



MOVIE: a Velocity-aided IMU Attitude Estimator for Observing and Controlling Multiple Deformations on Legged Robots

Matthieu Vigne, Antonio El Khoury, Marine Pétriaux, Florent Di Meglio, Nicolas Petit

► To cite this version:

Matthieu Vigne, Antonio El Khoury, Marine Pétriaux, Florent Di Meglio, Nicolas Petit. MOVIE: a Velocity-aided IMU Attitude Estimator for Observing and Controlling Multiple Deformations on Legged Robots. IEEE Robotics and Automation Letters, 2022, 7 (2), pp.3969-3976. <10.1109/LRA.2022.3149025>. <hal-03511198>

HAL Id: hal-03511198

<https://hal.science/hal-03511198v1>

Submitted on 4 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

MOVIE: a Velocity-aided IMU Attitude Estimator for Observing and Controlling Multiple Deformations on Legged Robots

Matthieu Vigne^{1,2}, Antonio El Khoury¹, Marine Pétriaux¹, Florent Di Meglio² and Nicolas Petit²

Abstract—This paper presents a novel observer for estimating the state of a legged robot with deformations in the structure, taking into account the dynamic nature of walking motions. Specifically, it considers a robot with several punctual deformations, whose rotations are estimated using low-cost IMUs. To compensate for the linear acceleration corrupting the measurement of gravity in the accelerometer, this observer exploits the robot’s kinematics model. Namely, it combines gyroscope and accelerometer signals, encoder measurements, and attitude estimates from the previous IMUs to reconstruct the linear velocity of a given IMU. This reconstructed measurement is used to estimate the tilt (roll and pitch) of the current IMU, which is then exploited to propagate the linear velocity to the next sensor. The resulting cascade of observers, called MOVIE, is experimentally benchmarked on the exoskeleton Atalante, where it outperforms our previous observer, relying on a quasi-static assumption. This observer is then used in closed-loop control, to compensate for the effect of deformations while walking, resulting in an improved walk.

I. INTRODUCTION

Having an accurate estimation of the robot’s pose is a key element for the control of any robotic system. For rigid legged robots, instrumented by joint encoders, six degrees of freedom describing the absolute position and orientation (freeflyer joint) of the system are not directly instrumented. Of particular importance is the robot’s orientation with respect to gravity, called the tilt [1], which determines the robot’s ability to keep balance.

In order to estimate this orientation, inertial measurements units (IMUs) are very commonly used [2]. Following the complementary filter approach, the idea of IMU-based attitude estimation is to integrate the gyroscope angular velocity measurement, while exploiting the accelerometer measurement of gravity to prevent estimation drift [2]. However, this “measurement of gravity” is “corrupted” by the linear acceleration of the sensor, which acts as an additive time-varying bias in the accelerometer signal. The challenge then lies in separating these two terms.

Several solutions have been proposed to address this on a legged robot walking. A first possibility consists in neglecting the linear acceleration against gravity [3], [4], [5], thus forming the “zero-on-average acceleration hypothesis”. While this approach results in a very simple observer formulation, its performance remains limited by the presence of linear accelerations while walking. Instead, using the fact that a walking robot remains in contact with its environment



Fig. 1. Picture of the exoskeleton Atalante with a paraplegic user. On the right, CAD reconstruction under the assumption of full rigidity (blue), and ground truth from motion capture (red): the real robot is leaning forward due to deformations.

leads to a coupling between translation and rotation motions, through the robot’s kinematic model: this can be used to enhance the estimation of the tilt [6], [7], [8]. More precisely, these observers rely on the perfect knowledge of the robot’s kinematics (full rigidity hypothesis) to relate encoder measurements to IMU motion. In particular, in [1] it is shown that for a rigid robot in punctual contact with its environment, a measurement of the IMU linear velocity in the sensor frame can be obtained through forward kinematics: this measurement is then used to build a globally convergent observer, robust to the presence of linear accelerations.

However, for many humanoid robots, the assumption of full rigidity is only an approximation. These robots indeed feature long, thin legs, carrying a relatively heavy torso. These mechanical constraints often result in the presence of deformations in the structure, which can be modeled as extra rotations in the structure. Such is the case of the exoskeleton Atalante (Figure 1), serving as illustration in this article [5]. The presence of multiple punctual deformations imply that the rigid model is no longer sufficient to describe the robot’s motion, as they introduce extra degrees of freedom: as such, the previous approaches cannot be used as-is.

In our previous work [5], we present an estimation technique relying on several IMUs to reconstruct the state of multiple deformations on a legged robot walking. This work however relies on neglecting the linear acceleration before gravity, using the complementary filter presented in [3] for tilt estimation. Instead, in this article, we extend the velocity-aided approach of [1] to design an observer for multiple deformations that takes these accelerations into account: this is the main contribution of the paper. Our approach is as follows: we use the robot’s flexible kinematic model to

¹Control team, Wandercraft, Paris, France
matthieu.vigne@mines-paristech.fr

²Centre Automatique et Systèmes, MINES ParisTech, PSL Research University, France

relate sensor information from one deformation to the next, starting from the stance foot. While an exact derivation leads to implicit equations, we carefully approximate the velocity term to yield an explicit triangular, easily solvable structure. The resulting algorithm, which we call MOVIE (Multiple-Object Velocity-aided IMU Estimator), is experimentally validated in open-loop experiments on the exoskeleton Atalante, where it significantly outperforms our previous quasi-static approach. We then use the data from this observer for closed-loop control, compensating the deflection due to the deformations while walking. This results in an improved trajectory tracking and gait stability.

This paper is organized as follows: Section II defines the state estimation problem, while Section III presents our observer, MOVIE. In Section IV, this observer is experimentally benchmarked on Atalante. Finally, Section V shows how to exploit this observer for feedback control.

II. SYSTEM MODELING

A. Problem statement

Consider a biped robot standing or walking on an arbitrary ground. We assume that the robot is equipped with joint encoders, giving a measurement of the joint angles θ and velocities $\dot{\theta}$ (through filtered numerical differentiation). We furthermore assume that the robot is equipped with force sensors in its feet, giving access to a measurement of the center of pressure in each foot. We exclude jumping and running gaits, and formulate the following assumption.

