup ...$ for any number of indexes. Notice two interesting properties: *i)* the locus of intersecting parallel arcs is $\mathcal{D}$ and *ii)* the subsets $\mathcal{R}_{147}$ and $\mathcal{R}_{369}$ are positively invariant under the controlled flow.

Next, the following result states that in the cone $\mathcal{C}_m$ (resp. $\mathcal{C}_M$), the flow corresponding to u_m (resp. u_M) reaches a point symmetric to the initial condition *w.r.t.* the line $\mathcal{D}$. This property is instrumental in the proof.

Lemma 3. For all x in the double cones $\mathcal{C}_m \cup \mathcal{C}_M$, we have

$$\phi(x, u, f(x, u)) = Sx$$

with $S = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ and $f(x, u) \triangleq \frac{1}{\omega} \log(\frac{u+x_2}{u-x_1})$ by taking $u = u_m$ if $x \in \mathcal{C}_m$ or $u = u_M$ if $x \in \mathcal{C}_M$.

Proof. In the double cones $\mathcal{C}_m$ and $\mathcal{C}_M$, f is well-defined as a function of its arguments. A direct calculus with (A.4) yields the conclusion. □

A.3 Boundedness of the set of feasible times $\mathbb{T}$

Depending on the values of x^0 and x^f, the set $\mathbb{T}$ can be empty ($\emptyset$), bounded ($\mathcal{B}d$), or unbounded (∞).

Lemma 4 (Boundedness of $\mathbb{T}$)**.** Conditions on x^0 and x^f corresponding to cases of non-empty $\mathbb{T}$ are listed in Figure A.2.

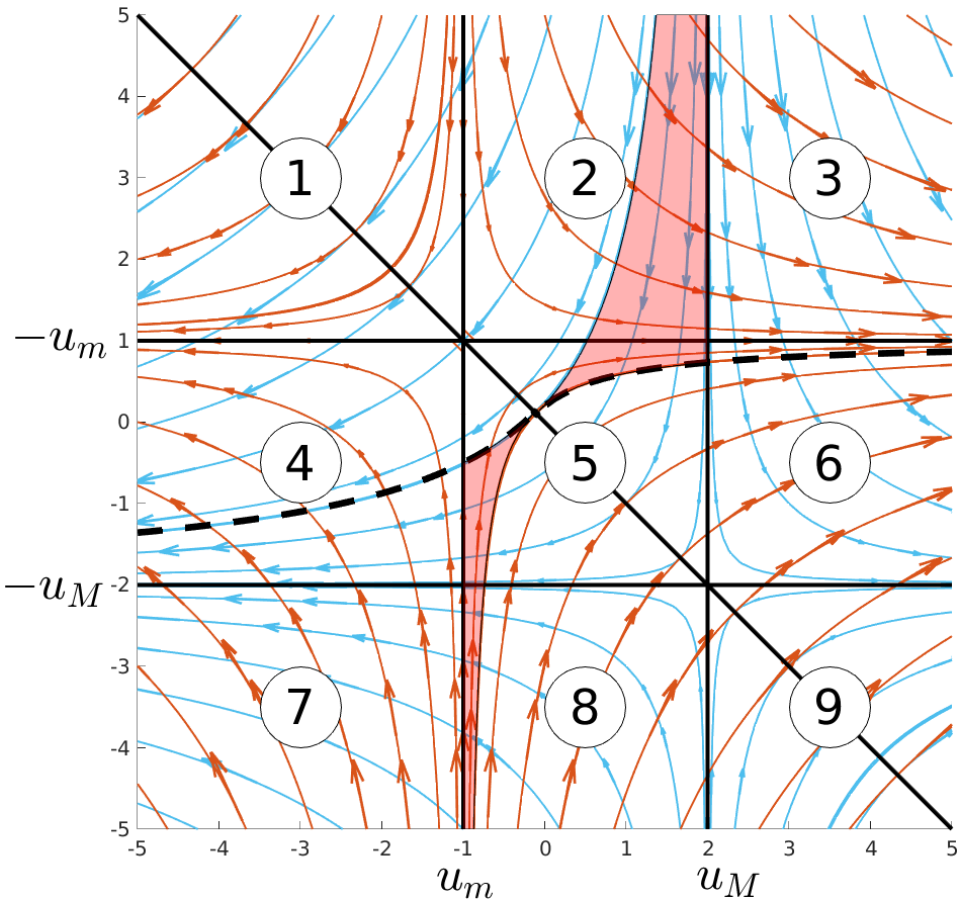


Figure A.1: Phase diagram for (A.2) with $u_m = -1$ (red) and $u_M = 2$ (blue). $\mathcal{C}_m$ covers $\mathcal{R}_1 \cap \mathcal{D}^+$, $\mathcal{R}_5 \cap \mathcal{D}^-$, $\mathcal{R}_9 \cap \mathcal{D}^-$ and $\mathcal{R}_8$. $\mathcal{C}_M$ covers $\mathcal{R}_1 \cap \mathcal{D}^+$, $\mathcal{R}_5 \cap \mathcal{D}^+$, $\mathcal{R}_9 \cap \mathcal{D}^-$ and $\mathcal{R}_2$.

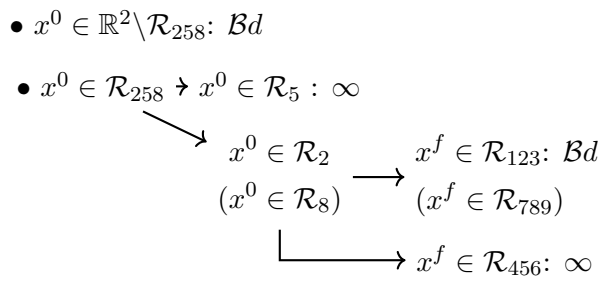


Figure A.2: Graph of all possible cases of non-empty $\mathbb{T}$.

Proof. We split the proof according to the location of x^0 in the phase plane and, when necessary, the location of x^f. Only cases corresponding to non-empty $\mathbb{T}$ are considered.

For $x^0 \in \mathbb{R}^2 \backslash \{\mathcal{R}_{258}\}$, the argument stems from the monotonic divergence of x_1. For instance $x^0 \in \mathcal{R}_{147}$, there exists $\epsilon > 0$, s.t. $x_1^u(0) \leq u_m - \epsilon$. Then, using (A.4), one easily shows that $\forall t > t_0,\ \dot{x}_1 = \omega(x_1 - u) \leq -\omega\epsilon$. Therefore, the final time is upper bounded by $\frac{x_1(t_0) - {x^f}_1}{\omega\epsilon}$. A similar inequality is obtained for $x_1^u(t_0) \geq u_M + \epsilon$ to cover $\mathcal{R}_{369}$. Hence, $\mathbb{T}$ is upper bounded.

For $x^0 \in \mathcal{R}_5 \backslash \mathcal{D}$, which is entirely covered by $\mathcal{C}_m \cup \mathcal{C}_M$, and is stable by symmetry *w.r.t.* $\mathcal{D}$. Lemma 3 permits to build a sequence that periodically returns to x^0, prolonging infinitely any solution from x^0. Hence, $\mathbb{T}$ is not upper-bounded.

For $x^0 \in \mathcal{R}_5 \cap \mathcal{D}$, for all possible values of u, the tangent vector field at x^0 is orthogonal to $\mathcal{D}$ and does not vanish. For any short time, the preceding rationale applies.

For $x^0 \in \mathcal{R}_2$ and $x^f \in \mathcal{R}_{123}$, one has $\dot{x}_2 < 0$, therefore x_2 is decreasing, hence $\dot{x}_2 \leq -\omega({x^f}_2 + u_m)$. Therefore, $\mathbb{T}$ is upper-bounded by $\frac{{x^0}_2 - {x^f}_2}{\omega({x^f}_2 + u_m)} < \infty$.

For $x^0 \in \mathcal{R}_2$ and $x^f \in \mathcal{R}_{456}$, there exists a sequence from any x^0 s.t., for some $t > 0$, $x_w \triangleq \phi(x^0, \text{seq}, t) \in \mathcal{R}_5$. In addition, any x^f can be accessed from this waypoint x_w through a sequence (u_m, a, b) or (u_M, a, b), with $a, b > 0$. Therefore, a transient from x^0 to x^f passing through x_w can be arbitrarily prolonged with sequences periodically returning to x_w. Hence, $\mathbb{T}$ is not upper-bounded.

The case $x^0 \in \mathcal{R}_8$ the analysis is similar to $x^0 \in \mathcal{R}_2$.

This completes the proof.

□

A.4 Convexity of bounded $\mathbb{T}$ cases

Lemma 5. When $\mathbb{T}$ is bounded, $\mathbb{T} = [T_{\min}, T_{\max}]$.

Proof. Lemma 4 shows that for $\mathbb{T}$ to be bounded either $x^0 \in \mathbb{R}^2 \backslash \mathcal{R}_{258}$, or $(x^0, x^f) \in \mathcal{R}_2 \times \mathcal{R}_{123}$, or $(x^0, x^f) \in \mathcal{R}_8 \times \mathcal{R}_{789}$.

As we only consider the case of bounded $\mathbb{T}$ in this section, Lemma 1 shows the existence of solutions of (4.19). In general, there are at most two bang-bang sequences with one switch between x^0 and x^f which are noted $\text{seq}_m \triangleq (u_m, a_m, b_M)$ and $\text{seq}_M \triangleq (u_M, a_M, b_m)$. Further, according to Lemma 2, the two controls $u_{\min}$ and $u_{\max}$ are bang-bang with at most one switch. Hence, either $u_{\min} = \text{seq}_m$ and $u_{\max} = \text{seq}_M$, or the other way around.

