Estimation and control of the deformations of an exoskeleton using inertial sensors

Matthieu Vigne

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Estimation et contrôle des déformations d'un exosquelette à partir de centrales inertielles

Estimation and Control of the Deformations of an Exoskeleton using Inertial Sensors

Soutenue par
Matthieu VIGNE
Le 06 Octobre 2021

École doctorale n°621
Ingénierie des Systèmes, Matériaux, Mécanique, Energétique

Spécialité
Mathématique et Automatique

Composition du jury :

Pascal MORIN Pr., Sorbonne Université, ISIR	*Président, rapporteur*
Jaeheung PARK Pr., Seoul National University	*Rapporteur*
Mehdi BENALLEGUE Dr, AIST Tsukuba	*Examinateur*
Antonio EL KHOURY Dr, Wandercraft	*Examinateur*
Nicolas MANSARD Directeur de recherche, LAAS-CNRS	*Examinateur*
Marine PETRIAUX Ingénieure, Wandercraft	*Examinateur*
Florent DI MEGLIO Dr, MINES ParisTech	*Co-directeur de thèse*
Nicolas PETIT Pr., MINES ParisTech	*Directeur de thèse*

MINES ParisTech | PSL

Abstract

This thesis addresses the general problem of the compensation of deformations on a walking robot, and considers it in the particular context of the medical exoskeleton Atalante. Structural deformations are unavoidable in anthropomorphic robots with long legs and a heavy torso. This is even more true for an exoskeleton, which has to support the weight of the user. Meanwhile, the presence of an uninstrumented human user leads to significant disturbances, that limit the performance of feedforward corrections.

We propose a sensor-based methodology, that feedbacks the data of several Inertial Measurement Units (IMUs) onto the actuators, in order to mitigate the effect of the multiple flexibilities on Atalante. To that end, we kinematically model the deformations as extra spherical joints. We study several observers to estimate the rotations induced by the flexibilities, by estimating the attitude of the IMUs. Three observers, based on various models, are considered: a (naive) zero-on-average acceleration model, a kinematic model and a dynamic model. We conclude that the best results are obtained by relying only on the kinematic model of the robot to build an approximate velocity measurement. This allows the design of an observer able to handle strong accelerations while being robust to the dynamical uncertainties linked to patient behavior.

These attitude estimates are then projected onto the joints of the robot, to adopt a joint elasticity model, which we use to perform decentralized high-gain feedback control. This methodology is experimentally validated on Atalante, where it yields improved disturbance rejection and improved trajectory tracking, enhancing the robustness of Atalante's walk with a user.

Résumé

Cette thèse porte sur la question générale de la compensation des déformations sur un robot marcheur, avec comme cas d'étude particulier l'exosquelette à usage médical Atalante. Les déformations des robots anthropomorphes sont des phénomènes indésirables résultant de leur conception mécanique. Ces robots sont en effet constitués de jambes longues devant soutenir un torse relativement massif. Ces effets sont d'autant plus présents sur un exosquelette, qui en plus de son propre poids doit supporter le poids de l'utilisateur. La présence d'un humain non instrumenté induit des perturbations significatives non mesurées, auxquelles il est difficile de s'adapter en temps-réel.

Nous proposons une approche en boucle fermée reposant sur des capteurs, qui effectue un retour d'état sur les signaux provenant de plusieurs centrales inertielles. A cette fin, nous modélisons cinématiquement les déformations par des liaisons rotules supplémentaires. Nous étudions plusieurs observateurs visant à estimer les rotations correspondantes, en reconstituant l'orientation des centrales inertielles. Trois observateurs, utilisant des modèles différents, sont étudiés: un modèle supposant que l'accélération moyenne des corps est nulle, un modèle cinématique et un modèle dynamique. Nous concluons que les meilleurs résultats sont obtenus en exploitant uniquement le modèle cinématique du robot afin de reconstruire une mesure approximative de la vitesse. Ceci permet la conception d'un observateur capable de supporter des accélérations élevées, tout en restant robuste à l'incertitude dynamique liée au comportement du patient.

Ces estimations d'attitude sont ensuite converties (par projection) sur les articulations du robot, comme dans un modèle d'actionneur élastique, et utilisées pour réaliser un contrôle par retour d'état grand gain décentralisé. L'ensemble de cette méthodologie est validé expérimentalement sur Atalante, où elle améliore le rejet de perturbation et le suivi de trajectoire, augmentant ainsi la robustesse de la marche d'Atalante avec utilisateur.

Remerciements

Quatre années de thèse, c'est finalement très vite passé ! Elle furent pleines de bons moments, et de raisons de dire merci.

Un grand merci à mes encadrants de m'avoir accompagné tout au long du trajet. A Nicolas, qui m'a aidé à prendre du recul et de la hauteur par rapport aux taches quotidiennes: tu as été pour un directeur de thèse très inspirant, et un modèle de chercheur. A Florent, toujours prêt à plonger avec moi dans les détails de l'analyse - et à déméler les nombreuses transformées de Fourier incompréhensibles que nous sort le robot ! A Antonio, avec qui j'ai partagé mes recherches au quotidien: tu as su m'écouter sans jamais me contraindre, m'aidant ainsi à formuler mon fil de pensées. A Marine, qui a réussi l'exploit de rejoindre cette thèse après trois ans de recherches, apportant un oeil neuf et un excellent accompagnement sur la dernière année. Merci à tous pour votre soutien et votre confiance.

Merci aux membres du jury, Mehdi Benallegue, Nicolas Mansard, Pascal Morin et Jaehung Park, pour l'intérêt que vous avez porté à mes travaux. Je remercie aussi Mehdi Benallegue pour nos échanges au cours de cette thèse: tes travaux ont été une riche source d'inspiration.

Merci à Wandercraft, pour m'avoir permis de faire cette thèse dans une super ambiance: à toute l'équipe, vous êtes géniaux ! Merci à Marco, Guillaume, Thomas et Yoann pour leur aide dans la réalisation d'expériences parfois insolites... Merci à Huy-Cong, Gustavo et Marine pour vous être déplacés jusqu'à La Courneuve pour de longues journées de motion capture: vous gagnez le droit d'être cité ici, ca valait le coup non ?

Bon courage à Omar, Alexis et Maxime pour la fin de votre thèse - et bienvenu à Fabio pour qui l'aventure commence !

Aucun animal réel n'a été blessé pendant la réalisation de cette thèse... mais je remercie mes vaillants compagnons, Alexis, Laurent, Huy-Cong, Anthony, Marco, Guillaume, Camille, Corentin et Victor, avec qui moults trolls, dragons et autres créatures fantastiques ont été occises !

Merci à tous les membres du CAS, ce fut un plaisir de travailler avec vous. La conversion aux jeux de société a été plus difficile, sous la surveil-

lance draconienne de Nicolas (même les grands hommes ont leurs défauts...). Heureusement que quelques cours d'analyse complexe loin de Paris sont venus égailler tout ca - merci Jean-Emmanuel !

Et bien évidemment, merci à tous mes amis, pour tous les bons moments qu'on a (et qu'on va!) partagé. Et à ma famille, pour tout l'amour et le soutien que vous me procurez.

Contents

Notations and useful formulas

The following pages give a brief overview of the notations used in the thesis, for quick reference. A detailed presentation and explanation of all these variables is provided as they are introduced throughout the thesis.

Vectors are written in bold font, matrices in capital letters.

Important frames

$\mathcal{W}$	World frame
$\mathcal{C}$	Contact foot frame
$\mathcal{L}$	Local frame, i.e. projection of the contact foot frame onto the ground

Spatial algebra

We define the skew-symmetric operator associated to the cross-product as

$$[.]_\times : \mathbb{R}^3 \to M_3(\mathbb{R}), \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \tag{1}$$

This operator is such that

$$\forall (\boldsymbol{a}, \boldsymbol{b}) \in (\mathbb{R}^3)^2, \boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b} \tag{2}$$

Given two orthogonal frames A and B, we write:

${}^B\boldsymbol{p}_A \in \mathbb{R}^3$	Position of the origin of frame A with respect to frame B, in frame B
${}^BR_A \in SO(3)$	Orientation of frame A with respect to frame B
${}^BM_A \in SE(3)$	Homogeneous transform matrix from frame A to frame B

$\boldsymbol{\omega}_{A/B} \in \mathbb{R}^3$	Angular velocity of frame A with respect to frame B, expressed in frame A

We recall that the angular velocity is defined by the following identity:

$$ {}^{B}\dot{R}_A = {}^{B}R_A[\boldsymbol{\omega}_{A/B}]_\times \tag{3} $$

We recall the definition of the roll-pitch-yaw decomposition, as the decomposition in Euler angle (Tait-Bryan convention) of a rotation matrix about axes ZYX. This writes:

$$ R \triangleq R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{4} $$

with R_x, R_y, R_z the rotations about axes first ($\boldsymbol{e}_x$), second ($\boldsymbol{e}_y$), and third ($\boldsymbol{e}_z$) axis of the orthonormal basis of reference. α, β, γ are called the roll, pitch and yaw angle respectively. This decomposition is singular when $\beta \cong \frac{\pi}{2}[\pi]$.

System coordinates

$\boldsymbol{p}_{ff}, R_{ff}$	Position and orientation of the freeflyer joint of the robot
$\boldsymbol{\theta}$	Vector of encoder measurements
$\boldsymbol{q}_r \triangleq (\boldsymbol{p}_{ff}, R_{ff}, \boldsymbol{\theta})$	Generalized position coordinates of the robot, assuming full rigidity
$D_j \in SO(3)$	Orientation of the spherical joints (deformations) of the extended flexible model
$\boldsymbol{q}_f \triangleq (\boldsymbol{p}_{ff}, R_{ff}, \boldsymbol{\theta}, D_j)$	Generalized position coordinates of the robot under the extended flexible model
$\boldsymbol{q}$	Joint output angle in the joint deformation model
$\boldsymbol{\alpha} \triangleq \boldsymbol{q} - \boldsymbol{\theta}$	Joint deformation angle
f_r	Forward kinematic model, assuming full rigidity: $f_r(\boldsymbol{q}_r)$ gives the pose of the robot bodies in the world frame)
f_f	Forward kinematic model of the extended flexible model

Chapter 1

Context

Chapitre 1 - Contexte: *Ce chapitre introductif présente le problème considéré dans cette thèse, à savoir la compensation des déformations au sein de structures robotiques, en particulier humanoïdes. L'utilisation d'une approche reposant sur des capteurs (par opposition à une pré-compensation reposant sur un modèle prédictif) est justifiée par les spécificités du dispositif expérimental considéré : l'exosquelette médical Atalante. La présence non instrumentée d'un humain induit en effet une grande incertitude sur les efforts exercés sur le système. Nous proposons d'utiliser des centrales inertielles pour mesurer (estimer) les déformations ayant lieu autour de plusieurs points de la structure d'Atalante, et pour les compenser.*

La seconde partie du chapitre décrit l'exosquelette Atalante, et explique son système de contrôle existante. Ces informations constituent la base de notre étude.

1.1 Introduction

1.1.1 Deformations in humanoid robots

As robotics technology develops, humanoids robots are becoming increasingly common. In a world designed for humans, having a human-like shape is a great asset for a robot whose task is to collaborate with a human, or to replace human workers in dangerous or tedious tasks.

A common practice for controlling robots, and humanoids in particular, is to servo the position of the actuators to a reference trajectory [Finet17, Westervelt18]. In this *position control* approach, the robot is assumed to be fully rigid, such that controlling the position of the end effector is equivalent to controlling the position of the joints.

However, one challenge in applying this method to humanoid robots is the presence of deformations in the structure. While these deformations may sometimes be the result of a voluntary design choice, e.g. to protect the structure from impacts [Hirai98] [De Magistris16], they are often the consequence of an unavoidable trade off in weight distribution and reduction [Stasse19]. Humanoid robots indeed feature long, thin legs, that must carry a rather heavy torso. To withstand the weight of the torso, the legs need to be very stiff, and thus quite heavy. On the contrary, to ease dynamic walking motion, which require accelerating the swing leg forward, the weight and inertia of the leg are desired to be as small as possible. The resolution of this conflict often results in robot designs with a non-negligible amount of deformation, which, if unaccounted for, decrease the overall performance of the robot [Ficht21] [Jung17]. This effect has been reported on various robots in the literature, such as those pictured in Figure 1.1.

These deformations are often taking place near the robot joints, which represent weaker points of the structure: namely, these deformations are undesired rotations of the robot's bodies with respect to each other. Being small rotations, these deformations are usually modeled by linear elasticity laws, i.e. springs of constant stiffness [Spong87].

Most often, existing robots do not carry any sensor to directly measure the state of these deformations. Consequently, the most common solution is to rely on model-based offsets to compensate the static effect of the deformation. In details, the general idea is to assume knowledge of a feedforward torque term $\boldsymbol{\tau}$: assuming that the deformation is a linear spring of stiffness k, the resulting deformation at rest is $\frac{\tau}{k}$. This value can then be applied as an offset to the position being tracked by the robot. This idea was originally presented in [Tomei91]; it was applied with various modifications and improvements on several humanoid robots: Atlas [Johnson15], DURUS [Reher16], THORMANG [Kim16].

While these sensor-less approaches can bring significant improvements to system performance, they often remain limited to countering the static deflection, without offering a possibility to efficiently damp oscillations. Furthermore, these approaches require an accurate model of the robot to compute a meaningful feedforward term $\boldsymbol{\tau}$, and a good knowledge of the stiffness parameter of the joints k - not even considering the fact that the linear spring model itself might be erroneous, and seriously jeopardize the compensation effectiveness.

In this thesis, we examine a different path. We use on-board IMUs to enable a direct measurement of the deformations, which can be used for closed-loop compensation. Such feedback-based approach has the potential of countering both the static and the dynamic effect of the deformations,

(a) ATLAS [Johnson15]

(b) THORMANG [Kim16]

(c) DURUS [Reher16]

(d) HRP-2 [Benallegue14]

Figure 1.1: An example of humanoid or biped robots on which the presence of significant deformations has been reported.

while being robust to modeling errors and parameter uncertainties. Meanwhile, MEMS IMU sensors are extremely cheap nowadays, small and easy to mount on existing robot hardware. Being sensitive to a change in orientation, the use of several IMUs located throughout the structure can, in principle, provide access to all internal angular deformations. This question is not as straightforward as it may seem, because IMUs are impacted by fictitious forces: the acceleration created by the motion of the robot and the deformations.

The use of IMU sensors on humanoid robot is quite common: virtually all humanoid robots use an IMU in their torso, with the aim of estimating the global attitude of the robot in space, i.e. the freeflyer coordinates [Bloesch12, Khandelwal13, Flayols17, Hartley18]. Using an IMU specifically to estimate a

deformation, and to stabilize it using feedback, has been the topic of several recent contributions from the LAAS laboratory on the HRP-2 humanoid robot, notably [Benallegue14, Benallegue15, Mifsud16]. This robot features a single compliant element at the ankle: its angle is estimated using the robot's IMU located in the torso, assuming that the rest of the structure is rigid.

However, many robots, including the exoskeleton Atalante described bellow, exhibit several points of deformation, at the ankle but also at the hips. In this context, a single IMU is not sufficient: several sensors must be placed to estimate each deformation individually. To our knowledge, this question of using several IMUs to estimate and control the internal flexibilities of a humanoid robot has not been addressed in previous works.

1.1.2 Use case: the exoskeleton Atalante

Our motivation for using a feedback-based approach is driven by the use case of this thesis: indeed, the system under study is not a humanoid robot, but the lower-limb medical exoskeleton Atalante (Figure 1.3).

The purpose of such an exoskeleton is to assist patients in performing standing and walking motion. Notably, it aims at giving back mobility and autonomy to patients suffering from walk disabilities, such as partial or complete paraplegia or hemiplegia, enabling them to stand and walk autonomously.

Pursuing this medical objective has been a long quest: works on this topic can be traced back as early as the 70s [Vukobratovic74], while commercial products have been available as early as 2010. Some well-known examples are the EksoGT (Ekso Bionics), the ReWalk (ReWalk Robotics) or the Indego (Parker Hannifin), illustrated in Figure 1.2. All these devices enable a paraplegic patient to stand and walk, though they require the use of crutches to keep balance. Their use through the years has shown considerable benefits for patients, both in terms of physical health (improved cardiac, respiratory and gastrointestinal function; decrease in pain and spasticity; improved sitting balance) as well as positive psychological effects: notably, standing up enables a direct eye-to-eye contact [Geigle17].

Nevertheless, the use of crutches reduces the realism of the walk, while leading to muscular fatigue of the arms and upper body. In 2011, REX Bionics commercialized the REX, a self-balanced exoskeleton. While it enables a variety of upper-body exercises, it is only able to perform slow, quasi-static walking.

In 2019, Atalante was certified as a medical device for the European market. Thus, it became the first commercial exoskeleton to allow for hands-

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

(b) Rewalk (https://rewalk.com/)

(c) Indego (http://www.indego.com)

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

Figure 1.2: An example of several commercial exoskeletons.

free, dynamic walking of paraplegic and hemiplegic patients, with realistic "human-like" walking gaits [Harib18]. The work done in this thesis is centered on this particular exoskeleton, being realized in partnership with Wandercraft, the company who designs and manufactures this exoskeleton.

A lower-limb exoskeleton shares many similarities with an armless humanoid robot. By design, it possesses the same anthropomorphic structure, since the kinematics of its legs must be compatible with that of the patient. Thus, with regard to deformations, the same difficulty in design arises: long, thin legs must support a heavy torso. This is particularly true for Atalante: being designed to be used without crutches, the structure must withstand its own weight (about 80 kg), plus the full weight of the patient.

Thus, like the humanoid robots presented in Figure 1.1, Atalante exhibits a significant amount of deformation. This effect is illustrated in a simple

Figure 1.3: The exoskeleton Atalante (https://www.wandercraft.eu/), walking with a paraplegic patient in a rehabilitation center .

experiment shown in Figure 1.4: there, the exoskeleton is asked to remain still, while one pushes or pulls on the foot. This results in a total displacement of about 14 cm of the foot, while only 4 cm of motion are measured by the joint encoders. The remaining motion can thus be attributed to the presence of deformations.

Working with an exoskeleton however presents a fundamental difference with classical robots: *the presence of a human in the loop*. The dynamics of the system indeed depend both on the exoskeleton motion, and that of the user. Since the weight of the user is comparable with that of the device, they can drastically impact the dynamics by moving their center of mass - thereby contributing to stabilizing or destabilizing the walk. However, while the patient is obviously not controlled, they are also not instrumented: as such, it is not possible to know the precise forces and moments applied onto the exoskeleton by the user. This causes a large uncertainty in the overall dynamics of the system, to which the control approach must be robust: indeed, the exoskeleton must be able to walk with various patients of diverse height and weight, but also with various pathologies (hemiplegia, paraplegia...). This uncertain nature of the disturbances in the dynamics translates into a design requirement for the present question of compensating the effect

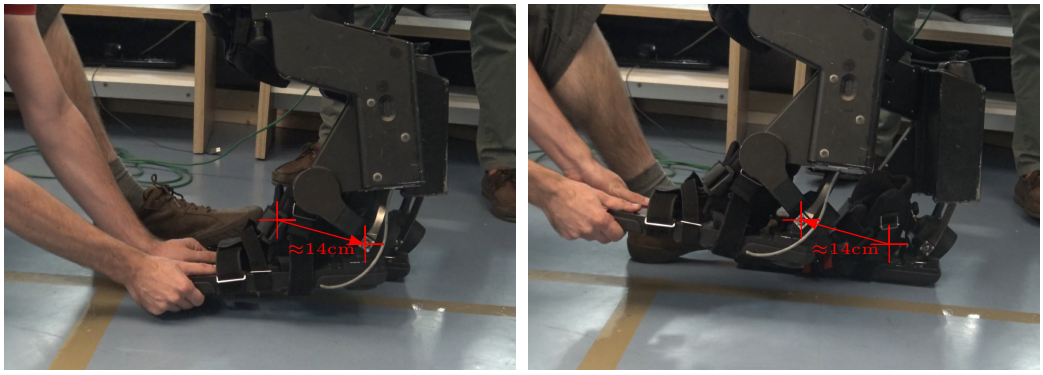


Figure 1.4: A direct observation of the presence of deformations: when pushing/pulling on the swing foot, the observed displacement (about 14 cm) is much larger than the motion measured by the encoders (only 4 cm), indicating the extra degrees of freedom, i.e. deformations.

of deformations: rather than using an ill-known model as feed-forward, we seek a sensor-based, feedback approach.

1.1.3 Problem under consideration in this thesis

The problem that we address in this thesis is formulated below.

How can strapdown IMU sensors be used to provide a feedback-based compensation of the deformations of a robot, in the hope of improving its positioning accuracy robustly with respect to uncertainties in the system dynamics and unknown disturbances? In particular, how can this be done on the exoskeleton Atalante, walking with a user?

Thesis outline

While the work carried out and presented in this manuscript is fully centered on Atalante, the methodology we propose is generic enough to be applicable to a variety of systems as well. We propose two main elements and contributions:

- an estimator of the attitude of the strapdown IMUs, robust to the presence of linear accelerations at the center of the IMU while walking. This observer is designed to rely only on the kinematic model of the robot and its deformation, but does not require any information about the system dynamics (in particular we do not exploit any constitutive equation to dynamically model the deformations).

- a modification of existing PID-based low-level controllers used on Atalante, to integrate a feedback term on the estimated deformation angles and velocities.

The outline of this manuscript follows the development of this methodology on Atalante. In the second part of this chapter, we give a more detailed presentation of this exoskeleton. In Chapter 2, we show how to model the deformations on this robot, to provide a convenient framework for estimation and control. Chapter 3 then focuses on the development of an observer to estimate these deformations, the output of which is used in Chapter 4 for feedback control.

Publications in this thesis

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

- Matthieu Vigne, Antonio El Khoury, Matthieu Masselin, Florent Di Meglio and Nicolas Petit. **Estimation of Multiple Flexibilities of an Articulated System Using Inertial Measurements**, *IEEE Conference on Decision and Control (CDC), 2018*

- Matthieu Vigne, Antonio El Khoury, Florent Di Meglio and Nicolas Petit. **State Estimation for a Legged Robot With Multiple Flexibilities Using IMUs: A Kinematic Approach**, *IEEE Robotics and Automation Letters, 2019 (presented in conference at Humanoids 2019)*

- Matthieu Vigne, Antonio El Khoury, Florent Di Meglio and Nicolas Petit. **Improving Low-Level Control of the Exoskeleton Atalante in Single Support by Compensating Joint Flexibility**, *IEEE Conference on Intelligent Robots and Systems (IROS), 2020*

- Matthieu Vigne, Marine Pétriaux, Antonio El Khoury, Florent Di Meglio and Nicolas Petit. **MOVIE: a Velocity-aided IMU Attitude Estimator for Observing and Controlling Multiple Deformations on Legged Robots**, *IEEE Robotics and Automation Letters, 2022 (presented in conference at ICRA 2022)*

A video describing the main experiments done in this thesis is available at https://youtu.be/c2Vdx81iu1A

1.2 A detailed description of the exoskeleton Atalante

In this section, we present the exoskeleton Atalante, and give a brief overview of its existing control architecture. Following the streamlined approach used for the control of Atalante, which has served as a basis to develop this architecture, all the elements presented in this section rely on the assumption of full rigidity.

1.2.1 General description

1.2.1.1 Vocabulary

To describe the kinematics of a walking human, the following planes are defined:

- the *sagittal plane* is a vertical plane that cuts the body into right and left halves. Walking motion mostly happens in this plane.
- the *frontal plane* is the vertical plane orthogonal to the sagittal plane, thereby defining a front and a back - used to describe lateral motion.
- the *transverse plane* is a plane parallel to the ground, separating the body into a superior and inferior part.

A representation of these planes is given in Figure 1.5. We associate to these planes the following coordinate system: $\boldsymbol{e}_z$ is the vertical axis (aligned with gravity), and $\boldsymbol{e}_x$ the forward-pointing axis, in the sagittal plane. $\boldsymbol{e}_y$ thus lies in the frontal plane, pointing from right to left, to form a direct orthogonal frame.

A biped robot is said to be in *single support* when only one foot is touching the ground, and in *double support* when both feet are in contact. We call *stance leg* a leg in contact with the ground, and *swing leg* a leg not in contact with the ground.

1.2.1.2 Hardware description

To replicate human motion, the kinematics of Atalante are similar to those of the human legs. Namely, Atalante is composed of 12 revolute joints, six for each leg, as represented in Figure 1.6. For one leg, it thus consists of

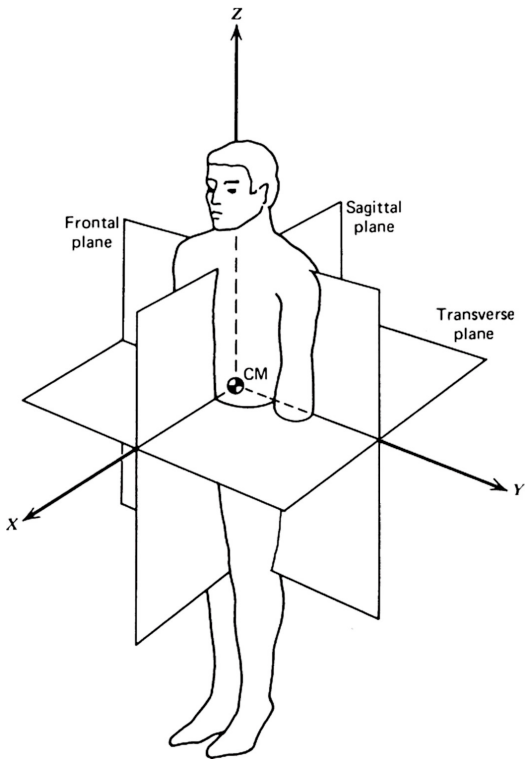


Figure 1.5: Definition of the sagittal, frontal and transverse plane (adapted from [Winter09]).

- three revolute joints allow for an arbitrary 3D rotation of the hip: they are called frontal, transverse and sagittal hip joint respectively (the name referring to the plane orthogonal to the joint axis, i.e. the plane in which they enable motion).

- a single sagittal knee joint.

- two concurrent revolute joints for the ankle enabling motion in the sagittal and frontal plane. The frontal motion of the human ankle is actually performed about an upward-angled axis, called Henke's axis, which makes an angle of 38° with the horizontal transverse plane. This kinematics is replicated on Atalante, hence these two pivot joints are called sagittal ankle and Henke ankle respectively.

Each joint is independently actuated by an electric motor, and instrumented by a joint encoder: the vector of joint encoder reading is noted $\boldsymbol{\theta} \in \mathbb{R}^{12}$.

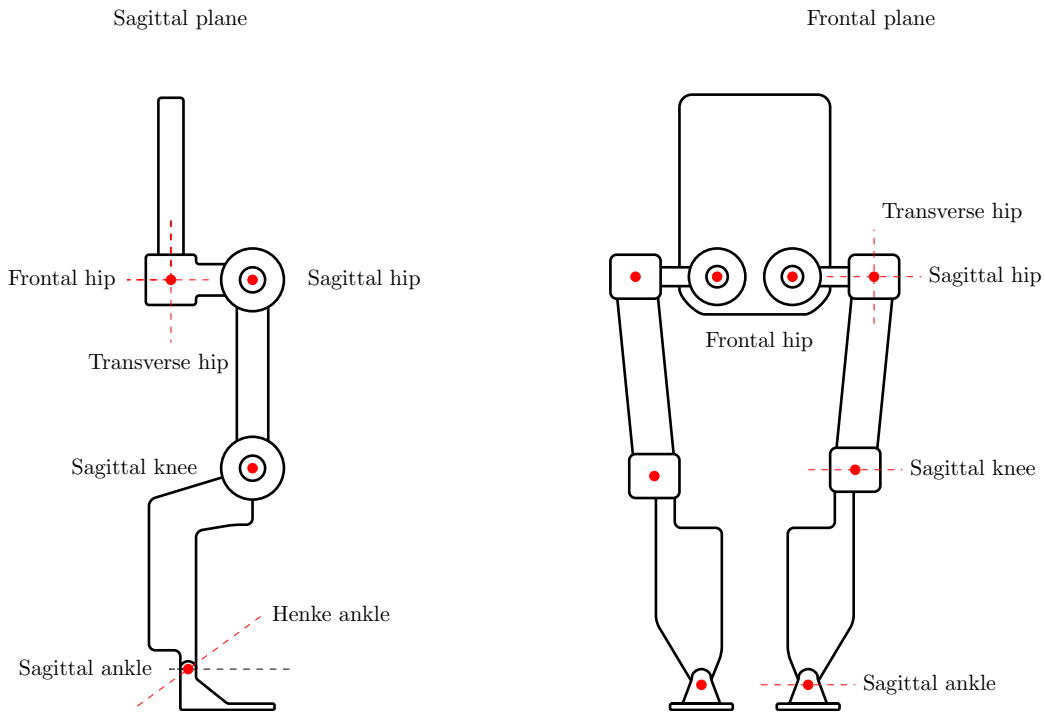


Figure 1.6: Kinematic model of Atalante.

Additionally, each foot is equipped with four 1D force sensors, one at each corner, to measure the vertical force applied by the foot on the ground, from which the position of the Center of Pressure (CoP) can be computed. Finally, 5 low-cost MEMS IMUs[1] are placed on the robot: one in each feet, one in each tibia, and one in the pelvis. These sensors will be used to provide a more accurate estimation of the position of the system in the presence of deformations.

A patient can be securely strapped onto the exoskeleton, using straps at the ankle, knee, thigh and waist. To prevent any pain or injury to the patient during motion, it is crucial that the axes of the joints of the exoskeleton remain aligned and concurrent with those of the patient. This is achieved by adjusting the length of the tibia and thigh segments of the exoskeleton, which can slide to match the user's size. This enables an inclusion of patients from approximately 1.50 m to 1.90 m of height, depending on the length of their leg segments. Note that this setting is done manually, by loosening and tightening a set of screws: once fastened, both parts of the slider behave like a single rigid body. This is thus not an extra degree of freedom of the system to be controlled - rather, it can be seen as a known change of parameters in

[1]The LSM6DSO, manufactured by STMicroelectronics.

the robot's kinematic model.

To control the robot, an additional IMU is placed on the patient's torso. This enables the patient to communicate intentions to the exoskeleton, by leaning forward to trigger the walk for instance.[2] Additionally, a remote control enables the transition through various high-level operating modes (standing, walking, turning...).

1.2.2 Kinematics and dynamics

1.2.2.1 Kinematics

As shown in Figure 1.6, Atalante is made of 13 bodies, or links, articulated about 12 joints: as such, the joint configuration (angles) $\boldsymbol{\theta}$ characterizes the relative pose (position and orientation) of these bodies. However, to describe the configuration of the system in the world frame $\mathcal{W}$, 6 additional degrees of freedom are needed, to represent the robot's ability to move freely in space. These degrees of freedom correspond to the position and orientation of an arbitrary body of the robot, called the *floating base* or *freeflyer*. We denote $\boldsymbol{p}_{ff}, R_{ff} \in \mathbb{R}^3 \times SO(3)$ these coordinates. This enables us to define the generalized position and velocity coordinates of the system [Featherstone08] as

$$\begin{cases} \boldsymbol{q}_r \triangleq (\boldsymbol{p}_{ff}, R_{ff}, \boldsymbol{\theta}) \\ \boldsymbol{v}_r \triangleq (\dot{\boldsymbol{p}}_{ff}, \boldsymbol{\omega}_{ff}, \dot{\boldsymbol{\theta}}) \end{cases} \tag{1.1}$$

where $\boldsymbol{\omega}_{ff}$ is the angular velocity vector associated to the rotation R_{ff}. The suffix r denotes the use of a rigid body model, as this parametrization will latter be changed to account for deformations.

Consequently, assuming full rigidity, the theoretical pose of any body $\mathcal{B}$ in the world frame is given as a known function of $\boldsymbol{q}_r$ through the forward kinematic model f_r: this writes

$${}^{\mathcal{W}}M_{\mathcal{B}} = f_r^{\mathcal{B}}(\boldsymbol{q}_r) \tag{1.2}$$

1.2.2.2 Dynamics without a patient

In this subsection, we express the dynamics of Atalante when no user is present, i.e. that of the robot alone, under the assumption of full rigidity.

[2] Note that this IMU is not rigidly linked to the exoskeleton, and thus will not be used in this work, as it does not provide any direct information about the robot's deformations.

The dynamics of a system of rigid bodies in interaction with its environment are well described in the literature [Featherstone08], and can be generically written as

$$M(\boldsymbol{q}_r)\dot{\boldsymbol{v}}_r + \boldsymbol{C}(\boldsymbol{q}_r, \boldsymbol{v}_r) = B\boldsymbol{u} + J_c(\boldsymbol{q}_r)\boldsymbol{\lambda}_c \tag{1.3}$$

where M is the generalized inertia matrix, $\boldsymbol{C}$ the vector of generalized bias forces (representing the combined effect of Coriolis, centrifugal and gravity forces), B is a selection matrix mapping the input torque $\boldsymbol{u}$ to the generalized velocity vector $\boldsymbol{v}_r$. Finally, J_c is the contact Jacobian, mapping the contact wrenches $\boldsymbol{\lambda}_c$ to the system's joints.

To compute the term in (1.3), we use the open-source kinematics and dynamics library *Pinocchio* [Carpentier19] developed by the LAAS laboratory, which provides an efficient implementation of these rigid body dynamics algorithms.

On Atalante, all the joints are actuated, i.e. $\boldsymbol{u}$ has the same dimension as $\boldsymbol{\theta}$ (12 degrees of freedom). We say that the exoskeleton is fully actuated. Note that despite this property, the system remains underactuated, as the six degrees of freedom corresponding to the freeflyer are not actuated. In order to control this joint, to prevent the robot from falling and to enable forward motion, the robot must therefore use the contact forces $\boldsymbol{\lambda}_c$ to apply a force and torque to the freeflyer.

1.2.2.3 Dynamics with a (passive) patient

The dynamics of the coupled system {patient + exoskeleton} are more complex than (1.3). Namely, the degrees of freedom corresponding to the motion of the patient need to be considered, together with the muscular torque induced by the patient, to express the full dynamics (i.e. to compute the forces expressed by the patient onto the exoskeleton). Such a model however can hardly be used in practice, considering that the patient is not instrumented, which means that no information about their pose and velocity is directly available.

As a simplification, we thus consider that the user is rigidly attached to the exoskeleton, and cannot perform any motion. This simplifies the system such that it has the same degrees of freedom as the exoskeleton alone. Furthermore, we assume that the patient is fully passive, and does not apply any torque (no muscular action). Then, the patient can then be seen, from a dynamic perspective, simply as an additional mass and inertia, attached to the bodies of the exoskeleton.

Specifically, we decompose the patient into 7 major segments - upper body, right/left thigh, right/left tibia, and right/left foot - and add the corre-

sponding mass and inertia to the robot's bodies. To obtain these parameters, we use an anthropometric description of an average human as presented in [Winter09] that models these parameters as a function of the total mass and height of the patient.

In the end, the dynamics of the system still takes the form of (1.3) Simply, the mass and inertia parameters in M and $\boldsymbol{C}$ in (1.3) are modified to include the patient, which we denote with the p suffix and write

$$M_p(\boldsymbol{q}_r)\dot{\boldsymbol{v}}_r + \boldsymbol{C}_p(\boldsymbol{q}_r, \boldsymbol{v}_r) = B\boldsymbol{u} + J_c(\boldsymbol{q}_r)\boldsymbol{\lambda}_c \tag{1.4}$$

Equation (1.4) simply translates the fact that we obtain a patient-specific model of an equivalent rigid body system, with the same kinematics as the exoskeleton but with a mass and inertia compatible with that of the coupled system. This equivalent model is then used in a classical trajectory generation and control framework used in humanoid robotics, as presented in the next subsection.

An important thing to keep in mind however is that the uncertainty in the value of the parameters in the coupled model (1.4) is vastly superior to what one might expect on a robot alone. Indeed, an average anthropometric model is used, but the precise distribution of mass of a given patient is not known. Likewise, the exact position of the patient in the exoskeleton is not measured: even in static conditions, the patient is never standing exactly as planned in the model.

Furthermore, the patient is able to move, in particular their upper body, while in the exoskeleton. This can be seen as the presence of an unknown disturbance added to (1.4). This disturbance can be quite significant, considering that the mass of the user is comparable to the mass of the exoskeleton: by leaning to the side for instance, the patient can move their center of mass several centimeters in any direction, with a strong impact on balance.

Finally, another source of disturbance lies in the patient's muscular activity. Atalante is indeed meant to be used with a wide variety of users suffering from different pathologies, from valid users to hemiplegics and paraplegics. The behavior of the exoskeleton must therefore remain robust to a variety of behaviors from the user.

In conclusion, it is important to keep in mind the uncertain nature of the true dynamics of the system when working with an exoskeleton. The nominal model of (1.4) is a mere approximation, typically not accurate enough to enable feedforward-only control: rather, robust feedback methods need to be employed. This observation is the source of our motivation for using a sensor-driven approach to tackle the question of compensating the deformations.

1.2.3 Overall control architecture

Atalante is controlled following the classical *position control* paradigm. This consists in considering that the position of the robot joints, $\boldsymbol{\theta}$, can be controlled to follow a reference trajectory[3], noted $\boldsymbol{q}^*$. This is performed through an inner control loop called the *low-level controller*, which runs at 1 kHz. Then, an outer loop is responsible for computing this reference: this is the *high-level controller*. This is illustrated in Figure 1.7.

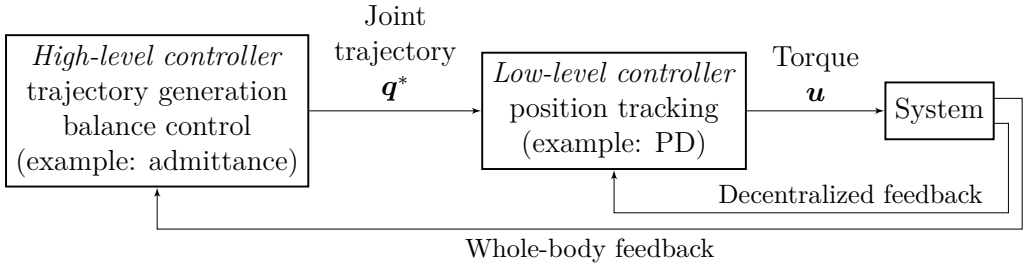


Figure 1.7: A simplified overview of "position-controll" for a robot like Atalante.

1.2.3.1 High-level controller

From a control system point of view, the role of the high-level controller is simply to provide a joint trajectory, $\boldsymbol{q}^*$, to execute the desired motion. For the case of legged robots, this trajectory is required to achieve an important objective: it must enable the robot to stand or walk without falling, i.e. to maintain balance.

Indeed, recall that a legged robot is under-actuated due to the presence of a freeflyer joint. This joint is only controlled through the application of external forces at the foot, as described in (1.3). This results in a complex trajectory planning problem, which has been broadly studied in the literature [El Khoury13, Finet17, Carpentier18, Mastalli20]. This problem is solved offline on Atalante through an optimization framework, similar to the one presented in [Hereid17], which provides a *nominal* trajectory theoretically balanced, according to (1.4).

However, simply following this trajectory is often not sufficient to provide a stable walk on the real system. As mentioned in Section 1.2.2.3, the dynamics is quite uncertain, meaning that the theoretical trajectory is not

[3]The use of the letter $\boldsymbol{q}$, instead of $\boldsymbol{\theta}$, is made to be consistent with the modeling of the robot to include the presence of deformations, as presented in Chapter 2

guaranteed to be feasible on the real system (i.e. it may lead to the robot falling). Furthermore, disturbances are unavoidable due to the presence of a user.

Thus, the nominal trajectory needs to be corrected online to adjust to the true system state. Such correction is done on position-controlled robot through *admittance frameworks* [Kajita10, Englsberger12, Caron19], which adjust the target trajectory to control the contact forces $\boldsymbol{\lambda}_c$ in (1.4), thereby providing balance. In the walking experiments presented in this manuscript, an admittance controller based on the control of the center of mass, similar to the one presented in [Caron19], is used to stabilize the robot - and ultimately obtain a joint target trajectory $\boldsymbol{q}^*$ for the low-level controller.

1.2.3.2 Low-level controller

The low-level controller is responsible for controlling the motor torque $\boldsymbol{u}$ in order to track a target joint trajectory ($\boldsymbol{q}^*$, $\dot{\boldsymbol{q}}^*$). One common choice of controller is the use of decentralized, high-gain PD or PID controller [Chung08] [Westervelt18]. In this decentralized approach, each motor is controlled individually.

The use of a decentralized approach is possible thanks to the high transmission gear ratio used on Atalante, which approximately decouple the dynamics of the system (see [Finet17, Section 1.6.3] for an analytical study of this property). Furthermore, a high gain approach proves to be robust to uncertainties in the dynamics of the system, as the large gain are meant to dominate these effects. More practically, a decentralized structure is often adopted as it greatly simplifies the gain tuning process, since it allows one to independently tune the gains for each actuator successively.

1.2.3.3 Impact of the presence of deformations

In a position-control framework, the presence of deformations can be seen as an issue in trajectory tracking, and thus, a problem to be handled by the low-level controller. Indeed, behind the joint trajectory $\boldsymbol{q}^*$ computed by the high-level controller really lies the rigid kinematic model of the robot (1.2). More than tracking the angle of the actuators (rotor position) *per se*, the true objective sent by the high-level controller is the relative pose of the exoskeleton's links: through the forward kinematic model, $\boldsymbol{q}^*$ indicates, for instance, where the right foot should be with respect to the left foot.

While on a rigid system, tracking the joint angles $\boldsymbol{\theta}$ is equivalent to tracking the pose of the links, this is no longer the case in the presence of deformations. This is illustrated in Figure 1.4: although the exoskeleton features

good tracking at joint level, the position of the left foot differs from the desired position.

One approach to solve this problem is to change the low-level control paradigm, to provide the desired link position tracking by including a feedback on a measurement of this position mismatch, i.e. the deformations. This is the solution studied in this thesis. This approach has the advantage of leaving the high-level controller unmodified.

The modification of the low-level controller is presented in Chapter 4. It requires a model and an estimation of the deformations: this is the topic of Chapter 2 and Chapter 3 respectively.

Chapter 2

Modeling of the deformations of Atalante

Chapitre 2 - Modélisation des déformations d'Atalante: *Ce chapitre présente la manière dont sont modélisées les déformations d'Atalante, à des fins d'observation et de contrôle. Une revue de la littérature fait ressortir trois modélisations prédominantes, dont deux que nous exploitons directement dans la suite. Le premier modèle utilisé consiste à étendre le modèle cinématique du robot en ajoutant des liaisons rotules au niveau des hanches et des chevilles de l'exosquelette, pour représenter les déformations. Ce "modèle flexible étendu" sera utilisé au chapitre suivant à des fins d'observations.*

Cependant, un tel modèle étendu complexifie le problème de contrôle, en ajoutant des degrés de liberté non directement actionnés à la cinématique du robot. Pour traiter ce problème, nous utilisons à la place une modélisation d'actionneur série élastique, colocalisant ainsi les déformations avec les axes des moteurs. L'opération de projection permettant de passer du modèle flexible étendu à ce modèle d'articulation flexible est présentée en détail. La précision de ces deux modèles est ensuite évaluée à l'aide d'un dispositif de capture de mouvement, qui valide leur pertinence.

In this chapter, we present the models used to describe the deformations of Atalante. We first review the classical models used in the literature to represent deformations in robots, from which we draw inspiration to design a model for Atalante. We propose an extended kinematic model, i.e. a model where extra degrees of freedom are added, in the form of spherical joints located at the weakest points of the structure, to represent the deformations. This model proves to be very convenient for observation purposes using IMUs, but is not directly usable to address the control problem. For this purpose,

we alternatively model the deformations as joint elasticity, projecting the previous model onto the joint's axes. The kinematic accuracy of both models is evaluated using motion capture, which confirms their relevance for, respectively, estimation and control purposes.

2.1 Description of flexibilities in the robotics literature

Modeling and control of flexible elements of a supposedly rigid robot have been topics of interest since the early days of robotics. Work on this topic can be easily traced back to the 80s, with work on industrial robotic manipulators [Nicosia81, Marino86, Spong87]. These robots are typically tasked with controlling the position of their end effector in space as accurately as possible: their performance is thus directly degraded by the presence of deformations. The need for such modeling only grew throughout the years, as the demand for cost reduction drives the development of lighter robots able to carry heavier payloads [Moberg10].

A robotic system is an extremely complex mechanical system, with tens or hundreds of parts interacting, from transmission elements to structural links. Mechanically modeling each part individually to try to precisely understand the behavior of every element is extremely complex and hardly feasible in real-time. Instead, phenomenological models have been developed to describe the overall behavior of such systems.

2.1.1 The flexible joint model

To model deformations on supposedly rigid robots, almost all works model them as a consequence of *joint elasticity*: this is the so-called *flexible joint* model [Spong87, Spong90].

The idea behind this model is to consider that the robot's links are infinitely rigid, and that the deformation only stems from the robot's transmission elements. Indeed, the use of belts, gears, harmonic drives... are all very plausible sources of compliance. This compliance results in a mismatch between the input (motor) angle, and the output angle, i.e. the true angle between the rigid links of the robot. Mathematically, this means that a robot with n elastic joints is described by $2n$ coordinates: the link angle $\boldsymbol{q} \in \mathbb{R}^n$ and the motor angle $\boldsymbol{\theta} \in \mathbb{R}^n$. We define $\boldsymbol{\alpha} \triangleq \boldsymbol{q} - \boldsymbol{\theta}$ the deformation angle. This is illustrated in Figure 2.1a

Dynamically speaking, the robot's transmission is modeled as a compliant element located between the motor's output and the driven link. By

compliant, we mean that the torque transmitted by this element, $\boldsymbol{\tau}_f \in \mathbb{R}^n$, is function of the deformation angle and velocity. This relationship is typically modeled as a linear spring damper [De Luca05a, Torabi08]:

$$\boldsymbol{\tau}_f \triangleq -K\boldsymbol{\alpha} - \nu\dot{\boldsymbol{\alpha}} \tag{2.1}$$

where K and ν are positive-definite diagonal matrices, representing the transmission's stiffness and damping.

Often, the damping term is ignored (i.e. $\nu = 0$), such that the deformation is modeled as an ideal spring instead [Marino86, De Luca98, De Luca05b], hence the name of elastic joints. Such a model in which the motor acts on the link by transmitting torque through a flexible element, also corresponds to the modeling of *series elastic actuator* [Pratt95, Buondunno18], although this terminology commonly refers to robots where a compliant element is placed in the transmission on purpose: example of such robots include the iCub [Tsagarakis11] or the Valkyrie [Paine15] humanoid robots.