Assumption 1 *At any given time, there is at least one foot in non-slipping contact with the ground.*

The contact foot is determined using force sensor measurements. In the case of double support (when both feet are touching the ground), an arbitrary foot can be chosen to act as reference: a typical choice is to take the foot with the highest force applied, as being the less likely to slide. We refer to this foot as the contact foot, and denote \mathcal{C} the associated frame. We call O_0 the contact point: by assumption the robot does not translate about this point, but can rotate around it as the support foot tilts on the ground. The position of this fixed point O_0 , in the contact foot frame \mathcal{C} , is determined using the robot's force sensors: more precisely, we locate O_0 at the center of pressure of the contact foot. This choice is motivated by the following observation: when the center of pressure is strictly inside the contact foot, this foot rests motionless on the ground, and every point is a fixed point. Conversely, when the foot starts to tilt on one of its edges, the center of pressure will be located on the corresponding edge.

To remove the odometry coordinates, we define the *local frame* \mathcal{L} as the frame translated from the world frame to lie below the contact foot frame \mathcal{C} , rotated about the vertical axis e_z to remain aligned with the robot's forward axis. This frame is represented in Figure 2.

Our goal is to estimate the position of the robot's bodies \mathcal{B} in the local frame \mathcal{L} (denoted ${}^{\mathcal{L}}M_{\mathcal{B}}$), taking into account the presence of deformations in the robot's structure, and the

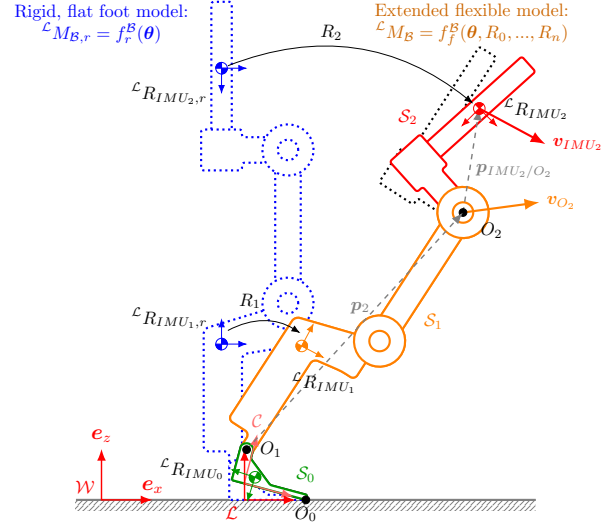


Fig. 2. A representation of the various frames and notations used in this article. This side view shows three rigid sets (in green, orange and red), with their associated IMUs, connected about two deformations at the ankle (O_1) and the hip (O_2).

fact that the contact foot is not horizontal (due to a non-flat ground, or to the foot tilting on the ground).

B. Rigid, flat foot model

A first approximation of the robot's pose in the local frame is obtained by neglecting all deformations, to assume a rigid model, and by considering that the contact foot rests flat on a horizontal ground. In this context, the only degrees of freedom of the system are the joints of the robot. This writes

$${}^{\mathcal{L}}M_{\mathcal{B},r} = f_r^{\mathcal{B}}(\theta) \quad (1)$$

where $f_r^{\mathcal{B}}$ is the kinematic model of the robot giving the pose (position and orientation) of body \mathcal{B} , assuming full rigidity (notice the suffix $_r$ used to indicate the use of a rigid model). This is represented by the blue dotted robot in Figure 2.

C. Flexible model

Equation (1) only approximately describes the true pose of a body \mathcal{B} in the local frame. A first source of discrepancy lies in the flat-foot assumption: in practice, the foot can rotate in roll and pitch with respect to the local frame. We call $R_0 \triangleq {}^{\mathcal{L}}R_{\mathcal{C}}$ this rotation, which is not measured by the joint encoders.

Secondly, full rigidity is for many robots, including Atalante, a mere approximation. In practice, punctual deformations are often present about the joints of the robot, which represent weaker points of the structure [9], [10]. In this paper, we consider the case where n such deformations are present, as formulated below.

Assumption 2 *The deformations of the robot correspond to n 3D rotations about known points in the structure (typically collocated with the joints), labeled down the kinematic chain O_i , $i = 1, \dots, n$ (O_1 being the closest to the support foot).*

These deformations decompose the robot into $n+1$ "rigid sets", labeled \mathcal{S}_i , $i = 0 \dots n$. Each set \mathcal{S}_i is composed of all the bodies between deformation i and deformation $i+1$

(\mathcal{S}_0 is composed of all bodies before the first deformation, while \mathcal{S}_n all the bodies after the last deformation). Each set \mathcal{S}_i is articulated about the joints of the robot, but includes no deformation, hence the name “rigid”. These sets are illustrated in different colors in Figure 2.

In order to ease the formulation of the observer, we parametrize the deformations in terms of the *total deviation with respect to the rigid, flat-footed model*. In particular, this enables for a unified handling of the deformations and the contact foot rotation. More precisely, we define $R_i \in SO(3)$ as the rotation taking the bodies of set \mathcal{S}_i from their rigid, flat footed configuration, to their real configuration in the local frame. This gives

$$R_i \triangleq {}^{\mathcal{L}}R_{\mathcal{B}_i} {}^{\mathcal{L}}R_{\mathcal{B}_i,r}^T \quad (2)$$

Thus, R_0 represents the contact foot’s angle, while $R_{i-1}^T R_i$ represents the deformation taking place at point O_i . Then, the configuration of the robot in the local frame is parametrized by the joint angles θ , and the $n+1$ rotations R_i to be estimated: this writes

$${}^{\mathcal{L}}M_{\mathcal{B}} = f_f^{\mathcal{B}}(\theta, R_0, \dots, R_n) \quad (3)$$

The function $f_f^{\mathcal{B}}$ is called the *extended flexible model* (this term is taken from [10]), i.e. the kinematic model of the robot taking into account the presence of deformations.

D. Instrumentation and handling of the yaw angle

In order to estimate the $n+1$ rotations R_i , we instrument the system by placing $n+1$ IMUs, one in each rigid set \mathcal{S}_i . We number these IMUs based on the rigid set they belong to. Then, using (2), estimating the rotations R_i is equivalent to estimating the orientation of the IMUs (as $R_i = {}^{\mathcal{L}}R_{IMU_i} {}^{\mathcal{L}}R_{IMU_i,r}^T$).