By definition, $\mathbb{T} \subset [T_{\min}, T_{\max}]$. When $T_{\min} = T_{\max}$, $\mathbb{T}$ is a singleton, hence is trivially convex. We now assume $T_{\min} < T_{\max}$. The rest of the proof depends on the location of (x^0, x^f) relative to $\mathcal{D}$.

$(x^0, x^f) \in \mathcal{D}^{-2}$ (on the same side)

Figure A.3 shows the trajectory steering the system from x^0 to x^f using seq_m and seq_M. The trajectories do not cross each other. The trajectory corresponding to seq_m and the reverse trajectory corresponding to seq_M define a positively oriented closed curve. The region $\mathcal{R}_{147} \cap \mathcal{D}^-$ being positively invariant, the curve is strictly included in $\mathcal{D}^{-2}$.

For all $x \in \mathcal{D}^-$,

$$\left\langle \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} (Ax + Bu_m),\ Ax + Bu_M \right\rangle > 0$$

Therefore, for all $a \in [0, a_m]$, there exists $b \geq 0,\ c \geq 0$ s.t.

$$\text{seq}_v(a, b, c) \triangleq (u_m, a, b, c) \in \Omega(x^0, x^f, a + b + c)$$

The solution x^{seq_v} is shown in Figure A.3. By definition, the solution x^{seq_v} is continuous, hence

$$x_1 \triangleq \phi(x, \text{seq}_v(a, b, c), a + b) = \phi(x^f, \text{seq}_v(a, b, c), -c)$$

We define the function

$$\mathcal{T} : [0, a_m] \ni a \mapsto \mathcal{T}(a) = a + b + c \in [T_{\min}, T_{\max}]$$

which maps the duration a of the first arc of x^{seq_v} to the total duration $a+b+c$. Define g as

$$g(a,b,c) = \phi(x^0, \text{seq}_v(a,b,c), a+b) - \phi(x^f, \text{seq}_v(a,b,c), -c)$$

From $u_m \neq u_M$, one has

$$\begin{aligned}\text{rank}(\begin{bmatrix}\frac{\partial g}{\partial b}, \frac{\partial g}{\partial c}\end{bmatrix}) &= \text{rank}(\begin{bmatrix}Ax^1 + Bu_M & -Ax^1 - Bu_m\end{bmatrix}) \\ &= \text{rank}(\begin{bmatrix}Ax^1 & B\end{bmatrix}) \\ &= \text{rank}(\begin{pmatrix}x_1^1 & 1 \\ -x_2^1 & 1\end{pmatrix}) = 2\end{aligned}$$

The intermediate point x^1 is defined by $g(a,b,c) = 0$. The full rank property above, associated with the injectivity of the function $(a,b,c) \mapsto \begin{pmatrix}a & g(a,b,c)\end{pmatrix}^T$, gives, through the global inversion theorem [Krantz2012, Theorem 6.2.3], the existence of $\psi \in \mathcal{C}^0$ s.t., over the domain of definition $[0, a_{\min}]$, $(b,c) = \psi(a)$.

Thus, the function $\mathcal{T}$ is continuous. Therefore, by the intermediate value theorem, $[T_{\min}, T_{\max}] \subset \mathcal{T}([0, a_m]) \subset \mathbb{T}$, which concludes the proof.

$(x^0, x^f) \in \mathcal{D}^{+2}$ (on the same side)

The proof is identical, replacing $\mathcal{D}^-$ by $\mathcal{D}^+$, the trajectory corresponding to seq_m and the reverse trajectory corresponding to seq_M defining a negatively oriented closed curve.

$(x^0, x^f) \in \mathcal{D}^+ \times \mathcal{D}^-$ (on opposite sides)

According to Figure A.2, $x^0 \in \mathcal{R}_1 \cap \mathcal{D}^+$ and $x^f \in \{\mathcal{R}_1 \cap \mathcal{D}^-\} \cup \mathcal{R}_4$.

If the (Euclidean) distance $d(x^f, \mathcal{D})$ between x^f and $\mathcal{D}$ is strictly lower than the distance $d(x^0, \mathcal{D})$ between x^0 and $\mathcal{D}$, then Lemma 3 states that $\phi(x^f, u_M, -t_0) = Sx^f = \phi(x^f, u_m, -t_1)$, for some $t_0, t_1 \geq 0$. We use the same constructive proof between Sx^f and x^f with the sequence $\text{seq}_{\mathcal{B}} \triangleq (u_m, a, b, c)$, with $a \in [0, \frac{t_0}{2}]$.

If $d(x^f, \mathcal{D}) > d(x^0, \mathcal{D})$, then Lemma 3 states that $\phi(x^0, u_M, t_0) = Sx^0 = \phi(x^0, u_m, t_1)$, for some $t_0, t_1 \geq 0$. We use the same constructive proof between Sx^0 and x^0 with $\text{seq}_{\mathcal{B}}$.

If $d(x^f, \mathcal{D}) = d(x^0, \mathcal{D})$, the proof directly follows from $x^0 = Sx^f$ to x^f, this situation is illustrated in Figure A.4.

$(x^0, x^f) \in \mathcal{D}^- \times \mathcal{D}^+$ (on opposite sides)

The case is identical to the previous case, x^0 belonging to $\mathcal{R}_9 \cap \mathcal{D}^-$ and x^f belonging to $\{\mathcal{R}_9 \cap \mathcal{D}^+\} \cup \mathcal{R}_6$.

This completes the proof. □

A.5 Convexity of unbounded $\mathbb{T}$ cases

Let us define $\mathcal{J}^+$ a subset of $\mathbb{R}^4$ as follows

$$\mathcal{J}^+ \triangleq \begin{cases} x^0 \in \mathcal{R}_{25} \cap \mathcal{D}^+, \ x^f \in \mathcal{R}_{56} \cap \mathcal{D}^+ \text{ s.t.} \\ \exists x_{\mathcal{D}}, x_d \in \mathcal{D} \cap \mathcal{R}_5, \ t_{\mathcal{D}} > 0, \ t_d > 0, \\ \begin{cases} x_{d1} > x_{\mathcal{D}1} \\ x^f = \phi(x_{\mathcal{D}}, u_m, t_{\mathcal{D}}) \\ x^0 = \phi(x_d, u_M, -t_d) \end{cases} \end{cases}$$

and $\mathcal{J}^-$ by symmetry of $\mathcal{J}^+$ *w.r.t.* $\mathcal{D}$. The set $\mathcal{J}^+$ is partially pictured in Figure A.1 (all possible values of x^0 are colored in red when x^f varies along the dashed line). It plays a particular role in Lemma 6 as it is the only one where boundary conditions yield a non-convex set $\mathbb{T}$.

Lemma 6. When $\mathbb{T}$ is unbounded, if $(x^0, x^f) \notin \mathcal{J}^+ \cup \mathcal{J}^-$, then $\mathbb{T} = [T_{\min}, +\infty[$, otherwise, there exists $A < B$ s.t. $\mathbb{T} = [T_{\min}, A] \cup [B, +\infty[$.

Proof. Following Lemma 4, a careful investigation of the graph in Figure A.2 reveals that for $\mathbb{T}$ to be unbounded we have $(x^0, x^f) \in \mathcal{R}_{258} \times \mathcal{R}_{456}$. By symmetry of the vector field (rotation of π about $(\frac{u_m+u_M}{2}, -\frac{u_m+u_M}{2})^T$), we now only consider a pair $(x^0, x^f) \in (\mathcal{R}_{25} \cap \mathcal{D}^+) \times \mathcal{R}_{456}$.

$(x^0, x^f) \notin \mathcal{J}^+$

In all such cases, there exists a sequence (u_m, a, b, c, d), with $a \geq 0, b > 0, c > 0, d \geq 0$ steering x^0 to x^f with a single intersection with $\mathcal{D} \cap \mathcal{R}_5$. This sequence can be easily extended in the vicinity of $\mathcal{D} \cap \mathcal{R}_5$ (which excludes equilibria) to increase the transient time by any desired arbitrarily small increment $\epsilon > 0$. Iteratively, this construction allows to infinitely increase the transient time by a continuous constructive process.

Also, the same type of sequence with other values for a, b, c, d can generate a smooth collection of trajectories approaching the minimum time trajectory. The proof of Appendix A.4 yields the conclusion with the continuous mapping $\mathcal{T} : (a, b, c, d) \mapsto a + b + c + d$.

$(x^0, x^f) \in \mathcal{J}^+$

There exist two sequences $\text{seq}_1 = (u_m, a_1, b_1)$ (e.g. corresponding to the minimum time $T_{\min}$) and $\text{seq}_2 = (u_M, a_2, b_2)$ (with time T_2) steering x^0 to x^f by two paths Γ_1 and Γ_2 entirely in $\mathcal{D}^+$. They are illustrated in Figure A.5.

$\text{seq} = (u_m, a, b, c)$ gives, by the continuity of $(a, b, c) \mapsto a + b + c$, that all feasible trajectories staying inside $\Gamma_1 \cup \Gamma_2$ have a transient time in $[T_{\min}, T_2]$, for $T_2 < \infty$.

Now, consider a trajectory from x^0 to x^f leaving $\Gamma_1 \cup \Gamma_2$. A detailed investigation of the phase portrait gives that this trajectory leaves $\Gamma_1 \cup \Gamma_2$ at a point $x^{ii} \triangleq \phi(x^f, u_m, -t), 0 < t \leq b_2$, strictly in $\mathcal{R}_5$, with control $u > u_m$.