An important property of this model of the deformations is that the robot's kinematic model, describing the motion of the links, remains unchanged. Indeed, the positions of the links remain constrained to rotations about the original axes of the joints, as a function of $\boldsymbol{q}$ only. This is a very convenient property for the control of the robot, where a reference joint trajectory $\boldsymbol{q}^*$ is to be tracked. Then, if each joint can be controlled such that $\boldsymbol{q}$ tracks $\boldsymbol{q}^*$, the links will be at the desired position.

However, considering that the deformations occur only about the actuated axes is not always sufficient to accurately describe the kinematics of the real system, as deformation may also happen about other points in the structure. This stresses the need for more complex models which include additional degrees of freedom to the robot link configuration.

2.1.2 The flexible link model

Instead of considering that the deformations are localized at the joints, the flexible link model discussed here considers that the deformations are *distributed* along the robot's link, as represented in Figure 2.1b. As such, the robot is no longer modeled as a set of articulated rigid bodies. Instead, the links are flexible elements capable of continuous deformation, which results in an infinite dimensional model described by partial differential equations [Carrera11].

While this modeling more accurately describes the physical behavior of a robot's link (considering that there is no such thing as an infinitely rigid body), it is much more difficult to mathematically model and manipulate.

This model will not be used on Atalante, where the links follow closely enough a rigid body assumption. Hence, we do not discuss it further, and refer the reader to [De Luca08, Bauchau11, Lismonde20] for a more complete description.

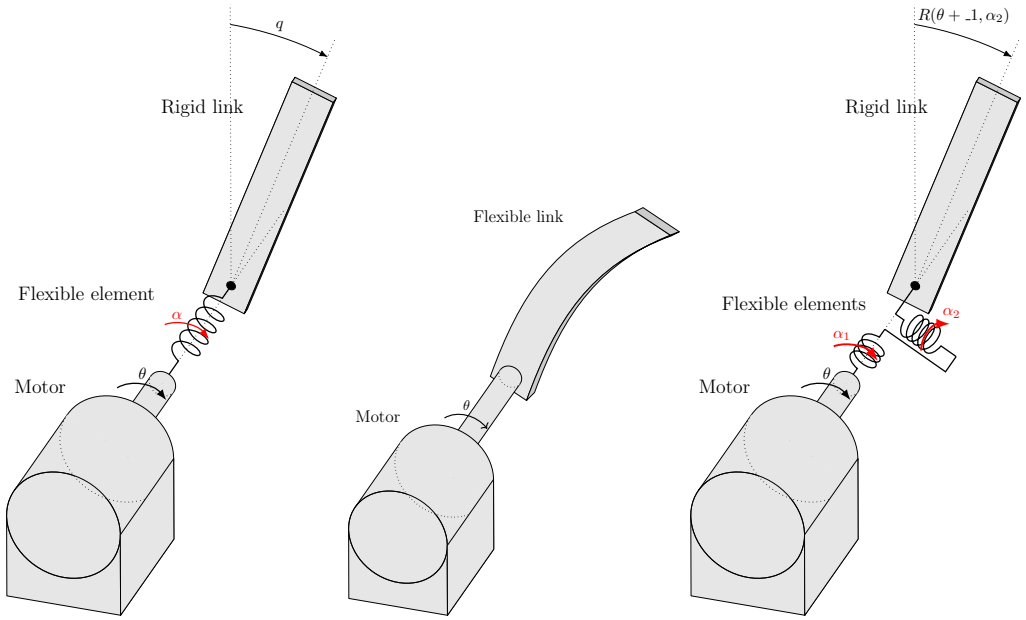
2.1.3 The extended flexible joint model

An intermediate solution to take into account the presence of deformations in the links, while keeping a rigid body approach, is to generalize the notion of joint flexibility to enable rotations about axes that are not parallel to the joint axis [Abele11, Moberg14, Zimmermann20] . This modeling increases the number of degrees of freedom of the system: hence, it is referred to as the *extended flexible model* [Moberg10].

This model introduces extra degrees of freedom to the robot's kinematic model, in the form of additional revolute joints collocated to an existing joint, but about an axis orthogonal to the actuated axis[1]. This is represented in shown in Figure 2.1c. These "virtual axes" can be seen as a modeling of the deformation of the joint's bearings ; they also represent a linear approximation of a distributed deformation along the robot's link. This model thus enables a more accurate kinematic description of the robot's motion than the constraint joint elasticity model, while still using a rigid body formalism: namely, the state of the system is still described by a finite set of degrees of freedom, $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\boldsymbol{\alpha} \in \mathbb{R}^m$, only this time $m > n$. These extra unactuated degrees of freedom often follow the same spring damper law as in the joint elasticity model, i.e. (2.1), with appropriate dimensions.

In the context of Atalante, we use two of these modeling approaches to describe the deformations of the robot. Namely, for observation purposes, we use an extended flexible model approach, representing the deformations as spherical joints and 3D springs. For the control part however, we use a joint deformation model, which provides a good approximation of the extended model, while making the formulation of the control problem much easier. In the rest of this chapter, we present both of these models, and validate them using motion capture.

[1]Note that a further extension considers that this joint does not need to be collocated with an existing joint, but can be placed anywhere on a link, splitting in in two halves[Bauchau11]. This alternative modeling is briefly discussed in Appendix D.



(a) Joint elasticity model (b) Flexible link model (c) Extended flexible joint model

Figure 2.1: A representation of the three proposed deformation models.

2.2 Extended flexible model for Atalante

The use of an extended flexible model came as a result of empirical physical considerations and observations about the structure of Atalante.

On one hand, the transmission elements of Atalante are extremely stiff: this can be observed on a simple experiment, using the fact that all joints are equipped, at their output, by a mechanical stop, to guarantee user's safety. By applying the maximal motor torque when the output rests against the stop, the rotor is able to travel further that the output, due to the deflection of the transmission element. By comparing the motion of the rotor (measured by the encoder) and the (known) distance between the mechanical stop, we obtain a direct measurement of the deformation of the transmission. This experiment shows that only a negligible amount of deformation, typically around 0.15°, occurs at the joint level[2]. The deformation of the transmission elements themselves are far from being sufficient to explain the total deformation observed, i.e. the additional 10 cm of displacement observed in Figure 1.4 for instance.

[2]This value is consistent with the very high theoretical stiffness of the harmonic drives used on Atalante (around $K = 50000$ N/rad)

On the other hand, the robot's largest links, the thigh and the tibia, were designed to be extremely rigid. Finite element simulations predict no significant deformation of these parts; this is validated using motion capture experiments, which measures no visible deformation. This means that the links can be modeled as rigid: the deformation remains a localized phenomenon happening around the robot joints, where the links are less thick and therefore less stiff.

This notion of localized deformation about the robot joints, not restricted to the transmission elements, is exactly what is captured by an extended flexible model. This is the approach we take, with a slight modification in terms of parametrization: instead of considering successive 1D deformations about 2 or 3 axis, we directly model a flexibility as a spherical joint, enabling an arbitrary 3D rotation about the joint center. For this purpose, we generalize the notion of 1D spring of (2.1) to a 3D torsion spring. More precisely, let $D \in SO(3)$ be the rotation representing the deformation. We note $\boldsymbol{\Omega}$ the associated vector of rotation [McCarthy90, Grassia98], i.e. the vector such that[3]

$$D = \exp\left([\boldsymbol{\Omega}]_\times\right) \tag{2.2}$$

Then the torque applied by the flexible element on its output frame writes

$$\boldsymbol{\tau} \triangleq -K\boldsymbol{\Omega} - \nu_i\boldsymbol{\omega} \tag{2.3}$$

where K and ν are positive-definite diagonal matrices representing the stiffness and damping of the flexibility, and $\boldsymbol{\omega}$ the local angular velocity associated to D.

Not all joints however need to be extended by a spherical joint to accurately describe the deformations on Atalante. Instead, we only introduce a spherical flexible joint at the weakest points of the structure. Namely:

- one deformation is located at the center of each ankle. This deformation is meant to capture the total deformation of Atalante's ankle transmission, which features a complex differential mechanism to actuate the ankle about two orthogonal axes.

- one deformation is located at each sagittal hip joint. This localization is chosen in regard of mechanical considerations, since the link between the sagittal and transverse hip is the weakest part of the robot's structure. Geometrically, it is also located at a midpoint between the pelvis

[3]This vector is also expressed in the well-known Rodrigues formula. More precisely, let $\theta \triangleq ||\boldsymbol{\Omega}||$ and $\boldsymbol{v} \triangleq \frac{\boldsymbol{\Omega}}{||\boldsymbol{\Omega}||}$, then Rodrigues formula gives the expression of D as the rotation of angle θ about the unit vector $\boldsymbol{v}$ as: $D = \cos\theta I_3 + \sin\theta[\boldsymbol{v}]_\times + (1-\cos\theta)\boldsymbol{v}\boldsymbol{v}^T$.

and the knee joint, and thus is intended to capture deformations that may happen through all these structural elements with minimal kinematic error.

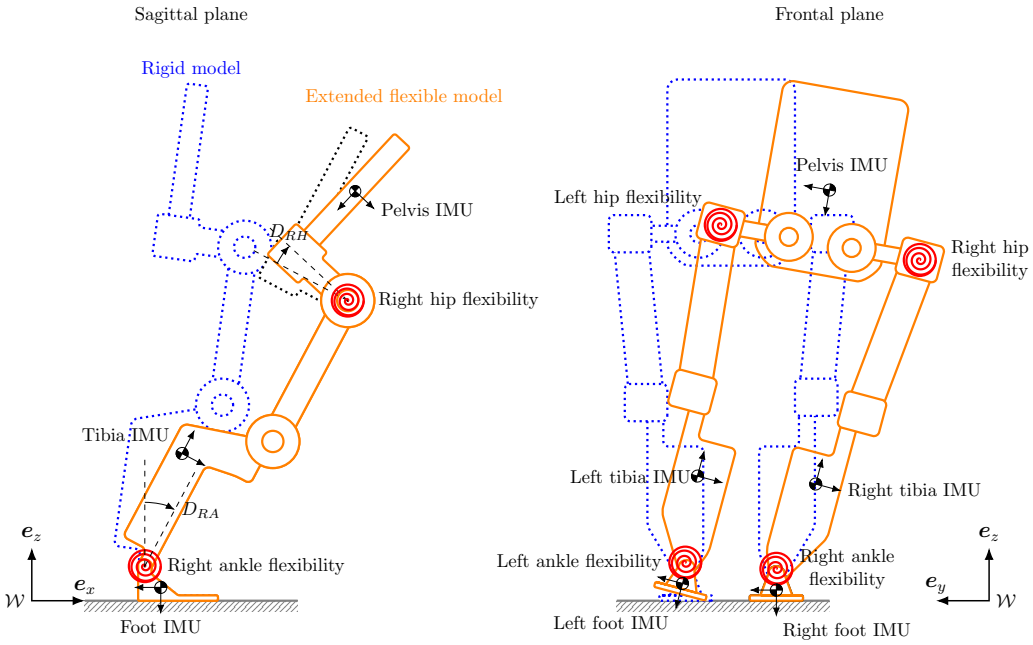


Figure 2.2: The punctual deformation model: four spherical joints (red spirals) are introduced into the robot's kinematic chain, with center point located at the ankle and sagittal hip joint.

The location of the flexibilities is represented on a schematic view of Atalante in Figure 2.2. In this model, the pose of the robot's bodies is now defined as a function of encoder angle $\boldsymbol{\theta}$, and the four rotation matrices parametrizing these spherical joints. We call these $(D_{RA}, D_{RH}, D_{LH}, D_{LA}) \in SO(3)^4$ to represent the deformations around the right ankle, right hip, left hip and left ankle respectively: this is illustrated in Figure 2.2. To simplify the notations, we use the letter j to index the joints ($j \in \{RA, RH, LH, LA\}$), and simply denote these deformations D_j. We call O_j the origin of the flexible joints (i.e. the center of rotation).

We call this kinematic model the *extended flexible model* with generalized position coordinates

$$\boldsymbol{q}_f \triangleq (\boldsymbol{p}_{ff}, R_{ff}, \boldsymbol{\theta}, D_{RA}, D_{RH}, D_{LH}, D_{LA}) \tag{2.4}$$

We denote f_f the forward kinematics function, giving the pose of a body

in the world frame as a function of these generalized coordinates

$$^{\mathcal{W}}M_{\mathcal{B}} = f_f^{\mathcal{B}}(\boldsymbol{q}_f) \tag{2.5}$$

Note that the main driving factor for the choice of this model, and in particular for parametrizing the deformations using rotation matrices, is the fact that the corresponding position $\boldsymbol{q}_f$ can be easily instrumented using IMUs placed on the robot - the location of the IMUs being in turn driven by the location of the flexibilities. Indeed, the generic idea of our methodology is to place an IMU on each side of every deformation, and to compute the deformation rotation D_j as a simple difference in IMU attitude. For Atalante, with four deformations to instrument, this process requires the use of 5 IMUs, located in the pelvis, both tibias and both feet. Indeed, the difference between the tibia and foot IMU on each side gives a measurement of the ankle deformation, and the difference between the pelvis and tibia IMU gives the hip deformation angle. Likewise, the difference in gyroscope measurements (correctly re-projected into the relevant frames of reference) gives the corresponding angular velocity. Note that this methodology can be generalized to a robot with any number of points of deformation: for a robot with n flexibilities modeled as spherical joints, $n + 1$ IMUs can be used to estimate the corresponding rotations.

2.3 Joint flexibility model for Atalante

2.3.1 Presentation

The extended flexible model presented in Section 2.2, though quite accurate and convenient for observation purposes, raises some difficulties in terms of control. To express this difficulty, we factorize the freeflyer joint (i.e. the odometry coordinates, common in both models), to consider only the relative pose of the bodies with respect to each other. Thus we rewrite the forward kinematics of the rigid model (1.2), and that of the flexible extended model (2.5) as

$$\begin{aligned} ^{ff}M_{\mathcal{B}} &= g_r^{\mathcal{B}}(\boldsymbol{\theta}) \\ ^{ff}M_{\mathcal{B}} &= g_f^{\mathcal{B}}(\boldsymbol{\theta}, D_{RA}, D_{RH}, D_{LH}, D_{LA}) \end{aligned} \tag{2.6}$$

In other words, the mapping g_r (respectively g_f) gives the pose of body $\mathcal{B}$ with respect to the freeflyer using the rigid (respectively extended flexible) model, while f_r (respectively f_f) gives the pose with respect to the world frame. g_r and f_r (and likewise g_f and f_f) are thus linked through the following identity: $^{\mathcal{W}}M_{\mathcal{B}} = {}^{\mathcal{W}}M_{ff}\,{}^{ff}M_{\mathcal{B}}$ i.e. $f_r^{\mathcal{B}}(\boldsymbol{p}_{ff}, R_{ff}, \boldsymbol{\theta}) = {}^{\mathcal{W}}M_{ff} g_r^{\mathcal{B}}(\boldsymbol{\theta})$.

Equation (2.6) outlines the fact that the extended flexible model possesses more degrees of freedom than the rigid model, and can thus reach configurations which are impossible for the rigid system. Conversely, given a target trajectory $\boldsymbol{q}^*$ and assuming non-zero deformations D_j, it is not possible to find motor angles $\boldsymbol{\theta}$ such that the position and orientation of every body of the system matches the target pose, i.e. such that

$$\forall \mathcal{B}, \qquad g_f^{\mathcal{B}}(\boldsymbol{\theta}, D_{RA}, D_{RH}, D_{LH}, D_{LA}) = g_r^{\mathcal{B}}(\boldsymbol{q}^*) \tag{2.7}$$

Instead of looking for a way to control this extended flexible model, we take another, much simpler approach: we collocate the deformations with the actuators, using the joint deformation model presented in Section 2.1.1. Indeed, for the case of Atalante, both models are kinematically quite close to each other. Both consider that the deformations are punctual rotations: simply, where the extended flexible model considers one 3D rotation at the sagittal hip, the joint deformation model uses three 1D rotations about the three orthogonal hip axes. Likewise, the ankle's spherical deformation can be decomposed about the ankle joint axes. Using a joint deformation model also enables for a physical interpretation of the computed angles in a joint elasticity model (unlike a generic inverse kinematics solution), which proves instrumental for the design of a decentralized controller, as presented in Chapter 4.

Mathematically, this means that we solve the inverse kinematics problem (2.7) by considering the rotations only - in order words, we look for alternative robot joint angles $\boldsymbol{q} \in \mathbb{R}^{12}$ that matches the orientation given by the extended flexible model (and thus the orientations given by the IMUs).

The computation of the joint angle $\boldsymbol{q}$ is done independently "joint by joint", associating each spherical joint of the extended flexible model to the corresponding motors. More precisely:

- No flexibility is associated to the knee joint in the extended flexible model. Thus, this joint is considered rigid, and the encoder value is simply kept as-is (i.e. $q_{knee} = \theta_{knee}$).

- The 3 angles of the hip are computed from the hip deformation - more specifically, using the rotation from the hip to the pelvis, ${}^{pelvis}R_{right\ hip}$ (respectively ${}^{pelvis}R_{left\ hip}$). This rotation is computed as a function of D_{RH} (respectively D_{LH}) using the flexible model, g_f in (2.6).

- Likewise, the 2 ankle angles are computed as a function of the ankle deformation through the rotation ${}^{right\ tibia}R_{right\ foot}$, again using g_f.

We thus obtain a projection, transforming these attitudes into joint angles

$$\begin{aligned}\phi\colon \quad & \mathbb{R}^2 \times SO(3)^4 \to \mathbb{R}^{12} \\ & \begin{pmatrix} \boldsymbol{\theta}_{knees} \\ {}^{pelvis}R_{right\ hip} \\ {}^{pelvis}R_{left\ hip} \\ {}^{right\ tibia}R_{right\ foot} \\ {}^{left\ tibia}R_{left\ foot} \end{pmatrix} \mapsto \begin{pmatrix} \boldsymbol{q}_{knees} = \boldsymbol{\theta}_{knees} \\ \boldsymbol{q}_{right\ hip} \\ \boldsymbol{q}_{left\ hip} \\ \boldsymbol{q}_{right\ ankle} \\ \boldsymbol{q}_{left\ ankle} \end{pmatrix}\end{aligned} \tag{2.8}$$

This function is built using Euler angle decomposition of rotation matrices, which enable for a close-form analytical expression of ϕ: this is presented in the next subsection.

2.3.2 Computing the joint deformation angles from the extended flexible model

In this subsection, we explicit the function ϕ (and its partial derivative for computing velocity). We present it successively for the hips and for the ankles, outlining the differences between the use of a joint deformation model and the extended flexible model. The process is implemented independently for the right and the left leg, hence we omit this distinction and simply refer to the "hip" and "ankle".

2.3.2.1 Projection of the hip deformation

For the hip deformation, we consider the rotation ${}^{pelvis}R_{thigh}$ between the thigh and the pelvis, given by the extended flexible deformation model (this is a function of the joint angular displacement $\boldsymbol{\theta}$ measured by the encoders, and the hip deformation D_j).

Recall that under the rigid kinematic model f_r, the hips of Atalante consist of three orthogonal pivot joints, the frontal (x), transverse (z) and sagittal (y) hip joints. This means that the rotation from the thigh to the pelvis is decomposed as:

$${}^{pelvis}R_{thigh} = R_x(q_f)R_z(q_t)R_y(q_s) \tag{2.9}$$

where $\boldsymbol{q}_{hip} \triangleq \begin{pmatrix} q_s & q_t & q_f \end{pmatrix}$ are the sagittal, transverse and frontal angle respectively.

From (2.9), computing the angles $(q_s, q_t, q_f) \in [-\pi, \pi) \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times (-\pi, \pi]$ to match the orientation ${}^{pelvis}R_{thigh}$ simply corresponds to a particular form of Euler angle decomposition[4] - following the X-Z-Y order. It is well known

[4]Such angles corresponding to a decomposition about three different orthogonal axes are also called Tait-Bryan angles.

[Craig89] that such a decomposition exists and is unique (the equations are presented in (A.4) of Appendix A), away from a singularity when the second angle reaches $\frac{\pi}{2}$ modulo π: this is (abusively) referred to as gimbal lock [Hemingway18]. In that particular case, gimbal lock corresponds to an angle of $\frac{\pi}{2}$ for the transverse hip (which would render the frontal and sagittal axis parallel to each other). Such condition will never be met in practice however, since the transverse hip has a mechanically limited range of about 15° about the origin, and the deformation can only account for a few extra degrees.

Thus, with this simple Euler angle decomposition, one creates a bijection (${}^{pelvis}R_{thigh} \mapsto \boldsymbol{q}_{hip}$) between the extended flexible model and the joint deformation model. By this definition, both reconstructions will give the same orientation. Likewise, the relative angular velocity vector can be decomposed into Euler angle derivatives to obtain $\dot{\boldsymbol{q}}_{hip}$). The precise equations are presented in Appendix A (Equation (A.6)).

While the kinematic structure of the hip enables a conservation of the rotation, both models are not equivalent in terms of position as one progresses along the kinematic chain. Indeed, where the extended flexible model rotates the rigid bodies of the robot around a single center point (the sagittal hip), the joint deformation model applies the same rotation but as successive rotations around non-collocated centers.

The transverse and sagittal hip are located very close to each other, rendering this difference negligible. However, this effect can be viewed on the frontal hip, illustrated in Figure 2.3. Notice there that the position of the swing foot is different between both models, despite the fact that the orientation of the hips and the pelvis are the same. Considering that the deformations are small rotations, this difference however remains limited. More precisely, at the first order, the two models differ only in the predicted height of the swing foot according to

$$e = l(\alpha_l - \alpha_r) \tag{2.10}$$

where $l = 170$ mm is the distance between the frontal and sagittal hip, and α_l, α_r are the deformation around the frontal hip of the left and right leg respectively. In practice when walking, the difference between both deformations is around 0.025 rad, the support hip (which withholds more weight) undergoes a larger deformation, leading to a difference of about 5 mm in the predicted foot height.

2.3.2.2 Projection of the ankle deformation

For the ankle, a similar decomposition into Euler angles can be performed. One difference worth mentioning is that only two of the three axes are ac-

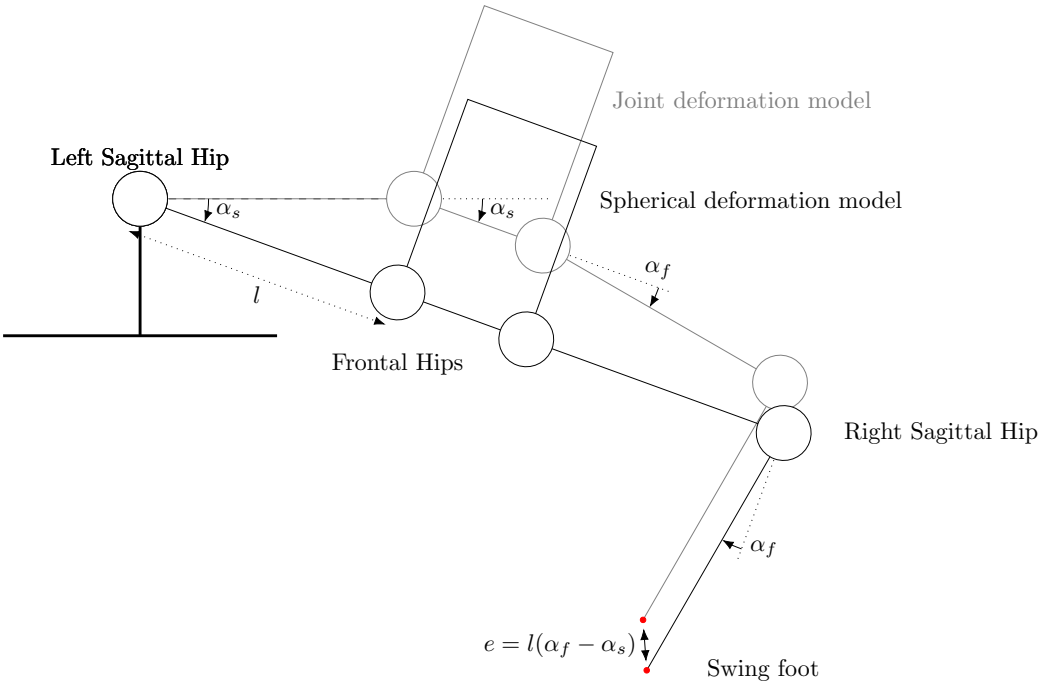


Figure 2.3: Illustration of the difference between the two models: although the angles of the successive rotations are equal, the position of the swing foot differs between both models.

tuated: the sagittal axis ($\boldsymbol{e}_y$) and the Henke axis $\boldsymbol{h}$. This means that the joint deformation model has one less degree of freedom (per leg) than the extended flexible model.

The joint angles are computed, like for the hip, by decomposing a rotation matrix given by the spherical model into Euler angles. A third vector $\boldsymbol{v} = \boldsymbol{e}_y \times \boldsymbol{h}$ is defined to complete an orthogonal basis with the two actuated axis. Then, the rotation from the foot to the tibia decomposes along these three axes as:

$$^{tibia}R_{foot} = R_v(q_v)R_s(q_s)R_h(q_h) \tag{2.11}$$

where q_s and q_h are the desired sagittal and Henke ankle angles. The last angle, q_v, corresponds to a rotation that cannot be accounted for in the joint deformation model, and is simply discarded ($q_v = 0$, i.e. $R_v = I_3$). The angular velocity is decomposed in the same fashion. Again, the formulas are given in Appendix A (Equations (A.9) and (A.10)).

2.3.3 Summary

The function ϕ is thus defined through the use of specific Euler-angle decompositions, performed at each hip and ankle. Notice that the definition of ϕ, as a function preserving the input rotations at the hip, and along the two actuated hip angles, is unique. This is a result of the uniqueness of the Euler angles in the range of interest[5]: thus, there is no ambiguity in the way the deformations are projected onto the joint[6].

Note also that, because only two degrees of freedom are present at the ankle, this operation is only a projection and not a bijection. It is straightforward to verify that, when applying the rigid kinematic model g_r to a configuration $\boldsymbol{q} \in \mathbb{R}^{12}$ to compute the rotation matrices in (2.8), then applying ϕ, we find back the same angles. Symbolically, this writes: $\phi(g_r(\boldsymbol{q})) = \boldsymbol{q}$. This implies that

- When no deformation are present, this mapping simply matches the original joint angles $\boldsymbol{\theta}$ (measured by the encoders).
- Having applied ϕ to the rotations given by the extend flexible model to obtain $\boldsymbol{q}$, if we iterate the process by computing these rotations through $g_r(\boldsymbol{q})$, we find back the same $\boldsymbol{q}$: this is why we call ϕ a projection.[7]

2.4 Comparison of the two models against motion capture

To quantitatively and qualitatively assess the merits of these two deformation models, we look at their kinematic accuracy, i.e. we investigate how accurately these models describe the motion of the robot's bodies. For this purpose, we use a motion capture device to provide ground-truth data: this provides us with a way to directly measure the deformations, and to estimate the accuracy of our models.

Motion capture consists of recording a scene using several infrared cameras, with infrared-reflecting markers placed on moving objects. Detecting the position of the markers in the images from several cameras enables reconstruction of the 3D position of this marker. By placing several markers on a

[5] All angles are of course defined modulo 2π, but since the robot joints are equipped with mechanical stops, only the angles closest to zero are relevant.

[6] This is made possible because the knee joint is modeled as rigid. Otherwise, the angle measured by the tibia IMU could be attributed equally to the sagittal knee or to the sagittal hip joint.

[7] $\phi \circ \phi$ is obviously not defined, but we have $\phi \circ g_r \circ \phi = \phi$

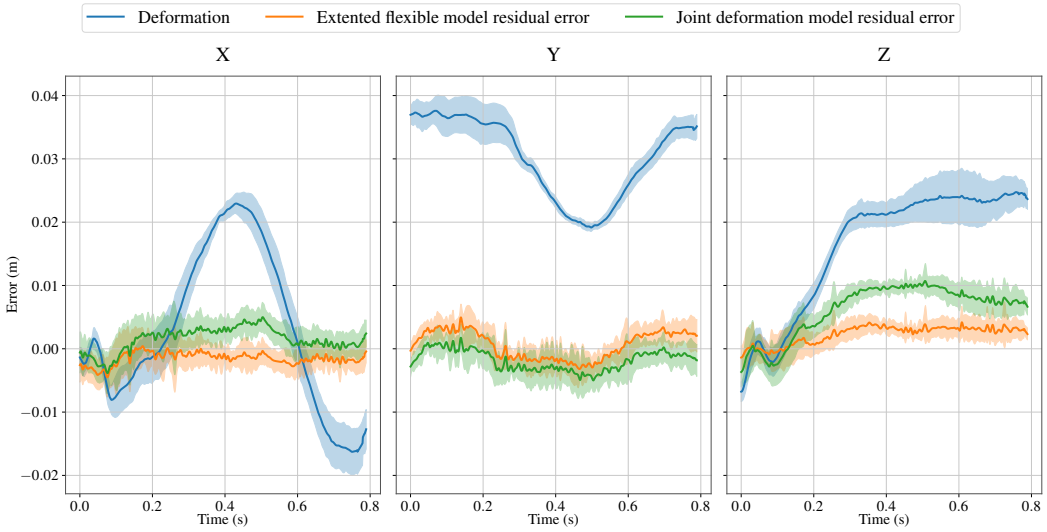


Figure 2.4: Blue curve: deformation during a step (average, standard deviation, over 10 steps) of the swing toe during a typical walking experiment. The orange and green curves are the residual error committed by the extended flexible model and the joint deformation model respectively.

given rigid body, the orientation of this rigid body can be reconstructed as well. In this thesis, we use a commercial motion capture system composed of 8 tracking cameras from OptiTrack[8], offering a theoretical accuracy of 1 mm.

In Figure 2.4, the blue curve represents the total deformation observed at the outermost toe of the swing foot, measured by motion capture, during a walking experiment with a valid user in the exoskeleton (average over the full walk). This deformation is measured as the difference between (i) the position of the swing toe with respect to the stance foot computed using the rigid model and encoder measurements, and (ii) the direct measurement done from motion capture.

The swing toe is chosen as reference for two reasons: one is that it is located at the very tip of the robot, furthest away from the stance foot, and thus is the point where the deformation is the largest, summing up the effect of all the deformations in the structure. The second, and most important argument, is the fact that the control of the foot motion is critical to correctly execute a walking motion, making it a variable of practical interest. Indeed, balance while walking is entirely determined by the contact forces at both feet. One of the most detrimental effect of the presence of deformations is the fact that the swing toe is lower than expected: this is shown by the

[8]OptiTrack Prime 13 cameras, https://optitrack.com/.

rightmost curve (Z axis) of Figure 2.4. This curve starts at $t = 0$ with the start of the double support phase, during which the vertical deformation is negligible. However once the swing foot leaves the ground (around $t = 0.2$ s), the deformation increases. During the swing phase, the swing toe is about 2.5 cm lower than predicted by the rigid model. This causes the foot to strike the ground earlier than predicted, greatly hindering walk stability.

The second most detrimental effect of the flexibilities is shown in the middle curve: the lateral deformation (Y axis) has the largest value, with up to 4 cm of deformation. This raises an issue in the form of potential self-collision of the robot's tibias, and therefore limits how close both legs can move to each other.

Finally, the X axis also reports visible deformations, in particular in the middle of the swing phase, as the leg is accelerated forward. Practically, this has less impact on flat foot walking than the two effects mentioned above. Indeed, it only slightly reduces the length of the steps, but does not cause any direct issue on stability. Though our generic approach seeks to control all three position axes (as well as the orientation of the swing foot), this means that in practice, the performance requirement will be lower in X than for the other two axes, of paramount importance.

Using motion capture data, we assess how accurately the swing toe position can be reconstructed using the extended flexible model, and the joint deformation model. For this purpose, we use motion capture orientation measurements to obtain the orientation of the robot's IMUs. From this orientation, we compute the deformation rotation matrices D_j as a difference in orientations between two IMUs and encoder measurements (as described later in Section 3.1.2), and therefore obtain an estimate of the swing toe position using the forward kinematics of the extended flexible model f_f. The residual error between this position and motion capture is represented in orange in Figure 2.4. This shows that this model quite accurately rebuilds the position of the swing toe, with an error of only a few millimeters on each axis. Note that the relatively large standard deviation, with a "spiky" pattern, is due to motion capture noise in orientation estimation, which is superior to the noise in position, which adds up for all four spherical joints of the model.

Finally, the green curve features the error of the joint deformation model, computed by projecting the extended flexible model angles onto the joints as presented in Section 2.3.2. This model causes a larger error, in particular for the vertical axis, where the value of the error almost doubles, from less than 5 mm to almost 10 mm. This is the consequence of having a different center of rotation for the frontal deformation, illustrated earlier in Figure 2.3. Nevertheless, this model remains sufficiently accurate to provide a meaningful

improvement over controlling the robot assuming a rigid model, as illustrated in Chapter 4.

2.5 Conclusion

Robots are complex mechanical systems, which feature a variety of potential sources for deformation: transmission elements, joint bearings, link element rigidity... The control of such system thus often relies on simplified linear models to empirically describe the global phenomenon: the most common of which being the joint deformation model, where flexibility is assumed to come from transmission elements alone.

In the context of Atalante, we propose two different models, which we examine at the light of their kinematic accuracy: our objective is to model the deformations in such a way as to be able to reconstruct the position of the swing toe while walking. These two models lay the foundation of the global approach used in this thesis.

The first model adds four spherical joints to the robot's kinematic model. This extended flexible model is shown by motion capture to be quite accurate, with a root mean square error of less than 5 mm on the swing toe position. It is also quite convenient to instrument using IMUs: this process is detailed in Chapter 3.

Then, to address the controller design problem, we project the deformation of this first model onto the joints of the robot, while keeping unchanged the IMU orientation. This will prove instrumental in the design of low-level controllers, as presented in Chapter 4.

Chapter 3

Online estimation of the deformations using IMUs

Chapitre 3 - Estimation en ligne des déformations à l'aide de centrales inertielles: *Ce chapitre aborde l'estimation des déformations du robot, telles que décrites par le modèle flexible étendu exposé précedemment. L'estimation des quatre rotations représentant les déformations, et de la rotation du pied de support, est équivalente à celle de l'orientation des cinq centrales inertielles du robot. Celles-ci, constituées d'un accéléromètre tri-axial et d'un gyroscope tri-axial, ne permettent pas (en l'absence de modélisation supplémentaire) d'observer l'angle de lacet : aussi, les déformations en lacet sont négligées et l'angle correspondant reconstruit sous une hypothèse de modèle rigide.*

L'estimation d'attitude d'une centrale inertielle repose sur l'intégration du signal du gyroscope, et sa fusion avec la mesure de l'accéléromètre, selon le principe du filtrage complémentaire. Cette opération est gênée, dans le cas d'un système dynamique en mouvement, par la présence d'une accélération linéaire qui s'ajoute à la mesure de la gravité. Dans le but d'obtenir une estimation précise des orientations des différents corps consituant le système, nous étudions trois approches différentes :

- *Un "observateur statique", reposant sur une hypothèse de stationnarité (accélération linéaire négligée devant la gravité). Il s'agit d'une implémentation directe de l'observateur présenté dans [Mahony08] sur l'exosquelette Atalante.*

- *Un "observateur cinématique", qui exploite le modèle cinématique du robot pour reconstruire une mesure fictive de la vitesse linéaire de la centrale inertielle, utilisée ensuite pour estimer l'attitude.*

- *Un "observateur dynamique", qui utilise le modèle dynamique complet du robot et des déformations pour calculer l'accélération comme la somme des forces modélisées s'exerçant sur le système.*

Ces trois approches sont comparées en détail sur des données expérimentales. Cette comparaison démontre la supériorité de l'"observateur cinématique", plus performante que les deux autres méthodes sur tous les critères considérés.

In this chapter, we address the problem of estimating the deformations of the robot, using Atalante's strap-down IMU sensors. In details, the objective is to estimate the four rotation matrices D_j of the extended flexible model presented in Section 2.2, and the corresponding angular velocities. In the process, we also obtain an estimate of the roll and pitch angles of the contact foot, which frees us from having to formulate a flat foot assumption.

While angular velocities are directly measured by the gyroscopes, and only need to be projected correctly to obtain an estimation of the velocity of the various bodies, strictly speaking IMU sensors offer no direct measurement of attitude, which must be estimated using an observer. This orientation can only be partially estimated, as the rotation around the gravitational field (yaw angle) is not observable. The reconstruction of this angle thus requires additional modeling assumptions or an extra measurement. Our approach is simply to neglect the yaw deformations, and rebuild the corresponding IMU rotation matrix using the rigid kinematic model, as detailed in Section 3.1.4. Although this impacts the accuracy of the reconstruction, the obtained performance is still satisfactory on Atalante, as shown by the results of Section 3.6.

The estimation of a roll and pitch angle from IMU measurements on an accelerating system, such as a robot walking, is also far from trivial. Following the complementary filter approach, the idea of IMU-based attitude estimation is to integrate the gyroscope's angular velocity measurement, while exploiting the accelerometer measurement of gravity to prevent estimation drift [Crassidis07]. However, this "measurement of gravity" is "corrupted" by the linear acceleration of the sensor, also measured by the accelerometer. The challenge then lies in separating these two terms. Different approaches have been proposed in the literature, and we here compare three different models:

- A "static observer", which assumes that $\boldsymbol{a}$ can be neglected before gravity. While being very simple to implement and use, the performance of

this observer is limited by this assumption of zero-on-average acceleration during dynamic motion. In this thesis, we simply call this observer the *StaticObserver*.

- A "kinematics-based observer" (called *KinematicObserver*), which uses only the kinematic model of Atalante to express the fact that the acceleration is the derivative of the linear velocity, which can be approximately measured with onboard sensors (encoder velocity and gyroscope measurement).

- A "dynamics-based observer" (the *DynamicObserver*) which uses the full kinematic and dynamic model of the system to compute the acceleration $\boldsymbol{a}$ as a result of the system dynamics, by exploiting the dynamical model of the flexibilities as spring dampers, see Equation (2.3).

These three observers are presented successively in this chapter, then extensively compared in Section 3.6, where various experiments enable us to quantitatively establish the accuracy of these methods on Atalante. The conclusion of this chapter is that the "kinematics-based observer" outperforms the other two in all cases.

3.1 Description and problem statement

In this section, we mathematically clarify the estimation problem that we seek to solve, and present a framework for exploiting (partial) IMU attitude estimations to reconstruct the state of the robot. This framework is then applied to three different observers, successively presented in the rest of this chapter.

3.1.1 Contact point assumption and definition of the local frame

In order to describe the state of the robot, we formulate the following assumption

Assumption 1. *At any given time, there is at least one foot in contact with hard ground. We furthermore assume that the contact foot does not slip on the ground.*

This assumption implies very little practical restrictions. Indeed, we do not assume that the ground is flat, nor that the contact foot rests flat on the

ground: the foot can rotate about its edges. Furthermore, we do not restrict ourselves to single support: double support also satisfies this condition.

Knowledge of which foot is in contact is readily deduced on the system from the force sensor measurements. In the case of double support, an arbitrary foot can be chosen: a typical choice is to take the foot with the highest force applied, as being the less likely to slide[1]. We refer to this foot as the contact foot, and denote $\mathcal{C}$ the associated frame. The other foot is then called the swing foot (even though it may also be in contact with the ground in the case of double support).

Then, the robot is represented by a kinematic chain, from the contact foot to the swing foot. The deformations are thus numbered in the order in which they appear, from 1 (the support ankle deformation) to 4 (the deformation of the swing ankle), instead of referring to them by their name. Since the floating base coordinates can be attached to any body on the system, we can also consider without loss of generality that the coordinates of the contact foot are used. Thus, the generalized position vector for the rigid model and the extended flexible model rewrite:

$$\begin{cases} \boldsymbol{q}_r \triangleq ({}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{C}}, {}^{\mathcal{W}}R_{\mathcal{C}}, \boldsymbol{\theta}) & \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^{12} \\ \boldsymbol{q}_f \triangleq ({}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{C}}, {}^{\mathcal{W}}R_{\mathcal{C}}, \boldsymbol{\theta}, D_1, D_2, D_3, D_4) & \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^{12} \times SO(3)^4 \end{cases} \tag{3.1}$$

The contact foot frame $\mathcal{C}$ also gives a reference point for defining a notion of odometry, to represent the total displacement of the robot from the world frame. More specifically, we define the *local frame* $\mathcal{L}$ as the frame translated along $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ from the world frame to lie below the origin of the contact foot frame, and rotated about $\boldsymbol{e}_z$ such that its $\boldsymbol{e}_x$ axis remains aligned with the sagittal plane of the robot[2]. This is represented in Figure 3.1.

By removing the odometry coordinates, the local frame offers a simpler context to fully characterize the robot's balance. Indeed, the robot's dynamics is independent of the absolute position of the robot in the world frame. It depends only on the pose of the various bodies in the local frame, and their velocities with respect to the world frame, expressed in the local frame.

[1] In practice, we use a time-based hysteresis to prevent rapid, erratic changes of the contact foot due to measurement noise when the weight distribution is near the threshold.

[2] In other words, the origin of the local frame share the same X and Y coordinate as the origin of the contact frame, and has a Z coordinate of 0. This frame however remains horizontal, i.e. its vertical axis is equal to the vertical axis of the world frame, $\boldsymbol{e}_z$.

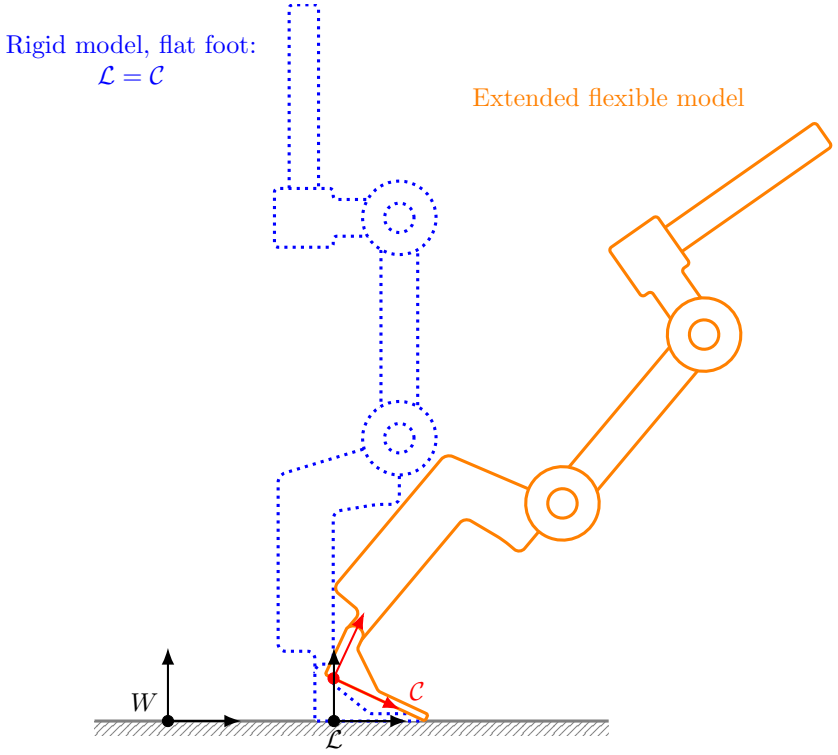


Figure 3.1: Representation of the world frame, contact foot frame and local frame.

3.1.2 Problem parametrization

In Section 2.2 and in Equation (3.1), we parametrize the extended flexible model using the local deformation matrices D_i. While this is indeed the relevant parametrization to use in the iterative computation of kinematics and dynamics algorithms, such as those presented in [Featherstone08], it is not the most convenient representation for writing observers from IMU data, which measure quantities relative to the absolute (and not relative) orientation. Thus, we use a change of coordinates to reparametrize the kinematic chain and consider the total deviation from the rigid, flat foot model (i.e. the sum of all deformations) instead of local deformations.

When considering only encoder measurements, the rigid body model gives the pose of any body $\mathcal{B}$ of the robot with respect to the contact foot frame, $\mathcal{C}$. Without additional information or measurement about the orientation of the contact foot, it is common to assume that the foot is flat on level ground - and thus that $\mathcal{C} = \mathcal{L}$. With this assumption, the rigid model and encoder measurements alone provide an estimate of the state of the robot, which we denote ${}^{\mathcal{C}}M_{\mathcal{B},r}$. This transform can be computed simply by setting the

freeflyer joint to identity in (3.1), i.e.

$$^{\mathcal{C}}M_{\mathcal{B},r} = f_r^{\mathcal{B}}(0, I_3, \boldsymbol{\theta}) \tag{3.2}$$

The flexible model of the robot then gives a pose in the local frame that differs from this rigid configuration by the application of 5 successive rotations:

- the rotation of the contact foot about the ground, which takes place at one of the corners of the foot, called O_0
- the four deformations, taking place around the center points O_j.

The robot can thus be described as being composed of 5 "rigid sets" (in the sense that the relative configuration of the bodies within the set can be described by the rigid model only, and is not affected by flexibilities nor the contact foot angle) S_i, with a spherical joint between each set (or between the first set and the world frame).

We then define the rotation matrix R_i, for $i \in [0, 4]$, as the total difference in orientation between the rigid and flexible model present at center O_i. More precisely, for $\mathcal{B}$ a body located in set $\mathcal{S}_i$, i.e. between O_i and O_{i+1} (or simply after O_4 for the case of the last deformation), R_i is defined as:

$$R_i \triangleq {}^{\mathcal{W}}R_{\mathcal{B}}\, {}^{\mathcal{C}}R_{\mathcal{B},r}^T \tag{3.3}$$

In other words, R_i takes body $\mathcal{B}$ from its rigid, flat-footed configuration, to its real orientation in the world frame. This rotation thus accounts for the support foot angle, and for all successive deformations up to body $\mathcal{B}$.