Each IMU is composed of a tri-axial accelerometer and gyroscope. Magnetometers are hardly usable on a robot, where electric motors generate a strong time-varying magnetic field. The sensor measurements of an IMU thus write

$$\begin{cases} \mathbf{y}_a = {}^{\mathcal{W}}R_{IMU}^T ({}^{\mathcal{W}}\ddot{\mathbf{p}}_{IMU} - \mathbf{g}) \\ \mathbf{y}_g = \boldsymbol{\omega}_{IMU} \end{cases} \quad (4)$$

where \mathbf{y}_a and \mathbf{y}_g are the accelerometer and gyroscope measurements, $\mathbf{g} = -g\mathbf{e}_z$ the gravity vector, and $\boldsymbol{\omega}_{IMU}$ the IMU angular velocity in the sensor frame, i.e. the vector such that

$${}^{\mathcal{W}}\dot{R}_{IMU} = {}^{\mathcal{W}}R_{IMU} [\boldsymbol{\omega}_{IMU}]_{\times} \quad (5)$$

where $[\cdot]_{\times}$ is the skew-symmetric operator associated to the cross product.

A well-known limitation of this sensor is the fact that the yaw angle (i.e. the angle about gravity) is not observable, because the sensor measurements are invariant by a rotation of the world frame about \mathbf{e}_z [3]. In particular, the IMU cannot distinguish the local frame from the world frame. This fact is conveniently expressed using the notion of *tilt* [1]:

Definition 1 Let \mathbf{e}_z be the world frame vertical axis. For any rotation $R \in SO(3)$, we call tilt the vector

$$\mathbf{t}(R) = R^T \mathbf{e}_z \quad (6)$$

We write \mathbf{t}_{IMU_i} the tilt of the i th IMU ${}^{\mathcal{W}}R_{IMU_i}$. Estimating this tilt is equivalent to estimating the tilt of R_i as

$$\begin{aligned} \mathbf{t}(R_i) &= {}^{\mathcal{L}}R_{IMU_i,r} {}^{\mathcal{L}}R_{IMU_i}^T \mathbf{e}_z = \\ &= {}^{\mathcal{L}}R_{IMU_i,r} {}^{\mathcal{W}}R_{IMU_i}^T \mathbf{e}_z = {}^{\mathcal{L}}R_{IMU_i,r} \mathbf{t}_{IMU_i} \end{aligned} \quad (7)$$

In order to reconstruct the rotation matrix R_i from an estimate of its tilt, $\hat{\mathbf{t}}$, we make the following approximation. We neglect the yaw deformations, which are less pronounced than the roll and pitch deformations since gravity forces have no physical impact on this angle. We thus nullify the non-observable component of R_i ¹. This is done by computing the rotation matrix of minimal norm matching the observed tilt: this process is detailed in our previous work [5], and ultimately leads to the following simple formula:

$$\hat{R}_i \triangleq \mathcal{R}(\arccos(\mathbf{e}_z^T \hat{\mathbf{t}}), \frac{\mathbf{e}_z \times \hat{\mathbf{t}}}{\|\mathbf{e}_z \times \hat{\mathbf{t}}\|}) \quad (8)$$

where $\mathcal{R}(\theta, \mathbf{v}) = \exp[\theta \mathbf{v}]_{\times}$ is the rotation of angle θ about the vector \mathbf{v} .

In summary, with this approximation, estimating the pose of the robot’s bodies in the local frame \mathcal{L} boils down to estimating the tilt of each IMU. This tilt is then transformed into an estimate of the rotations R_i through (8), which is then used in (3) to provide these poses (${}^{\mathcal{L}}\hat{M}_{\mathcal{B}} \triangleq f_f^{\mathcal{B}}(\theta, \hat{R}_0, \dots, \hat{R}_n)$). The challenge then lies in estimating these tilts: this is the purpose of our observer, MOVIE.

III. MOVIE: AN OBSERVER FOR ESTIMATING THE TILT OF THE IMUS

A. Core equations: velocity-aided tilt estimation

The main difficulty in estimating the tilt of an IMU in accelerated motion lies in separating the linear acceleration of the sensor from gravity. Indeed, (4) rewrites

$$\mathbf{y}_a = {}^{\mathcal{W}}R_{IMU}^T {}^{\mathcal{W}}\ddot{\mathbf{p}}_{IMU} + g\mathbf{t}_{IMU} \quad (9)$$

To separate these two terms, a possible solution is to use a measurement of the linear velocity of the IMU, in the sensor frame [11], [12]. Let $\mathbf{v} \triangleq {}^{\mathcal{W}}R_{IMU}^T {}^{\mathcal{W}}\dot{\mathbf{p}}_{IMU}$ be this velocity. Computing its time derivative and using (9) yields

$$\begin{aligned} \dot{\mathbf{v}} &= -[\boldsymbol{\omega}_{IMU}]_{\times} {}^{\mathcal{W}}R_{IMU}^T \dot{\mathbf{p}}_{IMU} + {}^{\mathcal{W}}R_{IMU}^T {}^{\mathcal{W}}\ddot{\mathbf{p}}_{IMU} \\ &= -[\boldsymbol{\omega}_{IMU}]_{\times} \mathbf{v} + \mathbf{y}_a + g\mathbf{t}_{IMU} \end{aligned} \quad (10)$$

Combined with the derivative of the tilt (6), this yields the following set of equations

$$\begin{cases} \dot{\mathbf{v}} = -[\boldsymbol{\omega}_{IMU}]_{\times} \mathbf{v} + \mathbf{y}_a + g\mathbf{t}_{IMU} \\ \dot{\mathbf{t}}_{IMU} = -[\boldsymbol{\omega}_{IMU}]_{\times} \mathbf{t}_{IMU} \end{cases} \quad (11)$$

Several observers have been proposed to estimate the tilt relying on (11) and on sensor values defined in (4), e.g. [12], [1], [8]. In this paper, we use the formulation proposed in [1] which writes

$$\begin{cases} \dot{\hat{\mathbf{v}}} = -[\mathbf{y}_g]_{\times} \hat{\mathbf{v}} + \mathbf{y}_a - g\hat{\mathbf{t}}_{IMU} + \alpha(\hat{\mathbf{v}} - \mathbf{v}) \\ \dot{\hat{\mathbf{t}}}_{IMU} = -[\mathbf{y}_g - \beta[\hat{\mathbf{t}}_{IMU}]_{\times}(\hat{\mathbf{v}} - \mathbf{v})]_{\times} \hat{\mathbf{t}}_{IMU} \end{cases} \quad (12)$$

¹Note that by definition, the local frame is aligned with the robot’s foot, thus R_0 also has a zero non-observable component, even though it is not strictly speaking a deformation.

where α, β positive observer gains such that $\beta g_0 < \alpha^2$. This observer is proven in [1] to be globally convergent, except on the zero-measure subset $\hat{\mathbf{t}}_{IMU} = -\mathbf{t}_{IMU}$.