From all solutions, starting from x^{ii}, reaching x^f and leaving $\Gamma_1 \cup \Gamma_2$ at x^{ii}, the one passing through Sx^{ii} displayed on Figure A.5 is the only bang-bang solution. Hence, it is the fastest according to Lemma 2. We denote Γ_{ii} the union of this solution and the fastest solution red from x^0 to x^{ii} and construct Γ_i similarly by passing through $x^i \triangleq \phi(x^f, u_m, -b_2)$ instead of x^{ii}.

We now show that Γ_i is faster than Γ_{ii}. As they share the arc from Sx^i to x^f, we only need to use that the one-arc solution from x^0 to Sx^i is faster than the 3-arc solution passing through x^{ii} and Sx^{ii}. This holds because the only bang-bang solution from x^0 to Sx^i is the minimum time solution (according to Lemma 2). This holds for all x^{ii}, hence all solution leaving $\Gamma_1 \cup \Gamma_2$ is slower than Γ_i. As Γ_i is crossing $\mathcal{D}$, it can be arbitrarily extended. This completes the proof. □

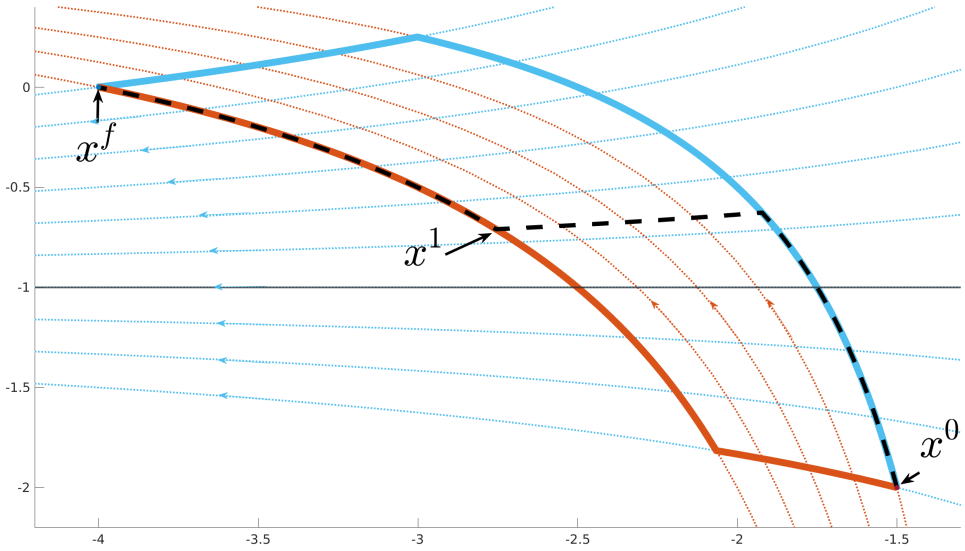


Figure A.3: Phase diagram for (A.2) with $u = u_m$ (dotted red), $u = u_M$ (dotted blue), minimum time trajectory $x^{\min}$ (solid blue line), maximum time trajectory $x^{\max}$ (solid red line), and x^{seq_v} for some $a \in [0, a_m]$ (dashed black line).

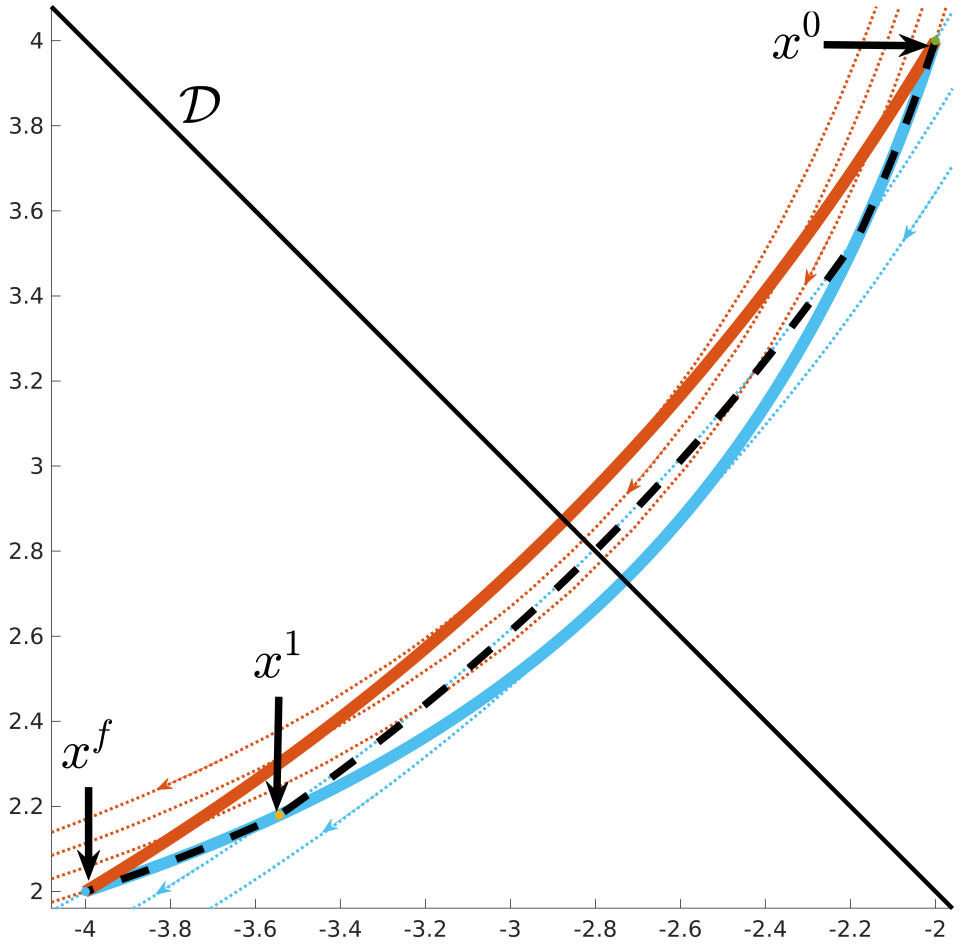


Figure A.4: Phase diagram for (A.2) with $u = u_m$ (dotted red), $u = u_M$ (dotted blue), minimum time trajectory $x^{\min}$ (solid red line), maximum time trajectory $x^{\max}$ (solid blue line), and $x^{\text{seq}_{\mathcal{B}}}$ for some $a \in [0, \frac{t_0}{2}]$ (dashed black line).

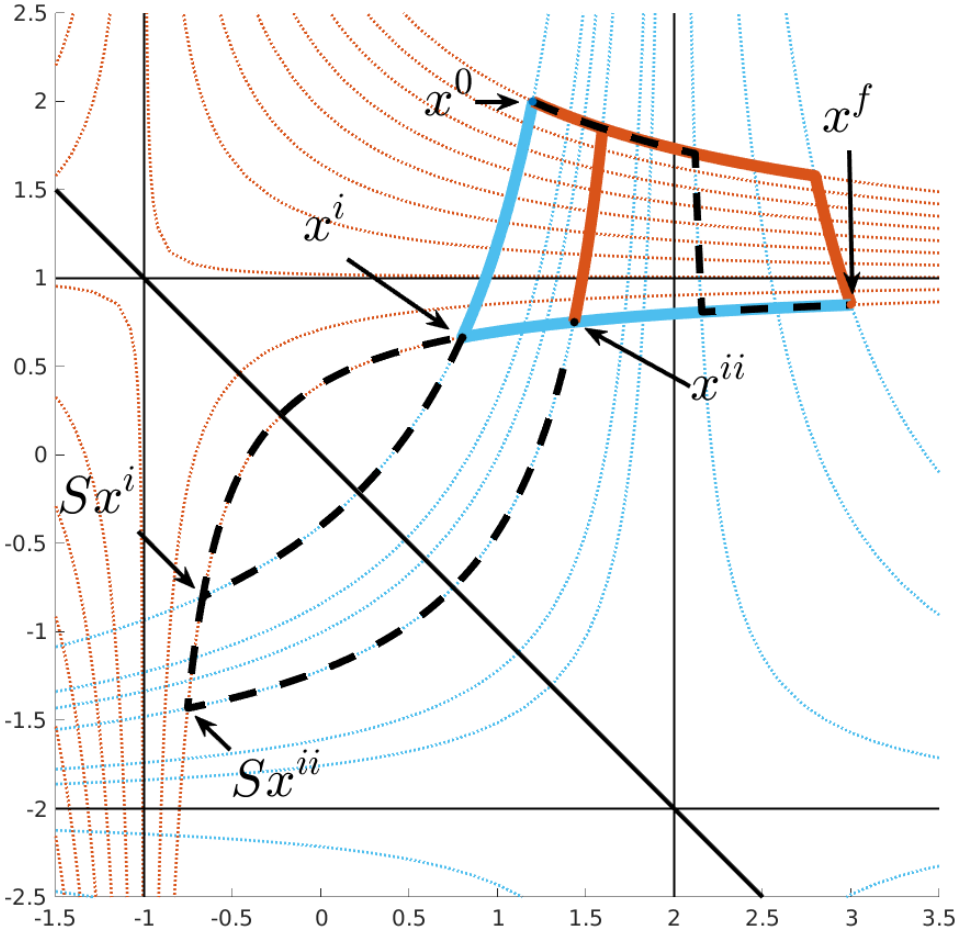


Figure A.5: Phase diagram for (A.2) with $u = u_m$ (dotted red), $u = u_M$ (dotted blue), minimum time trajectory $x^{\min}$ (solid red line), T_2 time trajectory x^{seq_2} (solid blue line), and points of interest for the proof of Lemma 6.