These rotations enable us to introduce a new generalized position vector for the flexible model as follows

$$\bar{\boldsymbol{q}}_f \triangleq ({}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{C}}, R_0, \boldsymbol{\theta}, R_1, R_2, R_3, R_4) \tag{3.4}$$

An explicit bijection between $\boldsymbol{q}_f$ in (3.1) and $\bar{\boldsymbol{q}}_f$ is readily defined by decomposing the rotations into individual elements:

- For the contact foot, we directly have $R_0 = {}^{\mathcal{L}}R_{\mathcal{C}}$ as ${}^{\mathcal{C}}R_{\mathcal{C},r}$ is just the identity matrix.
- For the deformation matrices, we can express D_i as a function of R_{i-1}, R_i and encoder measurements. For this purpose, we define $\mathcal{P}_i$ and $\mathcal{C}_i$ the parent and child frame linked to the deformation D_i. For the rigid model, these frames are equivalent as there is no deformation: it simply

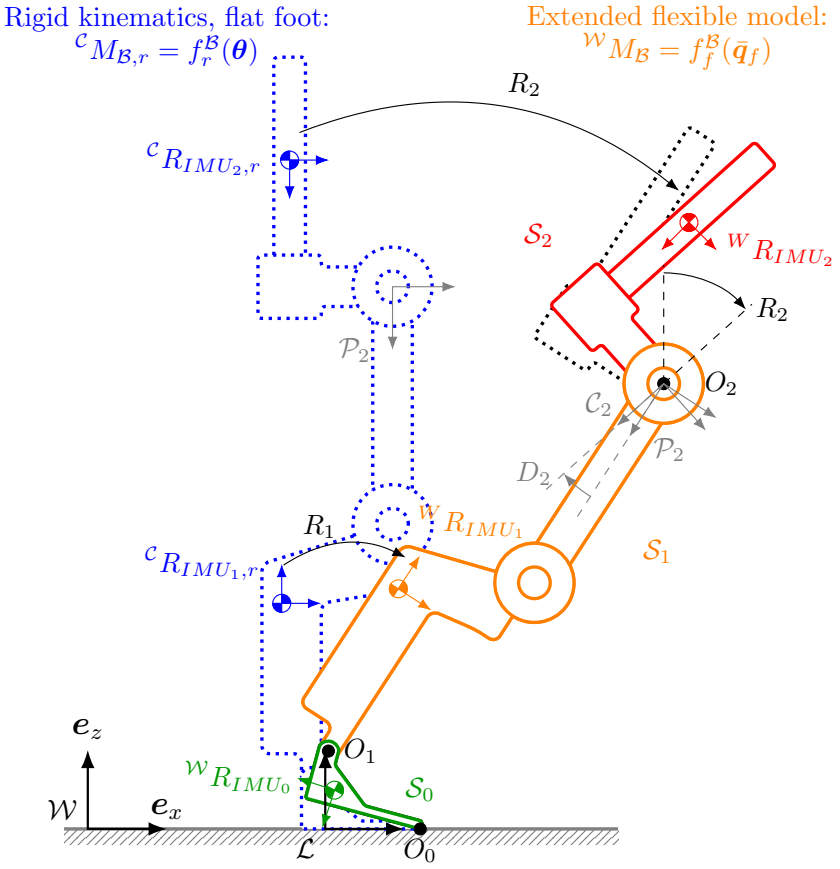


Figure 3.2: A representation of the various notations. The sets $\mathcal{S}_i$ of the flexible model, drawn in different colors, are connected at points O_i through spherical joints parametrized either by the D_i (local) or R_i (global) rotation matrices.

corresponds to the orientation of the joint onto which the deformation is attached. We note $R_r \triangleq {}^{\mathcal{C}}R_{\mathcal{P}_i,r}$ this orientation. By definition, we have

$$D_i \triangleq {}^{\mathcal{P}_i}R_{\mathcal{C}_i} \tag{3.5}$$

Since frame $\mathcal{P}_i$ is located before the deformation, it belongs to set S_{i-1} and thus

$${}^{\mathcal{L}}R_{\mathcal{P}_i} = R_{i-1}R_r \tag{3.6}$$

On the other hand, frame $\mathcal{C}_i$ is after the ith deformation, hence:

$${}^{\mathcal{L}}R_{\mathcal{C}_i} = R_iR_r \tag{3.7}$$

Combining these three equations, we get:

$$D_i = R_r^T R_{i-1}^T R_i R_r \tag{3.8}$$

This formula can be explained as follows: D_i, the deformation occurring at joint i, corresponds to the difference between the "total deformations" $R_{i-1}^T R_i$, expressed in the local frame of the parent joint $\mathcal{P}_i$.

Thus, estimating the state of the deformations D_i and the contact foot angle ${}^{\mathcal{W}}R_{\mathcal{C}}$ is equivalent to estimating the rotations R_i, which are characterized in a simple, unified manner by (3.3).

3.1.3 IMUs instrumentation on Atalante

To estimate the five rotations R_i, five low-cost MEMS IMUs are placed on Atalante, one in each of the rigid sets S_i, $i \in [0, 4]$. Specifically, one IMU is placed in each foot, one in each tibia, and one in the pelvis.

In the rest of this chapter, we refer to each IMU according to its location: thus, IMU i lies in set S_i. This means that according to (3.3):

$$ {}^{\mathcal{W}}R_{IMU_i} = R_i \, {}^{\mathcal{C}}R_{IMU_i,r} \tag{3.9} $$

Thus, since ${}^{\mathcal{C}}R_{IMU_i,r}$ is measured (being a function of the joint encoder position only according to (3.2)), the problem of estimating R_i is equivalent to that of estimating the orientation of the IMU, i.e. ${}^{\mathcal{W}}R_{IMU_i}$.

3.1.3.1 Description of IMU sensors

The IMUs used on Atalante are composed of a tri-axial MEMS accelerometer and a tri-axial MEMS gyroscope. A gyroscope measures the sensor's angular velocity with respect to the world frame, expressed in the sensor frame. The accelerometer measures the specific acceleration: the linear acceleration of the sensor in the world frame, minus gravity. Thus, the output of an idealized IMU writes

$$ \begin{cases} \boldsymbol{y}_a = {}^{\mathcal{W}}R_{IMU}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU} - \boldsymbol{g}) \\ \boldsymbol{y}_g = \boldsymbol{\omega}_{IMU} \end{cases} \tag{3.10} $$

with $\boldsymbol{g} \triangleq \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T$ is the gravity vector in the world frame (and $g = 9.81$ m/s^{-2}).

In practice, low-cost MEMS IMUs suffer from various errors: these are presented and discussed in Section 3.5.

3.1.3.2 Observability considerations: definition of the tilt.

As apparent in (3.10), only part of the rotation matrix ${}^{\mathcal{W}}R_{IMU}$ is observable. Indeed, consider that a *constant* rotation R_z around $\boldsymbol{e}_z$ is applied to the world frame, and defines a new frame $\mathcal{W}'$. Since $\mathcal{W}'$ is fixed with respect to $\mathcal{W}$, the angular velocity and linear acceleration of the IMU, expressed in the sensor frame, remain identical. Since $R_z^T \boldsymbol{g} = \boldsymbol{g}$, this means that $\mathcal{W}'$ can be substituted to $\mathcal{W}$ in (3.10).

Thus, the sensor's output is left invariant by this rotation, stressing that an IMU cannot distinguish between $\mathcal{W}$ and $\mathcal{W}'$ - or, in other words, that the orientation of the IMU about $\boldsymbol{e}_z$ cannot be reconstructed. In particular, this stresses that an IMU cannot observe the rotation between $\mathcal{L}$ and $\mathcal{W}$. This is a well-known result discussed e.g. in [Mahony08, Benallegue17], which is classically referred to as saying that the yaw angle cannot be estimated[3].

In order to avoid the ambiguity related to the roll-pitch-yaw decomposition, discussed in Appendix B, and the problem of gimbal lock, we instead formulate this property in a more precise way by defining the notion of *tilt* (taken from [Benallegue17]) as follows

Definition 1. *For any rotation $R \in SO(3)$, we call* tilt *the vector*

$$\boldsymbol{t}(R) \triangleq R^T \boldsymbol{e}_z \tag{3.11}$$

We define the tilt of a body $\mathcal{B}$ as the tilt of its rotation with respect to the world frame:

$$\boldsymbol{t}_{\mathcal{B}} \triangleq \boldsymbol{t}({}^{\mathcal{W}}R_{\mathcal{B}}) \tag{3.12}$$

The tilt can be physically interpreted as a measurement of verticality: it corresponds to the perceived gravity field in the body frame. Notice that by definition, the tilt is invariant by a yaw rotation, i.e. $\boldsymbol{t}(R_z R) = \boldsymbol{t}(R)$.

Unlike the seemingly closely related notion of roll and pitch[4], the tilt is singularity-free. This makes it a better parametrization for observer synthesis. For this purpose, the time derivative of this vector as a function of

[3] A common choice used for vehicle navigation to estimate this angle is to add a 3D magnetometer into MEMS IMUs to act as a compass, measuring Earth's magnetic field. Such device combining accelerometers, gyroscopes and magnetometers is called an attitude and heading reference system (AHRS). However, a magnetometer can hardly be exploited when located on a robot, i.e. inside a metallic structure near powerful electric motors with generate a time-varying field, much stronger that much stronger that of the Earth.

[4] Considering a roll-pitch-yaw decomposition (α, β, γ) of a matrix R, its tilt writes according to (A.9) $\boldsymbol{t} = \begin{pmatrix} -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{pmatrix}$. Hence when the decomposition is non-singular ($\sin\beta \neq 0$), the tilt is equivalent to knowing the roll and pitch angle about a given frame of reference.

the angular velocity will be used. This is obtained by computing the time derivative of (3.11), which yields

$$\dot{\boldsymbol{t}}_{\mathcal{B}} = -[\boldsymbol{\omega}_{\mathcal{B}}]_{\times}\boldsymbol{t}_{\mathcal{B}} \tag{3.13}$$

Using the notion of tilt, we can rewrite (3.10) as:

$$\begin{cases} \boldsymbol{y}_a = {}^{\mathcal{W}}R_{IMU}^T\, {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU} + g\boldsymbol{t}_{IMU} \\ \boldsymbol{y}_g = \boldsymbol{\omega}_{IMU} \end{cases} \tag{3.14}$$

The observability property of ${}^{\mathcal{W}}R_{IMU}$ can now be reinterpreted at the light of (3.14) by saying that, using IMU signals only, the best we can hope for is to reconstruct an estimate of the tilt of the IMU, and not the whole rotation matrix. In particular, we cannot reconstruct the orientation of the local frame with respect to the world frame. Note that, conversely, this analysis does not imply that the tilt is observable - this is not always true due to the presence of a linear acceleration term ${}^{\mathcal{W}}R_{IMU}^T\, {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU}$. In Section 3.2 and Section 3.3.1, we show two different models granting observability of the tilt, which we use for observer design. By contrast, in Section 3.4, we use a dynamic model of the flexibilities to render the last component of the rotation observable as well.

3.1.4 Reconstruction of a 3D rotation from a tilt estimate

In the case where the full rotation matrix of the IMU can be estimated (i.e. the *DynamicObserver* presented in Section 3.4), (3.9) directly gives the corresponding value of R_i:

$$\hat{R}_i = {}^{\mathcal{W}}\hat{R}_{IMU_i}\, {}^{\mathcal{C}}R_{IMU_i,r}^T \tag{3.15}$$

In this section, we show how to adapt this formula to the case where only an estimate of the tilt of the IMU, $\hat{\boldsymbol{t}}$, is available.

Estimating the tilt of the IMU, or the tilt of R_i, is actually equivalent: indeed using (3.9) yields

$$\boldsymbol{t}(R_i) = \boldsymbol{t}({}^{\mathcal{W}}R_{IMU_i}\, {}^{\mathcal{C}}R_{IMU_i,r}^T) = ({}^{\mathcal{W}}R_{IMU_i}\, {}^{\mathcal{C}}R_{IMU_i,r}^T)^T\boldsymbol{e}_z = {}^{\mathcal{C}}R_{IMU_i,r}\ \boldsymbol{t}_{IMU_i} \tag{3.16}$$

Since ${}^{\mathcal{C}}R_{IMU_i,r}$ is measured (as a function of encoder measurements only), we obtain an estimate of the tilt of R_i from the IMU estimate simply by multiplying it by ${}^{\mathcal{C}}R_{IMU_i,r}$.

We are thus left with the question of reconstructing R_i from an estimate of its tilt - and therefore require some assumption about the value of the non-observable component. Our approach is simply to neglect the deformations of Atalante around the vertical axis of the system - and to consider that it remains rigid.

Mathematically, this corresponds to setting the non-observable component of the rotation to zero. This process is not completely trivial: the intuitive solution of taking a roll-pitch-yaw decomposition and setting the yaw to zero, is actually ambiguous due to the choice of basis used for the decomposition, and suffers from gimbal lock. This is discussed in Appendix B. We instead propose a more robust solution based on a minimization criteria, presented below.

We first restate a classical result to define a norm on $SO(3)$[Park95, Huynh09]:

Definition 2. *Let $R \in SO(3)$; there exists a unit vector $\boldsymbol{v}$ and a scalar θ such that R is the rotation around $\boldsymbol{v}$ of angle θ. We write $R = \mathcal{R}_0(\boldsymbol{v}, \theta)$[5]. This defines a norm on $SO(3)$ as: $||R|| = |\theta|$.*

This norm enables us to define the smallest rotation matching a given tilt:

Theorem 1. *Let $\boldsymbol{t} \in S^3 \setminus \{-\boldsymbol{e}_z\}$, where S^3 is the unit sphere of $\mathbb{R}^3$. There exists a unique rotation of minimal norm with a tilt equal to $\boldsymbol{t}$: this rotation is given by*

$$R_s = \begin{cases} \mathcal{R}_0\Big(\dfrac{\boldsymbol{e}_z \times \boldsymbol{t}(R)}{||\boldsymbol{e}_z \times \boldsymbol{t}(R)||}, \alpha\left(\boldsymbol{e}_z,\ \boldsymbol{t}(R)\right)\Big) & \text{if} \quad \boldsymbol{t}(R) \neq \boldsymbol{e}_z \\ I_3 & \text{if} \quad \boldsymbol{t}(R) = \boldsymbol{e}_z \end{cases} \tag{3.17}$$

The proof of this result is provided in Appendix B.

This formula can then be applied to the tilt (3.16): note that in practice, the condition $\boldsymbol{t} \neq -\boldsymbol{e}_z$ is always met, since it would correspond to the unrealistic case of the robot being upside down[6].

Thus, from a estimate of the IMU tilt $\hat{\boldsymbol{t}}$ obtained by an observer, we reconstruct an estimate of R_i using (3.17), i.e.:

$$\hat{R}_i = \mathcal{R}_0\Big(\frac{\boldsymbol{e}_z \times {}^{\mathcal{C}}R_{IMU_i,r}\hat{\boldsymbol{t}}}{||\boldsymbol{e}_z \times {}^{\mathcal{C}}R_{IMU_i,r}\hat{\boldsymbol{t}}||}, \alpha\left(\boldsymbol{e}_z,\ {}^{\mathcal{C}}R_{IMU_i,r}\hat{\boldsymbol{t}}\right)\Big) \tag{3.18}$$

[5] $\mathcal{R}_0$ can be expressed using Rodrigues formula ($\mathcal{R}_0(\boldsymbol{v}, \theta) = \cos\theta I_3 + \sin\theta[\boldsymbol{v}]_\times + (1 - \cos\theta)\boldsymbol{v}\boldsymbol{v}^T$), or the matrix exponential ($\mathcal{R}_0(\boldsymbol{v}, \theta) = \exp\left([\theta\boldsymbol{v}]_\times\right)$).

[6] As shown in Appendix B, when $\boldsymbol{t} = -\boldsymbol{e}_z$, there exists an infinite number of rotations with minimal norm matching the tilt: namely, any rotation of angle π around a vector orthogonal to $\boldsymbol{e}_z$.

$\hat{R}_i$ can be interpreted as the smallest rotation in norm that needs to be applied to the rigid robot pose ${}^{\mathcal{C}}R_{\mathcal{B},r}$ to be consistent the tilt observed by the sensor.

Notice that by doing this, we not only neglect the yaw deformation, but also cancel out the robot's heading as well (i.e. the rotation ${}^{\mathcal{W}}R_{\mathcal{L}}$). This remains consistent with our claim of performing an estimation in the local frame only, without trying to reconstruct the odometry coordinates.

3.1.5 Estimating the angular velocities

To reconstruct the velocity of the various bodies of the robot, we are also interested in estimating the angular velocity $\boldsymbol{\omega}_i$ associated to the rotations R_i. This poses no practical difficulty however: indeed, this angular velocity is directly measured. This can be seen by computing the time derivative of (3.9):

$$ {}^{\mathcal{W}}R_{IMU_i}[\boldsymbol{y}_g]_\times = R_i[\boldsymbol{\omega}_i]_\times {}^{\mathcal{C}}R_{IMU_i,r} + R_i {}^{\mathcal{C}}R_{IMU_i,r}[\boldsymbol{\omega}_r]_\times \tag{3.19} $$

where $\boldsymbol{y}_g$ is the reading of the gyroscope of IMU i, and $\boldsymbol{\omega}_r$ is the angular velocity associated to ${}^{\mathcal{C}}R_{IMU_i,r}$. Since this rotation is a function of $\boldsymbol{\theta}$ only, $\boldsymbol{\omega}_r$ can be expressed as a function of $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ (the latter being estimated in practice by numerical differentiation of encoder readings) and is thus considered to be measured.

This rewrites

$$ \boldsymbol{y}_g = {}^{\mathcal{C}}R^T_{IMU_i,r}\boldsymbol{\omega}_i + \boldsymbol{\omega}_r \tag{3.20} $$

and thus

$$ \boldsymbol{\omega}_i = {}^{\mathcal{C}}R_{IMU_i,r}(\boldsymbol{y}_g - \boldsymbol{\omega}_r) \tag{3.21} $$

As appears above, $\boldsymbol{\omega}_i$ is expressed as a simple combination of gyroscope and encoder measurements. Note that, while for the rotation matrix we nullify the non-observable component, the angular velocity with respect to the world frame is fully measured: this is a valuable property as it enables us to express the velocities of the bodies of the robot with respect to the world referential - though only in local coordinates. This is one of the underlying principle of the kinematic observer we present in Section 3.3.

This formula can also be expressed in a more generic fashion: for any frame A in set S_i, we have

$$ {}^{\mathcal{W}}R_A = {}^{\mathcal{W}}R_{IMU}\,{}^{IMU}R_A \tag{3.22} $$

where ${}^{IMU}R_A$ is a function of encoder measurements only: its angular velocity $\boldsymbol{\omega}$ is thus measured as a function of $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$. Consequently, the angular

velocity of ${}^{\mathcal{W}}R_A$ is also measured:

$$\boldsymbol{\omega}_{A/\mathcal{W}} = {}^{A}R_{IMU}(\boldsymbol{y}_g - \boldsymbol{\omega}) \tag{3.23}$$

This can be seen as a simple angular velocity composition formula. This formula is used in Section 3.3.2 on for observer synthesis.

3.1.6 Conclusion

In this section, we have shown how IMU sensors can be employed to reconstruct the state of the robot in the local frame - estimating both the deformations and the contact foot angle. Namely, because only the tilt $\boldsymbol{t}_{IMU}$ of the IMU can be estimated, we neglect the yaw deformations using (3.18). When a full rotation matrix is obtained this hypothesis may be dropped, and (3.15) used instead. Meanwhile, the associated angular velocities are measured, as given by (3.21).

The next three sections of this chapter now focus on obtaining this tilt estimate, $\hat{\boldsymbol{t}}_{IMU}$, or a complete rotation estimate $\hat{R}_{IMU}$, from each IMU. Three successive methods are presented, and are compared at the end of the chapter.

3.2 Static observer: using a zero-on-average acceleration assumption

3.2.1 Observer formulation

In the context of attitude estimation, a classical assumption when working with an accelerometer is to consider that the sensor acceleration is, on average, negligible compared to gravity [Martin07, Mahony08, Khandelwal13, Barrau17] : in other words, to work in a quasi-static context. This assumption is qualitatively supported by personal experience of human walking. While fast running motion feature significant dynamical effects (e.g. centrifugal forces, that prevent sudden changes in direction while sprinting), these are not felt while walking: gravity remains the dominant force.

Under this quasi-static assumption, the accelerometer gives a direct measurement of the tilt we are trying to estimate, as (3.10) can be simplified by removing the linear acceleration into

$$\boldsymbol{y}_a = -{}^{\mathcal{W}}R_{IMU}^{T}\boldsymbol{g} = -g\boldsymbol{t}_{IMU} \tag{3.24}$$

One possible observer for computing a tilt estimate under this assumption is the nonlinear complementary filter presented in [Mahony08] - which we here

call the "*StaticObserver*". In the case where additional sensors render the full rotation observable, rotation matrices or quaternions are used to write the observer. This is how the observer is presented in this article. However, in the present case where, in addition to the gyroscope, only an accelerometer is available, this observer can be readily rewritten using our formulation of tilt, as presented below.

A convenient feature of this observer is that, alongside estimating the tilt, it is able to estimate the gyroscope additive bias. Indeed, MEMS gyroscope usually present a significant constant additive bias, $\boldsymbol{b}_g$ so that the sensor output writes

$$\boldsymbol{y}_g = \boldsymbol{\omega}_{IMU} + \boldsymbol{b}_g \tag{3.25}$$

To design an observer, the following error term is defined:

$$\boldsymbol{e} \triangleq -\frac{\boldsymbol{y}_a}{g} \times \hat{\boldsymbol{t}} \tag{3.26}$$

This term measures the misalignment between the accelerometer reading and the estimated tilt - which should be collinear according to (3.24). The observer then takes the form of a copy of the tilt dynamics (3.13) (removing the gyroscope's bias estimate), plus a feedback on $\boldsymbol{e}$ generating an additional rotational velocity of the estimate as long as the error is non-zero

$$\begin{cases} \dot{\hat{\boldsymbol{t}}} = -[\boldsymbol{y}_g - \hat{\boldsymbol{b}}_g + k_a \boldsymbol{e}]_\times \hat{\boldsymbol{t}} \\ \dot{\hat{\boldsymbol{b}}}_g = -k_b \boldsymbol{e} \end{cases} \tag{3.27}$$

where k_a, k_b are positive tuning parameters. Notice that this observer is non-linear in the sense that the error term $k_a \boldsymbol{e}$ is not added additively to the dynamics of the tilt, but rather inside the cross product, in the form $-[k_a \boldsymbol{e}]_\times \hat{\boldsymbol{t}}$. Mathematically, this guarantees that $\hat{\boldsymbol{t}}$ remains a unitary vector - and thus a valid estimate of the tilt[7]. The term $k_a \boldsymbol{e}$ can be interpreted as an extra angular velocity that is applied as a compensation of the measured angular velocity (gyroscope measurement minus bias) in order to realign the estimated gravity vector $g\ \hat{\boldsymbol{t}}$ with the accelerometer - thereby bringing $\boldsymbol{e}$ to zero.

The proof of convergence of this observer is presented in [Mahony08]. One notable property is that, though only the tilt of the rotation can be estimated, the full 3D bias of the gyroscope can be estimated, granted that a condition

[7]When implementing the observer in discrete time on a computer, the vector $\hat{\boldsymbol{t}}$ is renormalized at every time step to prevent the accumulation of rounding error on the norm.

of persistence of excitation is met. This can be intuitively explained by considering that $\hat{\boldsymbol{b}}$ is updated only along $\boldsymbol{e}$, i.e. a direction orthogonal to the accelerometer measurement $\boldsymbol{y}_a$. When the system remains motionless, only two of the three bias components can be observed, and the non-observable component is kept constant. But as soon as the orientation of the system changes (sufficiently in the sense of persistence of excitation), the third component gets to be estimated as well.

Though its great simplicity makes this observer very appealing, in practice its performance remains structurally limited by the "zero acceleration on average" hypothesis. Indeed, this observer constantly tries to align the estimated tilt $\hat{\boldsymbol{t}}$ with the accelerometer, something that is erroneous when linear accelerations are present.

The effect of a non-persistent linear acceleration can be mitigated by reducing the feedback gains k_a and k_b in (3.27). By proceeding so, only the low frequency content of the accelerometer signal is used. This is detrimental, and causes a very slow convergence speed. Furthermore, even with low gains, a linear acceleration still negatively impacts the behavior of this observer: this is illustrated in a simple example below.

3.2.2 Analysis of the effect of a linear acceleration on this observer: analogy with the inverted pendulum

To analyze the effect of linear acceleration on (3.27), and predict the behavior of this observer on the robot, we study the simplest representation of Atalante: a 1D inverted pendulum oscillating at small angles in the sagittal plane. This model nicely describes the behavior of the robot during a set of static single support experiments, which have been used as a benchmark scenario in this thesis. In this scenario, the exoskeleton is tasked to remain still in single support, and the exoskeleton's back is manually pushed. Due to the presence of flexibilities, the robot oscillates about its reference pose at about 1 Hz.

Interestingly, this model remains relevant even during walking phases. Indeed, the linear inverted pendulum (LIP) model is often used in bipedal robotics to describe the motion of the center of mass [Kajita01, Pratt06, Caron20]. Since, by construction, the robot's pelvis remains close to the center of mass, the LIP model describes well the motion of the pelvis IMU while walking.

Consider a pendulum of length l pivoting by an angle θ around the $\boldsymbol{e}_y$ axis, as illustrated in Figure 3.3. The observer (3.27) can be rewritten for

this planar case, where only one angle θ needs to be estimated, as

$$\begin{cases} e = -\dfrac{1}{g} \det\left(\boldsymbol{y}_a, \begin{pmatrix} -\sin\hat{\theta} \\ \cos\hat{\theta} \end{pmatrix}\right) \\ \dot{\hat{\theta}} = y_g - \hat{b} + k_a e \\ \dot{\hat{b}} = -k_b e \end{cases} \tag{3.28}$$

where y_g is the measurement of a 1D gyroscope about axis $\boldsymbol{e}_y$, and $\boldsymbol{y}_a$ the measurements of a planar accelerometer about $\boldsymbol{e}_x$ and $\boldsymbol{e}_z$.

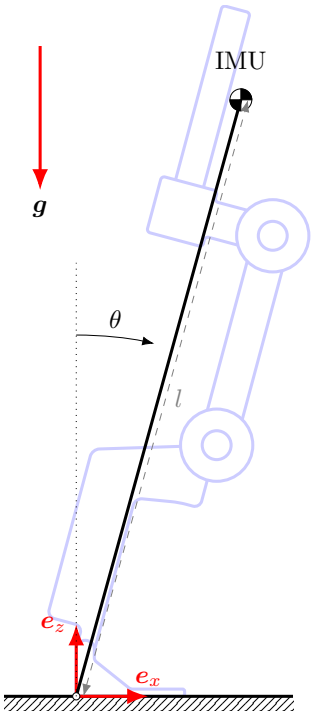


Figure 3.3: A simple inverted pendulum, which can be seen as a minimal representation of the motion of the pelvis IMU.

In the case of the pendulum of Figure 3.3, the sensor measurements write

$$\begin{cases} \boldsymbol{y}_a = \begin{pmatrix} -g\sin\theta + l\ddot{\theta} \\ g\cos\theta - l\dot{\theta}^2 \end{pmatrix} \\ y_g = \dot{\theta} + b_g \end{cases} \tag{3.29}$$

Combining (3.28) and (3.29) yields

$$e = \sin(\theta - \hat{\theta}) + \frac{l}{g}(\dot{\theta}^2 \sin\hat{\theta} - \ddot{\theta}\cos\hat{\theta}) \tag{3.30}$$

Assuming small displacements of the pendulum, we linearize this equation about $\theta = 0$[8] which yields

$$e = \theta - \hat{\theta} - \frac{l}{g}\,\ddot{\theta} \tag{3.31}$$

Notice that (3.31) clearly shows the nefarious effect of an acceleration: if no acceleration is present, then e is indeed a measurement of the error between θ and $\hat{\theta}$ - but the acceleration acts as an extra bias term.

Injecting (3.31) into (3.28), we get the following linear observer dynamics:

$$\begin{cases} \dot{\hat{\theta}} = \dot{\theta} + b - \hat{b} + k_a(\theta - \hat{\theta} - \dfrac{l}{g}\ddot{\theta}) \\ \dot{\hat{b}} = -k_b(\theta - \hat{\theta} - \dfrac{l}{g}\ddot{\theta}) \end{cases} \tag{3.32}$$

Differentiating the first equation to remove the constant b term, we obtain the following second-order dynamics for the estimated angle:

$$\ddot{\hat{\theta}} = \ddot{\theta} + k_b(\theta - \hat{\theta} - \frac{l}{g}\ddot{\theta}) + k_a(\dot{\theta} - \dot{\hat{\theta}} - \frac{l}{g}\dddot{\theta}) \tag{3.33}$$

Thus, the proposed observer (near the true value of θ) is simply a linear time-invariant filter, with transfer function

$$\mathcal{H}(s) \triangleq \frac{\hat{\theta}}{\theta} = 1 - \frac{l}{g}\frac{s^2 k_b + s^3 k_a}{s^2 + s k_a + k_b} \tag{3.34}$$

The corresponding Bode diagram is represented in Figure 3.4. As expected from the transfer function, at low frequencies the transfer function is approximately equal to 1 , and thus the observer gives a satisfactory estimate of θ. At high frequencies however the observer has an asymptote of $-\frac{l}{g}k_a s$ as the acceleration terms becomes dominant over the gyroscope's velocity. The observer therefore no longer gives a reliable orientation estimate, i.e. a transfer of 1 between θ and $\hat{\theta}$.

From (3.34), we can see that the bandwidth of the filter can be increased by decreasing the values of k_a and k_b. The lower the gains, the less we trust the accelerometer, and thus the more acceleration we can handle. But reducing the gains also means having a slower convergence time: indeed (3.33) shows that, in the absence of acceleration, the observer converges exponentially to θ with a characteristic time of $\tau = \frac{2}{k_a}$.

[8]The global convergence property of the observer justifies the fact that, if $\theta \approx 0$ is small, then eventually $\hat{\theta} \approx \theta$ and these terms can be linearized as well. We are simply studying the local behavior of the equations.

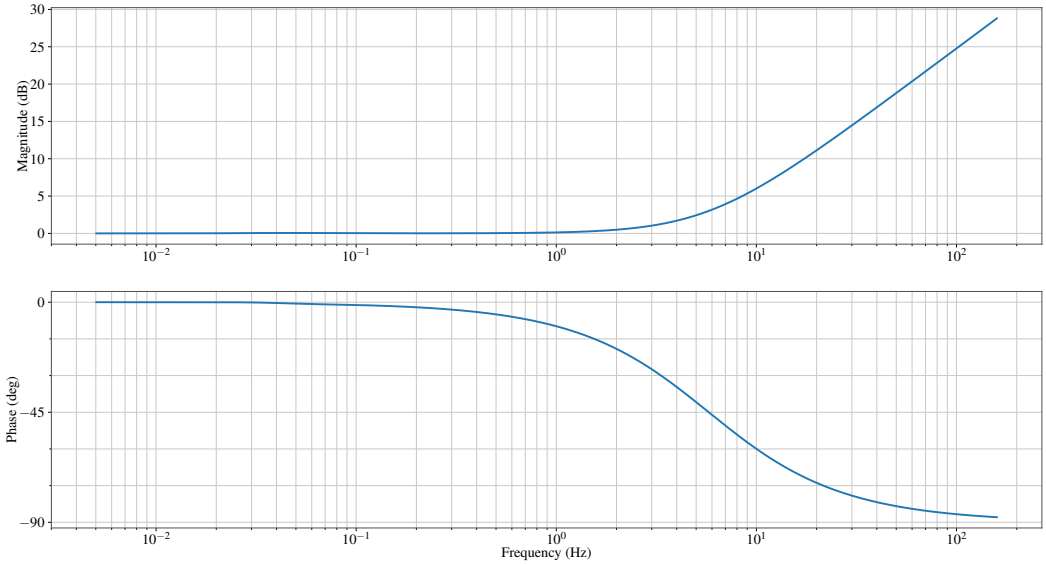


Figure 3.4: Bode diagram of the observer, for $k_a = 0.27$ and $k_b = 0.07$, the values used on the robot.

In practice, another phenomenon empirically limits the useful range of these gains: the overall quality of the gyroscope. Indeed, the lower the gains, the more we rely on pure gyroscope integration to obtain an attitude estimate. But MEMS gyroscopes are not perfect: in addition to the estimated constant bias $\boldsymbol{b}_g$, a gyroscope also suffers from other defects such as, among other, normalization and orthogonalisation errors, sensitivity to impacts, colored noise... [Titterton04, Woodman07] Thus, the attitude estimate computed through open loop integration will ultimately drift, even if an estimated bias is taken into account. This is why the accelerometer feedback term remains important, to anchor the attitude about a known direction.

Therefore, the tuning of the gains k_a and k_b on Atalante is performed not in terms of theoretical bandwidth, but rather using brute-force optimization, to minimize the reconstruction error on a chosen set of experiments, featuring both static experiments and walking motions. This leads to the choice of $k_a = 0.27$ and $k_b = 0.07$. As a sanity check, notice on the Bode plot of Figure 3.4 that this gain value corresponds to a 3dB bandwidth of 5.5 Hz, a seemingly reasonable choice considering that the walk pattern happens at 1 Hz.

Yet Figure 3.4 allows us to anticipate the effect linear accelerations will have on the walking pattern: though the gain at 1 Hz is almost equal to one, this observer will induce a significant phase shift of 10°, i.e. a lag of 27 ms at 1 Hz, while having a high characteristic time of $\boldsymbol{t}au = 7.4$ s which implies very

slow convergence. Both of these effects are indeed observed experimentally, as shown in Section 3.6 - and justify the need for a more advanced observer, not restricted by a static hypothesis.

3.3 Kinematics-based observer: acceleration as velocity derivative

Instead of neglecting the linear acceleration of the IMU, as done in the previous section, we here design an observer that exploits a simple fact: the linear acceleration is the time derivative of the linear velocity. Assuming that a measurement of the linear velocity in the IMU frame is available, this relationship can be used to build a tilt observer that is no longer limited by accelerated motions. This question first received interest in the case of outdoor ground or aerial vehicles, where the use of GPS or computer vision (optical flow [Barron94]) can provide such velocity measurement (see for instance [Bonnabel09, Herissé11, Hua16]).

In the context of a rigid robot pivoting about *a single fixed point* in its environment, [Benallegue17] shows that already available joint encoder and IMU measurements are sufficient to provide a measurement of this linear velocity. This enables the design of such an observer, without requiring additional sensors. This is the philosophy we extend, to enable the support of multiple deformations in an original algorithm which we call the *KinematicObserver*.

3.3.1 Velocity-aided observer for a system with only one unmeasured spherical joint

For tutorial purposes, in this section we recall the results of [Benallegue17], which serve as a basis for the general observer we design in Section 3.3.2.

Consider the rigid model of Atalante in single support, as presented in (1.2). We furthermore assume for now that the robot's foot can be modeled as punctual (this hypothesis will be fully relaxed later on). More generally than just Atalante, we consider a rigid robot fully equipped with joints encoders, capable of measuring the position and velocity $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ of the joints, in punctual contact with the ground. We call O_0 the contact point, which is assumed to be fixed (i.e. the foot is not slipping and the ground is hard), and set this as the origin of the world frame. This is presented in Figure 3.5.

Using the robot's forward kinematic model, the position and orientation of any body in frame $\mathcal{C}$ can be determined as a function of encoder measurement:

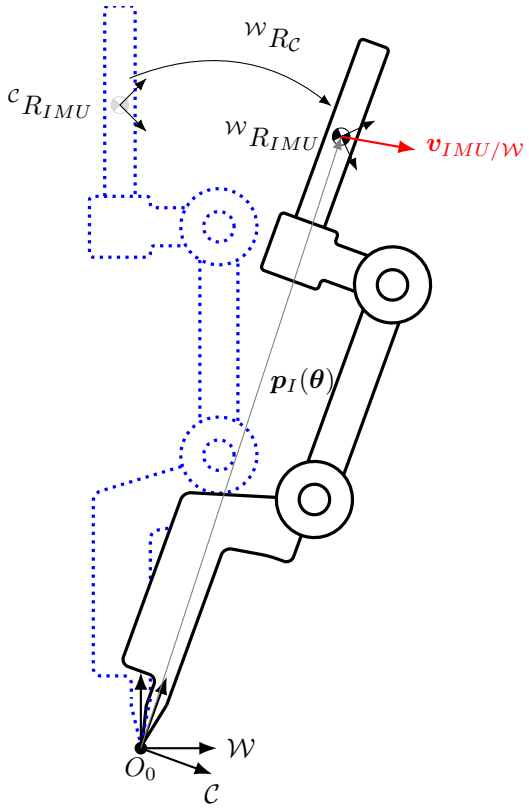


Figure 3.5: Representation of the system: a rigid robot with point feet, rotating about the world frame around a fixed point O_0.

in what follows, we explicit this dependency by writing these quantities as a function of $\boldsymbol{\theta}$ (and $\dot{\boldsymbol{\theta}}$ for the velocity terms).

As detailed earlier, our goal is to estimate the tilt of the rotation $^{\mathcal{W}}R_{\mathcal{C}}$, using a *single* IMU placed anywhere on the *rigid* robot[9]. This is equivalent to estimating the tilt of this IMU as

$$\boldsymbol{t}_{IMU} \triangleq \boldsymbol{t}(^{\mathcal{W}}R_{IMU}) = {}^{\mathcal{C}}R_{IMU}^{T}(\boldsymbol{\theta})\boldsymbol{t}(^{\mathcal{W}}R_{\mathcal{C}}) \tag{3.35}$$

and $^{\mathcal{C}}R_{IMU}(\boldsymbol{\theta})$, the orientation of the IMU with respect to the contact foot, is measured by joint encoders.

We begin by showing that the kinematics of the studied system allow for a measurement of the linear velocity of the IMU with respect to the world frame, in the sensor frame. We denote $\boldsymbol{v} \triangleq {}^{\mathcal{W}}R_{IMU}^{T}\,{}^{\mathcal{W}}\dot{\boldsymbol{p}}_{IMU}$ this velocity.

Let $\boldsymbol{p}_I(\boldsymbol{\theta}) \triangleq {}^{IMU}R_{\mathcal{C}}(\boldsymbol{\theta})\,{}^{\mathcal{C}}\boldsymbol{p}_{IMU}(\boldsymbol{\theta})$ be the position of the IMU with respect to the contact foot frame, expressed in the IMU frame: this "internal" position

[9]This rotation may be the result of the support foot tilting or resting on a non-horizontal ground, or the consequence of an ankle flexibility, as considered in [Benallegue17].

of the IMU, which represents the position of the IMU relative to another link of the robot (the contact foot) is known as a function of encoder readings, unlike the absolute position of the IMU in the world frame, which is not measured. We note $\dot{\boldsymbol{p}}_I(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ its time derivative, i.e. the velocity of the IMU with respect to the contact foot, in the sensor frame.

Since the contact foot frame and the world frame share the same origin (under the punctual foot assumption), ${}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{C}} = 0$. Thus, the position of the IMU in the world frame writes

$$ {}^{\mathcal{W}}\boldsymbol{p}_{IMU} = {}^{\mathcal{W}}R_{IMU}\boldsymbol{p}_I(\boldsymbol{\theta}) \tag{3.36} $$

Injecting (3.36) into the definition of $\boldsymbol{v}$, we obtain an explicit expression of this velocity:

$$ \begin{aligned} \boldsymbol{v} &= {}^{\mathcal{W}}R_{IMU}^T \frac{d}{dt}({}^{\mathcal{W}}R_{IMU}\boldsymbol{p}_I(\boldsymbol{\theta})) \\ &= {}^{\mathcal{W}}R_{IMU}^T \left({}^{\mathcal{W}}R_{IMU}[\boldsymbol{\omega}_{IMU}]_\times \boldsymbol{p}_I(\boldsymbol{\theta}) + {}^{\mathcal{W}}R_{IMU}\dot{\boldsymbol{p}}_I(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \right) \\ &= [\boldsymbol{y}_g]_\times \boldsymbol{p}_I(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_I(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{aligned} \tag{3.37} $$

This shows that the linear velocity $\boldsymbol{v}$ can be expressed in terms of encoder position, encoder velocity, and gyroscope measurement $\boldsymbol{y}_g$. This velocity is thus available as an extra measurement in any state estimation methodology.

To use it, we relate its dynamics to the accelerometer measurement. Indeed, from (3.14) we get

$$ \begin{aligned} \boldsymbol{y}_a &= {}^{\mathcal{W}}R_{IMU}^T {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU} + g\ \boldsymbol{t}_{IMU} \\ &= {}^{\mathcal{W}}R_{IMU}^T \frac{d}{dt}({}^{\mathcal{W}}\dot{\boldsymbol{p}}_{IMU}) + g\ \boldsymbol{t}_{IMU} \\ &= {}^{\mathcal{W}}R_{IMU}^T \frac{d}{dt}({}^{\mathcal{W}}R_{IMU}\boldsymbol{v}) + g\ \boldsymbol{t}_{IMU} \\ &= [\boldsymbol{y}_g]_\times \boldsymbol{v} + \dot{\boldsymbol{v}} + g\ \boldsymbol{t}_{IMU} \end{aligned} \tag{3.38} $$

Combined with (3.13), this gives the following system dynamics

$$ \begin{cases} \dot{\boldsymbol{v}} = -[\boldsymbol{y}_g]_\times \boldsymbol{v} + \boldsymbol{y}_a - g\ \boldsymbol{t}_{IMU} \\ \dot{\boldsymbol{t}}_{IMU} = -[\boldsymbol{y}_g]_\times \boldsymbol{t}_{IMU} \end{cases} \tag{3.39} $$

Note that (3.39) are not specific to the particular kinematics of our system, but are true for any motion. They are simply consequences of computing the time derivative of rotation matrices for any system undergoing motion with rotations (i.e. an application of the Frenet–Serret formulas).

In our opinion, a major contribution of [Benallegue17] is to show that, in the presented case of a robot with point feet pivoting about O_0, existing onboard instrumentation is sufficient to reconstruct this velocity at not extra hardware cost.

Equation (3.39) now enables the formulation of an observer, which unlike the previous *StaticObserver* takes into account the presence of linear accelerations. Several observers have been proposed relying on these equations: see for instance [Hua16, Benallegue17, Benallegue20]. In this thesis, we naturally use the formulation proposed in [Benallegue17], as their use case (the humanoid robot HRP-2) is very similar to ours. This observer writes

$$\begin{cases} \dot{\hat{\boldsymbol{v}}} = -[\boldsymbol{y}_g]_\times \hat{\boldsymbol{v}} + \boldsymbol{y}_a - g\hat{\boldsymbol{t}} + \alpha(\hat{\boldsymbol{v}} - \boldsymbol{v}) \\ \dot{\hat{\boldsymbol{t}}} = -[\boldsymbol{y}_g - \beta[\hat{\boldsymbol{t}}]_\times(\hat{\boldsymbol{v}} - \boldsymbol{v})]_\times \hat{\boldsymbol{t}} \end{cases} \tag{3.40}$$

with α, β positive observer gains such that $\beta g_0 < \alpha^2$.

A proof of convergence is provided in [Benallegue17]: just like the previous observer, this observer benefits from almost global convergence, in the sense that away from a zero-measure set of initial condition (the unstable equilibrium point $\hat{\boldsymbol{t}} = -\boldsymbol{e}_z$), the error converges to zero.

Notice that, unlike the previous *StaticObserver*, no approximation is made in the formulation of this observer, which can therefore handle arbitrarily large acceleration. A notable strength of this estimation technique is that it is achieved using only velocity measurements: the encoder acceleration $\ddot{\boldsymbol{\theta}}$ does not need to be estimated. This is significant in practice, given that computing this quantity from $\boldsymbol{\theta}$ using numerical differentiation of noisy and quantized measurements can lead to significant amplification of noises.

To summarize, (3.40) gives an observer for the tilt of an accelerated IMU, granted that a measurement of its linear velocity $\boldsymbol{v}$ is available. For the case of an articulated multi-body system rotating about a known fixed point O_0 (i.e. a rigid robot with point feet), this velocity is indeed measured, as a combination of encoder and gyroscope measurements.

3.3.2 The *KinematicObserver*: a cascade implementation on Atalante

In this section, we propose an extension of the work presented above, to adapt it to Atalante, i.e. to a system with *non-punctual foot* where *more than one uninstrumented rotation needs to be estimated*. The resulting algorithm is the observer we call the *KinematicObserver*. While the observer of the previous section can be directly applied to the contact foot IMU, provided that the center of rotation is properly defined, a more significant modification is

needed for subsequent IMUs along the kinematic chain. For tutorial purposes, we first present the case of the contact foot IMU in Section 3.3.2.1, before generalizing it in Section 3.3.2.2 to encompass all the IMUs in a single, unified formulation.

3.3.2.1 Direct application to the support foot IMU

The application of the observer (3.40) to the contact foot IMU requires a proper definition of the ground contact point, in order to compute the IMU velocity. Indeed, in the case of a non-punctual foot for which we still assume non-slipping motion, the center of rotation O_0 is time-varying, going from one edge of the foot to the other as the foot tilts on the ground: its position needs to be known in order to compute the linear velocity (3.37).

For that purpose, we use force sensor measurements, and locate O_0 at the center of pressure of the contact foot - in other words, we consider that the system is rotating about this center of pressure. Indeed, if the center of pressure is located strictly inside the support foot, then this foot rests flat on the ground and is not moving: the position of O_0 is then irrelevant. Conversely, as the foot starts to tilt about one of its edge, the center of pressure will be located on the corresponding edge. With this slight modification, (3.40) is used to estimate the tilt of the support foot IMU.

3.3.2.2 Generalization for the other IMUs

For the other IMUs, we run into the fact that the linear velocity $\boldsymbol{v}_i$ of these IMUs is no longer measured, due to the presence of several points of rotations O_i in the kinematic chain. Nevertheless, a very good approximation of this linear velocity can be obtained, by decomposing the motion of an IMU along the kinematic chain and exploiting the estimations coming from the previous observers in a cascade. More precisely, we use the estimations from IMU 0 to $i-1$ to estimate the velocity of IMU i, from which we estimate an attitude using (3.40). Before giving the mathematical details of this process, we illustrate it on an example. We take $i = 2$, and thus consider the pelvis IMU: this is shown in Figure 3.6.

The velocity of the pelvis IMU, expressed in the local frame, $\boldsymbol{v}_2$, is decomposed as the sum of two velocities:

- The velocity of point O_2, the origin of the previous deformations (here the right hip, in orange in Figure 3.6).
- The velocity due to the motion of the IMU with respect to O_2, i.e. the motion of set $\mathcal{S}_2$ ($\boldsymbol{p}_{I_2}$ and its derivative, in red in Figure 3.6).