B. MOVIE: a cascade implementation of (12)

In order to apply (12), the challenge lies in obtaining a measurement of the linear velocity \mathbf{v} . In the case of a rigid robot in punctual contact with the ground, [1] shows that this velocity can be algebraically reconstructed from gyroscope and encoder measurements. In the case where more than one rotation need to be estimated (i.e. when several deformations are present), this is no longer the case, and the computation presented in this article [1] cannot be used as-is.

The solution we propose is to successively rely on the signal and attitude estimation of the previous IMUs to reconstruct an accurate approximation of the linear velocity of the current IMU.

This results in an iterative algorithm, which we call MOVIE (Multiple Objects Velocity-aided IMU Estimator).

To formulate this algorithm, we associated two frames to each deformation: the parent frame \mathcal{P}_i located before the deformation (in set \mathcal{S}_i), and the child frame \mathcal{C}_i after the deformation; by convention we define $\mathcal{P}_0 = \mathcal{L}$ and $\mathcal{C}_0 = \mathcal{C}$ to account for the contact foot angle.

MOVIE algorithm

Input: Sensor measurements: IMUs, encoders, contact foot and location of O_0 (from force sensors)

For i from 0 to n :

- 1) use estimates from steps $[0, i-1]$ to compute the linear velocity of point O_i with respect to the world frame, in frame \mathcal{P}_i . This velocity is denoted $\hat{\mathbf{v}}_{O_i}$, and is given by (16).
- 2) use this velocity to compute an approximation of the velocity of IMU i in (17). Then, use the observer (12) to estimate its tilt.

Output: IMU tilt estimates, $\hat{\mathbf{t}}_{IMU_i}, i \in [0, n)$.

Both steps of the algorithm are detailed below:

1) *Step 1: computation of the linear velocity of O_i :* For the first IMU, the contact assumption simply gives $\hat{\mathbf{v}}_{O_0} = 0$.

For $i \geq 1$, we use the velocity and IMU tilt estimation from the previous IMU to compute this velocity. More precisely, we decompose the motion of O_i about O_{i-1} . Let \mathbf{p}_i be the position of point O_i with respect to point O_{i-1} , expressed in frame \mathcal{P}_i . This position is function of the configuration of the rigid set \mathcal{S}_{i-1} , and thus directly measured by the joint encoder. The position of point O_i in the world frame then writes

$${}^{\mathcal{W}}\mathbf{p}_{O_i} = {}^{\mathcal{W}}\mathbf{p}_{O_{i-1}} + {}^{\mathcal{W}}R_{\mathcal{P}_i}\mathbf{p}_i \quad (13)$$

Differentiating (13) and multiplying by ${}^{\mathcal{W}}R_{\mathcal{P}_i}^T$, the velocity of point O_i with respect to \mathcal{W} expressed in \mathcal{P}_i writes

$$\mathbf{v}_{O_i} = {}^{\mathcal{P}_i}R_{\mathcal{P}_{i-1}}\mathbf{v}_{O_{i-1}} + [\boldsymbol{\omega}_{\mathcal{P}_i}]_{\times}\mathbf{p}_i + \dot{\mathbf{p}}_i \quad (14)$$

where $\boldsymbol{\omega}_{\mathcal{P}_i}$ is the angular velocity of frame \mathcal{P}_i with respect to the world frame.

As shown in (15), this angular velocity is measured by the gyroscope of IMU $i-1$ and the encoders. Indeed, the rotation ${}^{IMU_{i-1}}R_{\mathcal{P}_i}$ is a function of the joint position only, as both frames belong to the same set \mathcal{S}_{i-1} . Its angular velocity $\boldsymbol{\omega}_{\mathcal{P}_i/IMU_{i-1}}$ is thus a function of $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ only. Meanwhile, differentiating the product ${}^{\mathcal{W}}R_{\mathcal{P}_i} = {}^{\mathcal{W}}R_{IMU_{i-1}}{}^{IMU_{i-1}}R_{\mathcal{P}_i}$ yields

$$\boldsymbol{\omega}_{\mathcal{P}_i} = \boldsymbol{\omega}_{\mathcal{P}_i/IMU_{i-1}} + {}^{IMU_{i-1}}R_{\mathcal{P}_i}^T \boldsymbol{\omega}_{IMU_{i-1}} \quad (15)$$

Hence, the last two terms of (14) can be expressed directly in terms of sensor measurements. Finally, the rotation ${}^{\mathcal{P}_i}R_{\mathcal{P}_{i-1}}$ is the product of the rotation ${}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}$ through the rigid set \mathcal{S}_{i-1} , measured by the encoders, and the $i-1$ th deformation, $D_{i-1} \triangleq {}^{\mathcal{C}_{i-1}}R_{\mathcal{P}_{i-1}}$. This matrix corresponds to the difference between previously estimated matrices: indeed using (2) yields $D_{i-1} = {}^{\mathcal{L}}R_{\mathcal{C}_{i-1},r}^T \hat{R}_{i-2}^T {}^{\mathcal{L}}R_{\mathcal{C}_{i-1},r}$ (defining by convention $R_{-1} \triangleq I$). Hence, using $\hat{D}_{i-1} \triangleq {}^{\mathcal{L}}R_{\mathcal{C}_{i-1},r}^T \hat{R}_{i-1}^T \hat{R}_{i-2}^T {}^{\mathcal{L}}R_{\mathcal{C}_{i-1},r}$, the estimate of this deformation provided by the previous iterations of the algorithm, and the estimate of \mathbf{v}_{O_i} computed at the previous iteration, we obtain the following formula for the velocity of point O_i :

$$\hat{\mathbf{v}}_{O_i} = {}^{\mathcal{P}_i}R_{\mathcal{C}_{i-1}}\hat{D}_{i-1}\hat{\mathbf{v}}_{O_{i-1}} + [\boldsymbol{\omega}_{\mathcal{P}_i/IMU_{i-1}} + {}^{IMU_{i-1}}R_{\mathcal{P}_i}^T \boldsymbol{\omega}_{IMU_{i-1}}]_{\times}\mathbf{p}_i + \dot{\mathbf{p}}_i \quad (16)$$

Note that the notation $\hat{\mathbf{v}}_{O_i}$ is simply used to emphasize the fact that this quantity depends on the estimated deformation \hat{D}_{i-1} . This velocity is not a new state estimate, but simply an algebraic computation of the velocity along the (flexible) kinematic chain of the robot, taking into account sensor measurements and tilt estimates.