Appendix B

Empower and monitor during the double support phase

Contents

In this appendix, we sketch an alternative control methodology for active gait training during double support phases which aims at enabling the patient to participate in the motion during these phases. It could be beneficial to patients' rehabilitation because transferring the weight from one leg to the other, which occurs during double support phases, is also important to regain ambulation capabilities. The sketched control law offers temporal freedom around the offline trajectory using the same VG methodology used during single support phases (see in Section 3.1), with a substantial difference: all the joints of the exoskeleton are constrained to the parametric curve P (computed from the offline trajectory $\mathcal{T}$ instead of $\mathcal{T}^{sw}$). Hence, no DoF is left free for the stabilization of the CoM dynamics. This is the opposite choice to the one described in Chapter 3, where all the joints are used for the stabilization of the CoM dynamics during double support phases.

In addition, we propose a saturation of the patient schedule, during double support, which ensures the feasibility of the next single support phase. This ensures the OP strategy can still be used during single support phases.

B.1 Whole-body VG: consequences on the CoM

We propose to use the VG methodology on all Atalante's joints. This requires only minimal changes in the control law described in Section 3.1. The curvilinear abscissa s and the parametric curve P are now defined from the complete joint trajectory $\mathcal{T}$

$$\mathrm{s} : \tau \in [0, T_f] \mapsto \int_0^\tau ||\dot{\mathcal{T}}(t)||_2 \, dt \in [0, L_{max}] \tag{B.1}$$

and

$$P : s \in [0, L_{max}] \mapsto \mathcal{T} \circ \mathrm{s}^{-1}(s) \in \mathbb{R}^{12} \tag{B.2}$$

The torque vector τ for all the joints of the exoskeleton is now computed as a high-gain PD controller plus gravity compensation

$$\tau(\sigma, \dot{\sigma}) = \tau_{PD}(\sigma, \dot{\sigma}, q^j, \dot{q}^j) + \tau_{GC}(q) \tag{B.3}$$

with

$$\tau_{PD}(\sigma, \dot{\sigma}, q^j, \dot{q}^j) = K_p(P(\sigma) - q) + K_d(T(\sigma)\dot{\sigma} - \dot{q}^j) \tag{B.4}$$

where $\sigma \in [0, L_{max}]$ is still the control variable and $q^j \in \mathbb{R}^{12}$ is the vector of joint positions. Following the VG approach, the control variable velocity reads

$$\dot{\sigma}(\sigma, q^j, \dot{q}^j) \triangleq \frac{T(\sigma)^\top \left[K_p(q^j - P(\sigma)) + K_d \dot{q}^j\right]}{T(\sigma)^T K_d T(\sigma)} \tag{B.5}$$

The control variable σ defines the patient schedule by the relation

$$\eta : t \in [0, t_f] \mapsto \mathrm{s}^{-1} \circ \sigma(t) \in [0, T_f] \tag{B.6}$$

with t_f the final time defined by the first time the control variable σ reaches the end of the curvilinear trajectory s *i.e.* $\sigma(t_f) = L_{max}$.

The CoM trajectory c is defined by the parametric curve P and the control variable σ according to

$$c(t) = \mathcal{FK} \circ P \circ \sigma(t) = \mathcal{FK} \circ \mathcal{T} \circ \eta(t) \tag{B.7}$$

By definition of $\mathcal{T}$,

$$c(t) = c^{nom} \circ \eta(t) \tag{B.8}$$

with $c^{nom} \triangleq \mathcal{FK} \circ \mathcal{T}$ the nominal CoM trajectory. Hence, the CoM velocity $\dot{c}$ reads

$$\dot{c}(t) = \dot{\eta}(t) \frac{dc^{nom}}{d\eta} \circ \eta(t) \tag{B.9}$$

Inspecting (B.8) and (B.9), one can easily notice that

- The position c of the CoM trajectory follows the nominal CoM path c^{nom}. Consequently, the final CoM position $c(t_f)$ is located at the final nominal CoM location $c^{nom}(T_f)$, *i.e.*

$$c(t_f) = c^{nom}(T_f) \tag{B.10}$$

- The velocity $\dot{c}(t)$ of the CoM trajectory is tangential to the nominal CoM velocity $\frac{dc^{nom}}{d\eta}$, time-rescaled by the patient schedule η. Consequently, the final CoM velocity $\dot{c}(t_f)$ is also tangential to the final nominal CoM velocity $\frac{dc^{nom}}{d\eta}(T_f)$, *i.e.*

$$\dot{c}(t_f) = \dot{\eta}(t_f)\frac{dc^{nom}}{d\eta}(T_f) \tag{B.11}$$

B.2 Monitor next step feasibility: a conservative approach

Motivated by the good results of the OP strategy, we ground our patient freedom saturation on the same reachability problem for the input-constrained LIP dynamics (2.17). The fixed-horizon OCP writes the same way as Problem 4.1, but the set of feasible commands Ω is adapted to include the CoM path constraint (B.10), yielding

$$\Omega(\chi^0, \chi^f, T) \triangleq \left\{ u \in U_{ad}(T),\ \begin{cases} \chi^u(0) = \chi^0 \\ \chi^u(T) = \chi^f \\ C\chi^u(t) \in \gamma,\ \forall t \end{cases} \right\}$$

with $C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ and γ the manifold defined by the nominal CoM path

$$\gamma = \left\{ x \in \mathbb{R}^2,\ \exists t \in [0, T_f]\ s.t.\ x = c^{nom}(t) \right\} \tag{B.12}$$

In general, γ is a non-trivial manifold. Hence, the constraint $C\chi^u(t) \in \gamma,\ \forall t$ can not be written as an affine constraint, which makes Problem 4.1 impossible to transcribe as a QP problem. This discards the solving methodology of Problem 4.2 proposed in Chapter 4.

Luckily, further investigations of the nominal CoM trajectory during double support[1], depicted on Figure B.1, reveal that γ is almost a segment.

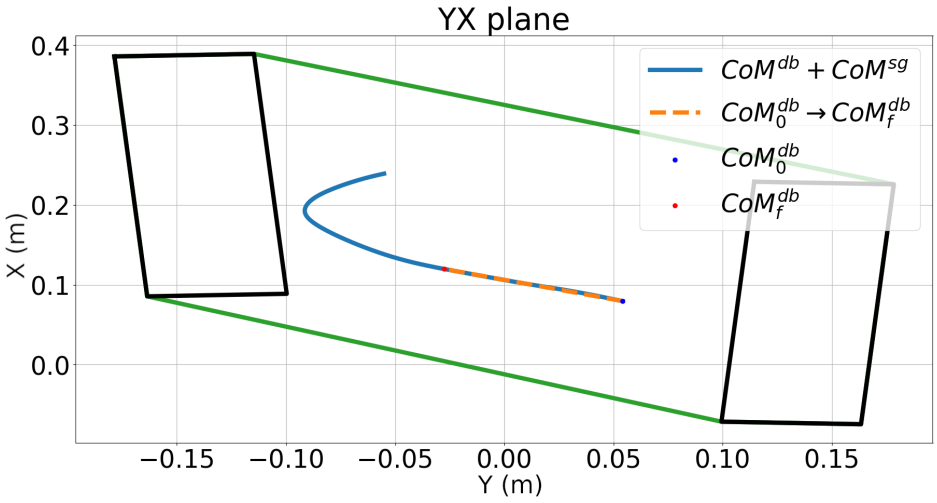


Figure B.1: Nominal CoM trajectory during one step (double and single support) and footprint of the support polygon during double support. The nominal CoM path during double support is almost a segment (the dotted orange segment).

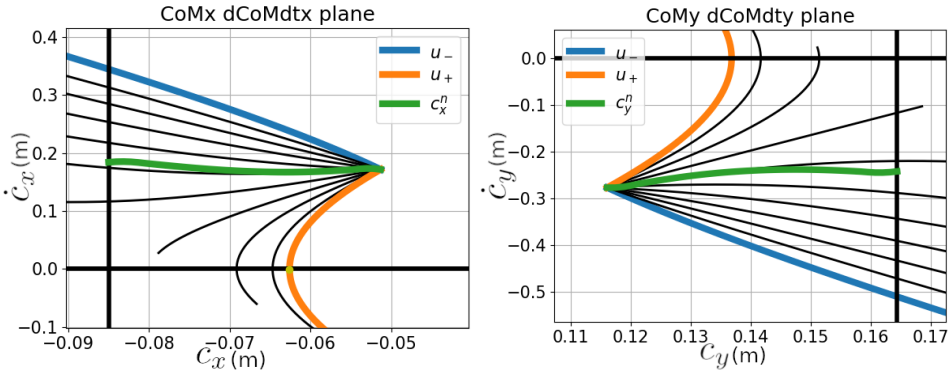
Therefore, the CoM path constraint (B.10) can be approximated by an affine constraint, consistent with the QP formulation.

This yields a few important remarks

- the X and Y CoM dynamics are linearly dependent, hence we only need to solve Problem 4.2 along one axis;
- the CoM velocity and acceleration are tangential to the $CoM_0^{db} CoM_f^{db}$ vector;
- the CoP location is constrained to lie on the segment defined by the intersection of, on one hand, the (infinite) line passing through CoM_0^{db} and CoM_f^{db}, and the double support polygon, on the other hand (by (2.17), the remark above and the unilateral contact constraints).

Rather than solving the OCP defined by Problem 4.2, we propose to integrate the LIP dynamics backward from the final nominal state χ_f, along each dimension X and Y, with the minimum and maximum CoP values u_-

[1]The OCP used to generate the flat-foot trajectory, described on Section 2.2 in the YX-plane, has been modified to include a double support phase for the purpose of the development presented in this appendix.