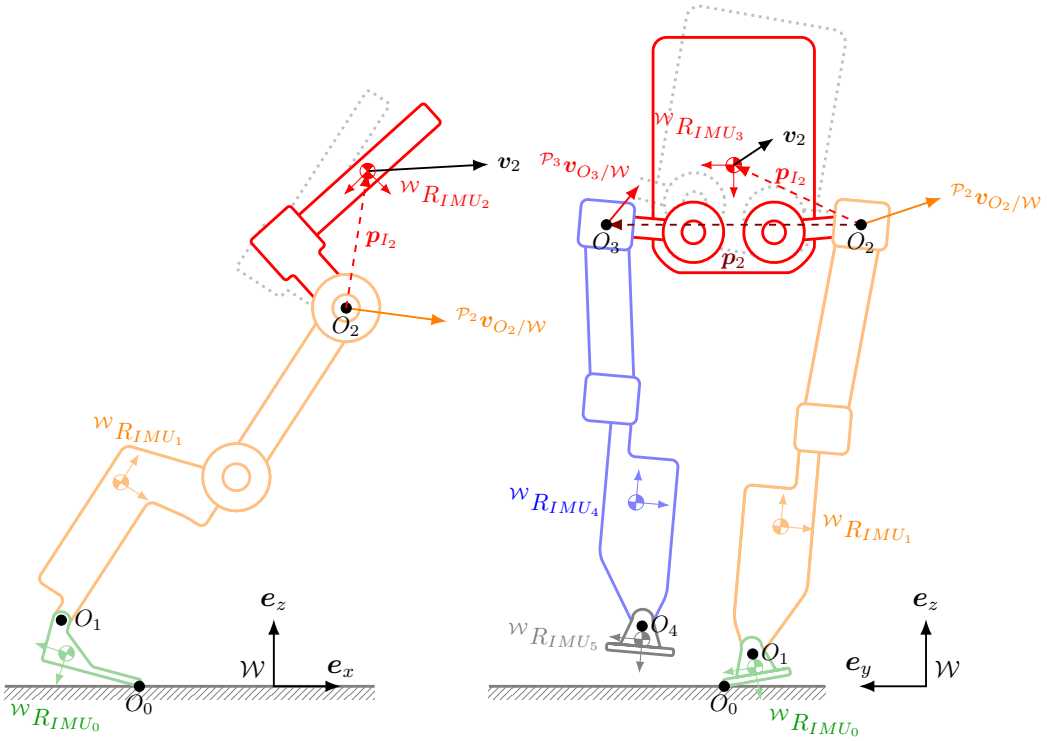


Figure 3.6: Illustration of the induction reasoning of the *KinematicObserver* for $i = 2$: the velocity of the pelvis IMU (black arrow) is the sum of the velocity of the deformation's origin O_2 (in orange), estimated by the previous observers, and the velocity induced by motions of the current rigid set (the hips, in red), measured by the joint encoders.

The motion of set $\mathcal{S}_2$ is measured by the joint encoder: thus, the associated velocity term is measured. Meanwhile, the velocity $\boldsymbol{v}_{O_2}$ can be estimated, in the world frame, using sensor measurements and an attitude estimation of the previous IMUs (IMU 0 and IMU 1). This velocity however needs to be rotated into the IMU frame: this rotation is precisely the orientation we seek to estimate, and is thus unknown. Instead, we approximate this rotation by neglecting the last deformation (D_2 about O_2): this gives us an estimation of the desired velocity, $\bar{\boldsymbol{v}}_2$ in (3.49), which depends only on the attitude estimation of all previous IMUs. This velocity is used to run the observer (3.40) on IMU_2 - and the process is then iterated on the next IMU.

Thus, the *KinematicObserver* we propose is constructed in successive steps, following an iterative procedure. At each step, we use the previous estimates to propagate the orientation and velocity estimation from one deformation to the next, and use it for attitude estimation.

This formulation is in fact valid for the contact foot IMU as well, which is only a specific, simpler subcase. To allow a consistent notation with the deformations, which are parametrized by the frames $\mathcal{P}_i$ and $\mathcal{C}_i$, we denote $\mathcal{C}_0 = \mathcal{C}$ and $\mathcal{P}_0 = \mathcal{L}$. Then, for the generic case of a robot with n deformations, the *KinematicObserver* algorithm writes

***KinematicObserver* algorithm.**
For i from 0 to n:

1. *compute an estimate of the linear velocity of point O_i with respect to the world frame, in the parent frame $\mathcal{P}_i$ (i.e. before the deformation). We denote this velocity ${}^{\mathcal{P}_i}\hat{\boldsymbol{v}}_{O_i/\mathcal{W}}$.*
2. *use this estimate to compute an approximation of the velocity of IMU i, and thus use the observer* (3.40) *to estimate its tilt.*

Both steps of the algorithm are detailed below:

Step 1: computation of the linear velocity of O_i For the first IMU, the contact assumption simply gives ${}^{\mathcal{P}_i}\boldsymbol{v}_{O_i/\mathcal{W}} = 0$.

For $i \geq 1$, we use the velocity and IMU tilt estimation from the previous IMU (i.e. step $i-1$ of the algorithm) to compute this velocity. More precisely, we decompose the position of point O_i about O_{i-1}: let $\boldsymbol{p}_i(\boldsymbol{\theta})$ be the position of O_i with respect to O_{i-1}, expressed in frame $\mathcal{P}_i$. This position is a function of the encoder measurement only, as it depends only on the configuration of set $\mathcal{S}_{i-1}$[10], and thus is measured. Then, the position of O_i in the world frame writes

$$ {}^{\mathcal{W}}\boldsymbol{p}_{O_i} = {}^{\mathcal{W}}R_{\mathcal{P}_i}\boldsymbol{p}_i(\boldsymbol{\theta}) + {}^{\mathcal{W}}\boldsymbol{p}_{O_i} \tag{3.41} $$

Differentiating this equation yields

$$ {}^{\mathcal{W}}\boldsymbol{v}_{O_i/\mathcal{W}} = {}^{\mathcal{W}}R_{\mathcal{P}_i}\Big([\boldsymbol{\omega}_{\mathcal{P}_i/\mathcal{W}}]_\times \boldsymbol{p}_i(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_i(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\Big) + {}^{\mathcal{W}}\boldsymbol{v}_{O_{i-1}/\mathcal{W}} \tag{3.42} $$

Multiplying both sides of (3.42) by ${}^{\mathcal{W}}R_{\mathcal{P}_i}^T$ thus yields

$$ {}^{\mathcal{P}_i}\boldsymbol{v}_{O_i/\mathcal{W}} = [\boldsymbol{\omega}_{\mathcal{P}_i/\mathcal{W}}]_\times \boldsymbol{p}_i(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_i(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + {}^{\mathcal{P}_i}R_{\mathcal{P}_{i-1}}\,{}^{\mathcal{P}_{i-1}}\boldsymbol{v}_{O_{i-1}/\mathcal{W}} \tag{3.43} $$

Finally, the rotation ${}^{\mathcal{P}_i}R_{\mathcal{P}_{i-1}}$ can be decomposed about the $i-1$ deformation into ${}^{\mathcal{P}_i}R_{\mathcal{P}_{i-1}} = {}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}(\boldsymbol{\theta})\,{}^{\mathcal{C}_{i-1}}R_{\mathcal{P}_{i-1}} = {}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}(\boldsymbol{\theta})D_{i-1}^T$. The rotation

[10] Recall that $\mathcal{P}_i$, the parent frame of the ith deformation, is simply the last frame of set $\mathcal{S}_{i-1}$.

${}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}(\boldsymbol{\theta})$, i.e. the rotation from the first frame to the last frame of set S_{i-1}, is thus a function of encoder measurement only. Thus, the velocity of point O_i writes

$$ {}^{\mathcal{P}_i}\boldsymbol{v}_{O_i/\mathcal{W}} = [\boldsymbol{\omega}_{\mathcal{P}_i/\mathcal{W}}]_\times \boldsymbol{p}_i(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + {}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}(\boldsymbol{\theta}) D_{i-1}^T \, {}^{\mathcal{P}_{i-1}}\boldsymbol{v}_{O_{i-1}/\mathcal{W}} \quad (3.44) $$

Since $\mathcal{P}_i$ and IMU $i-1$ belong to the same set ($\mathcal{S}_{i-1}$), $\boldsymbol{\omega}_{\mathcal{P}_i/\mathcal{W}}$ is measured according to (3.23). Meanwhile, the previous iteration of the algorithm (at step $i-1$) gives an estimate of D_{i-1} and ${}^{\mathcal{P}_{i-1}}\boldsymbol{v}_{O_{i-1}/\mathcal{W}}$. Hence, the following estimation of $\boldsymbol{v}_{O_i}$ is constructed

$$ {}^{\mathcal{P}_i}\hat{\boldsymbol{v}}_{O_i/\mathcal{W}} = [\boldsymbol{\omega}_{\mathcal{P}_i/\mathcal{W}}]_\times \boldsymbol{p}_i(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + {}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}(\boldsymbol{\theta}) \hat{D}_{i-1}^T \, {}^{\mathcal{P}_{i-1}}\hat{\boldsymbol{v}}_{O_{i-1}/\mathcal{W}} \quad (3.45) $$

This concludes step 1 of the algorithm.

Remark. *Note that ${}^{\mathcal{P}_i}\hat{\boldsymbol{v}}_{O_i/\mathcal{W}}$ is simply computed by applying the forward kinematics model f_f to the IMU measurements and the previous estimations, "propagating" the velocity accross set $\mathcal{S}_{i-1}$ from point O_{i-1} to point O_i. In other words, this is not a new state being estimated, but simply a mathematical byproduct of the previous estimates. We only detailed its computation in* (3.45) *because this velocity is instrumental to the implementation of the observer.*

Step 2: estimating the tilt of IMU i To estimate the tilt of IMU i using (3.40), we need an estimate of its linear velocity in the sensor frame, $\boldsymbol{v}_i$. This is obtained by performing the same computation as done in *Step 1*: this time, instead of propagating the velocity of O_{i-1} to O_i across set $\mathcal{S}_{i-1}$ and deformation D_{i-1}, we propagate the velocity of O_i to IMU_i across set $\mathcal{S}_i$ and deformation D_i. While this only corresponds to performing substitutions in (3.45), we briefly detail anew the computation for the sake of clarity.

Let $\boldsymbol{p}_{I_i}(\boldsymbol{\theta})$ be the position of IMU i with respect to O_i, expressed in the IMU frame. This position is a function of the configuration of set $\mathcal{S}_i$ only. The position of the IMU in the world frame then writes

$$ {}^{\mathcal{W}}\boldsymbol{p}_{IMU_i} = {}^{\mathcal{W}}R_{IMU_i}\boldsymbol{p}_{I_i}(\boldsymbol{\theta}) + {}^{\mathcal{W}}\boldsymbol{p}_{O_i} \quad (3.46) $$

Differentiating (3.46) yields

$$ {}^{\mathcal{W}}\boldsymbol{v}_{IMU_i/\mathcal{W}} = {}^{\mathcal{W}}R_{IMU_i}[\boldsymbol{\omega}_{IMU_i}]_\times \boldsymbol{p}_{I_i}(\boldsymbol{\theta}) + {}^{\mathcal{W}}R_{IMU_i}\dot{\boldsymbol{p}}_{I_i}(\boldsymbol{\theta}) + {}^{\mathcal{W}}\boldsymbol{v}_{O_i/\mathcal{W}} \quad (3.47) $$

Multiplying by ${}^{\mathcal{W}}R_{IMU_i}^T$ and decomposing ${}^{IMU_i}R_{\mathcal{P}_\rangle} = {}^{IMU_i}R_{\mathcal{C}_\rangle}(\boldsymbol{\theta}) D_i^T$, we obtain the expression of $\boldsymbol{v}_i$ in a form identical to (3.45):

$$ \boldsymbol{v}_i = [\boldsymbol{y}_{g_i}]_\times \boldsymbol{p}_{I_i}(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_{I_i}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + {}^{IMU_i}R_{\mathcal{C}_\rangle}(\boldsymbol{\theta}) D_i^T \, {}^{\mathcal{P}_i}\boldsymbol{v}_{O_i/\mathcal{W}} \quad (3.48) $$

Notice that Equation (3.48) is similar to (3.37) that describes the case where only one deformation is present, with the addition of one extra term: the linear velocity of point O_i expressed in the IMU frame. Again, the first two terms of (3.48) are measured. But this is not the case of the last term: though we have an estimate ${}^{\mathcal{P}_i}\hat{\boldsymbol{v}}_{O_i/\mathcal{W}}$ of the linear velocity, (3.48) also depends on D_i, i.e. the attitude of the IMU we are trying to estimate.

The reconstruction choice we make consists of neglecting the ith deformation in the computation of this velocity. We thus set $D_i = I$, and obtain an *approximate measurement* of the linear velocity $\boldsymbol{v}_i$ under the form

$$\bar{\boldsymbol{v}}_i \triangleq [\boldsymbol{y}_{g_i}]_\times \boldsymbol{p}_{I_i}(\boldsymbol{\theta}) + \dot{\boldsymbol{p}}_{I_i}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + {}^{IMU_i}R_{\mathcal{C}_\rangle}(\boldsymbol{\theta})\,{}^{\mathcal{P}_i}\hat{\boldsymbol{v}}_{O_i/\mathcal{W}} \tag{3.49}$$

We then apply the observer (3.40) using $\bar{\boldsymbol{v}}_i$ instead of $\boldsymbol{v}_i$ to obtain an estimate of the ith tilt - therefore completing Step 2 of the algorithm.

3.3.2.3 Summary

The algorithm we propose thus consists of applying the same observer (3.40) successively 5 times - each time using data from the previous observers to propagate the velocity computation to the next IMU. The first IMU, i.e. the contact foot, is the only one for which a measurement $\boldsymbol{v}$ is truly available. For the other, we use $\bar{\boldsymbol{v}}_i$ from (3.49) instead.

Technically speaking, $\bar{\boldsymbol{v}}_i$ is not a true measurement of the linear velocity, since we neglect the last deformation to propagate the velocity ${}^{\mathcal{P}_i}\hat{\boldsymbol{v}}_{O_i/\mathcal{W}}$, which is itself an estimate coming from the previous observers and not a true measurement. In particular, we do not have a proof to formally guarantee the convergence of the resulting observer.

Nevertheless, we find in experiments that this methodology gives an excellent approximation of the true linear velocity, yielding an observer not only stable, but also with very good performance. This is illustrated in Figure 3.7. There, the velocity of the swing foot IMU, estimated by (3.49), is compared with the velocity measured by motion capture (using finite differences on motion capture data). Because the swing foot is located at the tip of the kinematic chain, this is where the error will be maximal. Yet the velocity we estimate there is still very consistent with motion capture: the most visible difference is the high level of noise, coming from the combined noise of all 5 gyroscopes. Though this likely limits the value of the gains that can be used, in practice these high-frequency components are naturally filtered by integration, to result in practice in a smooth orientation estimate, as presented in Section 3.6.

Remark. *In practice, tuning the gains for the KinematicObserver is quite easy. In particular, we saw no need to account for the fact that each observer*

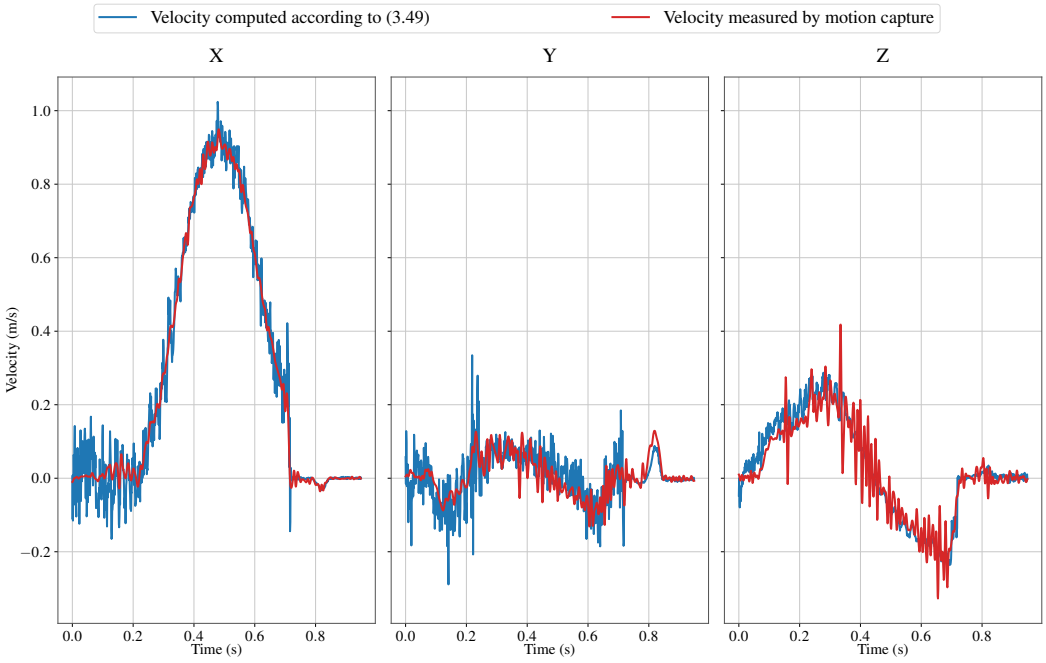


Figure 3.7: Velocity of the left foot, approximated by formula (3.49) and measured using motion capture.

depends on the previous one, which intuitively suggests that the convergence of IMU $i+1$ needs to be slower than that of the previous IMU. Instead, in simulation we observe that good results can be obtained by setting an identical gain value for all IMUs. In practice though, we notice a slight improvement by using a lower gain value on the support foot IMU: we believe this is linked to the saturation of the accelerometer that happens at impact, as discussed in Section 3.6.2.2. Thus, the curves in this manuscript are obtained using $\alpha = 0.75$ and $\beta = 0.057$ for the support foot IMU, and $\alpha = 1.5$ and $\beta = 0.229$ for the other IMUs.

3.4 Dynamics-based observer: acceleration as a sum of modeled forces

In the previous section, we considered the linear acceleration of the accelerometer as a kinematic consequence of a linear velocity $\boldsymbol{v}$. Another possibility to evaluate this term is to model it as the dynamic consequence of a sum of forces, according to Newton's law of motion. This leads to a formulation where the dynamic model of the exoskeleton, and of its flexibilities, is used: we thus call the corresponding observer the "*DynamicObserver*". Note

that a similar approach can be found in the literature for robotic manipulators with elastic joints, for instance in [Axelsson12, Staufer12] - but to our knowledge, this is the first time such approach is formulated on an extended flexible model, on a biped robot.

To formulate this observer, we consider the following assumption

Assumption 2. *The robot is standing in pure single support, with its swing foot not touching the ground. Additionally, the contact foot is considered to be flat on the ground, i.e.* ${}^{\mathcal{L}}R_{\mathcal{C}} = I_3$.

Note that this assumption is much more restrictive that the framework used for the previous two observers, which, by comparison, allows for double support phases, and makes no assumption about the angle of the swing foot (or the angle of the ground plane itself).

Due to the flat foot hypothesis, the robot now becomes an open kinematic chain parametrized by the joint angles $\boldsymbol{\theta}$ and the four deformations D_i. We denote $q_f \triangleq (D_1, D_2, D_3, D_4)$ the coordinates corresponding the flexible joints, $\boldsymbol{v}_f \triangleq (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3, \boldsymbol{\omega}_4)$ the associated velocities, to use as generalized coordinates: $\boldsymbol{q}_e \triangleq (\boldsymbol{\theta}, \boldsymbol{q}_f)$ and $\boldsymbol{v}_e \triangleq (\dot{\boldsymbol{\theta}}, \boldsymbol{v}_f)$.

Our goal is to estimate the vector of flexible state $\boldsymbol{x} \triangleq (\boldsymbol{q}_f, \boldsymbol{v}_f)$. For this purpose, we formulate these two assumptions

Assumption 3. *The vector* $\boldsymbol{u} \triangleq (\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})$ *representing the rigid joints position, velocity and acceleration, is known.*

Assumption 4. *The deformations are modeled as decoupled linear springs, similarly to the model presented in* (2.3)*, without damping. The torque transmitted by the ith flexibility onto the previous body, expressed in frame* $\mathcal{C}_i$*, thus simply writes*

$$\boldsymbol{\tau}_i \triangleq -K_j \, \boldsymbol{\Omega}_j \tag{3.50}$$

where K_j *and* $\boldsymbol{\Omega}_j$ *are, respectively, the stiffness matrix and vector of rotation associated to the deformation at joint* j*. The value of the joint stiffness* K_j *is known (through for instance system identification).*

An important feature of (3.50) is the fact that the torque depends only on $\boldsymbol{x}$. This enables us to write the derivative of $\boldsymbol{x}$ as a function of $\boldsymbol{x}$ and $\boldsymbol{u}$ only.

Indeed, thanks to Assumption 2, the system is simply an open kinematic chain with no external forces applied at the tip of the chain: thus, the equation of inverse dynamics can be generally written as

$$M(\boldsymbol{q}_e)\dot{\boldsymbol{v}}_e + C(\boldsymbol{q}_e, \boldsymbol{v}_e) = \boldsymbol{\tau} \tag{3.51}$$

with M the generalized inertia matrix, C the generalized bias force (inertial forces and gravity), and $\boldsymbol{\tau}$ the generalized forces acting at the joints.

Separating the rigid and flexible part, we obtain:

$$\begin{pmatrix} M_\theta(\boldsymbol{q}_e) & M_{\theta,f}^T(\boldsymbol{q}_e) \\ M_{\theta,f}(\boldsymbol{q}_e) & M_f(\boldsymbol{q}_e) \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{v}}_f \end{pmatrix} + \begin{pmatrix} C_\theta(\boldsymbol{q}_e, \boldsymbol{v}_e) \\ C_f(\boldsymbol{q}_e, \boldsymbol{v}_e) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau}_\theta \\ \boldsymbol{\tau}_f(\boldsymbol{x}) \end{pmatrix} \tag{3.52}$$

where $\boldsymbol{\tau}_\theta$ is the torque applied about the rigid joints, and $\boldsymbol{\tau}_f(\boldsymbol{x})$ is the torque at the flexible joints, given by (2.3). By solving only for the flexible part, we obtain the following dynamics:

$$\dot{\boldsymbol{v}}_f = M_f^{-1}(\boldsymbol{q}_e) \left[\boldsymbol{\tau}_f(\boldsymbol{x}) - C_f(\boldsymbol{q}_e, \boldsymbol{v}_e) - M_{\theta,f}(\boldsymbol{q}_e) \ddot{\boldsymbol{\theta}} \right] \tag{3.53}$$

Since $\boldsymbol{q}_e$, $\boldsymbol{v}_e$ and $\ddot{\boldsymbol{\theta}}$ are simply the elements of the vectors $\boldsymbol{x}$ and $\boldsymbol{u}$, (3.53) gives the dynamics of $\boldsymbol{v}_f$ as a function of $\boldsymbol{x}$ and $\boldsymbol{u}$. Coupled with the relationship between the generalized position $\boldsymbol{q}_f$ and the generalized velocity $\boldsymbol{v}_f$ (i.e. Equation (3) giving the derivative of a rotation matrix as a function of its angular velocity), the dynamics of the state $\boldsymbol{x}$ adopts the canonical form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{3.54}$$

The IMU measurements can in turn be expressed in terms of $\boldsymbol{x}$ and $\boldsymbol{u}$. Indeed, using the forward kinematics equation, the angular velocity and linear acceleration of the IMU can be reconstructed in term of $\boldsymbol{q}_e$, $\boldsymbol{v}_e$ and $\dot{\boldsymbol{v}}_e$ - or equivalently, $\boldsymbol{u}$, $\boldsymbol{x}$ and $\dot{\boldsymbol{v}}_f$. Thus, calling $\boldsymbol{y}$ the vector of all IMU measurements (both accelerometer and gyroscope), we have

$$\boldsymbol{y} = g(\boldsymbol{x}, \dot{\boldsymbol{v}}_f, \boldsymbol{u}) \tag{3.55}$$

In order to remove the dependency in $\dot{\boldsymbol{v}}_f$, we simply use the system's dynamic equation (3.53). This writes

$$\boldsymbol{y} = g(\boldsymbol{x}, M_f^{-1}(\boldsymbol{q}_e) \left[\boldsymbol{\tau}_f(\boldsymbol{x}) - C_f(\boldsymbol{q}_e, \boldsymbol{v}_e) - M_{\theta,f}(\boldsymbol{q}_e) \ddot{\boldsymbol{\theta}} \right], \boldsymbol{u}) \triangleq h(\boldsymbol{x}, \boldsymbol{u}) \tag{3.56}$$

Thus, having modeled the flexibilities according to (3.50) enables us, though the use of classical rigid body dynamics, to write the problem of estimating the deformation as a canonical non-linear observation problem:

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{u}) \end{cases} \tag{3.57}$$

The remarkable property of this formulation is stated as a theorem below:

Theorem 2. *Under Assumption 2 - 4, the state $\boldsymbol{x}$ is observable. More specifically, it is E-uniformly observable as defined in [Besançon07, Definition 2.4.1]: for any $T > 0$ and any input $\boldsymbol{u} : [0, T] \mapsto E$, $\boldsymbol{u}$ distinguishes every pair of initial state.*

The proof of this result is given in Appendix C: it relies on applying the recursive Newton-Euler algorithm for each set S_i between each flexibility.

Notice here an interesting feature of this observer: unlike the previous two observers, which only reconstructed the tilt of the IMU, here the observable state $\boldsymbol{x}$ includes the full 3D rotation of the IMU. The yaw deformation *can thus be reconstructed*: this is made possible by the modeling of the deformation as a spring. This observer is the only one to rely on such dynamical model. While this is conceptually a strength, this practically proves to be a great weakness, as shown by the experimental results of Section 3.6: the main culprit being the vast level of uncertainty associated to the flexibility model (3.50) (the value of the stiffness parameter K, but also uncertainty in the linear nature and localization of the deformation, as discussed in Section 3.6.1).

To optimize the estimation results, we use an Extended Kalman Filter to provide an estimate based on (3.57). Although the observability property does not guarantee the convergence of the EKF, the EKF is a natural first choice for the implementation of an asymptotic observer, as illustrated in several contributions in the literature [Khandelwal13, Benallegue15]. Though pure elasticity is instrumental in the observability proof that we propose in Appendix C, in practice we implement the dynamics of the flexibilities with a damping term, i.e. use (2.3) instead of (3.50) for added representativeness. The values of K_j and ν_j are identified experimentally, as explained in Section 4.1.2.2.

3.5 Handling sensor imperfections

In the formulation of our observers, we considered idealized IMUs, i.e. following the model of (3.10). In practice however, we use low-cost MEMS sensors, which are known to introduce a variety of errors (see for instance [Titterton04, Woodman07, Niu13]). While proper IMU calibration is out of the scope of this thesis, we here show how we account for the two main sources of error identified.

3.5.1 Handling the gyroscope bias

As mentioned in Section 3.2, a MEMS gyroscope is affected by an additive bias $\boldsymbol{b}_g$, which can be quite significant: values of up to $2°/s$ are observed on the IMUs of Atalante. This bias is handled only by the *StaticObserver* where it is estimated - but the *KinematicObserver* and *DynamicObserver* ignore it completely. In practice, this is a weakness which negatively affects the performance of both observers, by creating in static a constant bias on the estimated tilt (or rotation matrix): while this is less damaging than a linear drift that would result from open-loop integration, this is still something to take into account.

In fact, the value of this bias can be obtained simply by leaving the exoskeleton motionless, and computing the average of the gyroscope signal, over a period of time long enough to smooth out the effect of white noise (in practice, a few seconds is sufficient for this). In the literature, this bias is often modeled as being dependent on the temperature, and slowly time-varying [Niu13]. On the IMUs of Atalante however, both of these phenomena are very small in magnitude, as confirmed by some long-term experiments. Indeed, computation of biases done several months apart typically differ by no more than $0.3°/s$. Meanwhile, the temperature dependency is around $0.01°/s/°C$, and thus remains marginal in typical operating conditions.

Consequently, it is safe to assume that the bias remains constant over the duration of the robot's operation. In practice, we compute it manually at the start of an experimental session, on a static experiment, and apply the same bias value for all subsequent experiments. This process could easily be automated, for instance, by averaging the value of the gyroscope at the robot's startup, before any motion.

3.5.2 Correcting IMU alignment by a constant rotation matrix

In practice, when applying these observers, we observe a static offset between the tilt given by motion capture, $\boldsymbol{t}_m$, and the one estimated from the IMU signal - even in static, when this tilt is directly measured by the accelerometers.

We model this as a consequence of sensor misalignment with respect to the body it is attached to. Due to the tolerances in the mounting of the IMU on the robot (coming from robot assembly, soldering error, but also errors in the assembly of the IMU itself), the frame attached to the real sensor, S, is slightly different from the theoretical frame IMU[11]: this means that the

[11] We assume that the gyroscope and the accelerometer share the same frame S, though

measurement of the real IMU, $\boldsymbol{y}_s$, differs from the expected measurements $\boldsymbol{y}_{IMU}$, that should be obtained by a theoretical IMU, by a (constant) rotation matrix ${}^{IMU}R_S$ according to:

$$\boldsymbol{y}_{IMU} = {}^{IMU}R_S \; \boldsymbol{y}_s \tag{3.58}$$

This rotation is identified offline using motion capture data: in static the accelerometer directly measures the tilt, in the sensor frame: $\boldsymbol{t}_s = \frac{{}^S\boldsymbol{y}_a}{||{}^S\boldsymbol{y}_a||}$. Thus, we place the robot in double support in a reference position, then compute ${}^{IMU}R_S$ to realign this vector with motion capture, using a simple least-square approach:

$${}^{IMU}R_S = \underset{R \in SO(3)}{\arg\min} \sum_t ||R \; \boldsymbol{t}_s(t) - \boldsymbol{t}_m(t)||^2 \tag{3.59}$$

where $||.||$ is the Euclidean norm.

This results in small bias rotations[12], typically around 1°, that are then applied to the raw sensor signal according to (3.58).

Note that the presence of a rotational mounting offset is not the only reason of why the accelerometer does not give the same direction as motion capture in static. In practice, MEMS accelerometers suffer from a static bias, which can be modeled as a constant, yielding a more accurate model of the sensor

$$\boldsymbol{y}_{IMU} = {}^{IMU}R_S({}^S\boldsymbol{a}_{S/\mathcal{W}} + g \; \boldsymbol{t}_S + \boldsymbol{b}_a) \tag{3.60}$$

with $\boldsymbol{b}_a$ the additive bias. In static, this biases the observed direction: with bias value as high as 0.2 m/s^2, this bias can contribute to errors of 1° as well.

In practice, the impact of the additive bias is partially mitigated by our computation of ${}^{IMU}R_S$, which in the reference double support pose realigns the accelerometer measurement with gravity, thereby providing the correct direction despite the bias. This only works in the neighborhood of the reference position however: when the sensor rotates, the bias $\boldsymbol{b}_a$ rotates with it, whereas gravity does not. This means that we will observe a distortion of the tilt's estimate as the IMU rotates, with the correct tilt value being predicted only about the reference position.[13] However, in practice, this effect seems

in practice a (small) misalignment may exist there as well.

[12]We numerically solve this problem by gradient descent, parametrizing R by its rotation vector, $R = \exp[\boldsymbol{v}]_\times$. Note that the fact that R is a small rotation (i.e. $||\boldsymbol{v}|| << 1$) implies that the exponential can be linearized to obtain a true least square problem. Note that, in the more general context of large rotations, this problem corresponds to the well-known Wahba problem [Wahba65] for which analytical solutions have been proposed, e.g. [Shuster81].

[13]Interestingly, this effect can be used to estimate both ${}^{IMU}R_S$ and $\boldsymbol{b}_a$ if rotations with sufficiently large angles can be performed for the sake of calibration.

negligible, considering that, while walking, IMU rotation remains small (no more than 20°). In a more general context where large rotations need to be considered, the separate calibration of this additive bias should be considered as well, as presented for instance in [Frosio09, Tedaldi14].

3.6 Experimental results and discussion

In this section, we compare experimental results obtained with the three proposed approaches (*StaticObserver*, *KinematicObserver* and *DynamicObserver*) against motion capture. As a baseline, we use the rigid model of Atalante, with the flat-foot assumption: this gives an estimation of the pose of the robot that depends only on encoder measurements: we refer to this estimate as the *RigidModel*.

We illustrate the respective performances by performing two sets of experiments: a static experiment around a reference position in Section 3.6.1, and a more dynamic experiment featuring the exoskeleton walking in Section 3.6.2.

3.6.1 Static experiment

A simple static experiment is enough to tell us a lot about the properties of the presented observers. In this experiment, the exoskeleton is empty, i.e. without any user inside, and standing still in single support. The exoskeleton is then manually pushed generating disturbances causing excitation of the flexibilities. More precisely, in the next subsection, the exoskeleton is pushed in the back, which mostly excites a single flexibility (the support ankle). Then, in Section 3.6.1.2, the swing foot is pushed, which triggers all the flexibilities. Natural damping of the system yields a slow decay of the oscillations. A video of this experiment, and the associated 3D reconstruction, is available at https://youtu.be/c2Vdx81iu1A?t=60.

3.6.1.1 Pushing the exoskeleton's back

Figure 3.8 shows the estimated orientation of the pelvis IMU, when a disturbance is applied to the exoskeleton's back, and the corresponding error[14]. The pelvis IMU is indeed the most relevant output for this experiment: not only is it the closest to the localization of the disturbance, but it also acts

[14]This error is obtained by performing a roll-pitch-yaw decomposition of the rotation error $R_e \triangleq \hat{R}R_m^T$, (where R_m is the motion capture rotation and $\hat{R}$ the estimated rotation), and not a term by term difference of the roll-pitch-yaw coefficients.

as a proxy for the center of mass of the system, a key element for describing balance in static.

Without surprise, neglecting the deformations by using the *RigidModel* yields a significant static error, and does not capture well the oscillations which happen when the robot is pushed: these oscillations indeed mostly come from the flexibilities The moderate amount of oscillations seen on the *RigidModel* comes from the low-level tracking error induced by the perturbation, which is measured by the joint encoders.

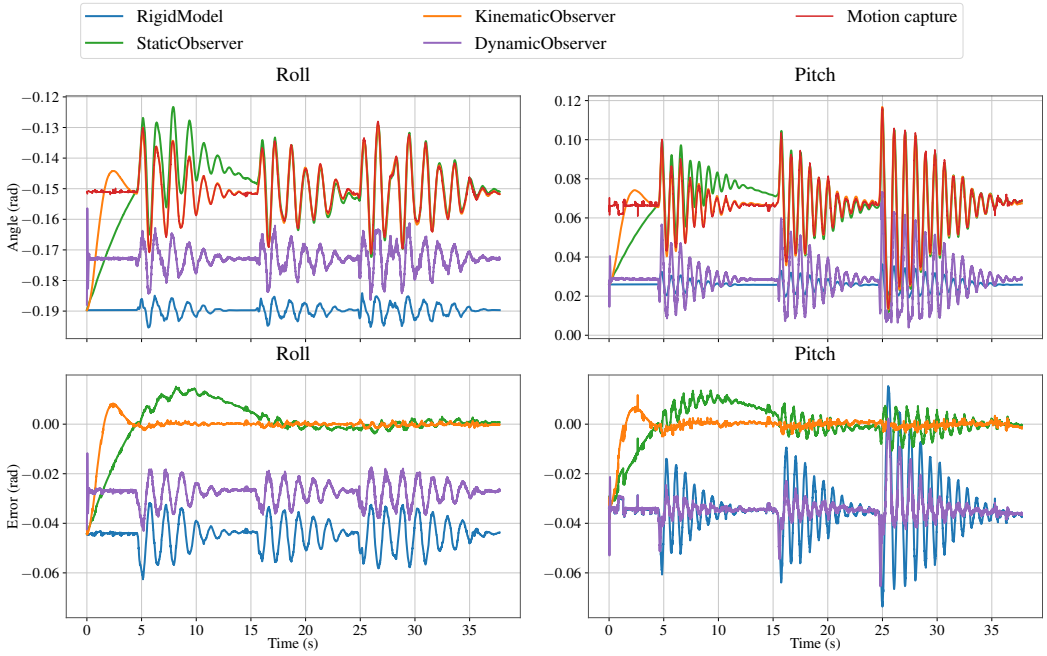


Figure 3.8: Pitch attitude estimation, on a static single support experiment, exoskeleton empty.

Out of the three proposed observers, the *DynamicObserver* behaves far worst than the other two. In particular, it suffers from a significant bias, mostly in pitch, where the estimated steady-state angle is almost the same as the rigid model. This comes from the fact that the use of spring dynamics to model the flexibilities dictates the equilibrium state: at rest, the deformation should be equal to the torque induced by gravity, divided by the stiffness. However, in this experiment, the position of the robot is such that, according to the CAD model of the robot, the center of mass of the robot is located at the vertical of the support ankle. Consequently, the (modeled) torque due to gravity about these points is close to zero, and our observer predicts no deformations. In practice however, the support ankle undergoes a static

deformation of 0.02 rad (which increases up to 0.05 rad when the exoskeleton is pushed), responsible for the robot's forward angle .

This static deformation underlines the approximative nature of our dynamical model. The most likely explanation of this shortcoming is that the deformations are not exactly collocated with the joints: considering that the center of the deformations is located a bit behind the ankle joint is mechanically sound, as this would correspond to a deformation of the rods of the differential transmission of the ankle. Changing the center of the deformation would result in a different equilibrium state for the dynamical model, which would bring us closer to the true attitude. Other causes of error include incorrect stiffness and viscous friction parameters, which would also explain the error observed in dynamic. Finally, the CAD model of the robot may also be inaccurate in terms of inertia and position of the center of mass of the various links. While, conceptually, the trade-off between model accuracy and sensor precision can be adjusted in a Kalman filter by modifying the associated covariance matrices, in practice we could not find a set of values that yielded results comparable to the other observers: adding more trust to the sensors would result in a strong amplification of the noise before significantly modifying the steady-state bias of this observer.

Generally speaking, we believe that a more thorough system identification of the mentioned deformation parameters could improve the results obtained by this observer. However, system identification, a reportedly tedious and difficult task, becomes intractable when a patient is present in the exoskeleton. On the contrary, this complex question of dynamic modeling is not necessary for both the *StaticObserver* and the *KinematicObserver*, which yield a significantly better performance. Both observers converge to a zero steady-state error, although the convergence phase of the *KinematicObserver* is much faster, lasting only 4 s, against 25 s for the *StaticObserver*. This slow convergence of the *StaticObserver* is related to the requirement of using low gains to reduce the error linked to a linear acceleration. Yet, despite this constrained tuning, the effect of the linear acceleration can still be seen during the oscillations, as shown in Figure 3.9 on a zoom on the third perturbation of this experiment. The error of the *StaticObserver*, an oscillation of 0.01 rad of amplitude, in fact comes from the presence of a 10° phase shift, as predicted by the linear analysis of Section 3.2.2. While reducing the gains of the observer would decrease this effect, this is done at the expense of an even slower convergence: structurally, it is not possible to achieve both fast convergence and good dynamical behavior with this observer. Conversely, the *KinematicObserver* does not suffer from this limitation, and thus easily outperforms the *StaticObserver*.

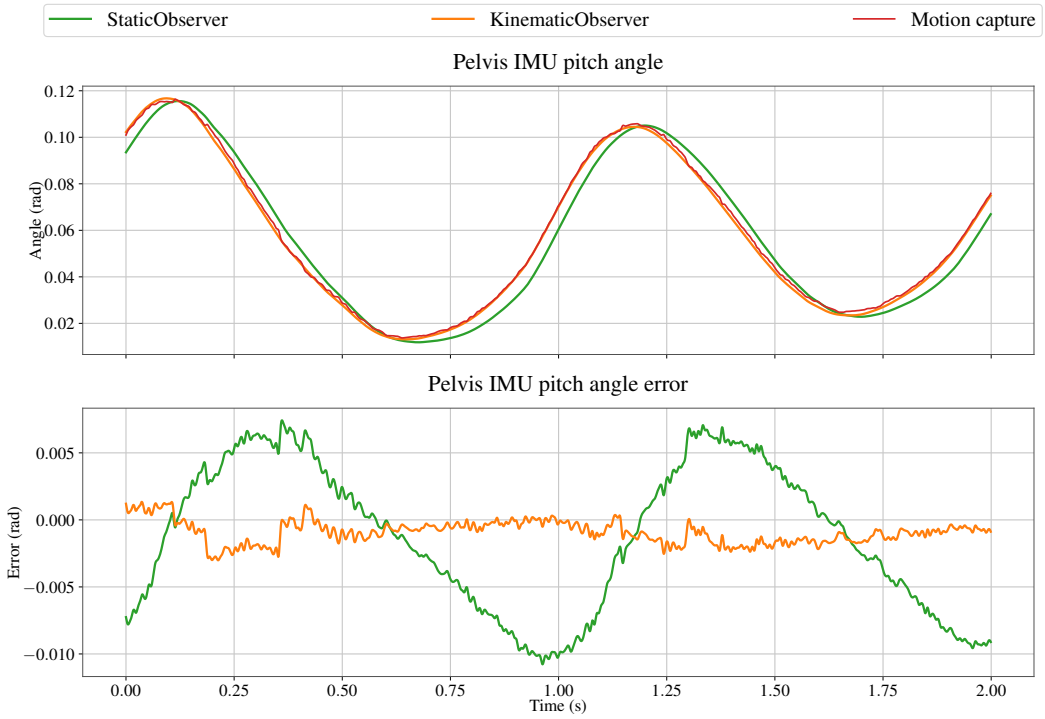


Figure 3.9: Pelvis pitch estimation during a perturbation: the *StaticObserver* suffers from a phase shift of 10°, as expected from the linear study of Section 3.2.2.

3.6.1.2 Pushing the exoskeleton's swing foot

Pushing the exoskeleton in the swing leg, in the back of the tibia, illustrates the presence of multiple deformations, and the importance of using more than one IMU. Figure 3.10 shows the deformation (in pitch angle, since the exoskeleton is pushed forward) of the support ankle, support hip and swing hip. The swing ankle receives no disturbance, and holds only very little weight (the weight of the foot alone) hence the deformation there remains negligible. However, the support ankle and both hips all show a comparable amount of deformations and oscillations as a response to the disturbance. In the end, the cumulative effect of these oscillations generates a significant motion of the swing foot, with an amplitude of 5 cm - something than cannot be captured by considering, for instance, the ankle deformation alone.

Notice that once again the *KinematicObserver* is very accurate at reconstructing the deformation observed by motion capture. Due to the linear acceleration, the *StaticObserver* is off during the oscillatory phase. Meanwhile, the *DynamicObserver* behaves more poorly, with both a large static

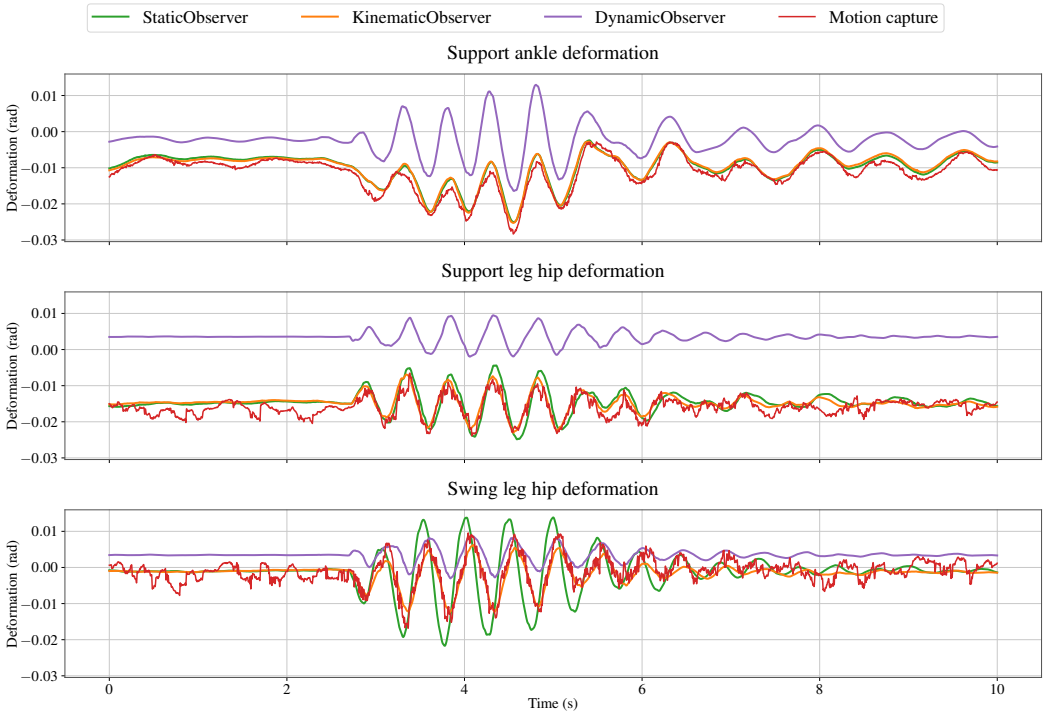


Figure 3.10: Deformation in pitch of the support ankle and both hips when pushing the tibia of the swing leg: this disturbance generates significant oscillations for all three considered flexibilities. Once again, the *KinematicObserver* best captures this motion.

bias and a poor handling of the oscillations. This is consistent with the observations made on the pelvis IMU alone in the previous subsection.

3.6.1.3 Conclusion

Testing the observers in a simple static, single support scenario, already shows a clear ranking between these observers. The *KinematicObserver* outperforms both other approaches. Indeed, the *StaticObserver* is negatively impacted by linear accelerations during the oscillatory phases, despite a low tuning of the gains which leads to a very long convergence phase. Meanwhile, the *DynamicObserver* proves to be quite sensitive to modeling errors, and uncertainties in the parameters representing the dynamical behavior of the deformations (i.e. the modeling of the torque in (2.3) or (3.50)).

These results lead us to abandon the dynamics-based approach. Improving its performance would require complex system identification, with a more advanced modeling of the deformations to try to account for the observed

bias. On the contrary, both the *StaticObserver* and the *KinematicObserver* are independent of the precise dynamics of the deformation (i.e. independent of the spring-damper model (2.3)). Rather, they are only impacted by kinematic parameters, i.e. the localization of the deformations. This makes both of these observers much easier to implement than the *DynamicObserver*. This dynamics-free property is all the more important to us considering that Atalante is an exoskeleton, meant to work with a user that acts as an unknown disturbance: this would yield patient-dependent results for the *DynamicObserver*. Finally, in the way it is designed, the *DynamicObserver* is not directly applicable to the case of dynamic walking: indeed, to correctly capture the dynamics of the system, we need to consider the presence of double support phases during which the contact forces are distributed on both feet, and the fact that the support foot can tilt on its edge during single support phases. An extension of this observer, using force sensor measurements and data from the support foot IMU in single support, is theoretically possible, but this was not performed in this thesis: considering the very good results of the *KinematicObserver*, we see no need to push this complex method any further.

3.6.2 State estimation while walking

3.6.2.1 Experimental results

We now consider the principal use case for Atalante: dynamic walking with a user. We here compare the two applicable observers, i.e. the *StaticObserver* and the *KinematicObserver*.