2) *Step 2: estimating the tilt of IMU i :* To use (12) on IMU i , we need a measurement of the linear velocity of this IMU. 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 . Performing the corresponding substitutions into (16) yields

$$\mathbf{v}_{IMU_i} = {}^{IMU_i}R_{\mathcal{C}_i}D_i\hat{\mathbf{v}}_{O_i} + [\boldsymbol{\omega}_{IMU_{i-1}}]_{\times}\mathbf{p}_{IMU_i/O_i} + \dot{\mathbf{p}}_{IMU_i/O_i} \quad (17)$$

where \mathbf{p}_{IMU_i/O_i} is the position of IMU i with respect to O_i in the IMU frame.

Unlike (16) however, (17) depends on D_i , i.e. the tilt of IMU_i , which is precisely what we are trying to estimate. This dependency implies that a true measurement of this velocity cannot be obtained independently of the tilt - except for the first IMU, where $\mathbf{v}_{O_0} = 0$.

To overcome this, we simply use the fact that D_i , which represents a mechanical deformation of the robot, is a small rotation (typically only a few degrees). We thus neglect this deformation, i.e. set $D_i = I$ in (17), and compute an *approximate* velocity measurement using the velocity provided by (16)

$$\bar{\mathbf{v}}_{IMU_i} = {}^{IMU_i}R_{\mathcal{C}_i}\hat{\mathbf{v}}_{O_i} + [\boldsymbol{\omega}_{IMU_{i-1}}]_{\times}\mathbf{p}_{IMU_i/O_i} + \dot{\mathbf{p}}_{IMU_i/O_i} \quad (18)$$

This reconstructed signal is then used in the observer (12) to obtain the tilt of IMU i , thereby concluding step 2.

To summarize, MOVIE consists in applying $n + 1$ times the observer (12) to estimate the deformations, each time using data from the previous observers to propagate the velocity to the next IMU. This propagation is done by neglecting each time the last deformation in (18). This simplification introduces an additive error term, which is bounded. The local exponential convergence property of the observer, proven in [1], suggests that practical stability could be established [13, Theorem XVIII p.123], i.e. the error is asymptotically arbitrarily small (by using large gains). This theoretical study is however out of the scope of this paper. Indeed, in practice, such high gains are not necessary: this is illustrated in the next section on the exoskeleton Atalante.

IV. EXPERIMENTAL RESULTS IN OPEN LOOP

A. Description of the experimental setup

The exoskeleton Atalante, depicted in Figure 1, is a lower-limb medical exoskeleton. Self-balanced, it enables paraplegics and hemiplegics to walk without crutches.

Having to carry its own weight together with that of the patient, Atalante exhibits a significant amount of deformations: for instance, when walking, the swing foot is typically 3 cm lower than expected by the rigid model. A direct observation of the robot using motion capture, acting as ground truth for all the experiments considered here, shows that a good pose reconstruction (with an accuracy of less than 5 mm) can be obtained by considering four punctual deformations: one at each ankle, and one at each hip (as shown in Figure 2). Thus, the exoskeleton is composed of 5 rigid sets: following the methodology advocated in this article, 5 low-cost IMUs (LSM6DSO) are mounted on the robot, one in each foot, one in each tibia and one in the robot's pelvis. Finally, the robot is equipped with 4 1D force sensors at the corner of each foot, enabling the measurement of the center of pressure. All sensors are sampled at 1 kHz.

In this section, we assess the performance of MOVIE, both around a static position and while walking, using motion capture as ground truth. The following gains are used: $(\alpha, \beta) = (0.75, 0.057)$ for the support foot IMU, and $(1.5, 0.229)$ for the other IMUs. We compare this observer to the use of the rigid, flat-footed model (1) - named RigidModel - and to our previous work [5], where we use a simple complementary filter [3] to reconstruct the tilt of the IMUs, neglecting the linear accelerations. We refer to this observer as the *ComplementaryFilter*².

For both observers, alignment biases between motion capture and IMU output (due to sensor mounting tolerance and accelerometer bias) are identified during a zero-load preliminary experiment. Furthermore, MEMS gyroscope are known to suffer from significant additive bias. While this bias is estimated by the *ComplementaryFilter*, this is not the case

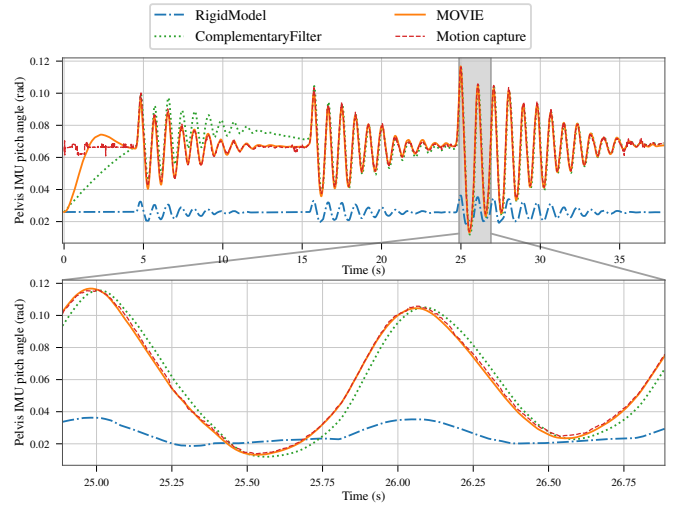


Fig. 3. Pelvis IMU pitch angle estimation during a static experiment. MOVIE converges much faster than the *ComplementaryFilter*, while being robust to the presence of linear acceleration during the oscillations, which induce a phase shift for the *ComplementaryFilter*.

for MOVIE. However, we observe that this bias varies only very little, even over periods of several months. Thus, we simply compute this bias on a preliminary static experiment and subtract it on all latter experiments.