(a) $(c_x, \dot{c}_x)$ (b) $(c_y, \dot{c}_y)$

Figure B.2: Backward trajectories from x_f (final state of flat-foot trajectory with double support). Nominal trajectory in green.

and u_+. These solutions, along with the solutions with 9 intermediate values between u_- and u_+, are displayed on Figure B.2.

Figure B.2 reveals that the set $\mathcal{S}$ of states from which the final nominal state is reachable is compact. Hence, from this observation, and the CoM velocity constraint (B.11), we can compute the saturation on the patient schedule which ensures the reachability of χ_f. This saturation is illustrated in Figure B.3, computed for each dimension X and Y. Note that

- the patient schedule is saturated to be equal to the nominal velocity 1 at the end of the double support phase, which is expected from (B.11) and the final CoM velocity constraint;
- the saturation computed using dimensions X and Y do not exactly overlap, which is a consequence of the real final nominal CoM velocity is not exactly tangential to the $CoM_0^{db}CoM_f^{db}$ vector.

This alternative control law for double support phases has only been tested in simulation with arbitrarily shaped patient schedules η to simulate the impact of a patient. It acts as expected and stabilizes simulations with simulated slowdowns during the double support phase which makes the system fall without the proposed saturation. Experiments supporting these simulation observations remain to be done.

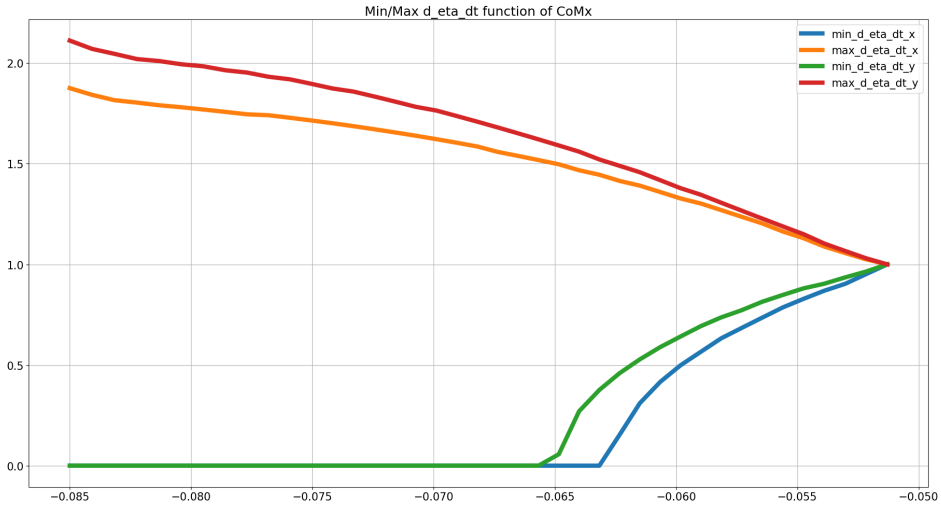


Figure B.3: Lower and upper saturation bounds computed using backward CoM trajectories either along X or Y axis. Bounds differ with the dimension used for their computation because the nominal CoM path during the double support phase is not exactly a segment.

Bibliography

[Ahn2021] Junhyeok Ahn, Steven Jens Jorgensen, Seung Hyeon Bang, and Luis Sentis. "Versatile Locomotion Planning and Control for Humanoid Robots". In: *Front. Robot. AI* 8 (Aug. 13, 2021), p. 712239.

[Ames2014] Aaron D. Ames. "Human-Inspired Control of Bipedal Walking Robots". In: *IEEE Transactions on Automatic Control* 59.5 (2014), pp. 1115–1130.

[Aoyagi2007] Daisuke Aoyagi, Wade E. Ichinose, Susan J. Harkema, David J. Reinkensmeyer, and James E. Bobrow. "A Robot and Control Algorithm That Can Synchronously Assist in Naturalistic Motion During Body-Weight-Supported Gait Training Following Neurologic Injury". In: *IEEE Trans. Neural Syst. Rehabil. Eng.* 15.3 (Sept. 2007), pp. 387–400.

[Banala2007] Sai K. Banala, Suni K. Agrawal, and John P. Scholz. "Active Leg Exoskeleton (ALEX) for Gait Rehabilitation of Motor-Impaired Patients". In: *2007 IEEE 10th International Conference on Rehabilitation Robotics*. 2007 IEEE 10th International Conference on Rehabilitation Robotics. June 2007, pp. 401–407.

[Banala2009] Sai K. Banala, Seok Hun Kim, Sunil K. Agrawal, and John P. Scholz. "Robot Assisted Gait Training With Active Leg Exoskeleton (ALEX)". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 17.1 (Feb. 2009), pp. 2–8.

[Benallegue2014] Mehdi Benallegue and Florent Lamiraux. "Humanoid Flexibility Deformation Can Be Efficiently

Estimated Using Only Inertial Measurement Units and Contact Information". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014 IEEE-RAS International Conference on Humanoid Robots. Nov. 2014, pp. 246–251.

[Benallegue2015] Mehdi Benallegue and Florent Lamiraux. "Estimation and Stabilization of Humanoid Flexibility Deformation Using Only Inertial Measurement Units and Contact Information". In: *Int. J. Human. Robot.* 12.03 (Sept. 2015), p. 1550025.

[Bock1984] H. G. Bock and K. J. Plitt. "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems". In: *IFAC Proceedings Volumes*. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984 17.2 (July 1, 1984), pp. 1603–1608.

[Bortole2015] Magdo Bortole, Anusha Venkatakrishnan, Fangshi Zhu, Juan C Moreno, Gerard E Francisco, Jose L Pons, and Jose L Contreras-Vidal. "The H2 Robotic Exoskeleton for Gait Rehabilitation after Stroke: Early Findings from a Clinical Study". In: *J NeuroEngineering Rehabil* 12.1 (Dec. 2015), p. 54.

[Brunet2022] Maxime Brunet, Marine Pétriaux, Florent Di Meglio, and Nicolas Petit. "Fast Replanning of a Lower-Limb Exoskeleton Trajectories for Rehabilitation". In: 2022 IEEE 61st Conference on Decision and Control (CDC). Dec. 2022.

[Brunet2023] Maxime Brunet, Marine Pétriaux, Florent Di Meglio, and Nicolas Petit. *Enabling Safe Walking Rehabilitation on the Exoskeleton Atalante: Experimental Results*. Apr. 17, 2023. preprint.

[Budhiraja2019] Rohan Budhiraja, Justin Carpentier, and Nicolas Mansard. "Dynamics Consensus between Centroidal and Whole-Body Models for Locomotion of Legged Robots". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). IEEE, May 2019, pp. 6727–6733.

[Caron2016] Stéphane Caron and Abderrahmane Kheddar. "Multi-Contact Walking Pattern Generation Based on Model Preview Control of 3D COM Accelerations". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). Nov. 2016, pp. 550–557.

[Caron2017] Stéphane Caron and Abderrahmane Kheddar. "Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-Base Inverted Pendulum". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2017, pp. 5017–5024.

[Caron2019] Stéphane Caron, Abderrahmane Kheddar, and Olivier Tempier. "Stair Climbing Stabilization of the HRP-4 Humanoid Robot Using Whole-body Admittance Control". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). May 2019, pp. 277–283.

[Caron2020] Stéphane Caron, Adrien Escande, Leonardo Lanari, and Bastien Mallein. "Capturability-Based Pattern Generation for Walking With Variable Height". In: *IEEE Trans. Robot.* 36.2 (Apr. 2020), pp. 517–536.

[Carpentier2019] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. "The Pinocchio C++ Library : A Fast and Flexible Implementation of Rigid Body Dynamics Algorithms and Their Analytical Derivatives". In: *IEEE International Symposium on System Integration (SII)*. IEEE International Symposium on System Integration (SII). 2019.

[Castillo2019] Guillermo A. Castillo, Bowen Weng, Ayonga Hereid, Zheng Wang, and Wei Zhang. "Reinforce-

ment Learning Meets Hybrid Zero Dynamics: A Case Study for RABBIT". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). May 2019, pp. 284–290.

[Castillo2020] Guillermo A. Castillo, Bowen Weng, Wei Zhang, and Ayonga Hereid. "Hybrid Zero Dynamics Inspired Feedback Control Policy Design for 3D Bipedal Locomotion Using Reinforcement Learning". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020 IEEE International Conference on Robotics and Automation (ICRA). May 2020, pp. 8746–8752.

[Castillo2021] Guillermo A. Castillo, Bowen Weng, Wei Zhang, and Ayonga Hereid. "Robust Feedback Motion Policy Design Using Reinforcement Learning on a 3D Digit Bipedal Robot". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2021, pp. 5136–5143.

[Ciocca2017] Matteo Ciocca, Pierre-Brice Wieber, and Thierry Fraichard. "Strong Recursive Feasibility in Model Predictive Control of Biped Walking". In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids). IEEE, Nov. 2017, pp. 730–735.

[Dantec2022] Ewen Dantec, Maximilien Naveau, Pierre Fernbach, Nahuel Villa, Guilhem Saurel, Olivier Stasse, Michel Taix, and Nicolas Mansard. "Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot". In: *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids). Nov. 2022, pp. 638–644.

[Doppmann2015] Corinne Doppmann, Barkan Ugurlu, Masashi Hamaya, Tatsuya Teramae, Tomoyuki Noda, and Jun Morimoto. "Towards Balance Recovery Control for Lower Body Exoskeleton Robots with Variable Stiffness Actuators: Spring-loaded Flywheel Model". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015 IEEE International Conference on Robotics and Automation (ICRA). May 2015, pp. 5551–5556.