IMU attitude estimation Once again, the *KinematicObserver* outperforms the *StaticObserver*, which remains limited by the zero average acceleration hypothesis. This assumption is obviously not fully satisfied during a walking motion. Indeed, the forward motion of the swing foot typically generates accelerations of 4 m/s^2: thus, during the swing phase, the accelerometer direction significantly differs from gravity, resulting in an attitude error for the *StaticObserver*. This is illustrated in Figure 3.11.

While both observers give an equivalent attitude estimation during the stance phase, the error in pitch angle of the *StaticObserver* greatly increases during the walk, due to the phase shift caused by the forward acceleration of the foot. Notice that this time, the phase shift is negative, unlike the case of the pelvis IMU in Figure 3.9. This can be explained as follows. While the motion of the pelvis can be approximated by that of an inverted pendulum, the robot's foot swings about the hips following the motion of a regular

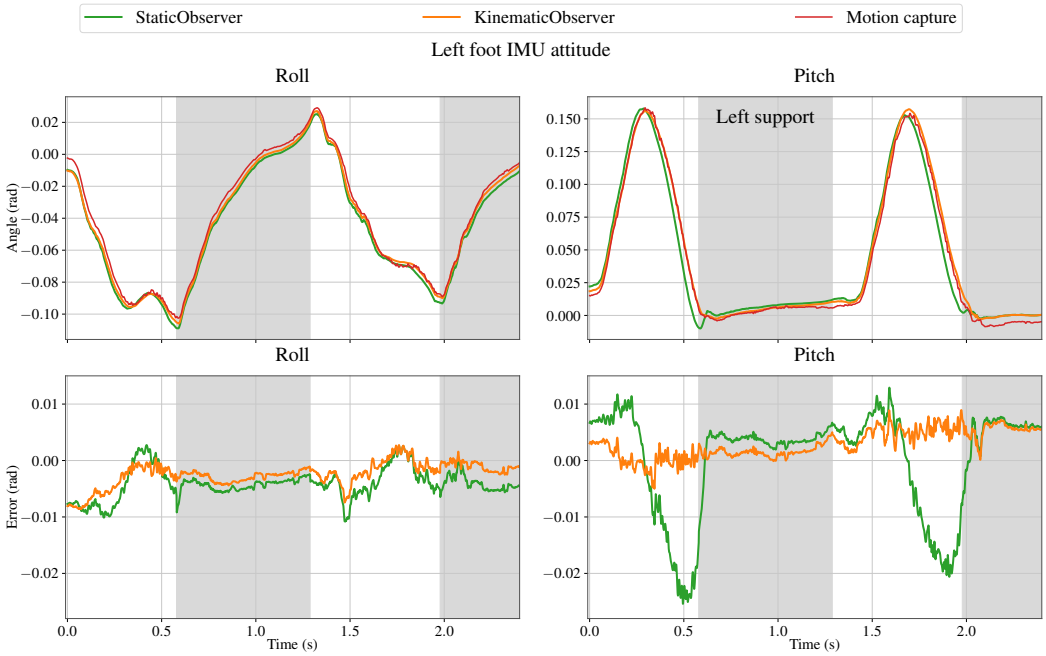


Figure 3.11: Estimation of the orientation of the left foot, and corresponding error: notice the phase shift in the *StaticObserver*, created by the linear acceleration during the swing phase.

pendulum instead. Applying the study of Section 3.2.2 to a regular pendulum instead indeed changes the sign of the obtained phase shift, compatible with the present observations[15]

The motion of the tibia IMU is very similar to that of the foot, and therefore shows the same error pattern. The pelvis however performs a different motion: unlike the leg which alternates between an stance and swing phase, the pelvis is continuously in motion. In particular, it oscillates from right to left as the weight is transferred from one foot to the next. The resulting acceleration again cause a significant phase shift for the *StaticObserver*, this time mostly visible on the roll angle, as shown in Figure 3.12.

Contrary to the *StaticObserver*, the *KinematicObserver* is far less affected by the presence of linear accelerations, as this term is kinematically taken into account in this observer. The observer thus keeps a far more constant estimate, with a root mean square error typically around 0.007 rad for all IMUs, when computed over a complete walk experiment.

[15]More precisely, linearizing (3.30) about $\theta = \pi$ leads to a modification of the transfer function from (3.34) into $\mathcal{H}(s) = 1 + \frac{l}{g}\frac{s^2 k_b + s^3 k_a}{s^2 + s k_a + k_b}$.

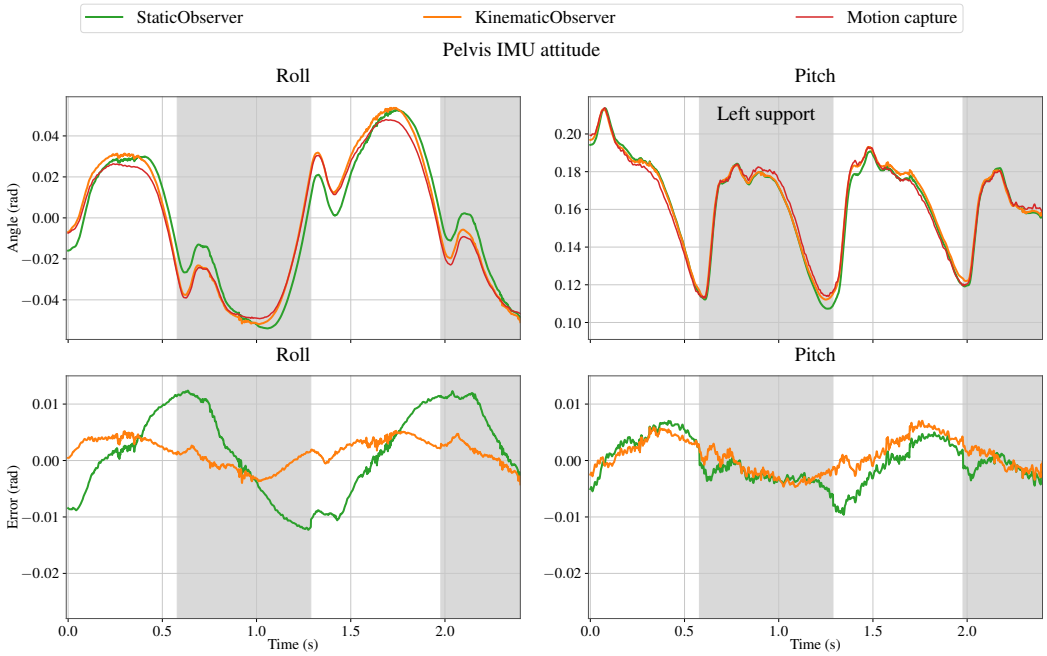


Figure 3.12: Estimation of the attitude of the pelvis IMU while walking: once again, the *KinematicObserver* outperforms the *StaticObserver*, which is continuously out of phase due to the exoskeleton's acceleration.

Link position estimation The IMU tilt estimates are in turn used to estimate the position of the robot's link in the local frame, using the extended kinematic model. Of particular interest during walking is the position of the swing foot toe: indeed, controlling when and where the swing foot impacts the ground is critical to realizing a stable walk. Figures 3.13 shows the corresponding position estimates during a full stride, while Table 3.1 gives the root-mean-square error of these quantities over a complete walk sequence.

Recall that we are only interested in local information (the deformations, and the angle of the contact foot), and not odometry: this comparison is done in the local frame, i.e. the frame following the support foot. Thus, the first half of the figure corresponds to the position (and orientation) of the right foot with respect to the local frame under the left foot, while the second half shows the position of the right foot with respect to the left foot - hence the discontinuity and symmetry for the Y axis in these curves.

The switch between left support and right support is done just as the swing foot impacts the ground: this is detected using the robot's force sensors. Thus, each step starts by a double support phase during which weight transfer is performed. When the step ends, the rigid, flat foot model, overestimates the height of the swing foot, which is in fact lower than expected

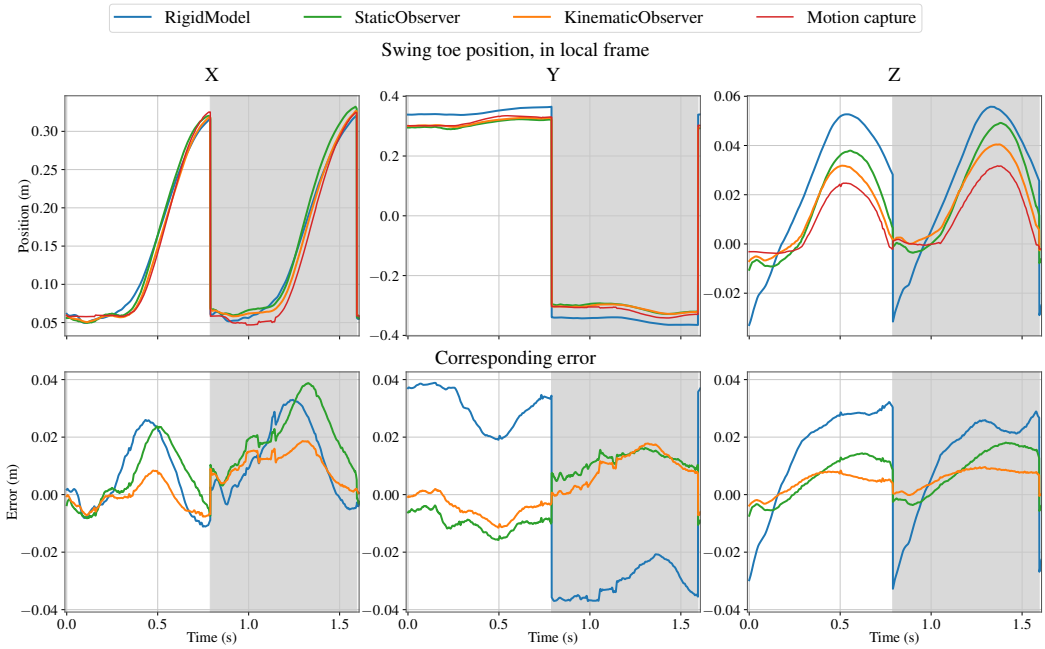


Figure 3.13: Estimation of the position of the swing foot's toes in the local frame while walking, and corresponding error.

due to the deformations and the tilt of the support foot. This mismatch in height between both feet remains constant through impact, hence a negative height estimate just after impact.

As can be expected from the IMU attitude estimation error, the *StaticObserver* is outperformed by the *KinematicObserver*: in particular, it yields a rather large error in the X direction: this is a consequence of the large error in the pitch estimate of the swing foot committed by this observer.

	X (mm)	Y (mm)	Z (mm)
RigidModel	15	32	21
StaticObserver	16	11	9
KinematicObserver	9	9	7

Table 3.1: Root mean square error of the estimation of the swing foot's toe position (ground truth being provided by motion capture).

By contrast, using the *KinematicObserver* provides a significantly more accurate estimation of the toe's position, with an error typically below 1 cm: as will appear later on in the next chapter, this level of precision is sufficient to perform IMU-based feedback and provide a meaningful impact on the

system's performance.

3.6.2.2 Investigating performance limitations using motion capture data

While the performance obtained by the *KinematicObserver* are already quite satisfactory, in this section we provide a deeper analysis to explain the sources of the residual error observed. For this purpose, we use motion capture data to simulate "perfect" IMU signals, in order to obtain a lower bound on the residual error.

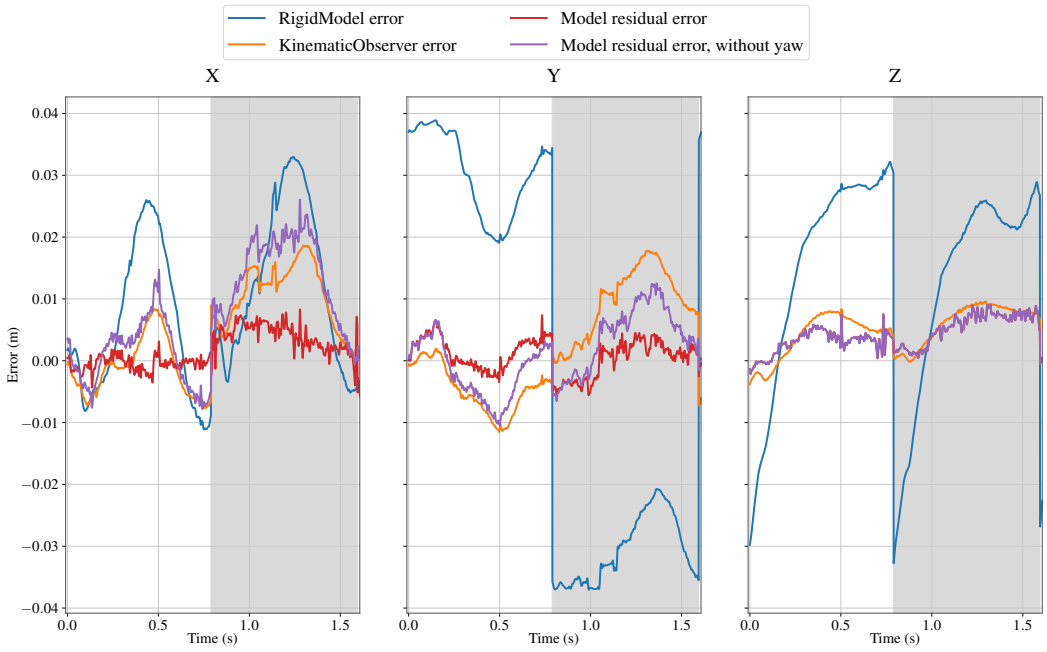


Figure 3.14: Error in swing toe reconstruction: the error committed by the *KinematicObserver* mainly comes from the deformation model and the fact that the yaw cannot be estimated.

Accuracy of the toe position estimate: the limitation induced by the kinematic model and non-observability of the yaw The residual error committed on the position of the toe can be mostly explained by the accuracy of our deformation model itself, and the fact that we neglect the yaw deformations. This is evaluated in Figure 3.14, where we use motion-capture to obtain a direct estimate of the IMU attitude. Namely, the red curve shows the error in the toe position when using the full orientations given by the motion capture in place of the IMU. This full estimate of the attitude however

cannot be obtained in practice, since the yaw deformation is not observable by IMUs. Thus, the yaw deformations are removed from these motion-capture rotations, as presented in Section 3.1.4. The resulting estimate is represented by the purple curve, which corresponds to the error that would be obtained if all IMUs gave a perfect tilt (roll and pitch) estimation. The IMU-based estimation given by the *KinematicObserver* is given in orange, while the blue curve is the *RigidModel* estimate (i.e. the deformation), for reference.

The difference between the red and purple curve indicate the presence of yaw deformations: these deformations, which amount for about 0.02 rad, cause an additional error of up to 2 cm in the foot X position, and a slight increase (less than 1 cm) along the Y axis. On the contrary, the vertical position is not affected.

The fact that the *KinematicObserver*'s estimate (in orange) remains quite close to the red curve shows that we are reaching the limit in precision that can be obtained by using tilt estimates from IMUs only. This is consistent with the fact that the error in IMU tilt given by this observer is quite small (RMSE of 0.007 rad). To further improve the accuracy, we would need additional sensors to estimate the yaw deformation (e.g. computer vision [Scandaroli11]) - or a more complex model featuring more deformations (and also more sensors) to further reduce the error committed by the model itself.

Behavior during a full walk: IMU signal quality as the limiting factor. While the curves presented in Section 3.6.2.1 show the dominant features of both observer (namely, a significant error for the *StaticObserver* in the presence of acceleration, which barely affects the *KinematicObserver*), a look at a complete walk experiments reveals that both observers exhibit slow variations, over the course of several steps, which decrease the quality of the estimation. This is shown in Figure 3.15.

While the error committed by the *KinematicObserver* remains quite small (it rarely goes above 0.01 rad, and the RMSE remains around 0.007 rad for all the considered IMUs), it exhibits a strange low-frequency behavior, similar to a random walk. The *StaticObserver* additionally shows a very slow oscillation: notably, the right tibia roll or the pelvis roll angle seem to be slowly drifting away from zero.

We attribute this error, in the case of the *KinematicObserver*, to artifacts in the IMU signals themselves. To reach this conclusion, we use motion capture to generate data representing fictitious IMU, in place of that of the real robot's IMUs. Namely, taking the time derivative of the position and orientation of the motion capture, we generate a gyroscope and an accelerometer signal for each IMU. We then use this data as input for both observers:

Figure 3.16 reports the resulting tilt estimation error.

While the *StaticObserver* uses only IMU data to perform its estimation of IMU orientation[16], the *KinematicObserver* relies on the kinematic model of the robot, the non-slipping foot hypothesis, and encoder and force sensor data from the robot. Finally, it is intrinsically built on an approximation to compute a linear velocity in (3.49), by neglecting the last deformation being estimated. All these elements are possible sources of error for this observer. As shown in Figure 3.16 however, they only contribute to a negligible error. This validates the overall soundness of the *KinematicObserver*, demonstrating that this observer can be expected to give an almost perfect attitude estimation of a walking Atalante.

By contrast, the *StaticObserver* exhibits the expected sharp error increase at each step for the leg IMUs (tibia and foot), and a significant oscillation at 0.7 Hz (the frequency of the walk) for the pelvis IMU roll. But on top of this, it also exhibits a slow oscillation, with a period of about 25 s: the error drifts away during the first 15 s of the walk, and converges back very slowly at the end of the excitation. This oscillation can be attributed to the fact that, while walking, the average linear acceleration is not equal to zero. This effect stems from the presence of Coriolis and centrifugal accelerations, which result in a non-zero average acceleration despite the fact that the walk is performed at a constant speed[17]. This results in a non-zero average bias term being added to the accelerometer direction, hence to the error term of the *StaticObserver* in (3.26). The observer then slowly converges to the direction of the bias acceleration - with the current gain value, the convergence lasts about 25 s, as seen in the static experiment, which is about the duration of the walk, hence this impression of a slow drift. As the walk ends and the excitation vanishes, the observer starts to converge back to the right angle, again taking a considerable time to do so. Unlike the *KinematicObserver*, the *StaticObserver* is thus intrinsically limited in terms of performance, and bound to exhibit larger error patterns.

Since the only difference between Figure 3.16 and Figure 3.15 is the use of true IMU data versus motion-capture derivatives, this indicates that the higher error observed on real experiment indeed stems from the quality of the sensor signal. Low-cost MEMS IMUs are know to suffer from a variety of

[16]Encoder measurements are only used when doing the reconstruction using the extended flexible model (2.5) to obtain the link position.

[17]This somewhat counter intuitive effect can be illustrated by a simple pendulum as shown in Section 3.2.2. Consider a pendulum oscillating as $\theta = \sin(t)$: this motion has an average velocity of zero. Yet, as shown by (3.29), the centrifugal acceleration $-l\dot{\theta}^2$ has a non-zero average $-\frac{l}{2}$.

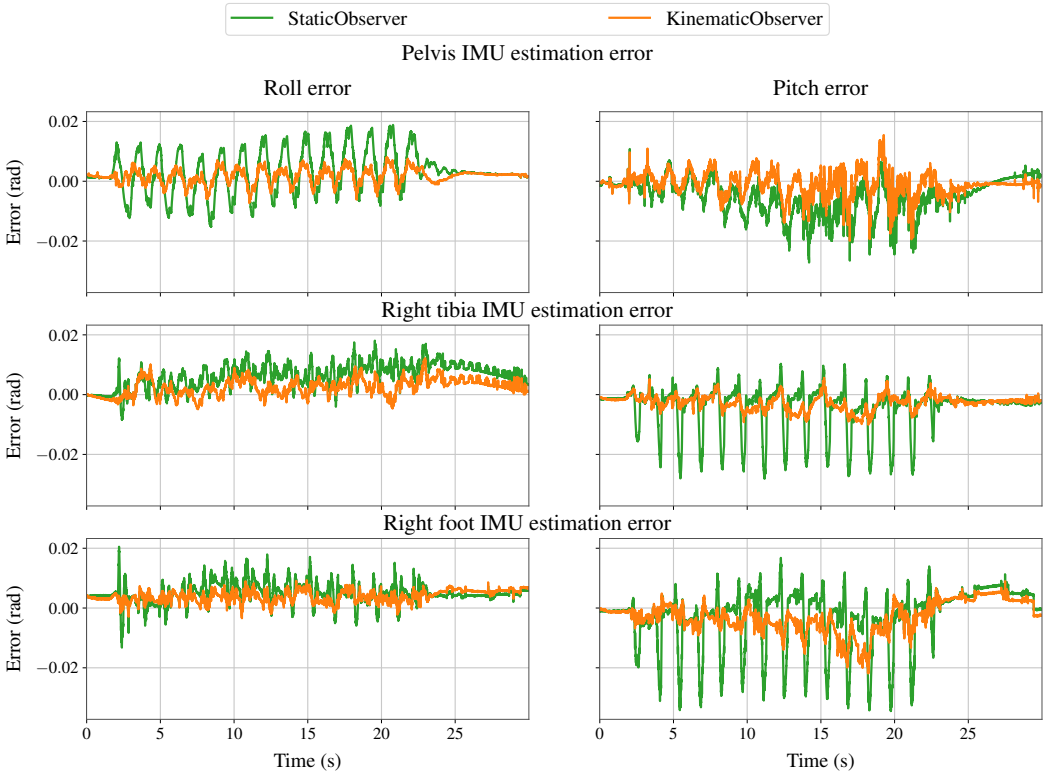


Figure 3.15: Attitude estimation error of the pelvis, right tibia and right foot IMU during a full walk: notice that the error is not constant throughout the experiment, but varies from one step to the next.

errors: axis misalignment resulting in a non-orthogonal measurement frame; scaling errors; temperature sensitivity; accelerometer additive bias, among others. These effects are not accounted for: as presented in Section 3.5, we only approximately cancel the gyroscope bias, and realign the accelerometer with the motion capture vertical axis.

An extra source of error in the IMU signal is the presence of impacts due to the foot striking the ground while walking. Indeed, impacts results in strong vibrations, which often cause the accelerometer to briefly saturate. Additionally, we experimentally observe the presence of high-frequency vibrations on the gyroscope at impact, sometimes with non-zero average creating an error of up to 0.01 rad in a few tenths of milliseconds. We believe this to be the result of signal aliasing: indeed while the accelerometer is equipped with an analog low-pass filter, this is not the case of the gyroscope. Finally, inconsistencies in the sampling of the multiple sensors are known to be present, as the internal clocks and timers are not synchronized. Each

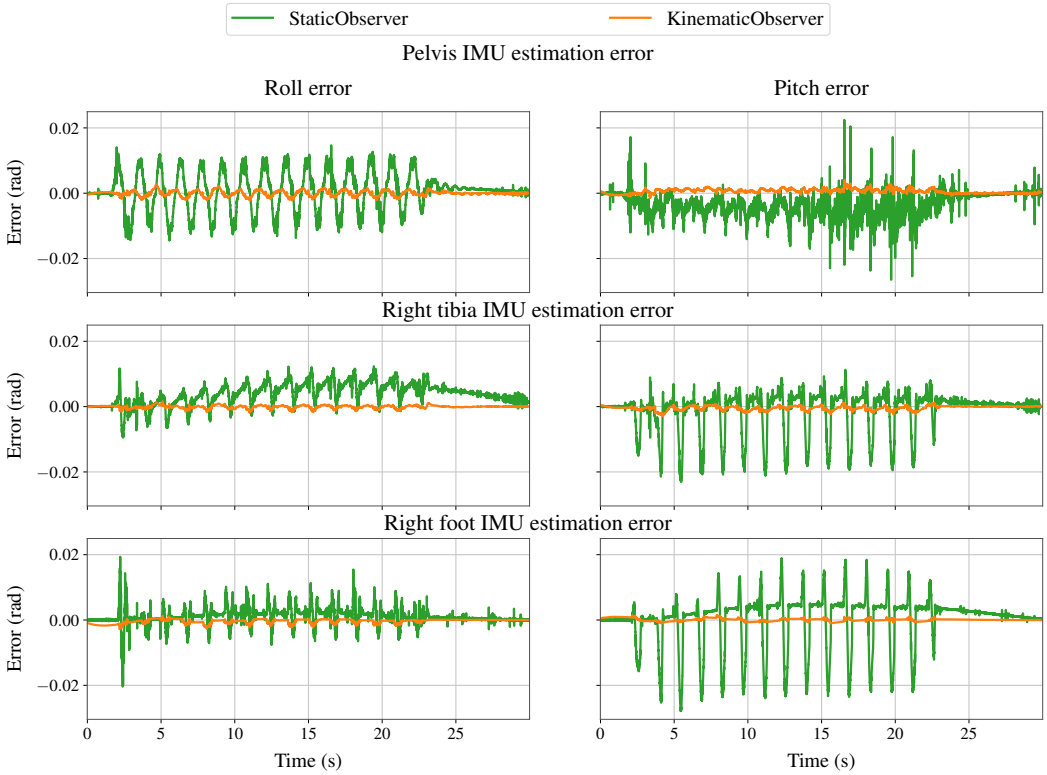


Figure 3.16: Attitude estimation error on the same experiment as Figure 3.15, but using motion capture derivatives instead of IMU signal: the error in the *KinematicObserver* vanishes, while the *StaticObserver* keeps both a per-step error, and converges very slowly towards a biased value.

IMU, encoder and force sensor thus reports its sample individually, creating a different time delay and jitter for each signal, which negatively affects the global reconstruction we perform, relying on the comparison of all of these signals.

Thus, we believe that to further improve the results, one needs to take a closer look at the sensor signals themselves, and in particular look at the various calibration methods documented in the literature: see for instance [Frosio09, Niu13, Tedaldi14]). The current results, obtained without these extra calibration steps, remain quite satisfying: most importantly, the quality of the estimation is sufficient to perform feedback control, and obtain meaningful results, as presented in Chapter 4. Conversely, we wish to stress the fact that the realignment of the IMU onto the motion capture, presented in Section 3.5.2, is not required for the methodology presented in this thesis to be applicable. Indeed, this step is only added as a final touch to improve

the results on the motion capture experiments used to validate this thesis, but the development was done on several exoskeletons before any motion capture was performed, without requiring any bias identification.

3.7 Conclusion

In this chapter, we have studied the problem of estimating the deformation of Atalante using IMUs. This is equivalent to estimating the orientation of the IMUs, as the deformations are simply characterized by the difference between two IMUs. To answer this problem, we have looked at three different formulations. The *StaticObserver* assumes that no acceleration is present to estimate the tilt of each IMU independently. The *KinematicObserver* consists in using the robot's kinematic model to estimate the linear velocity of each IMU sequentially, the estimation of each IMU serving as input for the next. Finally, the *DynamicObserver* uses the complete dynamical model of the robot, with the flexibilities modeled as spring dampers, to estimate all the deformations.

We have found the *KinematicObserver* to be the best approach, yielding an estimation of the IMU tilt with a root mean square error of 0.007 rad on all IMUs. Indeed, the erroneous nature of the dynamical model underlying the *DynamicObserver* yields poor performance, even in the simplest single support, exoskeleton empty scenario. While system identification would probably render this methodology viable for a humanoid robot, its dependency on the system's dynamics remains a point of concern for real-world use on an exoskeleton. Meanwhile, the zero-acceleration hypothesis underlying the *StaticObserver* creates significant artifacts: despite having tuned the observer with modest gain to provide a slow convergence, the attitude estimation remains sensitive to linear acceleration, causing a significant phase shift at each step, as well as an attitude bias that slowly appears during the walk.

By contrast, the *KinematicObserver* does not suffer from the dynamic nature of the motion, and gives a much more accurate tilt estimate. Experimentally, we explain the residual error as a consequence of sensor signal quality more than a limitation of the observer. In the end, this approach gives access to an estimate of the swing toe position with a precision of 1 cm. The position error mostly comes from the kinematic model of the flexibilities, and the fact that the yaw deformations are not observable when using only IMU sensors. Nevertheless, the quality of the estimation provided by the *KinematicObserver* proves to be sufficient to be used in feedback control: this is presented in the next chapter.

Chapter 4

Improving Atalante's low-level controllers to account for deformations

Chapitre 4 - Amélioration des contrôleurs bas-niveau d'Atalante par la prise en compte des déformations: *Ce chapitre aborde le contrôle en boucle fermée des déformations, en exploitant les données d'estimation produites par l'observateur cinématique présenté au chapitre précédent. L'approche choisie est celle d'un retour d'état grand gain décentralisé. Le réglage de gain est réalisé articulation par articulation, par la méthode LQR appliquée à un modèle linéaire d'actionneur série élastique. Le contrôleur résultant est tout d'abord testé expérimentalement sur plusieurs scénarios quasi-statiques en simple support, où il démontre une capacité de rejet de perturbation bien supérieure à celle du contrôleur PID classiquement utilisé sur Atalante.*

Ce contrôleur de compensation de flexibilités est ensuite adapté au cas de la marche dynamique, par la prise en compte de la nature hybride du mouvement (alternance phase de support / phase de vol), et l' ajustement des gains pour un meilleur suivi de trajectoire. Celui-ci est ensuite à nouveau comparé au contrôleur PID sur plusieurs expériences de marche, avec patient et avec différents mannequins. Ces expériences témoignent d'une amélioration significative des performances de suivi et de la robustesse de la marche.

The previous chapter has provided us with an observer to estimate the deformations on Atalante. Here, we seek to use this estimation to improve the position-tracking performance of the low-level controllers.

As explained in Section 2.3, we use for the controller a slightly different model of the deformations than for the observer: namely, we consider joint flexibilities instead of the extended kinematic model. The objective of the low-level controller can thus be formulated once again at joint level: given a reference trajectory $\boldsymbol{q}^*$ coming from the high-level controller, we wish for the output (flexible) angle $\boldsymbol{q}$, estimated by the IMUs, to track this reference as accurately as possible.

Atalante's current low-level controller is a classical high-gain PID, which assumes a rigid system and thus simply servoes the motor variable $\boldsymbol{\theta}$ to the target trajectory:

$$\boldsymbol{u} = -K_p(\boldsymbol{\theta} - \boldsymbol{q}^*) - K_d(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{q}}^*) - K_i \int (\boldsymbol{\theta} - \boldsymbol{q}^*) \tag{4.1}$$

where K_p, K_d, K_i are diagonal gain matrices (carefully tuned by trial and error).

We call (4.1) the *rigid controller*, which acts as a benchmark controller: indeed, even though this controller does not take into account the presence of deformations, it is sufficient to enable stable walking motions on Atalante.

To design our so-called *flexibility compensation controller*, we first study the behavior of Atalante in single support, about a reference static position, in Section 4.1. Our controller stems from (4.1), and simply consists in adding a feedback term on $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, to perform full state feedback, while keeping a decentralized, high-gain approach. We propose to perform gain tuning by applying LQR design to the linearized dynamics at every joint. This controller is successfully tested on Atalante, where it provides significantly improved disturbance rejection, quickly damping the oscillations which are the dominant behavior in this setup.

We then adapt this controller to a walking scenario: while the overall approach remains the same, this requires two main modifications:

- A discrete switching in the gains and integrator values when transitioning from right support to left support, to account for the hybrid nature of the system.
- A novel tuning of the gains, to account for the highly dynamic nature of the walk. This process is partially manual: the gains are slightly adjusted to mitigate the presence of high-frequency vibrations which otherwise appear when applying a pure LQR design, something we explain as the consequence of additional, modeled deformations in the system.

The performance of the resulting controller is successfully demonstrated, both on valid users and on experiments with a dummy.

4.1 Controlling the flexibilities in static: enhancing disturbance rejection

In this section, we consider the case of Atalante standing in pure single support, and we assume that its support foot remains flat on the ground. This effectively removes the freeflyer coordinates from the state of the system: in fact, the robot becomes similar to a fixed-based robotic manipulator. Furthermore, we model the deformations as joint flexibilities, as explained in Section 2.3.

4.1.1 Literature review of feedback controllers

The control of a robotic manipulator with elastic joints is an old question which has received significant attention since as early as the 1980s. We refer the reader to [Spong90, Benosman04, Moberg10] for a more complete review of the problem and different techniques used. Here, we simply give a brief overview of some of the most common approaches.

The dynamics of a manipulator with joint flexibilities, assuming that each rotor is symmetric about its rotation axis, simply writes [Spong87]

$$\begin{pmatrix} M(\boldsymbol{q}) & 0 \\ 0 & I_m \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{\theta}} \end{pmatrix} + \begin{pmatrix} C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \\ 0 \end{pmatrix} + \begin{pmatrix} -\boldsymbol{\tau}_f(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) \\ \boldsymbol{\tau}_f(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{u} \end{pmatrix} \tag{4.2}$$

where $M(\boldsymbol{q})$ is the link inertia matrix, I_m the diagonal matrix of rotor inertia, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ contains Coriolis and centrifugal torques, and $\boldsymbol{g}(\boldsymbol{q})$ is the torque due to gravity. Finally, $\boldsymbol{\tau}_f(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}})$ is the torque generated by the joint flexibility, which we assume to follow a spring-damper law according to (2.1).

As mentioned in the introduction of this thesis, several methods have been proposed to compensate part of the effect of the flexibilities in the absence of a direct measurement, by using the linear spring model and the feedforward torque to apply an offset to the error term of existing PD controllers. This was first proposed in the so-called gravity compensation method of [Tomei91]. This method was refined in [Zollo05], while different variations are proposed on humanoid robots in [Johnson15, Reher16, Kim16]. A more advanced approach is proposed in [Kim18], where a disturbance observer is used to provide compliance and thus reduce vibrations.

When a measurement of the deformation is available, as in the case of our IMU-based approach, we naturally turn toward more direct feedback-based approaches. In the absence of a viscous damping term (i.e. when the flexibility is modeled as a pure elasticity), the system (4.2) is statically

feedback-linearizable [Spong87, De Luca95], leading to the following feedback controller [De Luca08, eq. (13.45)]

$$\begin{aligned}\boldsymbol{u} =& I_m K^{-1}\Big(M(\boldsymbol{q})\boldsymbol{v} + \dddot{M}(\boldsymbol{q})\ddot{q} + 2\dot{M}(\boldsymbol{q})\boldsymbol{q}^{(3)} + \frac{d^2}{dt^2}\left(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\right)\Big) \\ &+ \left(M(\boldsymbol{q}) + I_m\right)\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\end{aligned} \tag{4.3}$$

where $\boldsymbol{v}$ is the new control input. However, feedback linearization controllers are known to be quite sensitive to modeling errors [Moberg08]: for instance, computing the control law (4.3) requires the second derivative of the terms of (4.2). This computation is not suitable in the case of an exoskeleton, where motion of the user introduces large unknown variations in the dynamics.

Even in the context of robotic manipulator, the use of high-gain decoupled PID controllers is often preferred over feedback linearization [Chung08, Finet17]. Indeed, this controller is quite simple to implement and tune (as each joint can be tuned individually), and is known to be quite robust. With this consideration in mind, the natural extension of the PID controller is to perform linear full-state feedback on both $\boldsymbol{\theta}$ and $\boldsymbol{q}$, as proposed in [Albu-Schaffer00] or in [Moberg10, Section (5.3.5)]. This can be generically written as:

$$\begin{aligned}\boldsymbol{u} =& -K_\theta(\boldsymbol{\theta} - \boldsymbol{\theta}^*) - K_{d\theta}(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}^*) \\ &- K_q(\boldsymbol{q} - \boldsymbol{q}^*) - K_{dq}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^*) - K_i\int(\boldsymbol{q} - \boldsymbol{q}^*) \\ &+ \boldsymbol{u}_{ff}\end{aligned} \tag{4.4}$$

where $K_\theta,\ K_{d\theta},\ K_q,\ K_{dq},\ K_i$ are positive diagonal gain matrices, and $\boldsymbol{u}_{ff}$ a feedforward torque.

4.1.2 Proposed controller design

The controller we propose for Atalante takes the form of the linear controller (4.4).

In this equation, the feedforward term is computed using the full dynamics of the system (4.2). Neglecting the torque due to the rotor acceleration $I_m\ddot{\boldsymbol{\theta}}$ as compared to the gravity torque due to the much heavier links, the feedforward term writes

$$\boldsymbol{u}_{ff} \triangleq M(\boldsymbol{q}^*)\ddot{\boldsymbol{q}}^* + C(\boldsymbol{q}^*,\dot{\boldsymbol{q}}^*)\dot{\boldsymbol{q}}^* + \boldsymbol{g}(\boldsymbol{q}^*) \tag{4.5}$$

This enables a simple computation requiring only link-side information, without needing an explicit value of the parameters of the flexibilities The

presence of a patient is another source of uncertainty - however, this term is only a feedforward torque: ultimately, the error is supposedly mitigated by the presence of an integral term on $\boldsymbol{q}$.

One significant challenge in implementing the controller 4.4 however lies in the fact that this controller requires both a link trajectory $\boldsymbol{q}^*$ and a motor trajectory $\boldsymbol{\theta}^*$ - or equivalently, a trajectory for the flexibility $\boldsymbol{\alpha}^*$.

Mathematically speaking, this trajectory can in principle be computed by integration of the inverse dynamics equation (4.2) along the trajectory $\boldsymbol{q}^*$. Yet this requires precise knowledge of the dynamics, which is not realistic considering the presence of an uninstrumented patient. Practically on Atalante, the uncertainty on the localization of the center of mass of the patient, but also on the precise localization of the deformations as outlined in Section 3.6.1, means that this computation, when done on the nominal model (1.4), can result in values of $\boldsymbol{\alpha}^*$ having the wrong sign, as the patient leans forward while our dynamic model predicts a backward deformation.

Instead, we use a simpler, coarse approximation: we set $\boldsymbol{\alpha}^* = 0$, i.e. $\boldsymbol{\theta}^* = \boldsymbol{q}^*$. This way, we let the integrator capture the missing feedforward term, i.e. the average value of the deformation. In the end, our controller writes

$$\begin{aligned} \boldsymbol{u} = &- K_\theta(\boldsymbol{\theta} - \boldsymbol{q}^*) - K_{d\theta}(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{q}}^*) \\ &- K_q(\boldsymbol{q} - \boldsymbol{q}^*) - K_{dq}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^*) - K_i \int (\boldsymbol{q} - \boldsymbol{q}^*) \\ &+ \boldsymbol{u}_{ff} \end{aligned} \tag{4.6}$$

Notice that by doing this, we have removed any explicit dependency on the dynamic model of the system in the feedback terms, to rely only on constant feedback gains to tune. This is done with the objective to keep a controller as robust as possible to changes in the parameters, namely the presence of different users: the dynamical model is only present in the feedforward term.

4.1.2.1 Controller gain tuning methodology

To compute the gain matrices of (4.6), we use a simplified model for each joint. We consider the robot standing still in a reference single support posture, and we model the dynamics of joint i to compute the corresponding gain. Given that the decentralized controller structure of (4.6) does not include any coupling between joints, we simplify the dynamics by considering that all other joints are fixed. Thus, the model reduces to a simple planar series-elastic actuator: motor i is fixed, and is linked to a single body $\mathcal{B}$,

aggregating all links downstream of joint i, through a spring damper. This single joint model is represented in Figure 4.1.

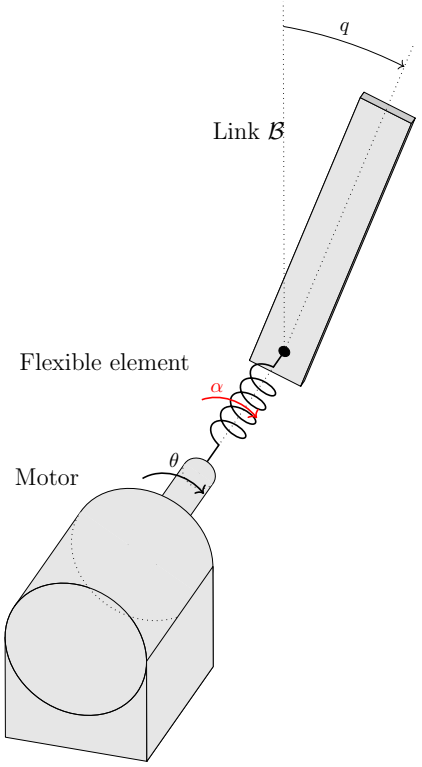


Figure 4.1: Model of a single joint of the robot: a flexible element links a single body to the rotor shaft.

Let α be the flexibility angle (i.e. the ith component of $\boldsymbol{\alpha}$), and q the angle between the vertical axis and the center of mass of $\mathcal{B}$, that is, the link angle $\boldsymbol{q}_i$ plus a constant angle. The dynamics of the system writes

$$\begin{cases} J\ddot{\alpha} = -u + \nu_m(\dot{q} - \dot{\alpha}) - k(1 + \frac{J}{I})\alpha - \nu(1 + \frac{J}{I})\dot{\alpha} + \frac{J}{I}mlg\sin(q) \\ I\ddot{q} = -k\alpha - \nu\dot{\alpha} + mlg\sin(q) \end{cases} \tag{4.7}$$

where J is the rotor inertia, m and I the mass and inertia of body $\mathcal{B}$, k the spring stiffness, and ν, ν_m viscous friction coefficients for respectively the flexibility and the motor joint.

We choose a reference single support pose $\boldsymbol{q}$ for the robot. Let q_0 be the corresponding reference position for the joint angle in this simplified model (q_0 is in general different from 0). The associated equilibrium state of (4.7)

is

$$\begin{cases} \alpha_0 = \dfrac{mlg \sin q_0}{k} \\ u_0 = -mlg \sin q_0 \end{cases} \tag{4.8}$$

Linearizing (4.7) about this equilibrium yields a linear time-invariant system

$$\delta \dot{\boldsymbol{x}} = A\delta \boldsymbol{x} + B\delta u \tag{4.9}$$

where

$$\boldsymbol{x} \triangleq \begin{pmatrix} \delta q & \dot{\delta q} & \delta \alpha & \dot{\delta \alpha} \end{pmatrix}^T \tag{4.10}$$

$$A \triangleq \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{mlg \cos q_0}{I} & 0 & -\frac{k}{I} & -\frac{\nu}{I} \\ 0 & 0 & 0 & 1 \\ \frac{mlg \cos q_0}{I} & \frac{\nu_m}{J} & -k\left(\frac{1}{I} + \frac{1}{J}\right) & -\nu\left(\frac{1}{I} + \frac{1}{J}\right) - \frac{\nu_m}{J} \end{pmatrix} \tag{4.11}$$

$$B \triangleq \begin{pmatrix} 0 & 0 & 0 & -\frac{1}{J} \end{pmatrix}^T \tag{4.12}$$

The given linear system is controllable, except in the singular case $Ik^2 = mlg \cos q_0 \nu^{21}$. Then, a classical solution to design a robust linear controller for a stationary system is to use a steady-state LQR [Lublin10]. We use diagonal weighing matrices, which are manually tuned to obtain the desired response for each axis.

4.1.2.2 Identification of the joint stiffness and damping

In order to apply LQR design to compute the controller gains, one needs to know the coefficients of the matrices A and B of (4.11)-(4.12). The main motivation for using a LQR controller is the known robustness of this controller against model uncertainties: this is key for use on Atalante, where we use a single set of gain independently of the weight of the user. Knowing that this inherent uncertainty is present, we have only carried out minimal system identification experiments with the aim of getting simply an order of magnitude on the value of each parameter, more than a precise value; experimental results have shown this to be sufficient to obtain a working controller.

For the mass and inertia parameters (m, I, J), we directly use theoretical CAD values for a user of average mass. Motor-side friction was previously

[1]This case is never encountered in practice on the system, as numerically k is much larger than the other parameters. This condition would require a particularity large damping and/or low stiffness to be encountered.

identified during actuator design on a simple testbed, following an approach similar to [Kelly00] to identify a viscous and Coulomb (dry) friction term[2].

Remark. *Notice that we do not perform robot-specific identification. In practice, the friction coefficients vary from one robot to the next due to tolerances in actuator assembly. However, the high-gain LQR approach we follow is robust enough to only rely on this "nominal value", identified once on a testbed. Likewise, the flexibility stiffness and damping which we identify below is performed only once, and does not need to be redone whenever a new robot is built.*

This leaves us with only two parameters to identify: the flexibility stiffness and damping, k and ν.

In order to identify the stiffness, we use a simple calibration experiment with the objective of plotting the deformation versus torque curve. Indeed, in static the spring-damper model simplifies to $\tau = k\alpha$: thus, if measurement of τ and α are available, then k can be simply computed as the slope of the associated linear regression.

To perform this experiment, the exoskeleton is set to a reference standing position. Then, we manually apply a gradually increasing force on the swing foot, to generate as much deformation as possible, each time holding a constant position (and thus, a constant torque) for a few seconds. We then use our IMU-based observer to obtain a direct measurement of the deformations α. For the torque, since Atalante is not equipped with joint torque sensors, we instead use the motor's current measurement as a torque measurement, assuming that the torque constant k_t is accurately known[3].

When doing this identification, the resulting point cloud features a visible offset, with a non-zero deformation at zero torque. This is attributed to IMU bias: indeed, at the time of these experiments, the IMU alignment offset presented in Section 3.5.2 had not been calibrated. We thus remove this offset to keep only the linear part: this is presented in Figure 4.2, which reports on six different experiments, carried out on different robots, to identify the stiffness of the sagittal hip.

The resulting fit shows a significant dispersion of the results, with variation of more than 20% between experiments. This reflects possible differences between exoskeletons, but also the inaccuracy of the identification procedure, where we consider that a measurement of the motor's current is equivalent to measuring the joint's torque at the output of the transmission: in practice, the motor's torque constant or the presence of dry friction introduce

[2]Only the viscous friction coefficient is kept in our linear model (4.9)

[3]We use the manufacturer's nominal value with no further identification.

additional sources of error. Nevertheless, these experiments are sufficient to obtain a nominal value of the joint's stiffness: this is sufficient to perform gain tuning, as the resulting feedback controller proves to be robust to these modeling errors.