B. Experiment around a static position

A simple experiment around a static position is enough to stress the superior behavior of MOVIE over the *ComplementaryFilter*. In this experiment, shown in the associated video at [14], the empty exoskeleton is placed in single support, in a posture similar to that of Figure 1 (right), and tasked to remain motionless. The exoskeleton is then repeatedly pushed in the back, which triggers the deformations, resulting in slowly damped oscillations. Figure 3 shows the resulting pitch angle of the pelvis IMU - the roll angle is not shown as there is less excitation in the frontal plane.

Without surprise, neglecting the deformations by using the RigidModel yields a significant static error, and does not capture well the oscillations happening when the robot is pushed: these oscillations mostly come from the deformations, while the RigidModel only sees a moderate oscillation on the encoders, resulting from low-level controller tracking error.

As for the observers (both of which are initialized to match the tilt of the RigidModel), MOVIE converges in about 4 s, and then gives an extremely accurate pitch estimate, with less than 0.002 rad of error. By contrast, the *ComplementaryFilter* exhibits a much slower convergence of about 25 s. Furthermore, during the oscillations, this observer suffers from a significant phase shift of 10° at 1 Hz. These limitations are a consequence of neglecting the linear acceleration in the observer formulation. Indeed, the oscillations of the deformations generate a forward acceleration, in quadrature with the angular velocity. This causes a phase shift of the estimate, as the accelerometer is used for attitude feedback in the *ComplementaryFilter*. To limit the value of the phase shift, i.e. the sensitivity of this observer to the linear acceleration,

²Note that at the time of [5], Atalante was not equipped with IMUs in the foot, thus a flat-foot hypothesis was formulated. This hypothesis is here relaxed, and the same complementary filter [3] is used for the foot IMUs

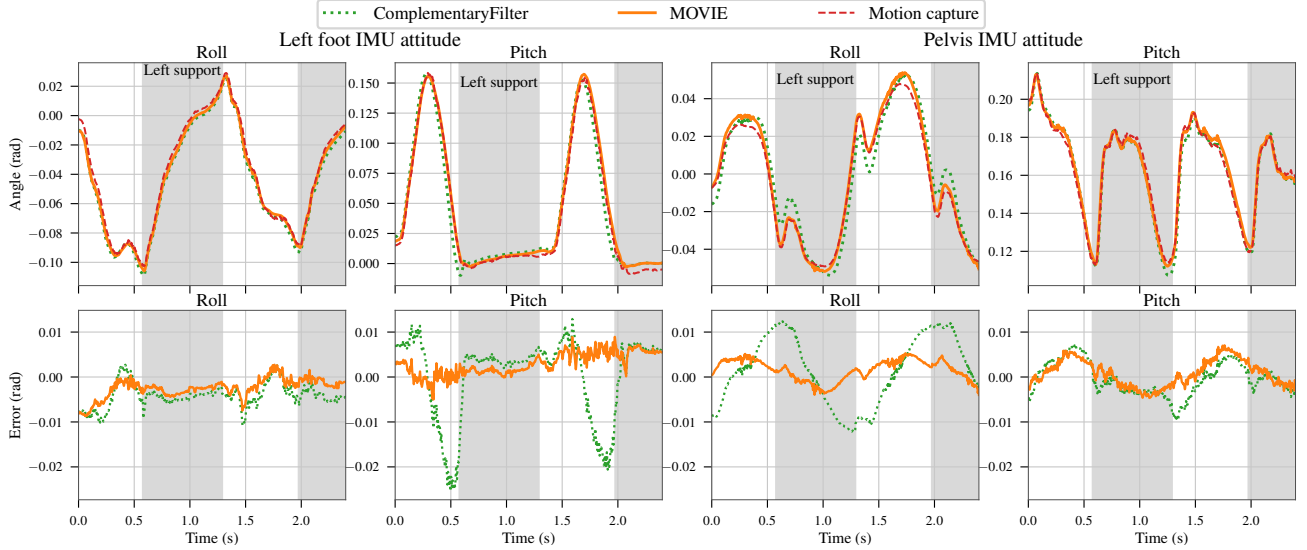


Fig. 4. Roll and pitch estimate of the pelvis IMU and left foot IMU while walking, and corresponding error.

one needs to use low observer gains, which explains the slow convergence. Reducing this phase shift would require even lower gains, at the cost of a slower convergence. By contrast, MOVIE, by using a velocity measurement to separate the sensor linear acceleration from the tilt, achieves both fast convergence and accurate reconstruction of the oscillations.

C. Dynamic walking

We now consider the use case of main practical interest for Atalante: dynamic walking with a user. Once again, MOVIE outperforms the ComplementaryFilter, which remains limited by the presence of linear accelerations. This is illustrated in Figure 4, which shows the roll and pitch estimates of the pelvis and left foot IMU. The impact of acceleration on the ComplementaryFilter is particularly visible in the pattern of error for the left foot IMU pitch. During the stance phase (gray background), the left leg moves only very little, and the ComplementaryFilter gives an accurate attitude estimate. However, during the swing phase, the leg is strongly accelerated forward: this results in a significant phase shift, just like in the static case, causing an error of up to 0.025 rad. By contrast, MOVIE remains insensitive to this effect, giving a more accurate estimate, with a root mean square error (RMSE) of only 0.005 rad. Likewise, the ComplementaryFilter is negatively affected by the lateral accelerations of the pelvis, as the exoskeleton transfers its weight from one leg to the next, resulting in a larger roll error than MOVIE.

The performance of these observers can also be evaluated by looking at the reconstruction of the body pose, computed using the *extended flexible model* (3). In Figure 5, we show the reconstruction of the position of the external toe of the swing foot, in the local frame. As expected from the accurate tilt estimation it provides, MOVIE efficiently reconstructs this position, with a RMSE of less than 1 cm on each axis (this error being mostly caused by the accuracy of the extended flexible model, which assumes punctual deformations³). Meanwhile, the ComplementaryFilter leads

³This can be readily checked using motion capture to compute “perfect” estimations of the tilts.