[Duburcq2019] Alexis Duburcq. *Jiminy: A Fast and Portable Python/C++ Simulator of Poly-Articulated Systems for Reinforcement Learning*. 2019.

[Duburcq2020] Alexis Duburcq, Yann Chevaleyre, Nicolas Bredeche, and Guilhem Boéris. "Online Trajectory Planning Through Combined Trajectory Optimization and Function Approximation: Application to the Exoskeleton Atalante". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020 IEEE International Conference on Robotics and Automation (ICRA). May 2020, pp. 3756–3762.

[Duburcq2022a] Alexis Duburcq. "Apprentissage et Optimisation de la Locomotion pour un Exosquelette à Destination des Patients Paraplégiques". Université Paris Sciences et Lettres, Dec. 9, 2022.

[Duburcq2022b] Alexis Duburcq, Fabian Schramm, Guilhem Boéris, Nicolas Bredeche, and Yann Chevaleyre. "Reactive Stepping for Humanoid Robots Using Reinforcement Learning: Application to Standing Push Recovery on the Exoskeleton Atalante". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Oct. 2022, pp. 9302–9309.

[Duschau-Wicke2010] Alexander Duschau-Wicke, Joachim von Zitzewitz, Andrea Caprez, Lars Lunenburger, and Robert Riener. "Path Control: A Method for Patient-Cooperative Robot-Aided Gait Rehabilitation". In:

IEEE Transactions on Neural Systems and Rehabilitation Engineering 18.1 (Feb. 2010), pp. 38–48.

[Ekkelenkamp2005] Ralf Ekkelenkamp, Jan F. Veneman, and Herman van der Kooij. "LOPES : Selective Control of Gait Functions During the Gait Rehabilitation of CVA Patients". In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005. IEEE, 2005, pp. 361–364.

[Englsberger2012] Johannes Englsberger and Christian Ott. "Integration of Vertical COM Motion and Angular Momentum in an Extended Capture Point Tracking Controller for Bipedal Walking". In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012).* 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012). IEEE, Nov. 2012, pp. 183–189.

[Englsberger2014] Johannes Englsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A. Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, and Alin Albu-Schäffer. "Overview of the Torque-Controlled Humanoid Robot TORO". In: *2014 IEEE-RAS International Conference on Humanoid Robots.* 2014 IEEE-RAS International Conference on Humanoid Robots. Nov. 2014, pp. 916–923.

[Erez2013] Tom Erez, Kendall Lowrey, Yuval Tassa, Vikash Kumar, Svetoslav Kolev, and Emanuel Todorov. "An Integrated System for Real-Time Model Predictive Control of Humanoid Robots". In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids).* 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids). Oct. 2013, pp. 292–299.

[Escande2014] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. "Hierarchical Quadratic Programming: Fast Online Humanoid-Robot Motion Generation".

In: *The International Journal of Robotics Research* 33.7 (June 2014), pp. 1006–1028.

[Eveld2021] Maura Eveld, Shane King, Karl Zelik, and Michael Goldfarb. "Design and Implementation of a Stumble Recovery Controller for a Knee Exoskeleton". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2021, pp. 6196–6203.

[Featherstone2008] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2008. 272 pp.

[Feng2013] Siyuan Feng, X. Xinjilefu, Weiwei Huang, and Christopher G. Atkeson. "3D Walking Based on Online Optimization". In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2013). IEEE, Oct. 2013, pp. 21–27.

[Fernbach2020] Pierre Fernbach, Steve Tonneau, Olivier Stasse, Justin Carpentier, and Michel Taïx. "C-CROC: Continuous and Convex Resolution of Centroidal Dynamic Trajectories for Legged Robots in Multicontact Scenarios". In: *IEEE Transactions on Robotics* 36.3 (June 2020), pp. 676–691.

[Finet2017] Sylvain Finet. "Contribution to the Control of Biped Robots". PhD thesis. Université Paris Sciences et Lettres, 2017.

[Galliker2022] Manuel Y Galliker, Noel Csomay-Shanklin, Ruben Grandia, Andrew J Taylor, Farbod Farshidian, Marco Hutter, and Aaron D Ames. "Planar Bipedal Locomotion with Nonlinear Model Predictive Control: Online Gait Generation Using Whole-Body Dynamics". In: (2022), p. 9.

[Gassert2018] Roger Gassert and Volker Dietz. "Rehabilitation Robots for the Treatment of Sensorimotor Deficits: A Neurophysiological Perspective". In: *J Neuro-Engineering Rehabil* 15.1 (Dec. 2018), p. 46.

[Goldfarb1983] D. Goldfarb and A. Idnani. "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs". In: *Mathematical Programming* 27.1 (Sept. 1983), pp. 1–33.

[Goswami2019] Ambarish Goswami and Prahlad Vadakkepat. *Humanoid Robotics: A Reference*. Springer Netherlands, 2019.

[Griffin2017] Robert J. Griffin, Georg Wiedebach, Sylvain Bertrand, Alexander Leonessa, and Jerry Pratt. "Walking Stabilization Using Step Timing and Location Adjustment on the Humanoid Robot, Atlas". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2017, pp. 667–673. arXiv: 1703.00477 [cs].

[Grizzle2017] Jessy W. Grizzle and Christine Chevallereau. "Virtual Constraints and Hybrid Zero Dynamics for Realizing Underactuated Bipedal Locomotion". In: *Humanoid Robotics: A Reference*. Springer. Humanoid Control. 2017.

[Gurriet2018] Thomas Gurriet, Sylvain Finet, Guilhem Boeris, Alexis Duburcq, Ayonga Hereid, Omar Harib, Matthieu Masselin, Jessy W. Grizzle, and Aaron D. Ames. "Towards Restoring Locomotion for Paraplegics: Realizing Dynamically Stable Walking on Exoskeletons". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, May 2018, pp. 2804–2811.

[Gurriet2020] Thomas Gurriet, Maegan Tucker, Alexis Duburcq, Guilhem Boeris, and Aaron D. Ames. "Towards Variable Assistance for Lower Body Exoskeletons". In: *IEEE Robotics and Automation Letters* 5.1 (Jan. 2020), pp. 266–273.

[Hargraves1987] C. R. Hargraves and S. W. Paris. "Direct Trajectory Optimization Using Nonlinear Programming and Collocation". In: *Journal of Guidance, Control, and Dynamics* 10.4 (1987), pp. 338–342.

[Haynes2006] G.Clark Haynes and Alfred A. Rizzi. "Gaits and Gait Transitions for Legged Robots". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006, pp. 1117–1122.

[Heinrichs1996] B. Heinrichs, N. Sepehri, and A.B. Thornton-Trump. "Position-Based Impedance Control of an Industrial Hydraulic Manipulator". In: *Proceedings of IEEE International Conference on Robotics and Automation.* Proceedings of IEEE International Conference on Robotics and Automation. Vol. 1. Apr. 1996, 284–290 vol.1.

[Herdt2010] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. "Online Walking Motion Generation with Automatic Footstep Placement". In: *Advanced Robotics* 24.5-6 (2010), pp. 719–737.

[Hereid2015] Ayonga Hereid, Christian M. Hubicki, Eric A. Cousineau, Jonathan W. Hurst, and Aaron D. Ames. "Hybrid Zero Dynamics Based Multiple Shooting Optimization with Applications to Robotic Walking". In: *2015 IEEE International Conference on Robotics and Automation (ICRA).* 2015 IEEE International Conference on Robotics and Automation (ICRA). May 2015, pp. 5734–5740.

[Hereid2016] Ayonga Hereid, Eric A. Cousineau, Christian M. Hubicki, and Aaron D. Ames. "3D Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics". In: *2016 IEEE International Conference on Robotics and Automation (ICRA).* 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 1447–1454.

[Herzog2016] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimminger, Stefan Schaal, and Ludovic

Righetti. “Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid”. In: *Auton Robot* 40.3 (Mar. 1, 2016), pp. 473–491.

[Hogan1984] Neville Hogan. “Impedance Control: An Approach to Manipulation”. In: *1984 American Control Conference*. 1984 American Control Conference. June 1984, pp. 304–313.

[Hornby2019] T. George Hornby, Christopher E. Henderson, Abbey Plawecki, Emily Lucas, Jennifer Lotter, Molly Holthus, Gabrielle Brazg, Meghan Fahey, Jane Woodward, Marzieh Ardestani, and Elliot J. Roth. “Contributions of Stepping Intensity and Variability to Mobility in Individuals Poststroke”. In: *Stroke* 50.9 (Sept. 2019), pp. 2492–2499.

[Huynh2021] Vaiyee Huynh, Guillaume Burger, Quoc Viet Dang, Raphaël Pelgé, Guilhem Boéris, Jessy W. Grizzle, Aaron D. Ames, and Matthieu Masselin. “Versatile Dynamic Motion Generation Framework: Demonstration With a Crutch-Less Exoskeleton on Real-Life Obstacles at the Cybathlon 2020 With a Complete Paraplegic Person”. In: *Front. Robot. AI* 8 (Sept. 24, 2021), p. 723780.

[Ivanenko2002] Yuri P. Ivanenko, Renato Grasso, Velio Macellari, and Francesco Lacquaniti. “Two-Thirds Power Law in Human Locomotion: Role of Ground Contact Forces:” in: *Neuroreport* 13.9 (July 2002), pp. 1171–1174.