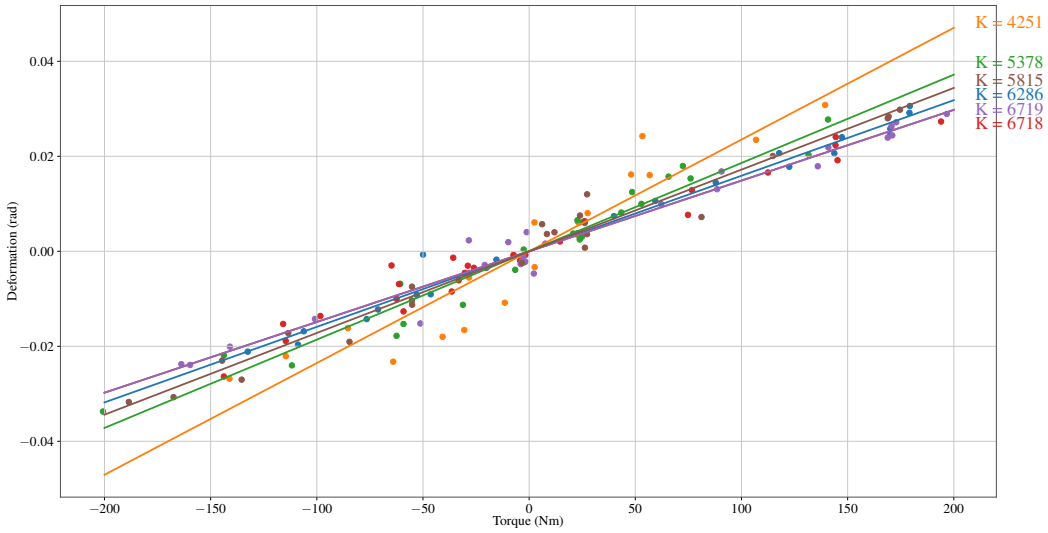


Figure 4.2: Identification of the sagittal hip stiffness: each color represents a different experiment, carried out on the right and left hip of different robots. The offset is attributed to IMU bias and is removed: the slope of the corresponding linear regression gives an estimate of the stiffness.

Finally, the last parameter, ν, is estimated to make the simulation of (4.9) match the decay rate of the oscillations observed as a response to a perturbation in static, similar to the experiment shown in Section 3.6.1. Again, this does not give access to a precise value of this damping coefficient, but only gives the correct order of magnitude, which is all that is needed to obtain meaningful gains with a LQR.

4.1.2.3 Controller implementation summary

The implementation of our *flexibility compensation controller* thus ultimately comes down to computing new gains for the hips and ankle joints, to implement (4.6). Indeed, for the knee joints which are modeled as rigid, the original PID controller (4.1) is kept as-is.

The gain tuning is done joint per joint using LQR design as presented in Section 4.1.2.1. When tuning the gain, we had to pay attention to the presence of delays in the control loop, which could render some joint controllers unstable. This is handled by making sure in simulation that the

gains computed by the LQR have a sufficiently large delay margin to be safely implemented on the hardware. Both the observer and the controller are implemented on the robot's hardware in the low-level control routine, running at 1 kHz.

Note that while the controller for each joint is tuned to be stable and robust on the linear single joint system, this does not guarantee that the resulting controller (4.6), applied on the whole system, remains stable. To numerically validate our controller, we linearize the dynamics (4.2) and check that the resulting close-loop dynamics is indeed stable, despite having neglected joint coupling for gain tuning.

4.1.3 Experimental results

To validate the effectiveness of this new controller, we perform two different experiments: disturbance rejection, on an empty exoskeleton, about a static pose, and a slow stepping trajectory, both with a user and a dummy. Each time, we benchmark the performance of our *flexibility compensation controller* against the *rigid controller* (4.1).

Note that these experiments were done in the middle of the PhD, as work on state estimation and controller were carried out in parallel. They constitute an important milestone in the development of a feedback controller applicable to Atalante's dynamic walking, as presented hereafter in Sections 4.2-4.3.

At the time of these experiments, the *KinematicObserver* was not fully developed, thus we used the *StaticObserver* instead[4]. Note also that the robot was not equipped with foot IMUs at the time: these IMUs were replaced by a flat foot hypothesis for the stance foot, while the deformation of the swing foot ankle is simply set to zero (considering that no weight is supported by this joint). In the evaluation of these experiments, we simply use the observer as ground truth[5].

4.1.3.1 Disturbance rejection

In a first test case, we assess how each controller reacts to external perturbations. We place the empty exoskeleton in single support on its right leg. We then apply perturbations by manually pushing the exoskeleton, at two different points: pushing the swing foot downward and pushing the back of the exoskeleton forward. A similar experiment is carried out for both the

[4]The *KinematicObserver* will be used for the final walking experiments in Section 4.3

[5]A more formal evaluation using motion capture is done in Section 4.3

rigid controller (4.1) and the *flexibility compensation controller* (4.6). This experiment is shown in the associated video [6].

Figure 4.3 shows the comparison of both of these experiments when a push is applied to the swing foot. Because the deflection due to the flexibilities are not taken into account, the *rigid controller* suffers from a large static error, despite accurate tracking on the motor side. This error is mostly compensated by our *flexibility compensation controller*, with a residual error coming from the fact that the kinematic model used for the observer, spherical deformations at the hip, does not exactly match the model used for control, which considers three successive deformations at the hip motors.

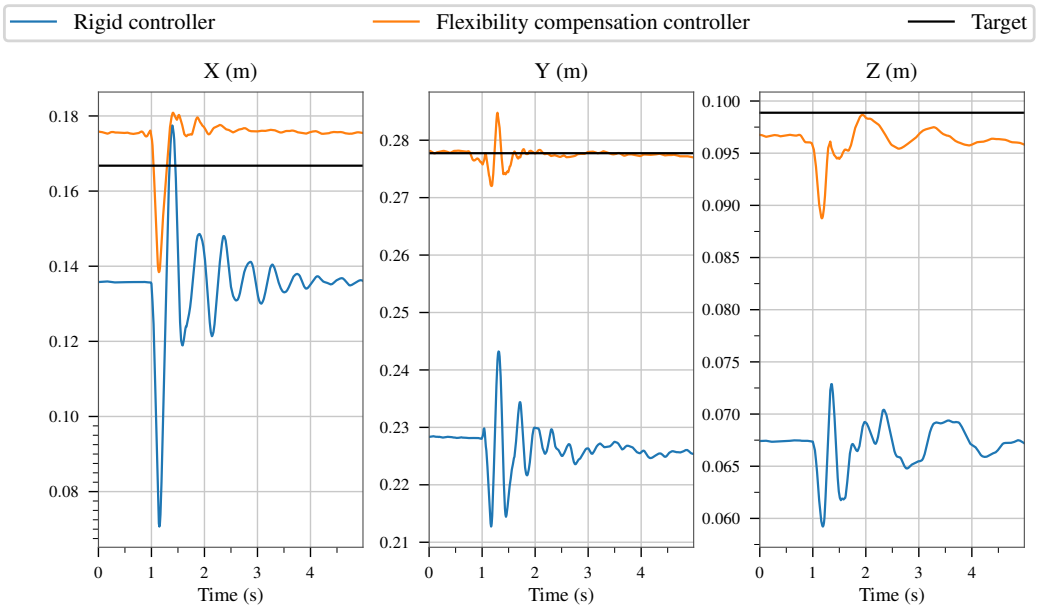


Figure 4.3: Position of the swing foot, when using the *rigid controller* (first experiment, blue) or the *flexibility compensation controller* (second experiment, orange). The target position is the same, and in both experiment an external force is applied to the swing foot at t=1s.

Furthermore, our controller provides a much better damping of the oscillations, which are mostly absorbed in two periods, whereas the *rigid controller* oscillates for several seconds, and slows down mostly thanks to the natural damping of the system. This improved damping is not present in feedforward-based gravity compensation techniques, but is here provided by the full-state feedback approach.

Since the disturbances applied are pushes in the sagittal plane, the joint

[6] https://youtu.be/c2Vdx81iu1A?t=128

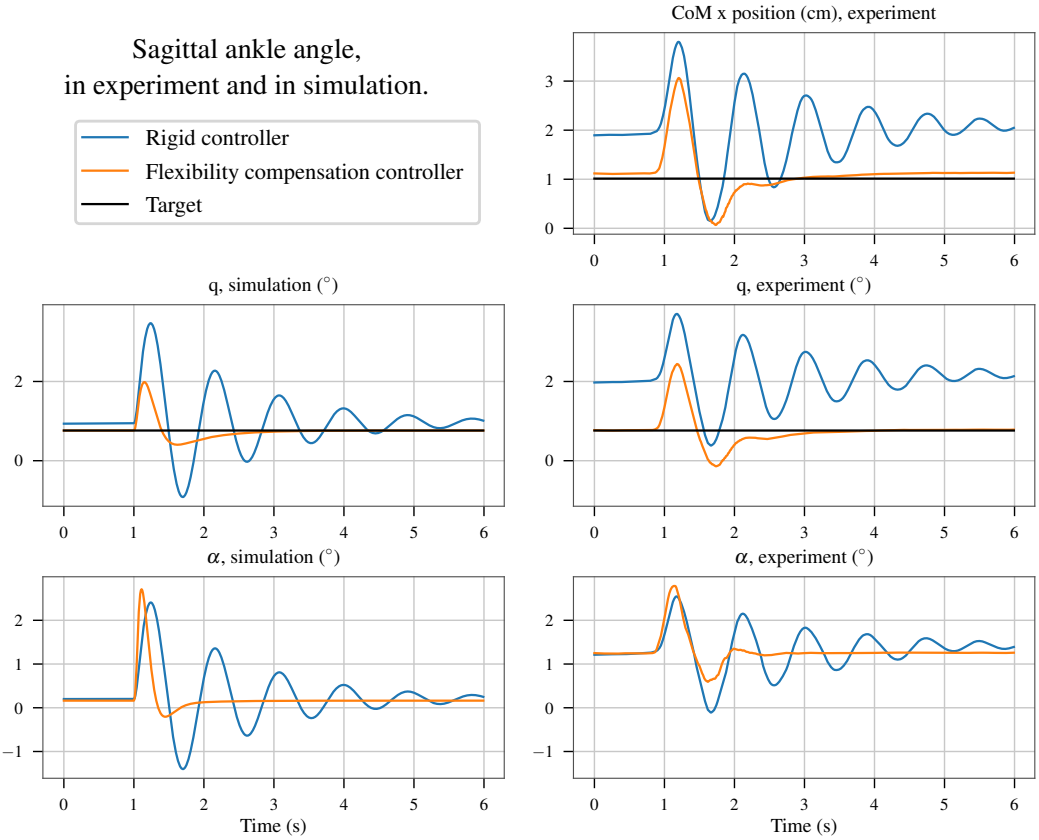


Figure 4.4: Sagittal ankle joint position (top) and flexibility angle (bottom). The right column shows experimental results, the left simulation results of the proposed linear model. In both cases, an external force is applied at t=1s.

with the most impact on the overall response of the system is the sagittal ankle, which bears the largest load. Figure 4.4 shows the response of the sagittal ankle flexible joint, when pushing the back of the exoskeleton, both in simulation and experiments. The simulation model used here is the linearized joint model (4.9) used for gain tuning. Notice that the curve of the center of mass indeed closely matches the oscillations observed at the ankle, proof that this is the predominant dynamics in this experiment.[7] This illustrates that our linear model manages to capture the overall dynamics of the system, by predicting a qualitatively correct response.

The main difference that can be seen is the value of the flexibility at

[7]These observations are consistent with those made during similar experiments while doing observer design, in Section 3.6.1.1.

rest: while a linear spring models predicts only 0.16° of deflection, the real deformation is estimated to be 1.25°. The model of linear elasticity, though sufficient to obtain an oscillation at the correct frequency, is thus inaccurate: a non-linear stiffness plus backlash would be necessary to bring the model closer to reality.

4.1.3.2 Stepping with a user

Our second test case features a more realistic scenario: a slow stepping trajectory with a user inside. Atalante is of course not meant to be used empty, but together with patients with various morphologies. A very convenient property of the *rigid controller* is that the same controller can be used both on the empty exoskeleton and when bearing any user. This robustness is very desirable, as doing individual gain tuning for each user would be impractical in a real world scenario.

The same robustness is achieved with our *flexibility compensation controller*. This is illustrated in the experiment depicted in Figure 4.5, where we perform a slow stepping motion. This experiment is conducted with both a dummy and a valid user, who is asked to remain still in the exoskeleton[8]. In both cases, the exact same controller is used, with no further gain tuning compared to the empty exoskeleton case (thus, our controller is completely agnostic of the mass of the user). Not only is the controller still stable, but it again outperforms by far the *rigid controller*.

Both experiments start with the exoskeleton in single support, with the *rigid controller* enabled. In the first experiment (in blue in Figure 4.5), this controller is kept active, whereas in the second (orange) one we switch at $t = 0$ to the *flexibility compensation controller*. For the first 5 seconds of the experiments, the target position remains constant, in order to allow the integrator in (4.6) to cancel the static error created by the flexibility. Then, a stepping trajectory is performed: the motion is slowed down compared to a real walking pattern to remain statically stable, so as to enable an experiment starting at the middle of the step, with the swing foot already airborne and the robot statically balanced on its right foot. To keep the single support assumption valid, weight transfer onto the left foot is not fully performed either.

When using the *rigid controller*, a large static error remains throughout the trajectory. In particular, the foot is approximately 2.5 cm lower than it should be. This discrepancy causes the robot to strike the ground much earlier than expected, after only 2.5 s of stride. As the target continues to

[8]The video of these experiments is available at https://youtu.be/c2Vdx81iu1A?t=202.

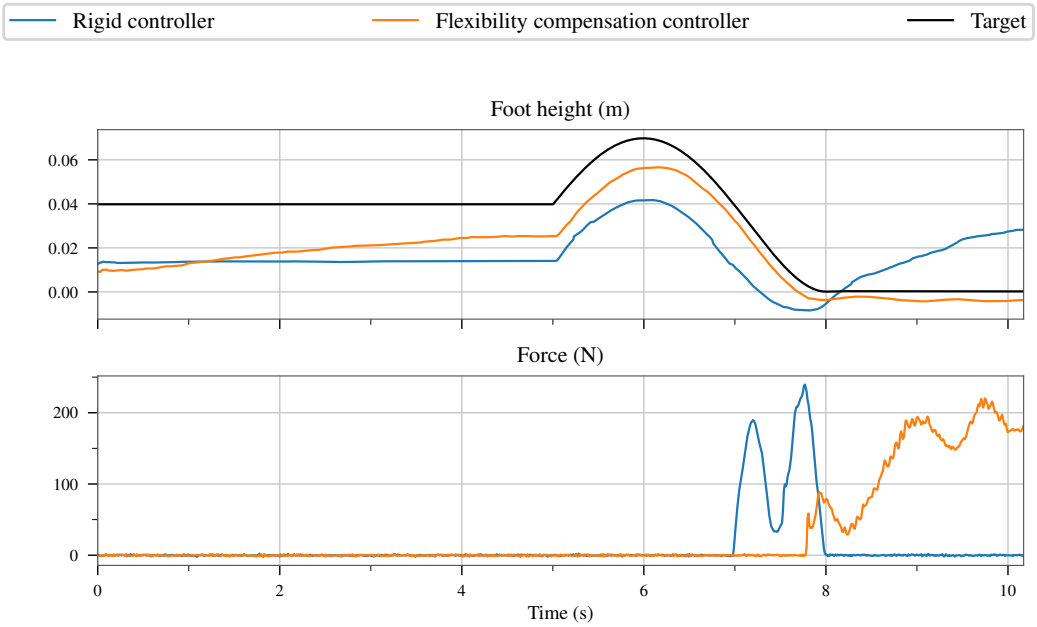


Figure 4.5: Height of the swing foot during a stepping trajectory with a dummy, and swing foot force sensor readings showing ground impact. Due to early ground strike, the exoskeleton falls (the force goes back to zero) when using the *rigid controller*.

push the foot lower, the exoskeleton tips over and falls.

On the contrary, the *flexibility compensation controller* greatly reduces the static error and improves trajectory tracking. Even though the position of the swing foot is not perfectly compensated, this correction makes the exoskeleton strike the ground as planned at $t = 9$ s. The exoskeleton then no longer falls, and is able to complete the step up to the start of the double support phase.

This experiment shows the importance of being able to control the position of the swing foot: when the flexibilities are not mitigated, early impact can indeed cause the exoskeleton to fall. In practice, other stabilization strategies such as the use of an admittance controller at the high-level [Caron19] enable Atalante to walk stably even with the *rigid controller* - but the control of the swing foot position during a walking gait remains a key element.

4.2 Modifications of the controller to handle a walking gait

The static experiments presented in Section 4.1.3 are a promising proof of concept of a flexibility-compensating controller. However, the implemented controller is not directly applicable to a walking gait. While the overall approach remains the same, three modifications are carried out to perform walking experiments.

4.2.1 Removal of the feedforward term

The computation of a feedforward term is more complex while walking than in single support, due to the presence of double support phases, during which the motor torque greatly depends on the repartition of effort between both feet. The trajectories currently generated for Atalante theoretically feature an instantaneous double support phase, with a direct transfer of the center of pressure from one foot to the next through the application of a discontinuous torque.

In practice however, such motion is not feasible: this would require a perfect knowledge of the instant at which the exoskeleton strikes the ground, and an unlimited control bandwidth. Instead, the real system thus features a true double support phase, lasting about 0.2 s. During this phase, the feedforward term computed from the trajectory, which assumes single support, is very inaccurate. Coupled with its discontinuous nature, this term proves to be nefarious to the system's performance.

The computation of a more meaningful feedforward term would require an adaptation of the input trajectory, to include a double support phase with a continuously varying force under both feet. This adaptation however is beyond the scope of the thesis. Instead, we simplify our controller by removing this feedforward term altogether, considering that the use of high feedback gains, and the presence of an integrator, is sufficient to obtain good tracking. Indeed, even without feedforward, our controller can be tuned to obtain a good quality tracking, with root mean square error below 0.01 rad.

4.2.2 Handling the change in stance leg

The choice of the feedback gains in Section 4.1.2.1 relies on the assumption that the exoskeleton is in single support. The gains obtained are thus asymmetrical, with the stance leg having larger gain values than the swing leg. This is particularly true of the ankle: where the support ankle supports the

full weight of the system, i.e. about 150 kg, the swing ankle only needs to hold a few kilos, the weight of the foot.

As the exoskeleton alternates between right support and left support during the walk, the gains need to be modified accordingly. More generically, one could imagine changing the value of the gains continuously during the weight transfer phase in double support. To keep things simple however, we decided to rely only on two sets of gains, and a discrete switch. Namely, we use the force sensors under each foot, and set a threshold to determine the stance phase from the swing phase. We then change the value of the gains of the ankle controller as we transition from one phase to the next.

On the contrary, for the hips controller, we use the same value of the gain for both phases. Indeed, the gain values provided by the LQR are quite close for the stance and swing hip: there is no more than a factor 2 in the difference of inertia between both joints, whereas this difference is close to 50 for the ankle joints. This simplifies gain tuning by reducing the number of parameters, requiring only a single set of gains for both hips. Note that using equal gains for both legs is also the approach of the existing *rigid controller*.

Although the feedback gains of the hip controllers are not changed, another element of this controller is modified between the swing and stance phase: the value of the integral term. This is linked to the fact that, in our controller, we set the reference value of the flexibility to zero (thus $\boldsymbol{\theta}^* = \boldsymbol{q}^*$): hence, the value of the integral actually captures the average deformation. The deformation however can be quite different between the stance phase and the swing phase: this is in particular the case of the frontal hip (i.e. the roll angle of the hips deformation), the largest source of deformation during the walk. This is shown in Figure 4.6.

To capture this, we thus compute and store two integrals: one is applied and updated during the swing phase only, the other during the stance phase. Mathematically, our controller thus rewrites

$$\begin{aligned}\boldsymbol{u} = &- K_\theta(\boldsymbol{\theta} - \boldsymbol{q}^*) - K_{d\theta}(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{q}}^*) \\ &- K_q(\boldsymbol{q} - \boldsymbol{q}^*) - K_{dq}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^*) \\ &- K_i\Big(\mathbb{1}_{sw} \int_t [\mathbb{1}_{sw}(s)(\boldsymbol{q} - \boldsymbol{q}^*)(s)ds] \\ &\qquad + (1 - \mathbb{1}_{sw}) \int_t [(1 - \mathbb{1}_{sw})(s)(\boldsymbol{q} - \boldsymbol{q}^*)(s)ds]\Big)\end{aligned} \tag{4.13}$$

where $\mathbb{1}_{sw}$ is the characteristic function of the swing phase, i.e. the function equal to 1 during the swing phase, and 0 during the stance phase, as determined by thresholds on the force sensor data.

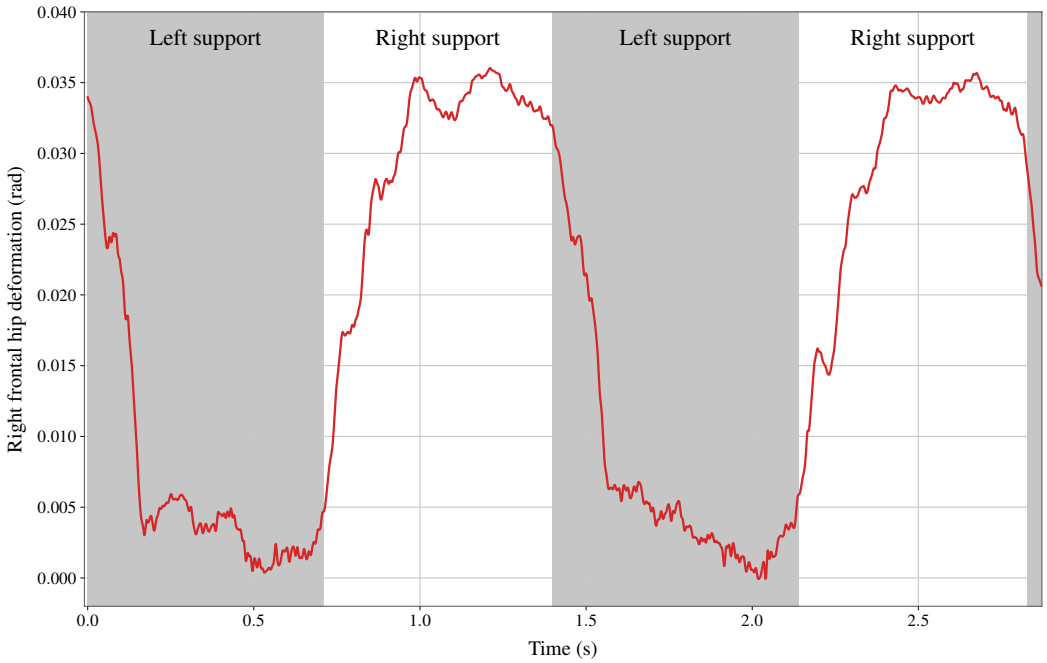


Figure 4.6: Deformation of the right frontal hip while walking, as seen by motion capture. The deformation angle is much larger during the stance phase than the swing phase, motivating the use of a different integrator to capture this pattern.

4.2.3 Manual adjustment to LQR gain tuning

Finally, the last modification carried out on the controller is a novel tuning of the gains, increasing their values to enhance tracking performance. Indeed, the controller as used in single support is mostly designed for disturbance rejection more than trajectory tracking. This is visible for instance in the very slow convergence visible in Figure 4.5, a consequence of using a low integral gain: fast convergence was not required for this experiment.

To retune the gains, we simply modify the weights of the LQR controller (reducing the weight on the input u i.e. a "cheap control" problem [Seron99]) following the method of Section 4.1.2.1. When doing so however, we observe significant high-frequency vibrations, at frequencies around 20 Hz (the exact frequency being dependent on the joint considered), that render the controller unusable. These vibrations cannot be explained or reproduced in simulation when using the elastic joint model - even when considering the non-linear equation (4.2) which introduces a coupling between the joints, or taking into account the presence of a control delay on the real system.

Instead, we believe that these vibrations are the consequence of the pres-

ence of *additional deformations* in the system. Recall indeed that the joint deformation model is only an approximation, a kinematically convenient formulation for control. In practice, the deformation comes from the exoskeleton's structure, and it is not unrealistic to consider that several parts contribute to the total deformation observed. While this can be kinematically represented as a single deformation by summing up the angles, dynamically speaking the presence of another deformation modifies the high-frequency response of the system, which proves to be detrimental to the stability of the controller.

In Appendix D, we show how the addition of a second spring to the series-elastic actuator model (4.7) can explain the vibrations observed on the robot. This study also suggests a heuristic workaround to remove these troublesome vibrations: since the presence of a second deformation mostly modifies the phase of the gyroscope signal, we reduce the corresponding feedback gain, i.e. K_{dq}.

Thus, we perform gain tuning by computing a complete set of gains using an LQR on the nominal, one-deformation model, then manually decreasing the gyroscope gain until no more vibrations are observed. This decrease ranges from a factor 1 (no modification) to a factor 10, depending on the joint considered. The resulting controller proves to be sufficient to provide very good tracking while walking, as shown in the next section. The main drawback of this modification however is a severe decrease in disturbance rejection around a static position: when compared to the results of Section 4.1.3, this novel *flexibility compensation controller* yields oscillations as large as the *rigid controller*. In the end, this means that different controllers (or rather different feedback gains) need to be applied depending on whether the exoskeleton is static or walking. Unifying these controllers, by providing a gain switching strategy, a more robust gain tuning method, or a different controller altogether, could be the topic of future research.

4.3 Experimental results on walking motion

In this section, we present the results obtained on walking experiments, with a valid user and with a dummy. Valid users, although asked to remain motionless so as not to purposefully help the exoskeleton walk, yield more stable walking patterns than paraplegic patients, let alone dummies. Keeping balance is indeed a deeply anchored reflex, and both valid users and paraplegics always work to keep their torso and head straight. Testing our controller on both a valid user and various dummies is therefore a good way to test its robustness.

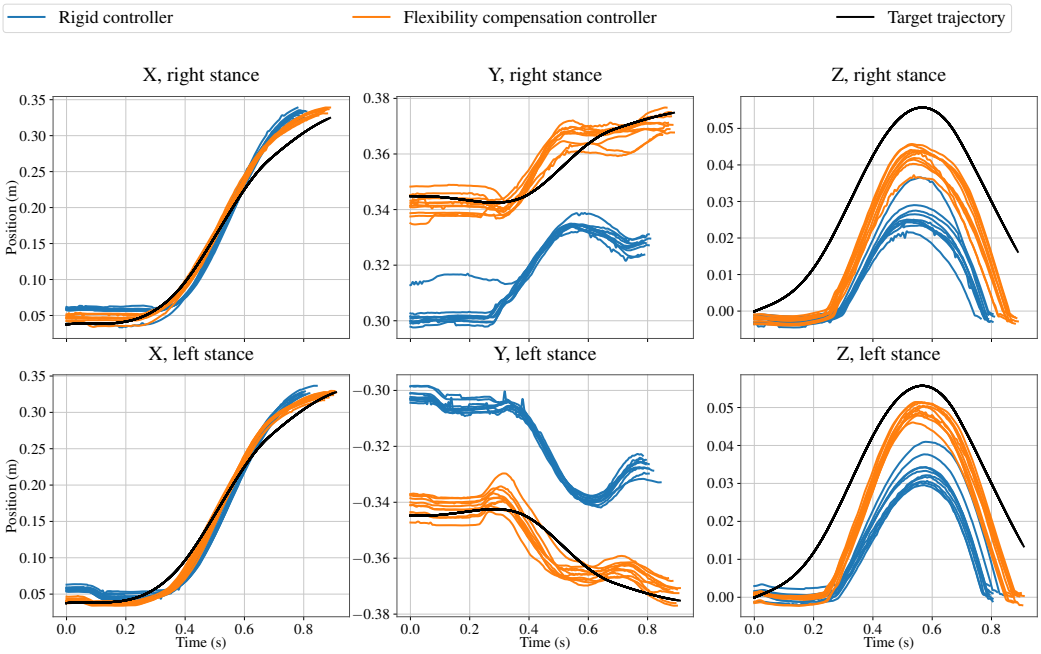


Figure 4.7: Motion of the swing foot's toe in the local frame, as seen by motion capture, when using the PID controller (blue) and the *flexibility compensation controller* (orange). Notice how the steps are shorter when using the *rigid controller*, due to the early ground impact linked to the presence of flexibilities.

In these experiments, the *KinematicObserver* is used. We use motion capture to evaluate the combined accuracy of the controller and the observer, and perform a detailed error analysis.

4.3.1 Walking with a valid user

The nominal development and test case at Wandercraft consists in having a valid user inside the exoskeleton.

We perform two walking experiments with the same user, following the same trajectory and using the same high-level controller. Between these experiments, we only change the low-level controller, from the nominal *rigid controller* to our *flexibility compensation controller* (4.13). Figure 4.7 shows the corresponding motion of the swing toe, as seen by motion capture, during the entire experiment: the right and left steps are plotted separately, each line corresponds to a single step. The black line corresponds to the nominal trajectory being tracked.

Using the *flexibility compensation controller* enables a much more accu-

rate trajectory tracking: most notably, it removes the 4 cm lateral offset, and raises the foot by about 2 cm during the swing phase. Another significant difference is the duration of each step: notice that, when using the *flexibility compensation controller*, the steps are longer, lasting 0.88 s on average, instead of 0.79 s. In both cases however, the steps are not fully completed (notice that the height of the nominal trajectory is not zero): in theory, a full step should last 1 s. This behavior comes from the high-level controller, which performs an impact-triggered trajectory switch. This is an empirical stabilization mechanism, which considers that, once the swing foot has reached the ground, the exoskeleton must not try to lower it anymore: this would mean pushing harder on the ground, and could make the exoskeleton fall[9]. Instead, the high-level controller directly jumps to the start of the next step, performing trajectory interpolation to yield a continuous target for the low-level controller.

One of the reasons for which the swing foot strikes the ground earlier is the presence of deformation, which implies that the foot is lower than expected, and angled forward. Hence, when using the *rigid controller*, the toes almost always impact the ground before the heels. This is why we use the toe position as metric to evaluate the performance of our approach. By correcting the foot position and orientation, compensating the flexibilities, we effectively raise the toe, thereby delaying impact and realizing a walk closer to the desired trajectory: as mentioned above, the average trajectory completion goes up from 79% to 88%, corresponding to a trajectory height of 18 mm at impact, against 31 mm for the *rigid controller*.

4.3.1.1 Analysis of the residual error

Consider again the trajectory tracking performance illustrated on Figure 4.7. We here perform a detailed analysis of the residual error committed by the *flexibility compensation controller*. Indeed, our approach features the use of an observer-controller for the low-level control. It is thus important to distinguish between the error caused by the observer, and that cause by the control, to see what is the limiting factor. Two other sources of error are also present in Figure 4.7, which are not caused by the low-level control but are the result of the complete system behavior:

- a lack of balance causing the support foot to tilt on the ground, impacting the position of the swing foot (by modifying the freeflyer coordinates) despite good low-level tracking.

[9]This is what happens in the quasi-static stepping experiment of Section 4.1.3.2, where this mechanism is not active.

- an interpolation of the nominal trajectory done by the high-level controller to handle the early ground impact, explaining the double support phase at the start of the step.

These various effects are discussed below.

Controller tracking error In this paragraph, we evaluate the tracking error, i.e. the error due to the control part alone. For this purpose, we need to look at the quantity on which the feedback is performed. Indeed, feedback is not done on motion capture, but on the output of the *KinematicObserver*, followed by a projection of the deformation angles onto the joint axes. This process gives an estimated flexible angle, $\hat{\boldsymbol{q}}$, which the low-level controller tries to servo to $\boldsymbol{q}^*$.

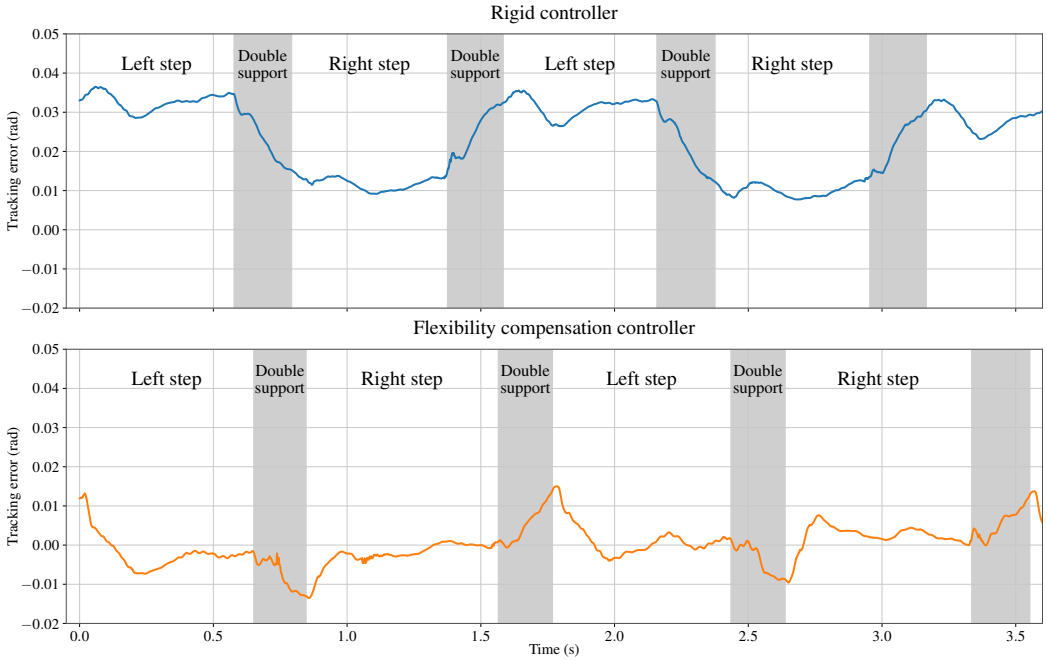


Figure 4.8: Tracking error of the right frontal hip, when using the *rigid controller* (top) or the *flexibility compensation controller* (bottom).

The controller's tracking error can be evaluated at joint level as $\boldsymbol{e}_{joint} \triangleq \hat{\boldsymbol{q}} - \boldsymbol{q}^*$: this is presented in Figure 4.8 for the exoskeleton's frontal hip, the largest deformation during the walk. The top curve shows the tracking of the *rigid controller*: since this controller works on $\boldsymbol{\theta}$ only, the resulting error on the flexible angle is quite large, as the flexibility is not compensated. Conversely, our controller provides almost perfect tracking during the swing

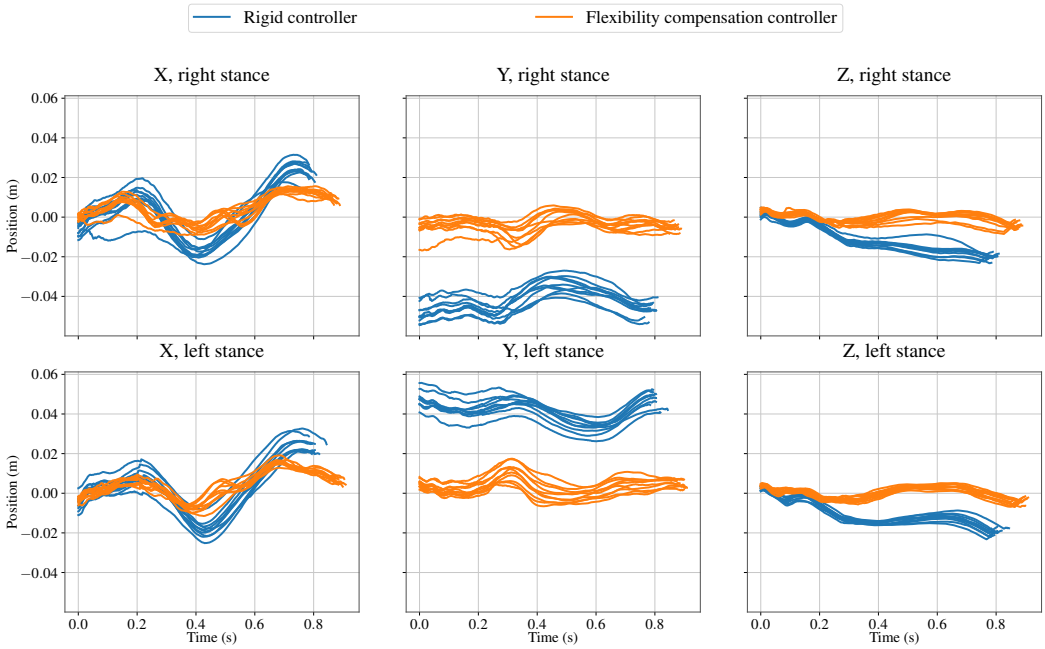


Figure 4.9: Impact of the tracking error on the swing toe position, evaluated using the joint deformation model, when using the *rigid controller* (blue) and the *flexibility compensation controller* (orange).

phase. The error is largest during the double support phase: indeed, during double support, the constraints applied on the system change rapidly, as the load is transferred from one leg to another. This results in a rapid variation of the deformation and the desired feedforward torque, two elements our controller is missing: the error thus momentaneously increases, but quickly drops back to zero at the end of double support.

Such a small tracking error (with root mean square error of 0.005 rad, against 0.024 rad for the *rigid controller*) is far from explaining the total error observed on Figure 4.7. To illustrate this, Figure 4.9 reports the effect of this tracking error on the swing toe tracking: more precisely, we compute the swing toe position using the robot's (rigid) forward kinematics g_r, thus comparing $g_r(\hat{\boldsymbol{q}}) - g_r(\boldsymbol{q}^*)$.

While the *rigid controller* causes a large error due to the presence of deformations, the error linked to the *flexibility compensation controller* is quite small. This is particularly true in the vertical direction, where the error always remains below 5 mm. This means that there remains little to gain in term of foot height simply by enhancing the low-level controllers (increasing the gains), and that the cause for the early impact must be looked for elsewhere.

Other sources of error To clarify the error analysis, we focus the study on the vertical axis only: indeed, controlling the height of the swing foot is what has the most impact on the walk behavior, in particular by determining the frequency of the steps due to the impact-triggered switch. Figure 4.10 shows various curves that represent the height of the swing foot toe during the experiment with flexibility compensation. The meaning of each curve and its significance is detailed below.

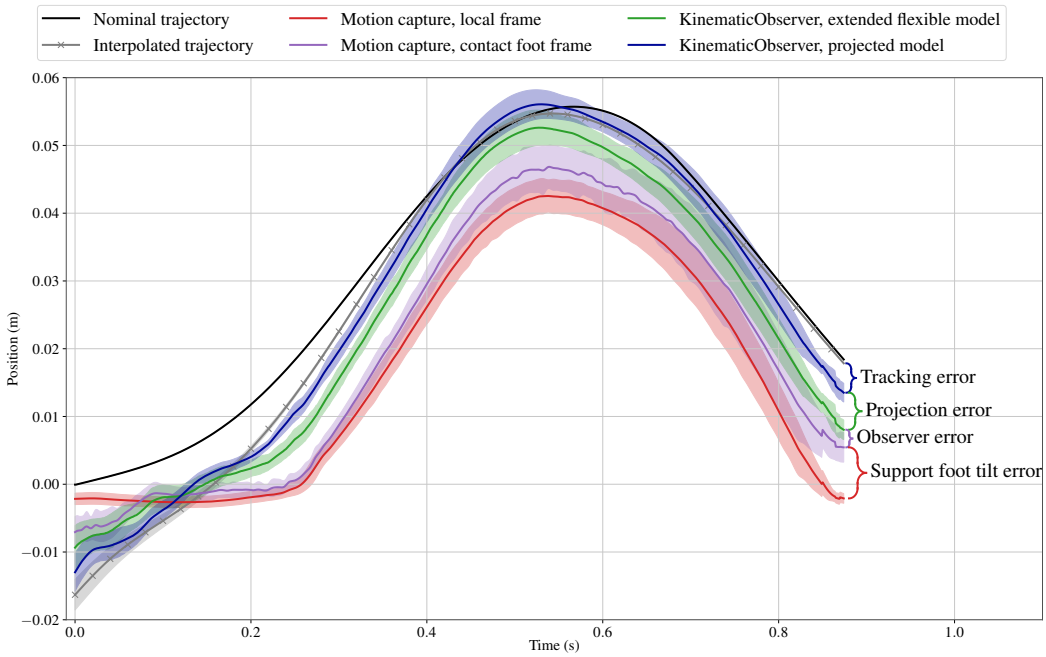


Figure 4.10: Various measurements or estimation of the foot height, which shows that the position error is the result of various effects of comparable amplitude.

The black curve represents the nominal trajectory, i.e. the input trajectory we seek to follow. This is however not the target trajectory $\boldsymbol{q}^*$ sent to the low-level controllers, represented in gray in Figure 4.9. The discrepancy comes from the fact that it is not realistic to directly track the nominal trajectory, due to the discontinuity at impact. Indeed, when the swing foot impacts the ground at $t = 0.88$ s in this case, the height of the swing foot in the trajectory is still about 18 mm. This results in a discontinuity with the next step, which starts with equal feet heights. This discontinuity is handled by the high-level controller, which performs an interpolation to smooth out the trajectory: from this computation, we obtain the true target trajectory $\boldsymbol{q}^*$, represented in gray in Figure 4.10. Notice that the interpolated swing foot height is negative at the start of the step: this comes from the

change in reference frame between both steps (the new swing foot is the old stance foot, which is located 18 mm below the other foot at the end of the previous step). This process explains the double support phase observed at the start of the step: it indeed takes 0.16 s for the target trajectory to reach an equal height for both feet, so the swing phase cannot start before this time.

The red curve shows the position of the swing toe in the local frame, as seen by motion capture: this curve should track the gray target trajectory curve. This position however is determined by two different types of motion: the relative motion of the robot's swing foot with respect to the stance foot, which we directly control with a low-level controller, and the motion of the stance foot with respect to the ground. While a control objective of the high-level controller is to keep the stance foot flat on the ground, as is the case in the target trajectory, in practice the stance foot can tilt on its edges. This is shown in the purple curve, which shows the position of the swing foot with respect to the stance foot, instead of the local frame. As such, the difference between the red and purple curve is the height loss due to the stance foot tilting on the ground, something for which the low-level controller is not directly accountable.

The difference between the purple and gray curve now truly represents the tracking accuracy of the low-level observer-controller, i.e. the difference in relative foot placement between the target trajectory $\boldsymbol{q}^*$, and motion capture measurement. This error can further be attributed separately to the control and observation part of our methodology. More precisely, the green curve is the height estimated by the *KinematicObserver* - and thus, the difference between the green and purple curve is the observer error, as quantified at the end of Chapter 3. This error however is computed by using the extended flexible model: instead, the blue curve shows the height estimate when projecting the deformation onto the joints. This causes an additional 5 mm of height error, as seen in Section 2.4. This error is not accounted for in our control framework, which works on the joint deformation model, and therefore only seeks to bring the corresponding foot position (blue curve) onto the target (gray curve). The error between these curves is what is shown in Figure 4.9, which confirms the fact that there is little to gain from trying to push the gains even further.

Summary: total tracking error imputable to the low-level control strategy The above study shows that the error in swing toe placement in the local frame comes from many different sources. In particular, the cause for the early ground impact (at 88% of the trajectory) is shared between four

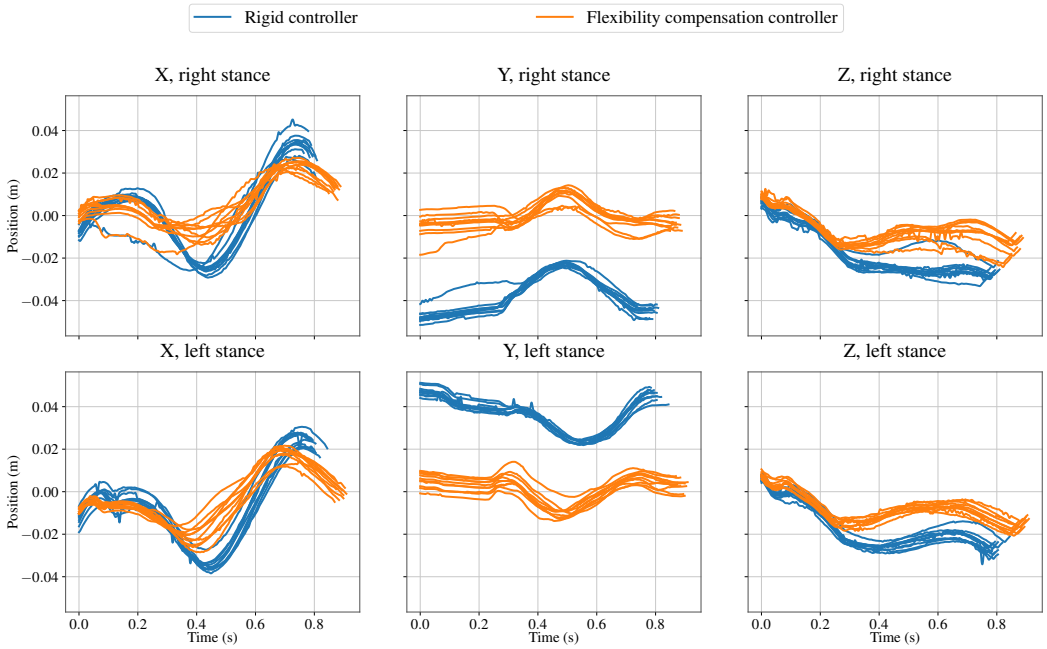


Figure 4.11: Complete low-level tracking error, measured by motion capture, for the rigid and *flexibility compensation controller*. This corresponds to the error in swing toe placement with respect to the stance foot frame.

elements (low-level controller tracking, observer performance on the extended flexible model, impact of the projection on the joint deformation model, and influence of the support foot loss of balance), which all contribute for comparable amounts, i.e. around 5 mm each.

To get a fair picture of the overall performance of our approach, it is important to separate the errors that can be attributed to the low-level control strategy from those that come from other sources. Namely, the interpolation process of the high-level controller explains why the nominal trajectory does not correspond to the realized motion, in particular at the start of the step. Meanwhile, the lack of balance characterized by the fact that the support foot tilts on the ground, though it stems from the motion of the robot and therefore the torque send by the low-level controller, is meant to be compensated for by high-level balance-keeping strategies. This is already the case in these experiments, where an admittance-based controller is used to keep balance, though it does not guarantee foot flatness. The corresponding error however is not relevant to benchmark a low-level controller, which only aims at controlling the relative position of the robot's links.

Thus, the total error imputable to our low-level control strategy can be formulated by removing both of these effects, and looking only at the error

in swing foot position relative to the support foot, compared to the target trajectory $\boldsymbol{q}^*$. This error can be directly measured by motion capture, and is shown in Figure 4.14. This shows the significant improvement in using a feedback *flexibility compensation controller*, over the nominal *rigid controller*.

4.3.2 Walking with a dummy

The second test case for this controller is to perform the same type of experiments with a dummy, instead of a valid user. As mentioned earlier, a dummy is by definition fully passive, while a human subject always applies reflex forces to stabilize themselves. As a consequence, a dummy is a more difficult setup for performing stable walking.

4.3.2.1 Walking with a light dummy

In this section, we use a "light" dummy, which weights 55 kg - a weight comparable to that of the user in the experiments of Section 4.3.1. Figures 4.12 shows the corresponding toe motion.