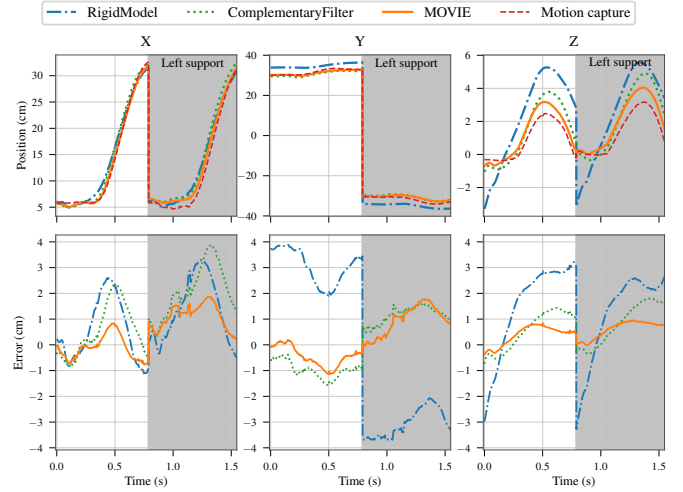


Fig. 5. Position of the swing foot external toe, in the local frame. X is a forward-pointing axis, and Z points upward.

to a much larger error, in particular on the X coordinate (related to the large pitch error on the swing foot, as shown in Figure 4), where the error is as large as that of the RigidModel, which does not consider any deformations.

V. EXPERIMENTAL RESULTS IN CLOSED-LOOP

In the previous section, we have shown that MOVIE provides an accurate estimation of Atalante’s deformations, both in static and when walking. In this section, we show how to use the estimation provided by MOVIE in feedback control, with the aim of compensating the negative effect of deformations on trajectory tracking accuracy.

A. Controller design

Our approach is directly derived from our previous work [15], which we briefly summarize here for completeness.

1) *Previous controller design for single support experiments*: As presented in [15] Atalante follows a *position-control* paradigm: a high-level controller computes a joint trajectory $q^* \in \mathbb{R}^{12}$, which is then tracked by a *low-level*

controller, using a high-gain decentralized PID:

$$\mathbf{u} = -K_p(\boldsymbol{\theta} - \mathbf{q}^*) - K_d(\dot{\boldsymbol{\theta}} - \dot{\mathbf{q}}^*) - K_i \int (\boldsymbol{\theta} - \mathbf{q}^*) \quad (19)$$

with K_p , K_d , K_i positive diagonal gain matrices.

The *extended flexible model* (3), used for the observer, introduces additional degrees of freedom (the spherical deformations), non-collocated with the actuators: this significantly complexifies the control problem. Thus, to design a flexibility-compensation controller, we instead use an alternative model to represent the deformations: the classical *flexible joint model* [16], which considers 1D deformations about the robot joints. More precisely, this model considers that the angle at the output of the transmission, $\mathbf{q} \in \mathbb{R}^{12}$, is no longer equal to the input (motor) angle $\boldsymbol{\theta}$. The deformation $\boldsymbol{\alpha} \triangleq \mathbf{q} - \boldsymbol{\theta}$ is attributed to a compliant element in the transmission, following a linear spring-damper model:

$$\boldsymbol{\tau} = -K\boldsymbol{\alpha} - \nu\dot{\boldsymbol{\alpha}} \quad (20)$$

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

An estimate of the angle \mathbf{q} is obtained from MOVIE, simply by decomposing each deformation of the *extended flexible model* (3D rotation) into successive 1D rotations using Euler angles. Thus, the resulting *flexible joint model* yields the same IMU orientation than the *extended flexible model*. The positions however are slightly different, since the rotations no longer happen about the same center (Atalante's hip, though composed of three orthogonal revolute joints, is not a true spherical joint as the axes are not concurrent). This difference typically amounts to only 5 mm, and is thus deemed acceptable, though it limits the accuracy of the controller.

Our controller stems from the natural extension of the decentralized, high-gain approach of (19) to include a feedback on both the rigid and flexible angle of the *flexible joint model*, as suggested in [17]:

$$\begin{aligned} \mathbf{u} = & -K_q(\mathbf{q} - \mathbf{q}^*) - K_{dq}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) - K_i \int (\mathbf{q} - \mathbf{q}^*) \\ & - K_\alpha(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - K_{d\alpha}(\dot{\boldsymbol{\alpha}} - \dot{\boldsymbol{\alpha}}^*) \\ & + \mathbf{u}_{ff} \end{aligned} \quad (21)$$

where K_q , K_{dq} , K_i , K_α , $K_{d\alpha}$ are positive diagonal gain matrices, $\boldsymbol{\alpha}^*$ is the trajectory of the deformations, and \mathbf{u}_{ff} a feedforward torque.

One difficulty in implementing (21) however is the computation of $\boldsymbol{\alpha}^*$, which to be done exactly requires an accurate dynamical model of the system and of the deformation. In practice however, the presence of an uninstrumented user on Atalante renders this computation quite complex; the linear dynamics of the flexibility (20) is also only an approximation. Instead, we take a simpler solution, and set $\boldsymbol{\alpha}^* = 0$, considering that the integral term will asymptotically compensate for the resulting static error.

As compared to our previous work [15], we also remove the feedforward term \mathbf{u}_{ff} , which is reasonable considering

that high-gains are used on the feedback part of the controller. Thus the corresponding controller writes:

$$\begin{aligned} \mathbf{u} = & -K_q(\mathbf{q} - \mathbf{q}^*) - K_{dq}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) - K_i \int (\mathbf{q} - \mathbf{q}^*) \\ & - K_\alpha\boldsymbol{\alpha} - K_{d\alpha}\dot{\boldsymbol{\alpha}} \end{aligned} \quad (22)$$

To compute the gains of (22), we consider the robot in single support. Neglecting joint coupling, the *flexible joint model* corresponds to modeling each joint as a 1D series-elastic actuator carrying a load equal to the total mass and inertia of the bodies downstream of this joint. We then linearize this dynamics and perform LQR tuning⁴.

2) *Controller modification for walking gaits*: The controller (22) was designed with single support in mind, and does not take into account the hybrid nature of the walk, which consists of alternating stance and swing phases, with intermediate double support phases. To handle this effect, we design a hybrid controller by adding the following modifications:

- *Gain scheduling*: For the ankle joints, we use two sets of gains: one for the stance phase, one for the swing phase. Indeed, the dynamics of both ankles are drastically different: while the support leg ankle carries the entire weight of the system (about 150 kg), the swing ankle only carries the swing foot (< 5 kg). The force sensors are used to determine if the foot is in contact or not, and change the gain accordingly. No such gain modification is done for the hip joints, as the difference between stance and swing phase is not as drastic.