[Johnson2015] Matthew Johnson, Brandon Shrewsbury, Sylvain Bertrand, Tingfan Wu, Daniel Duran, Marshall Floyd, Peter Abeles, Douglas Stephen, Nathan Mertins, Alex Lesman, John Carff, William Rifenburgh, Pushyami Kaveti, Wessel Straatman, Jesper Smith, Maarten Griffioen, Brooke Layton, Tomas de Boer, Twan Koolen, Peter Neuhaus, and Jerry Pratt. “Team IHMC’s Lessons Learned from the DARPA Robotics Challenge Trials”. In: *Journal of Field Robotics* 32.2 (2015), pp. 192–208.

[Joly1995] L. D. Joly and C. Andriot. "Imposing Motion Constraints to a Force Reflecting Telerobot through Real-Time Simulation of a Virtual Mechanism". In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. Proceedings of 1995 IEEE International Conference on Robotics and Automation. Vol. 1. May 1995, 357–362 vol.1.

[Kagami2002] Satoshi Kagami, Koichi Nishiwaki, Yasuo Kuniyoshi, Masayuki Inaba, and H. Inoue. "Online Generation of Humanoid Walking Motion Based on a Fast Generation Method of Motion Pattern That Follows Desired ZMP". In: *IEEE/RSJ International Conference on Intelligent Robots and System*. IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 3. IEEE, 2002, pp. 2684–2689.

[Kagami2011] Satoshi Kagami, Junichi Urata, Koichi Nshiwaki, Yuto Nakanishi, Kei Okada, and Masayuki Inaba. "Online Decision of Foot Placement Using Singular LQ Preview Regulation". In: *2011 11th IEEE-RAS International Conference on Humanoid Robots*. 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2011). IEEE, Oct. 2011, pp. 13–18.

[Kajita2003] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. "Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point". In: *2003 IEEE International Conference on Robotics and Automation*. 2003 IEEE International Conference on Robotics and Automation. Vol. 2. Sept. 2003, 1620–1626 vol.2.

[Kaneko2011] Kenji Kaneko, Fumio Kanehiro, Mitsuharu Morisawa, Kazuhiko Akachi, Go Miyamori, Atsushi Hayashi, and Noriyuki Kanehira. "Humanoid Robot HRP-4 - Humanoid Robotics Platform with Lightweight and Slim Body". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and*

Systems. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. Sept. 2011, pp. 4400–4407.

[Kang2023] Inseung Kang, Reese R. Peterson, Kinsey R. Herrin, Anirban Mazumdar, and Aaron J. Young. "Design and Validation of a Torque-Controllable Series Elastic Actuator-Based Hip Exoskeleton for Dynamic Locomotion". In: *Journal of Mechanisms and Robotics* 15.2 (Apr. 1, 2023), p. 021007.

[Katayama2022] Sotaro Katayama and Toshiyuki Ohtsuka. "Whole-Body Model Predictive Control with Rigid Contacts via Online Switching Time Optimization". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Oct. 2022, pp. 8858–8865.

[Khadiv2016] Majid Khadiv, Alexander Herzog, S. Ali. A. Moosavian, and Ludovic Righetti. "Step Timing Adjustment: A Step toward Generating Robust Gaits". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). Nov. 2016, pp. 35–42.

[Koenemann2015] J. Koenemann, Y. Tassa, A. Del Prete, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard. "Whole-Body Model-Predictive Control Applied to the HRP-2 Humanoid". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2015, pp. 3346–3351.

[Koolen2016] Twan Koolen, Sylvain Bertrand, Gray Thomas, Tomas de Boer, Tingfan Wu, Jesper Smith, Johannes Englsberger, and Jerry Pratt. "Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas". In: *Int. J. Human. Robot.* 13.01 (Mar. 2016), p. 1650007.

[Kour2019] Navleen Kour, Sunanda, and Sakshi Arora. "Computer-Vision Based Diagnosis of Parkinson's Disease via Gait: A Survey". In: *IEEE Access* PP (Oct. 28, 2019), pp. 1–1.

[Krantz2012] Steven G. Krantz and Harold R. Parks. *The Implicit Function Theorem: History, Theory, and Applications*. Birkhauser Boston, Inc., Boston, MA, Nov. 9, 2012. 173 pp.

[Kuindersma2016] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. "Optimization-Based Locomotion Planning, Estimation, and Control Design for the Atlas Humanoid Robot". In: *Auton Robot* 40.3 (Mar. 2016), pp. 429–455.

[Lang2015] C.E. Lang, K.R. Lohse, and R.L. Birkenmeier. "Dose and Timing in Neurorehabilitation: Prescribing Motor Therapy after Stroke". In: *Current Opinion in Neurology* 28.6 (2015), pp. 549–555.

[Li2012] Zhibin Li, Bram Vanderborght, Nikos G. Tsagarakis, Luca Colasanto, and Darwin G. Caldwell. "Stabilization for the Compliant Humanoid Robot COMAN Exploiting Intrinsic and Controlled Compliance". In: *2012 IEEE International Conference on Robotics and Automation*. 2012 IEEE International Conference on Robotics and Automation. May 2012, pp. 2000–2006.

[Liberzon2012] Daniel Liberzon. *Calculus of Variations and Optimal Control Theory*. Princeton University Press. 2012.

[Lohse2014] Keith R. Lohse, Catherine E. Lang, and Lara A. Boyd. "Is More Better? Using Metadata to Explore Dose–Response Relationships in Stroke Rehabilitation". In: *Stroke* 45.7 (July 2014), pp. 2053–2058.

[Maciejasz2014] Paweł Maciejasz, Jörg Eschweiler, Kurt Gerlach-Hahn, Arne Jansen-Troy, and Steffen Leonhardt. "A Survey on Robotic Devices for Upper Limb

Rehabilitation". In: *J NeuroEngineering Rehabil* 11.1 (Dec. 2014), p. 3.

[Mansard2009] Nicolas Mansard, Olivier Stasse, Paul Evrard, and Abderrahmane Kheddar. "A Versatile Generalized Inverted Kinematics Implementation for Collaborative Working Humanoid Robots: The Stack Of Tasks". In: *2009 International Conference on Advanced Robotics*. 2009 International Conference on Advanced Robotics. June 2009, pp. 1–6.

[Mansard2018] N. Mansard, A. DelPrete, M. Geisert, S. Tonneau, and O. Stasse. "Using a Memory of Motion to Efficiently Warm-Start a Nonlinear Predictive Controller". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 2986–2993.

[Martínez2018] Andrés Martínez, Brian Lawson, and Michael Goldfarb. "A Controller for Guiding Leg Movement During Overground Walking With a Lower Limb Exoskeleton". In: *IEEE Transactions on Robotics* 34.1 (Feb. 2018), pp. 183–193.

[Martínez2019] Andrés Martínez, Brian Lawson, Christina Durrough, and Michael Goldfarb. "A Velocity-Field-Based Controller for Assisting Leg Movement During Walking With a Bilateral Hip and Knee Lower Limb Exoskeleton". In: *IEEE Transactions on Robotics* 35.2 (Apr. 2019), pp. 307–316.

[Mastalli2022] Carlos Mastalli, Saroj Prasad Chhatoi, Thomas Corbères, Steve Tonneau, and Sethu Vijayakumar. *Inverse-Dynamics MPC via Nullspace Resolution*. Sept. 12, 2022. arXiv: 2209.05375 [cs, eess]. preprint.

[Mifsud2016] Alexis Mifsud, Mehdi Benallegue, and Florent Lamiraux. "Stabilization of a Compliant Humanoid Robot Using Only Inertial Measurement Units with a Viscoelastic Reaction Mass Pendulum Model". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Oct. 2016, pp. 5405–5410.

[Mombaur2001] Katja Mombaur, Hans Georg Bock, Johannes P. Schloder, and Richard W. Longman. "Human-like Actuated Walking That Is Asymptotically Stable without Feedback". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*. Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation. Vol. 4. May 2001, 4128–4133 vol.4.

[Moore2020] Jennifer L. Moore, Jan E. Nordvik, Anne Erichsen, Ingvild Rosseland, Elisabeth Bø, T. George Hornby, null null, Tonje Barkenæs, Hanne Bratlie, Miriam Byhring, Ingvild Grimstad, Magnus Hågå, Joakim Halvorsen, Chris Henderson, Julia-Aneth Mbalilaki, Stein-Arne Rimehaug, Kirsten Sæther, Thomas Tomren, and Karen Vergoossen. "Implementation of High-Intensity Stepping Training During Inpatient Stroke Rehabilitation Improves Functional Outcomes". In: *Stroke* 51.2 (Feb. 2020), pp. 563–570.

[Murray2015] Spencer A. Murray, Kevin H. Ha, Clare Hartigan, and Michael Goldfarb. "An Assistive Control Approach for a Lower-Limb Exoskeleton to Facilitate Recovery of Walking Following Stroke". In: *IEEE Trans. Neural Syst. Rehabil. Eng.* 23.3 (May 2015), pp. 441–449.

[Perry2010] Jacquelin Perry and Judith M. Burnfield. *Gait Analysis: Normal and Pathological Function*. 2nd ed. SLACK, 2010. 551 pp.

[Pinto-Fernandez2020] David Pinto-Fernandez, Diego Torricelli, Maria del Carmen Sanchez-Villamanan, Felix Aller, Katja Mombaur, Roberto Conti, Nicola Vitiello, Juan C. Moreno, and Jose Luis Pons. "Performance Evaluation of Lower Limb Exoskeletons: A Systematic Review". In: *IEEE Transactions on Neural Systems*

and Rehabilitation Engineering 28.7 (July 2020), pp. 1573–1583.

[Ponton2021] Brahayam Ponton, Majid Khadiv, Avadesh Meduri, and Ludovic Righetti. "Efficient Multicontact Pattern Generation With Sequential Convex Approximations of the Centroidal Dynamics". In: *IEEE Transactions on Robotics* 37.5 (Oct. 2021), pp. 1661–1679.