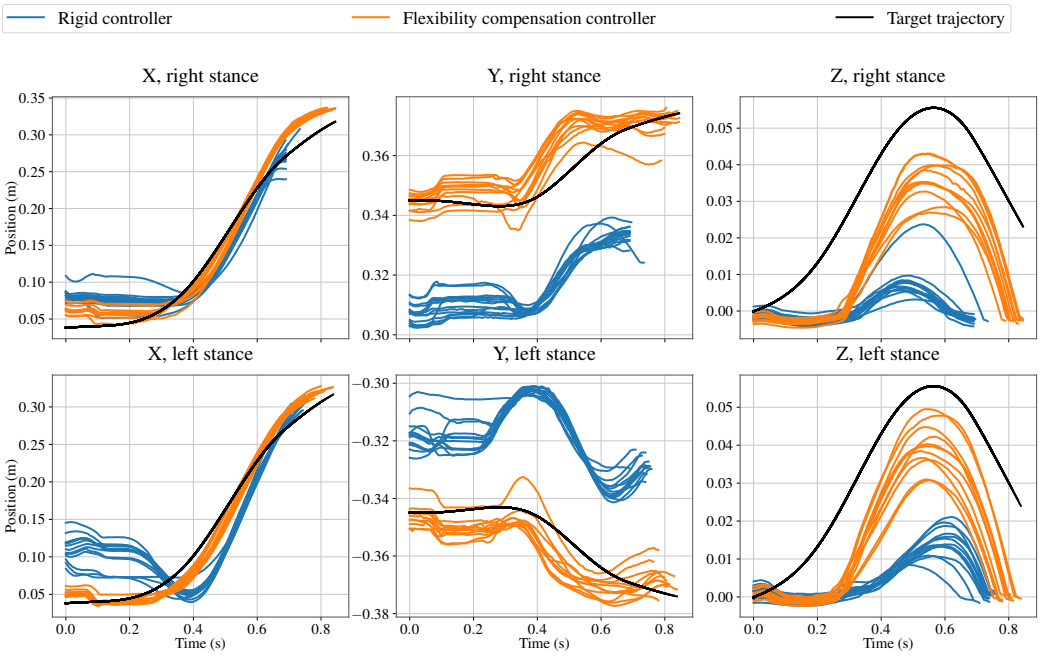


Figure 4.12: Walk with a 55 kg dummy: motion of the swing foot's toe in the local frame, as seen by motion capture, when using the *rigid controller* (blue) and the *flexibility compensation controller* (orange).

Once again, the *flexibility compensation controller* manages to significantly raise the swing foot: this effect, which can be viewed in the curves of Figure 4.12, is also clearly visible on the linked video[10], or on the images of Figure 4.13, which shows the position of the feet at the highest point in the trajectory. Notice how, when the flexibilities are not accounted for, the foot is not only lower, but also slanted, both in roll and pitch - an effect corrected by the *flexibility compensation controller*, which keeps the foot flat.

(a) *Rigid controller* (b) *Flexibility compensation controller*

Figure 4.13: Picture of the robot's feet when the swing toe is at the highest point of its trajectory.

The experiment done with the *rigid controller* shows significant differences with the one done with a valid user. Most strikingly, when using the *rigid controller*, the walk is a lot more asymmetrical than with a patient; the exoskeleton lifts its swing foot a lot lower (only 1 cm during the right steps, against 3 cm for the case of a valid user), and the steps are shorter, lasting on average only 0.65 s and 0.73 s for the right and left steps. Visually, the exoskeleton appears to be constantly tripping on the ground, with very long double support phases. This is related to the fact that the deformations are larger when using a dummy than with a user. The vertical deformation of the toe is around 3.5 cm, against 2.5 cm with the previous valid user. This difference does not come from a weight difference, as the user of the previous experiments is in fact heavier than the dummy[11]. Rather, we explain the difference from the posture adopted by the user in the exoskeleton. While a

[10]https://youtu.be/c2Vdx81iu1A?t=225

[11]The user is asked not to use their leg muscles to support their own weight - though no direct measurement is available to ensure that this is fully the case. This is a potential source of difference as well.

valid user (as well as many paraplegics with control of their abdominal muscles) stands straight against the exoskeleton, keeping their back close to the exoskeleton's back, a dummy falls forward and therefore pulls harder on the abdominal belt linking it to the exoskeleton, resulting in larger deformations. Note that this difference is not captured by our nominal dynamical model, which depends only on the mass of the patient and assumes that they are standing straight.

However, in a feedback based approach, this difference can be sensed by the IMUs, and absorbed by the controller. Indeed, applying our *flexibility compensation controller* raises the foot significantly, to a height comparable to that obtained with a user. Notice that the steps remain shorter than with a user (0.81 s on average), and that the toe height curve show a large dispersion. This comes not from the low-level controllers, but from a balance issue: keeping the support foot flat on the ground proves to be more difficult with a dummy. As such, the swing foot tilts a lot more on the edge, in a slightly different manner at each step, resulting in different foot height in the local frame.

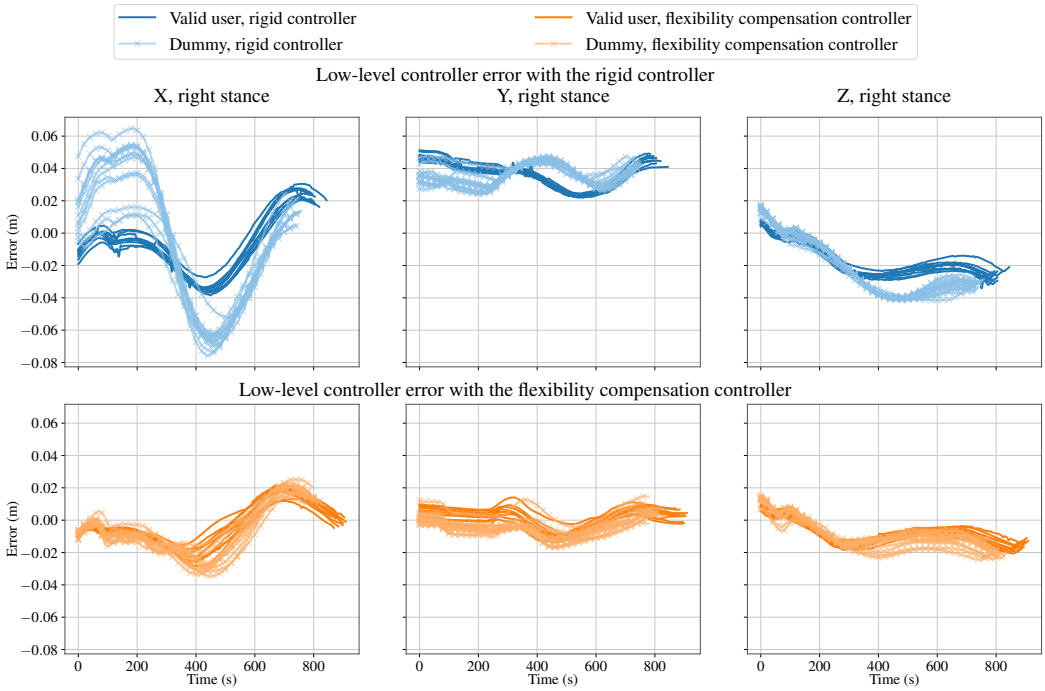


Figure 4.14: Comparison of the low-level error between both controllers, on a valid user and on a dummy (on the right stance steps only).

This effect can be seen in Figure 4.14 and Table 4.1, which compares the low-level accuracy (tracking of the swing toe in the stance foot frame), for

	X (mm)	Y (mm)	Z (mm)
Valid user, *rigid controller*	19	38	20
Valid user, *flexibility compensation*	13	6	10
	X (mm)	Y (mm)	Z (mm)
Dummy, *rigid controller*	31	37	28
Dummy, *flexibility compensation*	16	7	12

Table 4.1: Root mean square error of the swing foot tracking (as shown in Figure 4.14), measured by motion capture.

the rigid and *flexibility compensation controller*, for the case of a dummy and a valid user. The large difference between both rigid experiments (top curves) can be mostly attributed to the difference in terms of deformation when working with a user or with a dummy. In particular, the fact that the impact happens even earlier for the dummy causes a longer unplanned double support phase, during which the exoskeleton pushes forward as it tries to follow the reference trajectory. This results in larger deformations, and thus a much larger error in X.

Conversely, activating the flexibility compensation yields a comparable error: notably, in both cases the error in toe height is reduced to about 1 cm during the swing phase. While this does not mean that the walk of a valid user and a dummy are fully equivalent (a valid user remains more stable than a dummy), the difference between the two is significantly reduced.

4.3.2.2 Walking with a heavy dummy

Finally, we consider a third, more difficult test case: a heavier dummy, with a weight of 80 kg.

This experiment features an even more drastic conclusion: with this dummy, the exoskeleton is unable to walk when using the *rigid controller*. Indeed, we have seen that when using a 55 kg dummy, the exoskeleton barely manages to lift its toe off the ground, with a peak height of only 1 cm. With a heavier dummy, the *rigid controller* is no longer sufficient to fully lift the foot off the ground: the exoskeleton trips forward, and ends up falling. On the contrary, with the *flexibility compensation controller*, the swing foot is raised successfully, and the walk is stable. This effect is shown in Figure 4.15, and can be seen in video at https://youtu.be/c2Vdx81iu1A?t=245. This experiment thus further illustrates the significant benefits brought by compensating the deformations. Indeed, it makes Atalante's walk robust enough to function

with a heavy, 80 kg dummy.

(a) *Rigid controller* (b) *Flexibility compensation controller*

Figure 4.15: State of the robot at the middle of the swing phase, with a 80 kg dummy. When using the *rigid controller*, the exoskeleton does not manage to lift its toe off the ground: the exoskeleton then trips and fall.

4.4 Conclusion

In this chapter, we have presented a feedback-based approach for compensating the impact of the deformations on the positioning accuracy of the system. This is done by modeling the deformations as joint elasticity, using the projection of the deformations estimated by the *KinematicObserver* to obtain the corresponding angle, as presented in Section 2.3.

To control these angles, we perform full-state feedback on a series-elastic actuator model. Instead of doing a PID on the motor angle $\boldsymbol{\theta}$, we perform linear feedback on both $\boldsymbol{\theta}$ and $\boldsymbol{q}$. Gain tuning is initially done using an LQR approach, in single support. This controller lead to a correction of the static deformation, and a greatly improved disturbance rejection, canceling the oscillations linked to the flexibilities.

The application of this controller to a walking gait is done using a discrete switch, to handle the hybrid nature of the walk. Furthermore, the gains are retuned to improve tracking performance. This requires to manually reduce the feedback gain on $\dot{\boldsymbol{q}}$ to prevent strong vibrations around 20 Hz. We believe this to be linked to the presence of unmodeled deformations in the system, resulting in a mismatch between our controller model and the real exoskeleton. Nevertheless, the resulting controller provides a significant improvement in trajectory tracking while walking, handling equally well a valid user, or two dummies of different weights. By contrast, the existing

rigid controller leads to a large tracking error, with poor performances and significant discrepancies between experiments - notably, it is unable to make a 80 kg dummy walk in a stable manner. By reducing the tracking error to less than 2 cm, regardless of the use case, our proposed estimation and control approach thus provides enhanced performance and robustness to the walk of Atalante.

Chapter 5

Conclusion and perspectives

Chapitre 5 - Conclusion et perspectives: *Ce chapitre conclut notre étude, et présente des pistes de recherches futures. Un rapide résumé de la méthodologie développée est tout d'abord présenté. Son implémentation sur Atalante soulève plusieurs questions qui sont examinées, afin de rendre la méthode plus générique et robuste. Notamment, l'étude de l'impact d'une deuxième déformation sur le modèle d'actionneur série-élastique nous semble être un point important, pour comprendre, maitriser et expliquer les vibrations rencontrées lors du réglage des gains du contrôleur et proposer une solution plus performante.*

Enfin, nous concluons en rappelant la généricité de l'approche. Nous soulignons son importance dans les développements de robots et d'exosquelettes futurs, plus légers afin de réduire l'encombrement et de permettre des mouvements plus dynamiques. Cette réduction de poids rend bien souvent la structure plus flexible. Dans le cas d'un exosquelette, elle implique aussi que la dynamique de l'utilisateur devient de plus en plus prédominante sur celle du système. Dans ce contexte, une solution robuste de compensation des déformations par boucle fermée sur des IMUs est d'autant plus indispensable.

Summary

In this work, we have developed and successfully validated an IMU-based correction of the effect of structural deformations on the exoskeleton Atalante.

Our methodology considers punctual angular deformations, modeled as extra spherical joints inserted in the structure. The corresponding rotations are estimated using low-cost IMUs located on the robot's links, between each

deformation. We developed an observer to estimate the tilt of each IMU, relying on the sequential application of a velocity-aided observer. One major strength of this approach is to rely only on the exoskeleton kinematic model. Not limited by a static assumption, this observer is also independent of the dynamical model, and therefore is not affected by the uncertainty in the patient's inertial parameters, or by patient's movements. This observer gives excellent angular precision, with a root mean square error of only 0.007 rad on dynamic walking experiments, resulting in only 1 cm of error on the position of the swing foot.

To use this estimation for control purposes, we project these angular deformations onto the robot joints, preserving the IMU angles: this corresponds to modeling the deformations as joint elasticity. We then modify the existing PID controllers, by adding a feedback term on the deformation angle and velocity, keeping a decentralized, high-gain approach. To perform gain tuning, we use a LQR approach based on a linear series elastic actuator model for each joint. The resulting controller brings significant tracking improvement, reducing the error in swing foot position to less than 2 cm, while being robust to the presence of a user. This correction has a major impact on the overall performance of the system. Indeed, while the presence of deformations causes the swing foot to be lower than expected, and angled forward, which often result in the exoskeleton tripping and falling, this behavior is corrected by the proposed compensation technique.

Overcoming the encountered limitations

Though the results obtained experimentally on Atalante are quite satisfying, several leads can be proposed to further enhance the performance and usability of this methodology.

On the observation side, sensor calibration seems required to further enhance the tilt estimate, taking into account elements such as sensor misalignment and cross-coupling or temperature sensitivity. Furthermore, the proposed method to realign, at rest, the accelerometer with gravity, requires a motion capture setup. This is not practical in an industrial scenario, where a fast, in-situ calibration using only proprioceptive information would be a valuable asset. Moreover, our proposed IMU-based method only reconstructs the tilt of the sensors, and thus cannot estimate yaw deformations. The integration of additional sensors, such as cameras or LIDARs, would provide this missing information.

On the control side, we observed high-frequency vibrations which we believe are linked to the presence of a second deformation between the mea-

surement points, i.e. the IMUs. The solution we propose - to reduce the gain of the gyroscope - negatively affects low-frequency disturbance rejection. As such, we do not have a unified controller on Atalante yielding both satisfactory disturbance rejection in static single support experiments, and efficient tracking while walking. A more complete analysis of the control of a system with a second deformation, with poorly known parameters, could provide a more efficient solution for controlling this system using only one IMU. Alternatively, the use of an additional IMU to instrument both angles of deformation could be used to generalize our full-state LQR-based approach to a system with two deformations.

Perspectives

In this thesis, we developed and applied a flexibility-compensation methodology specifically for the exoskeleton Atalante. However, this approach of modeling deformations as a punctual phenomenon at the ankle and hip level is likely to be a good approximation for many anthropomorphic bipeds. More generally, we believe that this approach can be applied to a variety of systems for which a joint elasticity model is already used: robotic manipulators, humanoid robots and exoskeletons... We stress the fact that, once this hypothesis is met, and once IMUs have been mounted on the robot, our method is straightforward to implement, as it requires only minimal system identification. Namely, it only needs an approximate value for the joint elasticity stiffness and damping, for the tuning of feedback gains only. The robustness of a LQR-based controller is then exploited for the control side, while the observer is inherently independent of the dynamic parameters.

This approach was benchmarked on Atalante, a relatively stiff robot: with a total (patient + exoskeleton) weight of up to 160 kg, the swing foot is only lowered by 4 cm. Although this is sufficient to significantly hinder the absolute performance of the system, something which we improve on in this work, this value is small enough to enable the development of stable walks for many patients, without considering this question of deformations. However, the development of future exoskeletons is likely to feature lighter, less cumbersome robots - which will likely translate into more flexible structures, for which a deformation-aware control approach will be even more important. Meanwhile, the weight ratio of patient over exoskeleton will increase, meaning that robustness to patient parameters and behavior will be even more critical. In this context, we believe that using a sensor-based feedback approach is all the more important.

Appendices

Appendix A

Euler angle decomposition used for projecting the deformations

In this section, we derive the formulas used to decompose a rotation matrix R, and its corresponding angular velocity $\boldsymbol{\omega}$, into a set of Euler angles and their derivatives. These classical formulas [Craig89] are often readily implemented in linear algebra libraries, for instance the well-known C++ library *Eigen*[1] used for on-board implementation in this thesis.

Euler angles consist in decomposing a rotation matrix into unitary rotations, which correspond to rotations about the canonical basis's axes: this writes

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \quad R_y(\alpha) = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix} \quad R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.1}$$

There exists 12 different decompositions into Euler angle (or Tait-Bryan angles), depending on the choice of vectors (x, y or z) for each rotation. We here present the two versions used for the hip and ankle decomposition in the projection operation ϕ presented in Section 2.3.2.

We first recall the definition of the "2-argument arctangent" function

[1] https://eigen.tuxfamily.org

giving the principal value of the argument of the complex number $x + iy$ as

$$\begin{aligned} \text{atan2}: \quad & \mathbb{R}^2 \setminus 0 \to (-\pi, \pi] \\ & (x, y) \mapsto \begin{cases} 2 \arctan \frac{y}{\sqrt{x^2+y^2}+x} & \text{if } x > 0 \text{ or } y \neq 0 \\ \pi & \text{otherwise} \end{cases} \end{aligned} \tag{A.2}$$

A.1 Hip decomposition: X-Z-Y order

For the hip joint, we consider the succession of a frontal (X), transverse (Z) and sagittal (Y) rotation, yielding the X-Z-Y order

$$\begin{aligned} R &= R_x(\gamma) R_z(\beta) R_y(\alpha) \\ &= \begin{pmatrix} \cos\alpha\cos\beta & -\sin\beta & \sin\alpha\cos\beta \\ \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma \end{pmatrix} \end{aligned} \tag{A.3}$$

From (A.3), we thus compute the Euler angles as follow:

$$\begin{aligned} \alpha &= \begin{cases} 0 & \text{if } \cos\beta = 0 \\ \text{atan2}(R_{13}, R_{11}) & \text{otherwise} \end{cases} \in (-\pi, \pi] \\ \beta &= -\arcsin R_{12} \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \gamma &= \begin{cases} \text{atan2}(R_{31}, R_{21}) & \text{if } \cos\beta = 0 \\ \text{atan2}(R_{32}, R_{22}) & \text{otherwise} \end{cases} \in (-\pi, \pi] \end{aligned} \tag{A.4}$$

Note that when $\cos\beta = 0$, the decomposition is not unique[2], and (A.4) reflects a particular choice. This is the well-known singularities of Euler angles (gimbal lock). In practice, this case is never encountered on the robot where mechanical stops limit the value of β, and this formula is only included for generality.

Differentiating (A.3) and identifying the resulting matrix with the definition of the angular velocity ($\dot{R} = R[\boldsymbol{\omega}]_\times$), we obtain the following relationship

$$\boldsymbol{\omega} = \begin{pmatrix} 0 & -\sin\alpha & \cos\alpha\cos\beta \\ 1 & 0 & -\sin\beta \\ 0 & \cos\alpha & \sin\alpha\cos\beta \end{pmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} \tag{A.5}$$

The matrix in (A.5) is invertible if and only if $\cos\beta \neq 0$, in other words, outside of the singularity of the Euler angle decomposition. Then, by com-

[2] only the sum $\alpha + \gamma$ is defined.

puting the inverse we obtain the desired Euler angles derivatives

$$\begin{pmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{pmatrix} = \begin{pmatrix}\cos\alpha\tan\beta & 1 & \sin\alpha\tan\beta\\ -\sin\alpha & 0 & \cos\alpha\\ \frac{\cos\alpha}{\cos\beta} & 0 & \frac{\sin\alpha}{\cos\beta}\end{pmatrix}\boldsymbol{\omega} \tag{A.6}$$

A.2 Ankle decomposition: the classical "roll-pitch-yaw", Z-Y-X order

The decomposition done for the ankle actually follows the Z-Y-X order. This specific decomposition is often called "the roll-pitch-yaw decomposition". Indeed, when done about the inertial frame, this decomposition corresponds to the "intuitive" notion of what these angles mean for an aircraft. Namely, the roll is the rotation about the aircraft body, the pitch the angle about the wings, while the yaw gives the aircraft heading.

Note however that this decomposition can be done about any basis of reference, not necessarily aligned with the world frame: X-Y-Z must form an orthogonal basis, but X does not have to be the world $\boldsymbol{e}_x$ axis for this decomposition to make sense. This is the case for the ankle, and this is why in (2.11) we insisted on the fact that the decomposition is done about the sagittal and Henke axis.

Mathematically however, we decompose the rotation ${}^{tibia}R_{foot}$ about the Henke frame, defined as the frame rotated from the foot frame by $-38°$ about the Y axis (see Figure 1.6). Let $H \triangleq R_y(-38°)$ be this constant rotation, and define

$${}^{tibia}R_{henke} \triangleq {}^{tibia}R_{foot}H^T \tag{A.7}$$

In the Henke frame, the Henke axis is the X axis, while the sagittal axis is the Y axis. Thus, the motion of this frame with respect to the tibia follows the classical Z-Y-X order (with the rotation of the Z axis set to zero for the rigid model): this is the decomposition we perform on the rotation (A.7).

This decomposition writes

$$\begin{aligned}R &= R_z(\gamma)R_y(\beta)R_x(\alpha)\\ &= \begin{pmatrix}\cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma\\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma\\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta\end{pmatrix}\end{aligned} \tag{A.8}$$

This yields the following formulas for computing the Euler angles

$$\begin{aligned}
\alpha &= \begin{cases} 0 & \text{if } \cos\beta = 0 \\ \operatorname{atan2}(R_{32}, R_{33}) & \text{otherwise} \end{cases} \in (-\pi, \pi] \\
\beta &= -\arcsin R_{31} \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\
\gamma &= \begin{cases} \operatorname{atan2}(R_{12}, R_{22}) & \text{if } \cos\beta = 0 \\ \operatorname{atan2}(R_{12}, R_{11}) & \text{otherwise} \end{cases} \in (-\pi, \pi]
\end{aligned} \tag{A.9}$$

Once again, this decomposition is singular when $\cos\beta = 0$. Away from this singularity, computing the derivative and performing the same computation as in (A.5) yields

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} 1 & \sin\alpha\tan\beta & \cos\alpha\tan\beta \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \frac{\sin\alpha}{\cos\beta} & \frac{\cos\alpha}{\cos\beta} \end{pmatrix} \boldsymbol{\omega} \tag{A.10}$$

Appendix B

Reconstructing a rotation matrix from a tilt estimate

As we have seen in Section 3.1.3.2, an IMU is only able to observe its tilt, and not the rotation about $\boldsymbol{e}_z$, i.e.the yaw rotation. We thus need a method to decompose a rotation matrix into an observable and an unobservable part - which we use in our context to cancel out the unobservable part, leaving a rotation matrix "without yaw" that can explain the sensor's tilt. However, the intuitive solution of using Euler angles and the roll-pitch-yaw decomposition is ill-defined, as presented below. Instead, we propose a constrained minimization-based method that relies on another decomposition: the twist-swing decomposition.

B.1 Ambiguity of the roll-pitch-yaw decomposition

The first solution that comes to mind to define and remove the yaw angle of a rotation is to perform a roll-pitch-yaw decomposition, then remove the yaw rotation. For concision, we write the decomposition of a rotation matrix in roll-pitch-yaw as $R = rpy(\alpha, \beta, \gamma)$, i.e.

$$rpy(\alpha, \beta, \gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{B.1}$$

Then, we define the rotation "without yaw" as

$$R_{noyaw} \triangleq rpy(\alpha, \beta, 0) \tag{B.2}$$

While it is clear that this rotation has the correct tilt ($\boldsymbol{t} = R^T\boldsymbol{e}_z = R^T_{noyaw}R^T_z(\gamma)\boldsymbol{e}_z = R^T_{noyaw}\boldsymbol{e}_z$), this choice suffers from two significant drawbacks.

APPENDIX B. RECONSTRUCTING A ROTATION MATRIX FROM A TILT ESTIMATE

The first one is the issue of gimbal lock, always present when working with Euler angles. This decomposition is singular when $\theta_y \cong \frac{\pi}{2}[\pi]$, and thus cannot be applied. More importantly, because Euler angles are discontinuous when crossing the singularity, the resulting mapping from $\boldsymbol{t}$ to R_{noyaw} is also discontinuous: small changes in tilt can result in large changes in R_{noyaw}. Consider for example the rotation $R_1 = rpy(0, \frac{\pi}{2} - \epsilon, 0)$, for $\epsilon << 1$. Applying a rotation of angle 2ϵ about $\boldsymbol{e}_y$ to this rotation makes it cross the singularity, resulting in $R_2 = rpy(\pi, \frac{\pi}{2} - \epsilon, \pi)$: thus, their rotation "without yaw" according to (B.2) differ by an angle of π about $\boldsymbol{e}_x$, though these two rotations share approximately the same tilt, i.e. $\begin{pmatrix}1 & 0 & 0\end{pmatrix}$ at the first order. This is clearly undesirable.

The continuity about the singular configuration is only one problem of the roll-pitch-yaw decomposition. The second, less obvious phenomenon, is the fact that the yaw angle defined by this process is not intrinsic to the rotation matrix, but instead depends on the choice of the two horizontal vectors, $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$, leading to an ambiguous formulation. This is best illustrated by an example: we take as example

$$R \triangleq rpy(0.4, -0.7, 0.6)$$

The value of these angles was chosen at random to provide a readable 2D drawing in Figure B.1, where this rotation is drawn to scale.

Removing the yaw according to (B.2) yields $R_{noyaw} = rpy(0.4, -0.7, 0)$, drawn in red in Figure B.1. Notice that the rotated vector $R_1\boldsymbol{e}_x$ lies in the xOz plane: this is guaranteed by (B.2), as $(R_{noyaw}\boldsymbol{e}_x)\cdot\boldsymbol{e}_y = (R_y(\beta)\boldsymbol{e}_x)\cdot\boldsymbol{e}_y = 0$. This can be seen however as an arbitrary choice: indeed, while vector $\boldsymbol{e}_z$ has a true meaning to define the tilt (physically, it represents gravity), the choice of the direction for $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ for the world frame is arbitrary, and does not influence the definition of the tilt: as such, it should not influence the decomposition we perform.

The observation that $\boldsymbol{e}_x$ remains in the xOz plane, whereas $\boldsymbol{e}_y$ does not remain in the xOy plane, shows that this is not the case: somehow, $\boldsymbol{e}_x$ is being privileged over $\boldsymbol{e}_y$: changing the direction of these vectors thus changes the rotation given by this method. Mathematically, this translates into the following: let $\psi \in \mathbb{R}$, and define frame $\mathcal{W}'$ by a rotation from the world frame of angle ψ about $\boldsymbol{e}_z$. $\mathcal{W}$ and $\mathcal{W}'$ therefore share the same vertical axis $\boldsymbol{e}_z$, but their horizontal axes differ. Instead of performing a roll-pitch-yaw decomposition in $\mathcal{W}$, we can do it in $\mathcal{W}'$: this is done by computing the roll-pitch-yaw decomposition of $R_z(-\psi)RR(\psi)$ (this express the change of basis of the endomorphism R).

Doing this for instance for $\psi = \frac{\pi}{2}$ (such that the X axis of $\mathcal{W}'$ is the

world's Y axis), we obtain a yaw angle of 0.87 rad. By contrast, the yaw angle in $\mathcal{W}$ is 0.6 rad. Yet both values represent a rotation about *the same axis*. This shows that the yaw angle (or in fact any angle of a Euler angle decomposition) is not intrinsic to the rotation matrix and the chosen axis, but also depends on the full choice of the basis.

Back to our example, if we perform perform the roll-pitch-yaw decomposition in $\mathcal{W}'$ then remove the yaw, we remove an angle of 0.87 rad instead of 0.6 rad, and thus obtain a different rotation matrix. Re-expressed in the world frame, we thus obtain $R'_{noyaw} = (0.4, -0.7, -0.27)$ instead of R_{noyaw}: this is shown in orange in Figure B.1. Notice that this time, $\boldsymbol{e}_y$ lies in the yOz plane. More generally, by choosing any value for ψ, we can get different rotations ranging from R_{noyaw} to R'_{noyaw}: all are candidates for version of R "without yaw", and there is *a priori* no reason to choose one over the other.

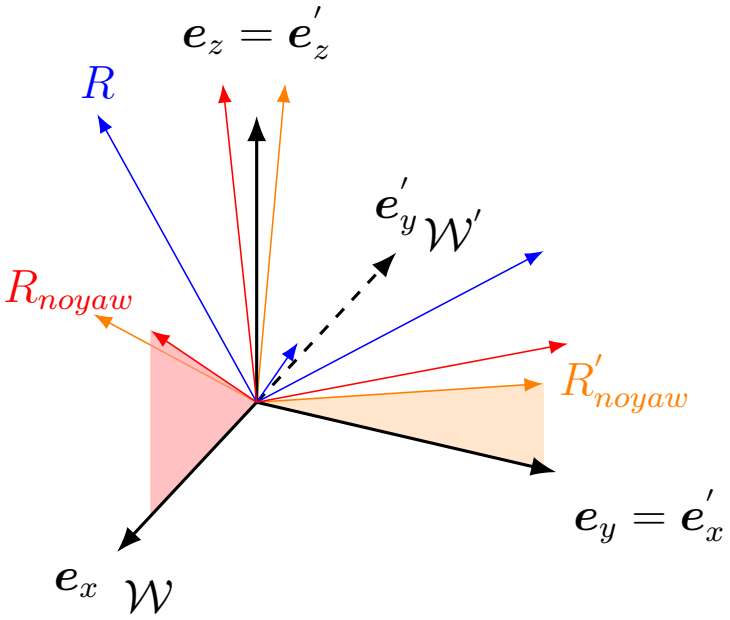


Figure B.1: Attempting to remove the yaw angle of a rotation R using roll-pitch-yaw decomposition: the rotation obtain depends on the definition of $\boldsymbol{e}_x$. The red frame shows the rotation obtained by computing Euler angles in $\mathcal{W}$, the orange frame in $\mathcal{W}'$: this gives two different candidate rotation for a version of R "without yaw".

In conclusion, this idea of canceling the yaw angle of a roll-pitch-yaw decomposition offers an ill-posed solution. It creates a bias toward the horizontal vector chosen to represent $\boldsymbol{e}_x$, though the notion of tilt is independent

of this vector. Furthermore, it offers a discontinuous solution about the singularity of the decomposition. Both of these issues can be solved by taking another route, no longer relying on Euler angles, as presented below.

B.2 A minimization-based approach: the twist-swing decomposition

To overcome the limitations of a Euler angle decomposition, we instead use another decomposition of rotation matrices, the so-called twist-swing decomposition [Baerlocher01]. This enables the reformulation of the problem of removing the unobservable component as a minimization problem for which a simple, closed-form solution exists.

We first restate a classical result to define a norm on $SO(3)$ [Park95, Huynh09]:

Definition 3. *For two non-zero vectors $\boldsymbol{a}$, $\boldsymbol{b}$, we write $\alpha(\boldsymbol{a}, \boldsymbol{b})$ the angle from $\boldsymbol{a}$ to $\boldsymbol{b}$.*

Let $R \in SO(3)$; there exists a unit vector $\boldsymbol{v}$ and a scalar θ such that R is the rotation around $\boldsymbol{v}$ of angle θ. We write $R = \mathcal{R}_0(\boldsymbol{v}, \theta)$[1]. This defines a norm on $SO(3)$ as: $||R|| = |\theta|$[2]. Furthermore, this norm is characterized by

$$||R|| = \max_{\{\boldsymbol{a} \in \mathbb{R}^3 \mid ||\boldsymbol{a}||=1\}} |\alpha(\boldsymbol{a},\ R\boldsymbol{a})| \tag{B.3}$$

Using this norm, the twist-swing decomposition can be defined as follows:

Theorem 3. *Any rotation matrix R can be decomposed as*

$$R = R_z R_s \tag{B.4}$$

where R_z (the twist) is a rotation around $\boldsymbol{e}_z$ and R_s (the swing) is the smallest (according to the norm (B.3)) rotation matrix such that $\boldsymbol{t}(R_s) = \boldsymbol{t}(R)$. In other words,

$$R_s = \underset{\{P \in SO(3) \mid \boldsymbol{t}(P)=\boldsymbol{t}(R)\}}{\arg\min} ||P|| \tag{B.5}$$

[1] $\mathcal{R}_0$ can be expressed using Rodrigues formula ($\mathcal{R}_0(\boldsymbol{v}, \theta) = \cos\theta I_3 + \sin\theta[\boldsymbol{v}]_\times + (1 - \cos\theta)\boldsymbol{v}\boldsymbol{v}^T$), or the matrix exponential ($\mathcal{R}_0(\boldsymbol{v}, \theta) = \exp([\theta\boldsymbol{v}]_\times)$).

[2] This norm corresponds to the classical Riemannian metric [Moakher02] $d(R_1, R_2) = ||\log(R_1 R_2^T)||$: indeed, taking the log of $\mathcal{R}_0(\boldsymbol{v}, \theta) = \exp([\theta\boldsymbol{v}]_\times)$ readily shows that $|\theta| = ||\log R||$.

If additionally $R^T \boldsymbol{e}_z \neq -\boldsymbol{e}_z$, this decomposition is unique, and is given by the following formula:

$$R_s = \begin{cases} \mathcal{R}_0\Big(\dfrac{\boldsymbol{e}_z \times \boldsymbol{t}(R)}{||\boldsymbol{e}_z \times \boldsymbol{t}(R)||}, \alpha\left(\boldsymbol{e}_z,\ \boldsymbol{t}(R)\right)\Big) & \text{if} \quad \boldsymbol{t}(R) \neq \boldsymbol{e}_z \\ I_3 & \text{if} \quad \boldsymbol{t}(R) = \boldsymbol{e}_z \end{cases} \tag{B.6}$$

Before giving the proof of this result, we first explain its significance.

The swing rotation given by (B.6) represents the rotation R with nullified non-observable component that we seek. Notice that this rotation can be computed from knowledge of $\boldsymbol{t}$ only, and does not need an intermediate computation of a rotation matrix R: from a tilt measurement, it directly gives the rotation with zero "yaw rotation".

Notice that (B.6) only depends on $\boldsymbol{t}$ and $\boldsymbol{e}_z$: this guarantees that the obtained rotation R_s is independent of the choice of vectors $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$, therefore solving the ambiguity encountered when using the roll-pitch-yaw decomposition. This rotation is now really intrinsic to the tilt being observed. Note also that this formula is continuous (as a composition of continuous functions, and because $\alpha\left(\boldsymbol{e}_z,\ \boldsymbol{t}(R)\right)$ converges to zero as $\boldsymbol{t}(R)$ converges to $\boldsymbol{e}_z$) on its domain of definition, i.e. the unit sphere minus the south pole, $-\boldsymbol{e}_z$. This is a significant improvement over the roll-pitch-yaw decomposition, which is singular over the whole equator (i.e. a rotation of $\frac{\pi}{2}$ in $\boldsymbol{e}_y$). The remaining singularity (this function has no limit as $\boldsymbol{t}(R)$ converges to $-\boldsymbol{e}_z$) is now truly intrinsic to the quantity being observed: when the tilt is equal to $-\boldsymbol{e}_z$, i.e. the IMU is upside down, there is no way to tell in which direction the IMU has rotated. This can also be understood by saying that, on a sphere, there exists a unique geodesic (i.e. unique minimal rotation), except when going from one pole to the other.

Proof of Theorem 3. Let $R \in SO(3)$ be an arbitrary rotation matrix.

We define $\theta \triangleq |\alpha\left(\boldsymbol{t}(R),\ \boldsymbol{e}_z\right)|$ and $\mathcal{A} \triangleq \{P \in SO(3) \ \mid \ \boldsymbol{t}(P) = \boldsymbol{t}(R)\}$ the set of all rotation matrices with the same tilt than R.

For any $P \in \mathcal{A}$, we define $R_z \triangleq RP^T$. Then as

$$\boldsymbol{t}(P) = \boldsymbol{t}(R) \iff P^T\boldsymbol{e}_z = R^T\boldsymbol{e}_z \iff \boldsymbol{e}_z = R_z^T\boldsymbol{e}_z$$

R_z is indeed a rotation of axis $\boldsymbol{e}_z$, yielding a decomposition of R according to (B.4). Thus, we only need to show that $\mathcal{A}$ has a unique minimum given by (B.6). Using (B.3) on $\boldsymbol{t}(R)$ yields

$$||P|| \geq |\alpha\left(\boldsymbol{t}(R),\ P\boldsymbol{t}(R)\right)| = |\alpha\left(\boldsymbol{t}(R),\ P\boldsymbol{t}(P)\right)| = |\alpha\left(\boldsymbol{t}(R),\ \boldsymbol{e}_z\right)| = \theta \tag{B.7}$$

APPENDIX B. RECONSTRUCTING A ROTATION MATRIX FROM A TILT ESTIMATE

If $R^T\boldsymbol{e}_z = -\boldsymbol{e}_z$, any rotation of angle π around a unit vector normal to $\boldsymbol{e}_z$ is a minimum over $\mathcal{A}$: thus the proposed decomposition exists but is not unique.

If $R^T\boldsymbol{e}_z = \boldsymbol{e}_z$, the identity matrix is trivially the only minimum over $\mathcal{A}$.

Finally, we consider the case where $R^T\boldsymbol{e}_z$ is not collinear to $\boldsymbol{e}_z$. Let R_s be an element of $\mathcal{A}$ of norm θ. We call $\boldsymbol{v}$ its axis of rotation, and use the property (proven in [Huynh09]) that, for any nonzero vector $\boldsymbol{a}$,

$$|\alpha\left(\boldsymbol{a},\ R_s\boldsymbol{a}\right)| = ||R_s|| \iff \boldsymbol{a} \text{ is orthogonal to } \boldsymbol{v} \tag{B.8}$$

Consequently, as

$$|\alpha\left(\boldsymbol{t}(R_s),\ R_s\boldsymbol{t}(R_s)\right)| = |\alpha\left(R_s^T\boldsymbol{e}_z,\ \boldsymbol{e}_z\right)| = ||R_s|| \tag{B.9}$$

$$|\alpha\left(\boldsymbol{e}_z,\ R_s\boldsymbol{e}_z\right)| = |\alpha\left(R_s^T\boldsymbol{e}_z,\ \boldsymbol{e}_z\right)| = ||R_s|| \tag{B.10}$$

$\boldsymbol{v}$ is orthogonal to both $\boldsymbol{e}_z$ and $\boldsymbol{t}(R_s)$. As these vectors are not collinear, this implies that $\boldsymbol{v} \in span(\boldsymbol{e}_z \times \boldsymbol{t}(R_s))$. Since $\boldsymbol{v}$ is a unit vector, there are only two possible choices

- $\boldsymbol{v}_1 = \frac{\boldsymbol{e}_z \times R^T\boldsymbol{e}_z}{||\boldsymbol{e}_z \times R^T\boldsymbol{e}_z||}$, in which case $R_s = \mathcal{R}_0(\boldsymbol{v}_1, \theta)$ to verify $R_s^T\boldsymbol{e}_z = \boldsymbol{t}(R)$.
- $\boldsymbol{v}_2 = -\frac{\boldsymbol{e}_z \times R^T\boldsymbol{e}_z}{||\boldsymbol{e}_z \times R^T\boldsymbol{e}_z||}$, in which case $R_s = \mathcal{R}_0(\boldsymbol{v}_2, -\theta)$ to verify $R_s^T\boldsymbol{e}_z = \boldsymbol{t}(R)$.

As $\boldsymbol{v}_2 = -\boldsymbol{v}_1$ and $\mathcal{R}_0(\boldsymbol{v}, \theta) = \mathcal{R}_0(-\boldsymbol{v}, -\theta)$, this analysis yields a unique candidate for the minimum, which is indeed the value given in (B.6). □

Remark. *It should be noted that, for the case of small rotations, the proposed twist-swing decomposition method, or the erroneous roll-pitch-yaw formulation are in fact equivalent. Indeed, R_s computed in* (B.6) *has a vector of rotation $\boldsymbol{v} = \boldsymbol{e}_z \times \boldsymbol{t}(R)$. This vector is orthogonal to $\boldsymbol{e}_z$, and thus has a zero vertical component.*

However, for small rotations, the roll-pitch-yaw angles are equal (at the first order) to the coordinates of the vector of rotation. This means that R_s has a zero yaw angle (according to the roll-pitch-yaw decomposition), and is thus, in this context, is equivalent to the solution computed by (B.2)*. This justifies the denomination of R_s as the rotation "without yaw", though the yaw is not defined in the twist-swing decomposition.*

Appendix C

Proof of observability for the *DynamicObserver*

In this appendix, we prove the observability of the state $\boldsymbol{x}$ defined in Section 3.4 under Assumption 2 - 4. Since the angular velocities can obviously be determined from the rotation matrices once they are known, the challenge is to prove that the deformations D_i are observable - or equivalently, that the vector of rotation $\boldsymbol{\Omega}_i$ is observable. For this purpose, we show that this vector can be reconstructed from $\boldsymbol{u}$, the sensor signal $\boldsymbol{y}$ and their derivative.

In this proof, we say that a given quantity is *measured* when it can be expressed as a function of $\boldsymbol{u}$, $\boldsymbol{y}$ and their derivative.

To explicitly express the dynamics of the system, we use the Newton-Euler formulation instead of the Lagrange formalism of (3.51). For this purpose, we use the notion of "rigid sets" defined in Section 3.1.2: namely, set S_i is composed of all bodies between the ith and $i+1$th flexibility (respectively the end of the kinematic chain for the final deformation). The relative configuration of the bodies inside set S_i is determined by encoder measurements $\boldsymbol{\theta}$ only: thus, the position, velocity and acceleration of any point of the set S_i with respect to the parent frame $\mathcal{C}_i$ is measured.

For $i \in [1, n]$, where n is the number of deformations, we call $\boldsymbol{f}_i$ and $\boldsymbol{\tau}_i$ the force exerted by set S_i on set S_{i-1} in frame $\mathcal{C}_i$. We recall that $\boldsymbol{\tau}_i$, the corresponding torque, is given by (3.50). By convention, we define $\boldsymbol{f}_{n+1} = 0$ and $\boldsymbol{\tau}_{n+1} = 0$ as no external force is applied to the last set[1].

Consider now set S_i. We call G_i its center of mass, and m_i the total mass of the set. This set is subject to three external actions: gravity, the interaction with S_{i-1} and the interaction with S_{i+1}. This is shown in Figure C.1.

[1]Note that the proof of observability would still hold if a known external force is applied (for instance, in the context of a robot manipulator arm with a force / torque sensor at the tip).

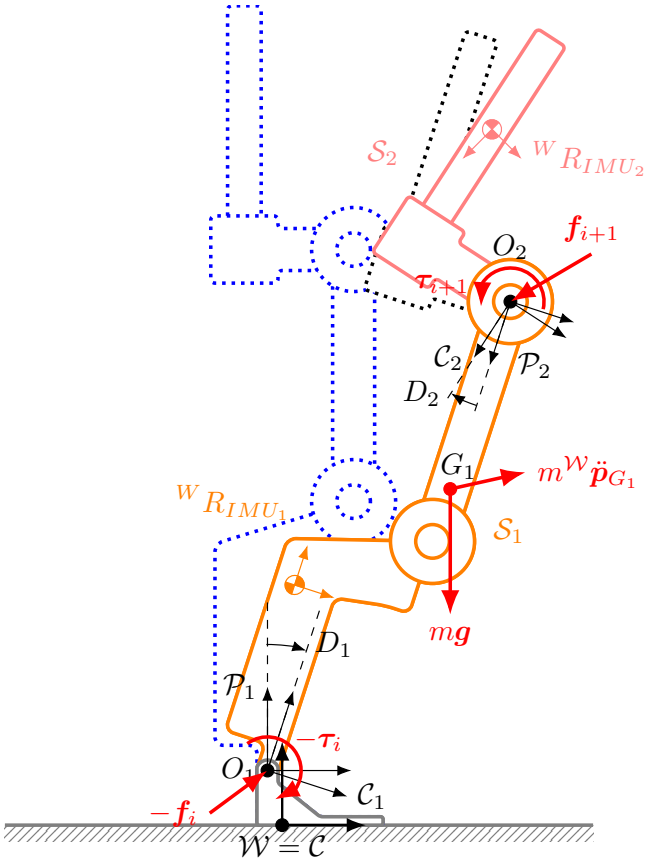


Figure C.1: Illustration of the system dynamics. The dynamics of set S_1 is the result of four actions: its acceleration, gravity, and the wrench from the previous $(-\boldsymbol{f}_i, -\boldsymbol{\tau}_i)$ and next $(\boldsymbol{f}_{i+1}, \boldsymbol{\tau}_{i+1})$ set.

Applying Newton's law of motion to this set thus yields

$$m_i {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{G_i} = m_i \boldsymbol{g} + {}^{\mathcal{W}}R_{\mathcal{C}_i} \boldsymbol{f}_i - {}^{\mathcal{W}}R_{\mathcal{C}_{i+1}} \boldsymbol{f}_{i+1} \tag{C.1}$$

Writing (C.1) for $i = 1...n$ and solving for the forces yields the expression of each force $\boldsymbol{f}_i$ as:

$$\boldsymbol{f}_i = {}^{\mathcal{W}}R_{\mathcal{C}_i}{}^{T} \sum_{k=i}^{n} ({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{G_k} - \boldsymbol{g}) \tag{C.2}$$

The remarkable property of $\boldsymbol{f}_i$ is that it can be expressed only in terms of sensor data, the input $\boldsymbol{u}$, and the deformation $D_j, j \geq i$. Namely, we define $Z_i \triangleq (\boldsymbol{y}_a, \boldsymbol{y}_g, \dot{\boldsymbol{y}}_g, \boldsymbol{u}, D_i, ...D_n)$ and show that $\boldsymbol{f}_i$ depends only on Z_i.

For that purpose, we use the following lemma:

Lemma 1. *Let A be a frame attached to a body of set S_i. The quantity*

$$\phi(A) \triangleq {}^{\mathcal{W}}R_{\mathcal{C}_i}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_A - \boldsymbol{g}) \tag{C.3}$$

is a function of $\boldsymbol{u}$, $\boldsymbol{y}_a$ and $\boldsymbol{y}_g$ and their derivatives - and is thus measured.

Remark. *$\phi(A)$ is the specific acceleration of frame A, expressed in frame $\mathcal{C}_i$.*

Proof. Since IMU i belongs to set S_i, the rotation ${}^{IMU_i}R_{\mathcal{C}_i}$ is measured (as a function of $\boldsymbol{\theta}$ only). The accelerometer of IMU_i measures

$$\boldsymbol{y}_{a_i} = {}^{\mathcal{W}}R_{IMU_i}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU} - \boldsymbol{g}) \tag{C.4}$$

Thus $\phi(A)$ rewrites

$$\phi(A) = {}^{IMU_i}R_{\mathcal{C}_i}^T\boldsymbol{y}_{a_i} + {}^{IMU_i}R_{\mathcal{C}_i}^T\,{}^{\mathcal{W}}R_{IMU_i}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_A - {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU}) \tag{C.5}$$

Consequently, since ${}^{IMU_i}R_{\mathcal{C}_i}^T\boldsymbol{y}_{a_i}$ is measured, we just need to show that ${}^{\mathcal{W}}R_{IMU_i}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_A - {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU})$ is measured.

For this purpose, we decompose the position of point A about the frame IMU_i as

$${}^{\mathcal{W}}\boldsymbol{p}_A = {}^{\mathcal{W}}\boldsymbol{p}_{IMU_i} + {}^{W}R_{IMU_i}\,{}^{IMU_i}\boldsymbol{p}_A \tag{C.6}$$

Since A and IMU_i belong to the same set, the position ${}^{IMU_i}\boldsymbol{p}_A$ is a function of $\boldsymbol{\theta}$ only. The first derivative of this equation gives

$${}^{\mathcal{W}}\dot{\boldsymbol{p}}_A - {}^{\mathcal{W}}\dot{\boldsymbol{p}}_{IMU} = {}^{W}R_{IMU_i}[\boldsymbol{y}_{g_i}]_\times\,{}^{IMU_i}\boldsymbol{p}_A + {}^{W}R_{IMU_i}\,{}^{IMU_i}\dot{\boldsymbol{p}}_A \tag{C.7}$$

Differentiating a second time, we obtain

$$\begin{aligned} {}^{W}R_{IMU_i}^T({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_A - {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{IMU}) = \;& [\dot{\boldsymbol{y}}_{g_i}]_\times\,{}^{IMU_i}\boldsymbol{p}_A \\ & + [\boldsymbol{y}_{g_i}]_\times^2\,{}^{IMU_i}\boldsymbol{p}_A \\ & + 2[\boldsymbol{y}_{g_i}]_\times\,{}^{IMU_i}\dot{\boldsymbol{p}}_A \\ & + {}^{IMU_i}\ddot{\boldsymbol{p}}_A \end{aligned} \tag{C.8}$$

All the terms on the right hand side are functions of $\boldsymbol{u}$, $\boldsymbol{y}_{g_i}$ and $\dot{\boldsymbol{y}}_{g_i}$ only - and thus, so is $\phi(A)$ □

By decomposing the rotations ${}^{\mathcal{W}}R_{\mathcal{C}_i}$ about frame $\mathcal{C}_k$ in (C.2), this equation yields

$$\boldsymbol{f}_i = \sum_{k=i}^{n} {}^{\mathcal{C}_k}R_{\mathcal{C}_i}{}^{T}\,{}^{\mathcal{W}}R_{\mathcal{C}_k}{}^{T}({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{G_k} - \boldsymbol{g}) \tag{C.9}$$

This enables us to use Lemma 1 on each term ${}^{\mathcal{W}}R_{\mathcal{C}_k}{}^{T}({}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{G_k} - \boldsymbol{g})$ in the sum. Meanwhile, ${}^{\mathcal{C}_k}R_{\mathcal{C}_i}$ can be written as a successive product of rotation

through set S_j, ${}^{\mathcal{P}_j}R_{\mathcal{C}_j}$, function of $\boldsymbol{\theta}$ only, and the jth deformation D_j. Thus, it is indeed a function of Z_i only. We write

$$\boldsymbol{f}_i = \Phi(\boldsymbol{Z}_i) \tag{C.10}$$

The same idea is now applied to the angular momentum, using Euler's law of motion applied at point O_i. For this purpose, we consider a body B in S_i. We call $\boldsymbol{L}(B)$ its angular moment with respect to point O_i, expressed in $\mathcal{C}_i$. This definition yields the following expression

$$\boldsymbol{L}(B) \triangleq {}^{\mathcal{C}_i}R_B I_B \boldsymbol{\omega}_B + {}^{\mathcal{C}_i}\boldsymbol{p}_B \times (m_B\, {}^{W}R_{\mathcal{C}_i}^T({}^{W}\dot{\boldsymbol{p}}_B - {}^{W}\dot{\boldsymbol{p}}_{O_i})) \tag{C.11}$$

where m_B is the mass of body B, and I_B its inertia with respect to its center of mass, in the body frame.

The angular velocity $\boldsymbol{\omega}_B$ can be expressed in terms of the gyroscope measurement, using (3.23). Furthermore, decomposing the position of point B as ${}^{W}\boldsymbol{p}_B = {}^{W}\boldsymbol{p}_{O_i} + {}^{W}R_{\mathcal{C}_i}\,{}^{\mathcal{C}_i}\boldsymbol{p}_B$, we obtain

$${}^{W}R_{\mathcal{C}_i}^T({}^{W}\dot{\boldsymbol{p}}_B - {}^{W}\dot{\boldsymbol{p}}_{O_i}) = [\boldsymbol{\omega}_{\mathcal{C}_i}]_\times\, {}^{\mathcal{C}_i}\dot{\boldsymbol{p}}_B + {}^{\mathcal{C}_i}\boldsymbol{p}_B \tag{C.12}$$

Using once again (3.23) shows that $\boldsymbol{\omega}_{\mathcal{C}_i}$ is measured - and thus ${}^{W}R_{\mathcal{C}_i}^T({}^{W}\dot{\boldsymbol{p}}_B - {}^{W}\dot{\boldsymbol{p}}_{O_i})$ is measured.

Hence, the angular momentum $\boldsymbol{L}(B)$ is measured. The total angular moment of set S_i, $\boldsymbol{L}(S_i)$, is simply the sum of the individual angular momentum of each body, and is also measured.

We now apply Euler's law of motion at point O_i to the set S_i, along the world frame axes. As O_i is not fixed in $\mathcal{W}$, we add the torque due to the inertia forces related to the translational motion of O_i with respect to the world. This writes

$$\begin{aligned}\frac{d}{dt}\Big({}^{W}R_{\mathcal{C}_i}\boldsymbol{L}(S_i)\Big) =& m_i[{}^{W}R_{\mathcal{C}_i}\,{}^{\mathcal{C}_i}\boldsymbol{p}_{G_i}]_\times(\boldsymbol{g} - {}^{W}\ddot{\boldsymbol{p}}_{O_i}) \\ &+ [{}^{\mathcal{W}}R_{\mathcal{C}_i}\,{}^{\mathcal{C}_i}\boldsymbol{p}_{O_{i+1}}]_\times\, {}^{\mathcal{W}}R_{\mathcal{C}_i}\boldsymbol{f}_{i+1} \\ &+ {}^{W}R_{\mathcal{C}_i}\boldsymbol{\tau}_i \\ &- {}^{W}R_{\mathcal{C}_{i+1}}\boldsymbol{\tau}_{i+1}\end{aligned} \tag{C.13}$$

This rewrites as

$$\begin{aligned}\boldsymbol{\tau}_i =& {}^{\mathcal{C}_i}R_{\mathcal{C}_{i+1}}\boldsymbol{\tau}_{i+1} \\ &+ [\boldsymbol{\omega}_{\mathcal{C}_i}] \times \boldsymbol{L}(S_i) + \frac{d}{dt}\Big(\boldsymbol{L}(S_i)\Big) \\ &- [{}^{\mathcal{C}_i}\boldsymbol{p}_{O_{i+1}}]_\times \boldsymbol{f}_{i+1} \\ &+ m_i[{}^{\mathcal{C}_i}\boldsymbol{p}_{G_i}]_\times\, {}^{W}R_{\mathcal{C}_i}^T({}^{W}\ddot{\boldsymbol{p}}_{O_i} - \boldsymbol{g})\end{aligned} \tag{C.14}$$

The remarkable property is that the right hand side of this equation depends on Z_{i+1} only - i.e. only the deformations downstream of S_i. Indeed, taking each line one by one:

- ${}^{\mathcal{C}_i}R_{\mathcal{C}_{i+1}}\boldsymbol{\tau}_{i+1}$ is a function of $\boldsymbol{u}$ and $\boldsymbol{\Omega}_{i+1}$ according to (3.50)
- $[\boldsymbol{\omega}_{\mathcal{C}_i}] \times \boldsymbol{L}(S_i) + \frac{d}{dt}\Big(\boldsymbol{L}(S_i)\Big)$ is composed only of measured terms
- ${}^{\mathcal{C}_i}\boldsymbol{p}_{O_{i+1}}$ is measured, while $\boldsymbol{f}_{i+1} = \phi(Z_{i+1})$ according to (C.10)
- For the last term, we use Lemma 1 on point O_i to obtain the desired result.

Hence, (C.14) rewrites as

$$\boldsymbol{\tau}_i = \Psi(Z_{i+1}) \tag{C.15}$$

Finally, using (3.50) and the fact that the stiffness matrix K_i is positive definite, we obtain

$$\boldsymbol{\Omega}_i = -K_i^{-1}\Psi(Z_{i+1}) \tag{C.16}$$

The proof then follows by induction:

- For the last set, $Z_{n+1} = (\boldsymbol{u}, \boldsymbol{y}, \dot{\boldsymbol{y}})$ is measured, hence $\boldsymbol{\Omega}_n$ is measured.
- For a set $i < n$, we assume that $\boldsymbol{\Omega}_j, j > i$ are all measured. Then Z_{i+1} is measured, and thus according to (C.16), $\boldsymbol{\Omega}_i$ is measured.

This ends the proof.

Appendix D

Impact of the presence of a second deformation on a feedback controller

When tuning the gains of the low-level controller in Section 4.2.3, we had to deal with the presence of strong vibrations at around $15 - 20$ Hz, that could not be explained by the joint elasticity model. The presence of a time delay in the system, the coupling between the various joints of the robot, or the impact of the *KinematicObserver*, were studied in simulation[1], but these effects alone could not reproduce the phenomenon.

Instead, we believe that these vibrations are caused by the fact that the deformations are not really punctual, but distributed about several points of the structure. This hypothesis is corroborated by the following study.

D.1 Presentation of the model

Modeling the deformation as a simple joint elasticity leads to the series-elastic actuator model, presented in Section 4.1.2.1 and used for gain tuning. This model is represented graphically on the left in Figure D.1. Namely, it considers that a single rigid link (in practice, the aggregation of several links of the robot) is attached to the motor through a spring-damper. In this section, we call this the single deformation model.

To account for the presence of more than one flexible element, we split the link into two parts, about a point located at a distance d from the motor's

[1]For wholebody simulation, we used the open-source simulator *Jiminy* [Duburcq20], which relies on the dynamics library *Pinocchio*[Carpentier19]. This simulator integrates a model of the flexibilities as spherical springs, as presented in (2.3)

axis. The two halves are connected by a spring-damper. We call this model the double deformations model.

The state of the system in the double deformations model is now characterized by three angles: the motor angle theta, θ, the deformation of the motor flexibility α_1 and that of the link flexibility α_2. This means that the order of the system increases, from 4 to 6 degrees of freedom. Our current instrumentation however only gives access to two angles (and the two associated flexibilities): θ, the motor angle, through the encoder, and $q \triangleq \theta + \alpha_1 + \alpha_2$, the angle of the IMU located at the tip of the second link element. Note therefore that the angle α_1 cannot be directly estimated kinematically[2]: an additional IMU located on the first half of the link would be necessary for that purpose.

The dynamics of the single deformation model is given by (4.7). Meanwhile, the dynamics of the double deformations, linearized about a reference position q_0, writes

$$\begin{aligned} M \begin{pmatrix} \ddot{\theta} \\ \ddot{\alpha_1} \\ \ddot{\alpha_2} \end{pmatrix} &= \begin{pmatrix} m_1 g l_1(\theta + \alpha_1) + m_2 g(d(\theta + \alpha_1) + l_2 q) \\ m_1 g l_1(\theta + \alpha_1) + m_2 g(d(\theta + \alpha_1) + l_2 q) \\ m_2 g l_2(\theta + \alpha_1 + \alpha_2) \end{pmatrix} \cos q_0 \\ &+ \begin{pmatrix} u - \nu_m \dot{\theta} \\ -k_1 \alpha_1 - \nu_1 \dot{\alpha_1} \\ -k_2 \alpha_2 - \nu_2 \dot{\alpha_2} \end{pmatrix} \end{aligned} \tag{D.1}$$

with

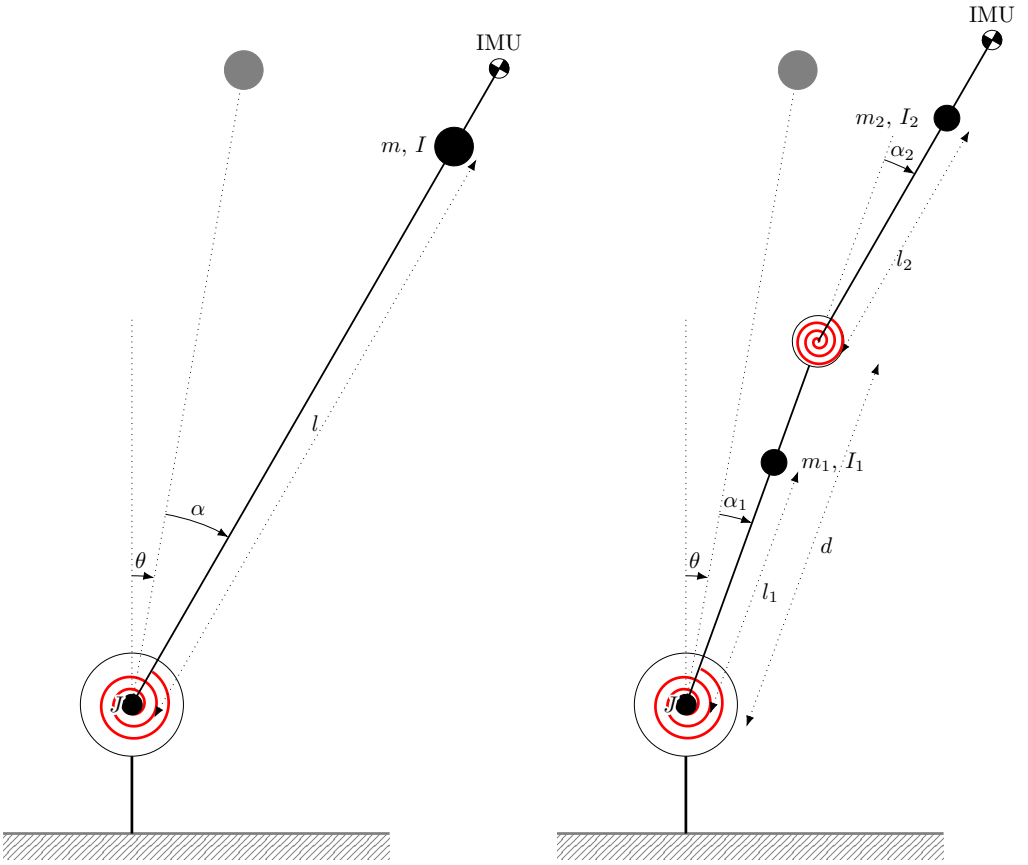
$$M \triangleq \begin{pmatrix} I + J & I & I_2 + m_2 l_2 d \\ I & I & I_2 + m_2 l_2 d \\ I_2 + m_2 l_2 d & I_2 + m_2 l_2 d & I_2 \end{pmatrix}$$

where I_1 and I_2 is the inertia of both half of the link about the previous joint, and $I = I_1 + I_2 + m_2 d^2 + 2 m_2 d l_2$ is the apparent inertia of the link.

D.2 Open-loop identification on Atalante

The difference between the single deformation model and the double deformations model only become visible in dynamic, when high-frequency excitation are present. Indeed, kinematically speaking, the models are equivalent, since

[2]The dynamical modeling of the flexibility as a spring-damper theoretically renders this state observable. But in practice the uncertain nature of this model is likely to greatly limit the accuracy of the reconstruction, as observed on the single deformation model for the *DynamicObserver* in Section 3.6.1.



(a) Single deformation model (b) double deformations model

Figure D.1: Representation of both models: the double deformations model splits the link in half, adding a second spring damper.

both the length of the segment and the angles of deformation are small. This is why such deformation may go unnoticed when looking only at the kinematic accuracy of the various models - in particular, this is how we designed the extended flexible model, considering only one spherical joint at the sagittal hips to represent potential deformations about the three hips and the knee.

Likewise, the static identification experiment performed in 4.1.2.2 does not distinguish between both models. Rather, in the case of the double deformations model, it gives access to the equivalent stiffness obtained by placing two springs in series, i.e. $\frac{k_1 k_2}{k_1+k_2}$.

In order to test the hypothetical presence of a second deformation, we performed a more advanced identification experiment, to obtain the open-loop transfer from the input to the sensor's reading. This identification was

carried out in the frontal plane, with the objective of seeing if the double deformations model gives a better match of the transfer from the frontal hip motor to the tibia IMU, i.e. the point where the deformations are measured. During this experiment, the exoskeleton's pelvis is rigidly attached to a wall to prevent it from moving. The frontal hip motor is then fed a sinusoidal torque input, at increasing frequencies. The motors of the other leg are deactivated; the other motors of the identified leg are kept active using a PID controller, to prevent erratic motion of the leg in the other planes.

In Figure D.2, the dots show the obtained Bode diagram for the open-loop transfer from u to θ (encoder measurements) and from u to $\dot{q}$ (gyroscope measurements). On top of this data, we plot the transfer function for "nominal" model values:

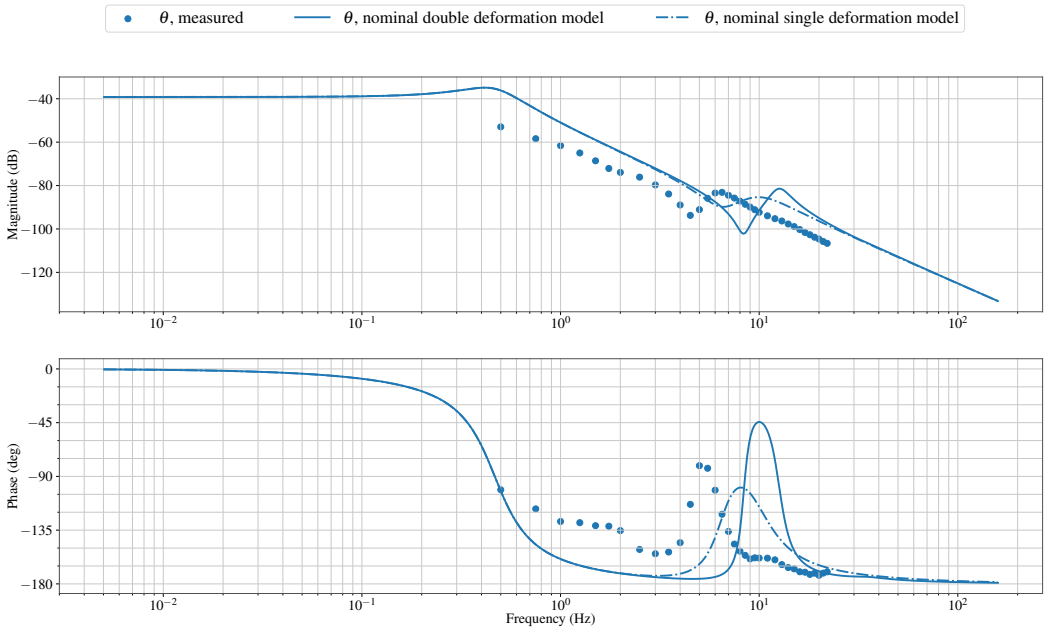
- For the single deformation model, we use the parameters used for LQR tuning as detailed in 4.1.2.2 - most importantly, we use the stiffness k identified in the described static experiment.

- For the double deformations model, we locate the second deformation at the robot's knee, a likely guess for the presence of a small deformation. We arbitrarily assign the same stiffness value, i.e. a value of $2k$ to the motor and link stiffness, so as to obtain the same equivalent stiffness in static, and use the same damping, $\frac{\nu_f}{2}$ for both flexible joints.

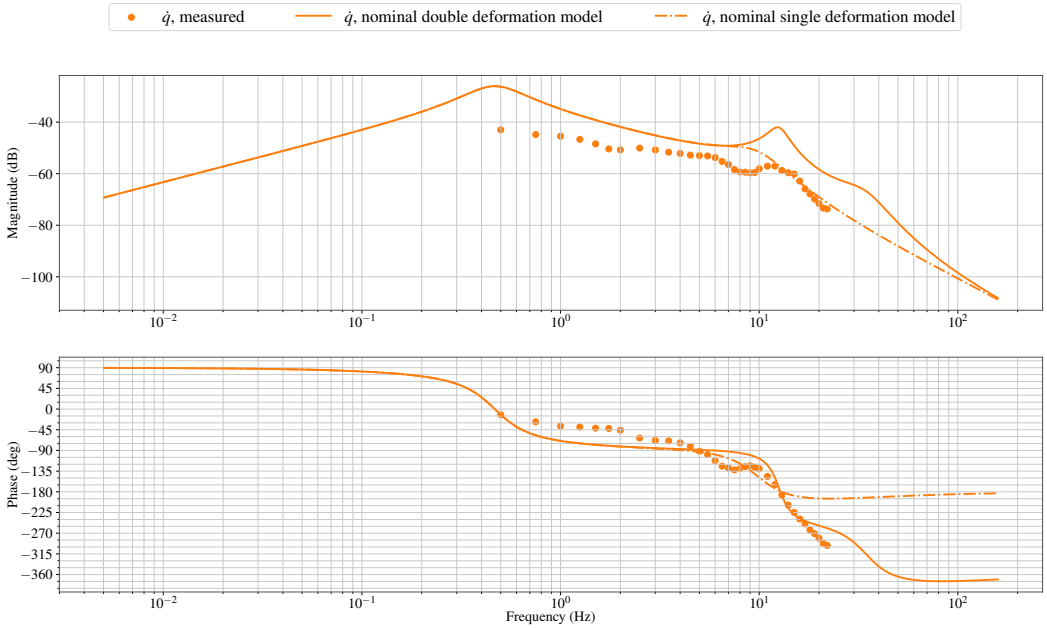
While the nominal models used do not accurately fit the data, they enable for a qualitative comparison between both models. As expected, at low frequencies (under 10 Hz), the single deformation model and double deformations model are equivalent. They however start to differ as the frequency increases: notably, the single deformation model exhibits a single resonance frequency, across which the phase of $\dot{q}$, the gyroscope, drops from $-90°$ to $-180°$. By comparison, the double deformations model exhibits two successive peaks in amplitude, and leads to a increase in phase of $-360°$ for $\dot{q}$.

This change in phase is likely to have a drastic impact on the stability of a controller. Indeed, a controller designed for the single deformation model expects a phase shift of $-180°$ on $\dot{q}$, whereas the actual shift will be $-360°$: in other words, for this frequency range, the sign of the gyroscope signal is not what we expect it to be ! Note that by contrast, the encoder signal θ is barely affected by the presence of a second deformation.

The experimental data (the dots in Figure D.2) qualitatively shows the features of a double deformations model: two peaks in the amplitude of the transfer toward $\dot{q}$, but most importantly a decrease in phase below $-180°$ between 10 Hz and 20 Hz. This behavior cannot be explained by a model



(a) Transfer from u to the encoder signal.



(b) Transfer from u to the gyroscope signal.

Figure D.2: Experimental determination of the open-loop transfer response of the system, and bode plot of the nominal model.

with only one deformation, and therefore indicates a more complex behavior of the system, for which the presence of a second deformation is a plausible explanation.

While it is possible to over-fit this data with a 6th order model, the resulting parameters are nonphysical. We believe this is due to the complexity of the system (which may be more complex than simply two linear springs), with relatively poorly controlled boundary conditions. Indeed, the exoskeleton's back was not perfectly constrained to remain fixed (as the identification experiment needed to be non-destructive). The fact that other motors were active in the system might also play a role in modifying the overall response by introducing or dissipating energy. Motor current is used in place of a true torque measurement, with associated uncertainty on the value of the torque constant. Finally, the range of available measurements proved to be limited in frequency, between 0.5 Hz and 25 Hz. Lower frequency points could no be acquired due to the increase in amplitude of the motion, while high-frequency points proved to be erratic and unrepeatable from one experiment to the next, in particular for the gyroscope data. This is likely due to the strong attenuation of the signal (-75 dB at 20 Hz) which leads to a very poor signal-to-noise ratio

In the end, from the experimental data in Figure D.2, we draw the conclusion that the system does not behave simply like a series-elastic actuator: at frequencies above 10 Hz, the gyroscope phase starts to significantly differ from this model. We see this as the explanation for the presence of vibrations in the $15 - 20$ Hz region, when applying the gains computed by a LQR on a one-deformation model. This is further confirmed by the fact that, when looking at the various terms used in feedback, these frequencies are almost only present on $\dot{q}$, but negligible on the other states.

To remove these vibrations, simply filtering the gyroscope above 10 Hz (with a low-pass or band-stop filter) does not work: the induced phase shift of the filter leads to more vibrations at different frequencies. Instead, we took the most simple yet drastic solution: to completely remove the gyroscope signal, at all frequencies. More precisely, we reduce the gain provided by the LQR, sometimes quite drastically (up to a factor 10), until these vibrations were no longer present. While this gives a satisfactory controller in particular for trajectory tracking, reducing this gains leads to a poorer low-frequency behavior: namely, disturbance rejection at 1 Hz is significantly reduced.

We believe that finding a better method to handle this discrepancy between our model, and the actual response of the system, is a topic of valuable research, with the objective of finding a unified controller than can perform both accurate trajectory tracking and efficient disturbance rejection.

Bibliography

[Abele11] E. Abele, J. Bauer, T. Hemker, R. Laurischkat, H. Meier, S. Reese, and O. von Stryk. Comparison and validation of implementations of a flexible joint multibody dynamics system model for an industrial robot. *CIRP Journal of Manufacturing Science and Technology*, 4(1):38–43, 2011.

[Albu-Schaffer00] A. Albu-Schaffer and G. Hirzinger. State feedback controller for flexible joint robots: a globally stable approach implemented on DLR's light-weight robots. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, volume 2, pages 1087–1093, Takamatsu, Japan, 2000. IEEE.

[Axelsson12] Patrik Axelsson, Rickard Karlsson, and Mikael Norrlöf. Bayesian state estimation of a flexible industrial robot. *Control Engineering Practice*, 20(11):1220–1228, 2012.

[Baerlocher01] P. Baerlocher and R. Boulic. Parametrization and Range of Motion of the Ball-and-Socket Joint. In *Deformable Avatars: IFIP TC5/WG5.10 DEFORM'2000 Workshop November 29–30, 2000 Geneva, Switzerland*, pages 180–190. Springer US, Boston, MA, 2001.

[Barrau17] Axel Barrau. *Non-linear state error based extended Kalman filters with applications to navigation*. PhD Thesis, Mines Paristech, 2017.

[Barron94] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[Bauchau11] O. A. Bauchau. *Flexible Multibody Dynamics*, volume 176 of *Solid Mechanics and Its Applications*. Springer Netherlands, Dordrecht, 2011.

[Benallegue14] 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*, pages 246–251, November 2014. ISSN: 2164-0572, 2164-0580.

[Benallegue15] M Benallegue and F Lamiraux. Estimation and Stabilization of Humanoid Flexibility Deformation Using Only Inertial Measurement Units and Contact Information. *International Journal of Humanoid Robotics*, 12(3), September 2015.

[Benallegue17] M. Benallegue, A. Benallegue, and Y. Chitour. Tilt estimator for 3D non-rigid pendulum based on a tri-axial accelerometer and gyrometer. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 830–835, November 2017.

[Benallegue20] Mehdi Benallegue, Rafael Cisneros, Abdelaziz Benallegue, Yacine Chitour, Mitsuharu Morisawa, and Fumio Kanehiro. Lyapunov-Stable Orientation Estimator for Humanoid Robots. *IEEE Robotics and Automation Letters*, 5(4):6371–6378, October 2020.

[Benosman04] M. Benosman and G. Le Vey. Control of flexible manipulators: A survey. *Robotica*, 22(5):533–545, September 2004.

[Besançon07] G. Besançon. *Nonlinear Observers and Applications*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin Heidelberg, 2007.

[Bloesch12] Michael Bloesch, Marco Hutter, Mark Hoepflinger, Stefan Leutenegger, Christian Gehring, C. David Remy, and Roland Siegwart. State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[Bonnabel09] Silvère Bonnabel, Philippe Martin, and Erwan Salaün. Invariant Extended Kalman Filter: theory and application

to a velocity-aided attitude estimation problem. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 1297–1304, 2009.

[Buondunno18] Gabriele Buondunno. *Numerical Solutions for Design and Dynamic Control of Compliant Robots*. PhD thesis, Sapienza Universita Di Roma, 2018.

[Caron19] Stéphane Caron, Abderrahmane Kheddar, and Olivier Tempier. Stair Climbing Stabilization of the HRP-4 Humanoid Robot using Whole-body Admittance Control. In *IEEE International Conference on Robotics and Automation*, Montréal, France, May 2019.

[Caron20] Stéphane Caron, Adrien Escande, Leonardo Lanari, and Bastien Mallein. Capturability-Based Pattern Generation for Walking With Variable Height. *IEEE Transactions on Robotics*, 36(2):517–536, 2020.

[Carpentier18] Justin Carpentier and Nicolas Mansard. Multicontact Locomotion of Legged Robots. *IEEE Transactions on Robotics*, 34(6):1441–1460, 2018.

[Carpentier19] 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 *SII 2019 - International Symposium on System Integrations*, Paris, France, January 2019.

[Carrera11] Erasmo Carrera, Gaetano Giunta, and Marco Petrolo. *Beam structures: classical and advanced theories*. Wiley, Chichester, West Sussex, U.K, 2011.

[Chung08] Wankyun Chung, Li-Chen Fu, and Su-Hau Hsu. Motion Control. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 133–159. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Craig89] John Craig. *Introduction to Robotics : Mechanics and Control*. Addison-Wesley, Reading, MA, second edition edition, 1989.

[Crassidis07] John L Crassidis, F Landis Markley, and Yang Cheng. Survey of nonlinear attitude estimation methods. *Journal of guidance, control, and dynamics*, 30(1):12–28, 2007.

[De Luca95] A. De Luca and L. Lanari. Robots with elastic joints are linearizable via dynamic feedback. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 4, pages 3895–3897, New Orleans, LA, USA, 1995. IEEE.

[De Luca98] A. De Luca and P. Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 1, pages 504–510 vol.1, May 1998.

[De Luca05a] A. De Luca, R. Farina, and P. Lucibello. On the Control of Robots with Visco-Elastic Joints. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4297–4302, Barcelona, Spain, 2005. IEEE.

[De Luca05b] Alessandro De Luca, Bruno Siciliano, and Loredana Zollo. PD Control with On-line Gravity Compensation for Robots with Elastic Joints: Theory and Experiments. *Automatica*, 41(10):1809–1819, October 2005.

[De Luca08] Alessandro De Luca and Wayne Book. Robots with Flexible Elements. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 287–319. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[De Magistris16] Giovanni De Magistris, Adrien Pajon, Sylvain Miossec, and Abderrahmane Kheddar. Humanoid walking with compliant soles using a deformation estimator. In *ICRA*, 2016.

[Duburcq20] Alexis Duburcq. Jiminy: a fast, lightweight open-source simulator for poly-articulated systems. `https://github.com/duburcqa/jiminy`, 2018–2020.

[El Khoury13] Antonio El Khoury. *Planning Optimal Motions for Anthropomorphic Systems*. Theses, Université Paul Sabatier - Toulouse III, June 2013.

[Englsberger12] 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)*, pages 183–189, 2012.

[Featherstone08] R. Featherstone. *Rigid Body Dynamics Algorithms*. Springer US, 2008.

[Ficht21] Grzegorz Ficht and Sven Behnke. Bipedal Humanoid Hardware Design: a Technology Review. *Current Robotics Reports*, 2(2):201–210, June 2021.

[Finet17] Sylvain Finet. *Contribution to the Control of Biped Robots*. Theses, Université Paris sciences et lettres, June 2017.

[Flayols17] T. Flayols, A. Del Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse. Experimental evaluation of simple estimators for humanoid robots. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 889–895, November 2017.

[Frosio09] I. Frosio, F. Pedersini, and N.A. Borghese. Autocalibration of MEMS Accelerometers. *IEEE Transactions on Instrumentation and Measurement*, 58(6):2034–2041, June 2009.

[Geigle17] Paula Richley Geigle and Marni Kallins. Exoskeleton-Assisted Walking for People With Spinal Cord Injury. *Archives of Physical Medicine and Rehabilitation*, 98(7):1493–1495, July 2017.

[Grassia98] F. Sebastin Grassia. Practical Parameterization of Rotations Using the Exponential Map. *J. Graph. Tools*, 3(3):29–48, March 1998.

[Harib18] O. Harib, A. Hereid, A. Agrawal, T. Gurriet, S. Finet, G. Boeris, A. Duburcq, M. E. Mungai, M. Masselin, A. D. Ames, K. Sreenath, and J. W. Grizzle. Feedback Control of an Exoskeleton for Paraplegics: Toward Robustly Stable, Hands-Free Dynamic Walking. *IEEE Control Systems Magazine*, 38(6):61–87, December 2018.

[Hartley18] Ross Hartley, Maani Ghaffari Jadidi, Jessy W. Grizzle, and Ryan M. Eustice. Contact-Aided Invariant Extended Kalman Filtering for Legged Robot State Estimation. In *Proceedings of Robotics: Science and Systems*, June 2018.

[Hemingway18] Evan G. Hemingway and Oliver M. O'Reilly. Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics*, 44(1):31–56, September 2018.

[Hereid17] Ayonga Hereid and Aaron D. Ames. FROST*: Fast robot optimization and simulation toolkit. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 719–726, Vancouver, BC, September 2017. IEEE.

[Herissé11] Bruno Herissé, Tarek Hamel, Robert Mahony, and François-Xavier Russotto. Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on robotics*, 28(1):77–89, 2011.

[Hirai98] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 2, pages 1321–1326 vol.2, 1998.

[Hua16] Minh-Duc Hua, Philippe Martin, and Tarek Hamel. Stability analysis of velocity-aided attitude observers for accelerated vehicles. *Automatica*, 63:11–15, January 2016.

[Huynh09] D. Huynh. Metrics for 3d Rotations: Comparison and Analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, October 2009.

[Johnson15] 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, and Jerry Pratt. Team IHMC's lessons learned from

the DARPA robotics challenge trials. *Journal of Field Robotics*, 32, March 2015.

[Jung17] Jaesug Jung, Soonwook Hwang, Yisoo Lee, Jaehoon Sim, and Jaeheung Park. Analysis of position tracking in torque control of humanoid robots considering joint elasticity and time delay. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 515–521, Birmingham, November 2017. IEEE.

[Kajita01] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 239–246, Maui, HI, USA, 2001. IEEE.

[Kajita10] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shin'ichiro Nakaoka, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4489–4496, 2010.

[Kelly00] R. Kelly, J. Llamas, and R. Campa. A measurement procedure for viscous and coulomb friction. *IEEE Transactions on Instrumentation and Measurement*, 49(4):857–861, 2000.

[Khandelwal13] S. Khandelwal and C. Chevallereau. Estimation of the Trunk Attitude of a Humanoid by Data Fusion of Inertial Sensors and Joint Encoders. In *Nature-Inspired Mobile Robotics*, pages 822–830. World Scientific, Singapour, 2013.

[Kim16] J. Kim, M. Kim, and J. Park. Improvement of humanoid walking control by compensating actuator elasticity. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 29–34, November 2016.

[Kim18] Mingon Kim, Jung Hoon Kim, Sanghyun Kim, Jaehoon Sim, and Jaeheung Park. Disturbance Observer Based Linear Feedback Controller for Compliant Motion of Humanoid Robot. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 403–410, 2018.

[Lismonde20] Arthur Lismonde. *Geometric modeling and inverse dynamics of flexible manipulators*. PhD thesis, Université de Liège, Liège, Belgique, 2020.

[Lublin10] Leonard Lublin and Michael Athans. Linear Quadratic Regulator Control. In William Levine, editor, *The Control Systems Handbook, Second Edition*, volume 20103237 of *Electrical Engineering Handbook*. CRC Press, December 2010.

[Mahony08] R. Mahony, T. Hamel, and J.M. Pflimlin. Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*, 53(5):1203–1217, June 2008.

[Marino86] R. Marino and M. Spong. Nonlinear control techniques for flexible joint manipulators: A single link case study. In *1986 IEEE International Conference on Robotics and Automation Proceedings*, volume 3, pages 1030–1036, April 1986.

[Martin07] Philippe Martin and Erwan Salaun. Invariant observers for attitude and heading estimation from low-cost inertial and magnetic sensors. In *2007 46th IEEE Conference on Decision and Control*, pages 1039–1045, 2007.

[Mastalli20] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard. Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2536–2542, Paris, France, May 2020. IEEE.

[McCarthy90] J. M. McCarthy. *Introduction to Theoretical Kinematics*. MIT Press, Cambridge, MA, USA, 1990.

[Mifsud16] A. Mifsud, M. Benallegue, and F. 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)*, pages 5405–5410, October 2016.

[Moakher02] Maher Moakher. Means and Averaging in the Group of Rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, January 2002.

[Moberg08] Stig Moberg and Sven Hanssen. On Feedback Linearization for Robust Tracking Control of Flexible Joint Robots. *IFAC Proceedings Volumes*, 41(2):12218–12223, January 2008.

[Moberg10] Stig Moberg, Linköpings universitet, and Institutionen för systemteknik. *Modeling and control of flexible manipulators*. PhD thesis, Department of Electrical Engineering, Linköping University, Linköping, 2010. OCLC: 939125546.

[Moberg14] Stig Moberg, Erik Wernholt, Sven Hanssen, and Torgny Brogårdh. Modeling and Parameter Estimation of Robot Manipulators Using Extended Flexible Joint Models. *Journal of Dynamic Systems, Measurement, and Control*, 136(3):031005, May 2014.

[Nicosia81] S. Nicosia, F. Nicolò, and D. Lentini. Dynamical control of industrial robots with elastic and dissipative joints. *IFAC Proceedings Volumes*, 14(2):1933–1940, 1981.

[Niu13] Xiaoji Niu, You Li, Hongping Zhang, Qingjiang Wang, and Yalong Ban. Fast Thermal Calibration of Low-Grade Inertial Sensors and Inertial Measurement Units. *Sensors*, 13(9):12192–12217, September 2013.

[Paine15] Nicholas Paine, Joshua S. Mehling, James Holley, Nicolaus A. Radford, Gwendolyn Johnson, Chien-Liang Fok, and Luis Sentis. Actuator Control for the NASA-JSC Valkyrie Humanoid Robot: A Decoupled Dynamics Approach for Torque Control of Series Elastic Robots. *Journal of Field Robotics*, 32(3):378–396, 2015.

[Park95] F. C. Park. Distance Metrics on the Rigid-Body Motions with Applications to Mechanism Design. *Journal of Mechanical Design*, 117(1):48–54, 03 1995.

[Pratt95] G.A. Pratt and M.M. Williamson. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 399–406 vol.1, 1995.

[Pratt06] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture Point: A Step toward Humanoid Push Recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, 2006.

[Reher16] Jacob P Reher, Ayonga Hereid, Shishir Kolathaya, Christian M Hubicki, and Aaron D Ames. Algorithmic Foundations of Realizing Multi-Contact Locomotion on the Humanoid Robot DURUS. *International Workshop on the Algorithmic Foundations of Robotics*, page 16, 2016.

[Scandaroli11] Glauco Garcia Scandaroli, Pascal Morin, and Geraldo Silveira. A nonlinear observer approach for concurrent estimation of pose, IMU bias and camera-to-IMU rotation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3335–3341, 2011.

[Seron99] M.M. Seron, J.H. Braslavsky, P.V. Kokotovic, and D.Q. Mayne. Feedback limitations in nonlinear systems: from Bode integrals to cheap control. *IEEE Transactions on Automatic Control*, 44(4):829–833, 1999.

[Shuster81] M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77, January 1981.

[Spong87] M. W. Spong. Modeling and Control of Elastic Joint Robots. *Journal of Dynamic Systems, Measurement, and Control*, 109(4):310–318, December 1987.

[Spong90] Mark W Spong. Control of flexible joint robots: a survey. *Coordinated Science Laboratory Report no. UILU-ENG-90-2203, DC-116*, 1990.

[Stasse19] Olivier Stasse and Thomas Flayols. An overview of humanoid robots technologies. In *Biomechanics of Anthropomorphic Systems*, pages pp.281–310. Springer, January 2019.

[Staufer12] Peter Staufer and Hubert Gattringer. State estimation on flexible robots using accelerometers and angular rate sensors. *Mechatronics*, 22(8):1043–1049, 2012.

[Tedaldi14] D. Tedaldi, A. Pretto, and E. Menegatti. A robust and easy to implement method for IMU calibration without external equipments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3042–3049, May 2014.

[Titterton04] David Titterton and John Weston. *Strapdown Inertial Navigation Technology*. Institution of Engineering and Technology, January 2004.

[Tomei91] P. Tomei. A simple PD controller for robots with elastic joints. *IEEE Transactions on Automatic Control*, 36(10):1208–1213, October 1991.

[Torabi08] Meysam Torabi and Mehran Jahed. A Novel Approach for Robust Control of Single-Link Manipulators with Visco-Elastic Behavior. In *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)*, pages 685–690, 2008.

[Tsagarakis11] Nikos G. Tsagarakis, Zhibin Li, Jody Saglia, and Darwin G. Caldwell. The design of the lower body of the compliant humanoid robot “cCub”. In *2011 IEEE International Conference on Robotics and Automation*, pages 2035–2040, Shanghai, China, May 2011. IEEE.

[Vukobratovic74] M. Vukobratovic, D. Hristic, and Z. Stojiljkovic. Development of active anthropomorphic exoskeletons. *Medical and biological engineering*, 12(1):66–80, January 1974.

[Wahba65] G. Wahba. Problem 65-1: a least squares estimate of spacecraft attitude. In *SIAM Review*, volume 7, page 409, 1965.

[Westervelt18] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 1 edition, October 2018.

[Winter09] David A. Winter. *Biomechanics and motor control of human movement*. Wiley, Hoboken, NJ, 4. ed edition, 2009.

[Woodman07] Olivier J. Woodman. An introduction to inertial navigation. Technical Report 696, University of Cambridge, Computer Laboratory, 2007.

[Zimmermann20] Stefanie A. Zimmermann, Tobias F. C. Berninger, Jeroen Derkx, and Daniel J. Rixen. Dynamic modeling of robotic manipulators for accuracy evaluation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8144–8150, 2020.

[Zollo05] Loredana Zollo, Bruno Siciliano, Alessandro De Luca, Eugenio Guglielmelli, and Paolo Dario. Compliance Control for an Anthropomorphic Robot with Elastic Joints: Theory and Experiments. *Journal of Dynamic Systems, Measurement, and Control*, 2005.

RÉSUMÉ

Cette thèse porte sur la question générale de la compensation des déformations sur un robot marcheur, avec comme cas d'étude particulier l'exosquelette à usage médical Atalante. Les déformations des robots anthropomorphes sont des phénomènes indésirables résultant de leur conception mécanique. Ces robots sont en effet constitués de jambes longues devant soutenir un torse relativement massif. Ces effets sont d'autant plus présents sur un exosquelette, qui en plus de son propre poids doit supporter le poids de l'utilisateur. La présence d'un humain non instrumenté induit des perturbations significatives non mesurées, auxquelles il est difficile de s'adapter en temps-réel.
Nous proposons une approche en boucle fermée reposant sur des capteurs, qui effectue un retour d'état sur les signaux provenant de plusieurs centrales inertielles. A cette fin, nous modélisons cinématiquement les déformations par des liaisons rotules supplémentaires. Nous étudions plusieurs observateurs visant à estimer les rotations correspondantes, en reconstituant l'orientation des centrales inertielles. Trois observateurs, utilisant des modèles différents, sont étudiés: un modèle supposant que l'accélération moyenne des corps est nulle, un modèle cinématique et un modèle dynamique. Nous concluons que les meilleurs résultats sont obtenus en exploitant uniquement le modèle cinématique du robot afin de reconstruire une mesure approximative de la vitesse. Ceci permet la conception d'un observateur capable de supporter des accélérations élevées, tout en restant robuste à l'incertitude dynamique liée au comportement du patient.
Ces estimations d'attitude sont ensuite converties (par projection) sur les articulations du robot, comme dans un modèle d'actionneur élastique, et utilisées pour réaliser un contrôle par retour d'état grand gain décentralisé. L'ensemble de cette méthodologie est validé expérimentalement sur Atalante, où elle améliore le rejet de perturbation et le suivi de trajectoire, augmentant ainsi la robustesse de la marche d'Atalante avec utilisateur.

MOTS CLÉS

Exosquelette, flexibilités, estimation d'attitude, centrale inertielle, contrôle boucle fermé

ABSTRACT

This thesis addresses the general problem of the compensation of deformations on a walking robot, and considers it in the particular context of the medical exoskeleton Atalante. Structural deformations are unavoidable in anthropomorphic robots with long legs and a heavy torso. This is even more true for an exoskeleton, which has to support the weight of the user. Meanwhile, the presence of an uninstrumented human user leads to significant disturbances, that limit the performance of feedforward corrections.
We propose a sensor-based methodology, that feedbacks the data of several Inertial Measurement Units (IMUs) onto the actuators, in order to mitigate the effect of the multiple flexibilities on Atalante. To that end, we kinematically model the deformations as extra spherical joints. We study several observers to estimate the rotations induced by the flexibilities, by estimating the attitude of the IMUs. Three observers, based on various models, are considered: a (naive) zero-on-average acceleration model, a kinematic model and a dynamic model. We conclude that the best results are obtained by relying only on the kinematic model of the robot to build an approximate velocity measurement. This allows the design of an observer able to handle strong accelerations while being robust to the dynamical uncertainties linked to patient behavior.
These attitude estimates are then projected onto the joints of the robot, to adopt a joint elasticity model, which we use to perform decentralized high-gain feedback control. This methodology is experimentally validated on Atalante, where it yields improved disturbance rejection and improved trajectory tracking, enhancing the robustness of Atalante's walk with a user.

KEYWORDS

Exoskeleton, flexibilities, attitude estimation, inertial measurement unit, feedback control