[Quintero2012] Hugo A. Quintero, Ryan J. Farris, and Michael Goldfarb. "A Method for the Autonomous Control of Lower Limb Exoskeletons for Persons With Paraplegia". In: *Journal of Medical Devices* 6.4 (Dec. 1, 2012), p. 041003.

[Reher2016] Jacob Reher, Eric A. Cousineau, Ayonga Hereid, Christian M. Hubicki, and Aaron D. Ames. "Realizing Dynamic and Efficient Bipedal Locomotion on the Humanoid Robot DURUS". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016 IEEE International Conference on Robotics and Automation (ICRA). 2016, pp. 1794–1801.

[Samson1991] Claude Samson, Bernard Espiau, and M. Le Borgne. *Robot Control: The Task Function Approach*. Vol. 9. Cambridge University Press, Dec. 1991.

[Sanchez Restrepo2018] Susana Sanchez Restrepo. "Intuitive, Iterative and Assisted Virtual Guides Programming for Human-Robot Comanipulation". PhD thesis. Université de Toulouse, Université Toulouse III - Paul Sabatier, Feb. 1, 2018.

[Scianca2020] Nicola Scianca, Daniele De Simone, Leonardo Lanari, and Giuseppe Oriolo. "MPC for Humanoid Gait Generation: Stability and Feasibility". In: *IEEE Transactions on Robotics* 36.4 (Aug. 2020), pp. 1171–1188.

[Shahrokhshahi2022] Ahmadreza Shahrokhshahi, Majid Khadiv, Ali Taherifar, Saeed Mansouri, Edward J. Park, and Siamak Arzanpour. "Sample-Efficient Policy Adaptation for Exoskeletons Under Variations in the Users and the Environment". In: *IEEE Robot. Autom. Lett.* 7.4 (Oct. 2022), pp. 9020–9027.

[Siekmann2021] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. "Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning". In: *Robotics: Science and Systems XVII*. Robotics: Science and Systems 2021. Robotics: Science and Systems Foundation, July 12, 2021.

[Singh2022] Rohan P. Singh, Mehdi Benallegue, Mitsuharu Morisawa, Rafael Cisneros, and Fumio Kanehiro. "Learning Bipedal Walking On Planned Footsteps For Humanoid Robots". In: *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids). Nov. 2022, pp. 686–693.

[Skinner1979] Nicholas F. Skinner. "Learned Helplessness: Performance as a Function of Task Significance". In: *The Journal of Psychology* 102.1 (May 1, 1979), pp. 77–82.

[Smaldone2021] Filippo M. Smaldone, Nicola Scianca, Leonardo Lanari, and Giuseppe Oriolo. "Feasibility-Driven Step Timing Adaptation for Robust MPC-Based Gait Generation in Humanoids". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 1582–1589.

[Sugihara2009] T. Sugihara. "Standing Stabilizability and Stepping Maneuver in Planar Bipedalism Based on the Best COM-ZMP Regulator". In: *2009 IEEE International Conference on Robotics and Automation*. 2009 IEEE International Conference on Robotics and Automation (ICRA). IEEE, May 2009, pp. 1966–1971.

[Vallery2008] Heike Vallery, Jan Veneman, Edwin van Asseldonk, Ralf Ekkelenkamp, Martin Buss, and Herman van Der Kooij. "Compliant Actuation of Rehabilitation Robots". In: *IEEE Robot. Automat. Mag.* 15.3 (Sept. 2008), pp. 60–69.

[Vallery2009] Heike Vallery, Alexander Duschau-Wicke, and Robert Riener. "Generalized Elasticities Improve Patient-Cooperative Control of Rehabilitation Robots". In: *2009 IEEE International Conference on Rehabilitation Robotics*. The Community (ICORR). IEEE, June 2009, pp. 535–541.

[Vigne2021] Matthieu Vigne. "Estimation and Control of the Deformations of an Exoskeleton Using Inertial Sensors". PhD thesis. Université Paris Sciences et Lettres, 2021.

[Vigne2022] Matthieu Vigne, Antonio El Khoury, Marine Pétriaux, Florent Di Meglio, and Nicolas Petit. "MOVIE: A Velocity-Aided IMU Attitude Estimator for Observing and Controlling Multiple Deformations on Legged Robots". In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), pp. 3969–3976.

[Villa2022] Nahuel A. Villa, Pierre Fernbach, Maximilien Naveau, Guilhem Saurel, Ewen Dantec, Nicolas Mansard, and Olivier Stasse. "Torque Controlled Locomotion of a Biped Robot with Link Flexibility". In: *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids). IEEE, Nov. 28, 2022, pp. 9–16.

[vStryk1993] Oskar von Stryk. "Numerical Solution of Optimal Control Problems by Direct Collocation". In: *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*. ISNM International Series of Numerical Mathematics. Birkhäuser, 1993, pp. 129–143.

[Wächter2006] Andreas Wächter and Lorenz T. Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming". In: *Math. Program.* 106.1 (Mar. 2006), pp. 25–57.

[Westervelt2007] Eric Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Taylor & Francis Group, 2007.

[Wieber2006a] Pierre-Brice Wieber. "Holonomy and Nonholonomy in the Dynamics of Articulated Motion". In: *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*. Lecture Notes in Control and Information Sciences. Springer, 2006, pp. 411–425.

[Wieber2006b] Pierre-Brice Wieber. "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations". In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*. 2006 6th IEEE-RAS International Conference on Humanoid Robots. IEEE, Dec. 2006, pp. 137–142.

[Wieber2016] Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. "Modeling and Control of Legged Robots". In: *Springer Handbook of Robotics*. Springer International Publishing, 2016, pp. 1203–1234.

[Wieber2019] Pierre-Brice Wieber. "Model Predictive Control". In: *Humanoid Robotics: A Reference*. Springer Netherlands, 2019, pp. 1077–1097.

[Winter2009] David A. Winter. *Biomechanics and Motor Control of Human Movement*. 4th ed. Wiley, 2009. 370 pp.

[Wool1980] R. N. Wool, D. Siegel, and P. R. Fine. "Task Performance in Spinal Cord Injury: Effect of Helplessness Training". In: *Arch Phys Med Rehabil* 61.7 (July 1980), pp. 321–325.

[Yokoi2003] Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Shuji Kajita, and Hirohisa Hirukawa. "Experimental Study of Biped Locomotion of Humanoid Robot HRP-1S". In: *Experimental Robotics VIII*. Springer Tracts in Advanced Robotics. Springer, 2003, pp. 75–84.

[Young2017] Aaron J. Young and Daniel P. Ferris. "State of the Art and Future Directions for Lower Limb Robotic Exoskeletons". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.2 (Feb. 2017), pp. 171–182.

RÉSUMÉ

Les exosquelettes auto-équilibrés, grâce à leurs algorithmes de contrôle qui permettent une marche autonome et stable, offrent aux personnes handicapées des capacités de marche retrouvées. Cette thèse vise à permettre aux patients d'utiliser ces systèmes à des fins de rééducation selon le concept de rééducation active de la marche. Selon notre interprétation, ce concept implique de redonner une certaine liberté de mouvement aux patients. Cet objectif est en conflit avec la garantie de la sécurité, qui implique de prendre le contrôle du mouvement. Le principal sujet abordé dans cette thèse est le défi que représente la conciliation de ces deux objectifs.
Notre solution est double : donner de la liberté au patient et en garantissant l'équilibre. Tout d'abord, nous introduisons une division dans l'architecture de contrôle existante, permettant aux patients de participer activement au mouvement de la jambe de vol. Parallèlement, l'algorithme de contrôle de la jambe de support régule les forces de contact. Deuxièmement, nous résolvons de manière embarquée des problèmes de contrôle optimal en temps final libre sur un modèle réduit afin de générer des trajectoires stabilisables. La durée de ces trajectoires sert à déterminer le moment où le système doit reprendre le contrôle de l'exécution de la marche.
Le manuscrit présente les mises à jour nécessaires de l'architecture de contrôle ainsi que le développement de l'algorithme de gestion de la sécurité. Des résultats expérimentaux sont fournis pour évaluer l'efficacité de notre approche lors d'exercices de rééducation de la marche en utilisant une trajectoire sans et une trajectoire avec déroulé du pied.

MOTS CLÉS

Rééducation robotique, Exosquelette, Stabilisation, Marche bipède, Contrôle optimal

ABSTRACT

Self-balanced exoskeletons offer individuals with disabilities recovered ambulation capabilities thanks to their embedded control algorithms which realize autonomous and stable walking. This thesis is concerned with enabling patients to use these systems for rehabilitation purposes according to the concept of active gait rehabilitation. In our interpretation, this concept entails giving back some freedom of motion to the patients. This objective is conflicting with ensuring safety, which implies taking control over the motion. The main topic addressed in the thesis is the challenge of reconciling these two objectives.
Our solution is twofold: empower the patient and monitor safety. Firstly, we introduce a split in the control architecture, allowing patients to actively participate in the swing leg motion. Concurrently, the control algorithm of the support leg regulates the unilateral contact forces. Secondly, we solve optimal control problems in free final time on a reduced model to generate stabilizable trajectories. The duration of these trajectories serves as a determinant for when the control system should take back control over the execution of the walking pattern.
The manuscript presents the necessary control architecture updates along with the development of the safety management algorithm. Experimental results are provided to assess the effectiveness of our approach in both flat-foot and foot-rolling walking exercises.

KEYWORDS

Rehabilitation Robotics, Exoskeleton, Stabilization, Humanoid and Bipedal Locomotion, Optimal control