- *Integrator switching*: Both for the hips and ankles, the value of the integral is computed separately for the swing phase and the stance phase, to capture the difference in feedforward between both phases (and in particular, the difference in terms of deformation, as the integral is meant to capture the average value of $\boldsymbol{\alpha}^*$). Mathematically, this writes

$$\begin{aligned} \mathbf{u} = & -K_\theta(\boldsymbol{\theta} - \mathbf{q}^*) - K_{d\theta}(\dot{\boldsymbol{\theta}} - \dot{\mathbf{q}}^*) \\ & - K_q(\mathbf{q} - \mathbf{q}^*) - K_{dq}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) \\ & - K_i \left(\mathbb{1}_{sw} \int_t [\mathbb{1}_{sw}(s)(\mathbf{q} - \mathbf{q}^*)(s)ds] \right. \\ & \left. + (1 - \mathbb{1}_{sw}) \int_t [(1 - \mathbb{1}_{sw})(s)(\mathbf{q} - \mathbf{q}^*)(s)ds] \right) \end{aligned} \quad (23)$$

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.

B. Feedback control experimental results

In Figure 6 we compare the performance of the traditional PID controller of Atalante (19), which we call the *rigid controller*, against our *flexibility compensation controller* (23). In this experiment, shown in video at [14], a 55 kg dummy is placed on the exoskeleton: using a dummy enables for a reproducible setup by preventing any user-induced

⁴Note that, while the methodology is the same as in [15], different weights are used in the LQR problem to obtain a more reactive controller, suitable for fast walking gaits.

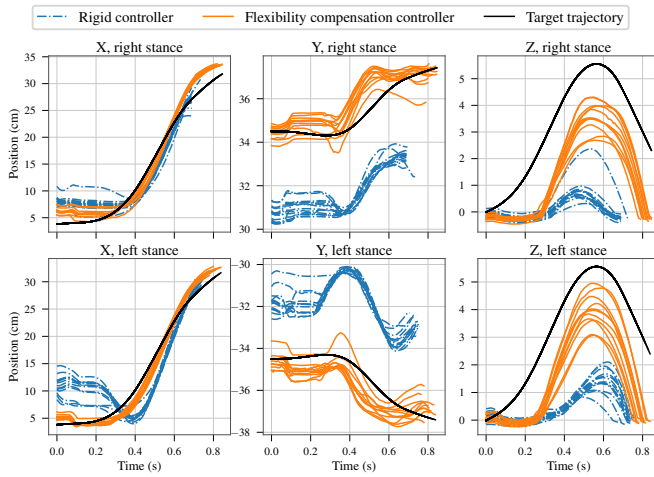


Fig. 6. Comparison of the motion of the swing foot in the local frame, as seen by motion capture, when using the rigid controller (blue) and the flexibility compensation controller (orange). Each curve is a single step, and the black curve represents the desired trajectory.

motion. The exoskeleton is then tasked to walk using either *i*) the rigid controller or *ii*) the flexibility compensation controller, performing feedback on MOVIE. Motion capture is used to access the resulting tracking precision of our observer-controller.

Using the flexibility compensation controller significantly reduces the tracking error committed by the rigid controller. In particular, the swing foot now rises much higher, culminating at 4 cm at the apex of the trajectory: by contrast, the rigid controller barely manages to lift the right foot off the ground, with a ground clearance of only 1 cm.

This improved tracking leads to a greater stability and robustness of the walk. This effect is well illustrated by replacing the 55 kg dummy by a heavier, 80 kg dummy, as shown in the associated video [14]. Then, the rigid controller is no longer able 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.

VI. CONCLUSION

In this paper, we presented a novel observer for estimating the state of a biped robot with multiple punctual deformations while performing accelerated motions such as walking. This observer, named MOVIE, uses a “velocity-aided approach” to estimate the tilt of the IMUs. It exploits the kinematic model of the robot, accounting for its deformations, to build an approximate measurement of velocity. This measurement is used for robust tilt estimation, then propagated forward in a cascade approach.

We benchmark this observer on Atalante, where it gives a very accurate tilt estimate, with RMSE of only 0.005 rad. In particular, this observer outperforms our previous approach, which relies on a zero-on-average acceleration hypothesis. This observer is then used for feedback control of the position, in a decentralized, high-gain approach. This flexibility compensation controller is successively tested on walking gaits, where it improves trajectory tracking and therefore

walk stability, in particular enabling the exoskeleton to walk with a 80 kg dummy.

In the future, we aim at removing the need for motion-capture based calibration, by looking for a solution to identify sensor bias without needing external measurement. Furthermore, the robustness of the observer to strong linear accelerations is a significant asset to enable the realization of faster walking gaits on Atalante. Indeed, our goal is to reach human-like walking speeds of 1 m/s (3.6 km/h), while the walk featured in this article happens at the modest speed of 0.25 m/s (0.9 km/h).

REFERENCES

- [1] 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 International Conference on Humanoid Robotics (Humanoids)*, Nov. 2017, pp. 830–835.
- [2] J. L. Crassidis, F. L. Markley, and Y. Cheng, “Survey of nonlinear attitude estimation methods,” *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.
- [3] R. Mahony, T. Hamel, and J. Pflimlin, “Nonlinear Complementary Filters on the Special Orthogonal Group,” *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1217, June 2008.
- [4] 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*. Singapore: World Scientific, 2013, pp. 822–830.
- [5] M. Vigne, A. El Khoury, F. Di Meglio, *et al.*, “State Estimation for a Legged Robot With Multiple Flexibilities Using IMUs: A Kinematic Approach,” *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 195–202, 2020.
- [6] M. Bloesch, M. Hutter, M. Hoepfner, *et al.*, “State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU,” in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.
- [7] R. Hartley, M. Jadidi, J. Grizzle, *et al.*, “Contact-Aided Invariant Extended Kalman Filtering for Legged Robot State Estimation,” in *Proceedings of Robotics: Science and Systems*, May 2018.
- [8] M. Benallegue, R. Cisneros, A. Benallegue, *et al.*, “Lyapunov-Stable Orientation Estimator for Humanoid Robots,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6371–6378, Oct. 2020.
- [9] A. De Luca and W. Book, “Robots with Flexible Elements,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 287–319.
- [10] S. Moberg, “Modeling and control of flexible manipulators,” Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Linköping.
- [11] S. Bonnabel, P. Martin, and E. Salaün, “Invariant Extended Kalman Filter: theory and application to a velocity-aided attitude estimation problem,” in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 1297–1304.
- [12] M.-D. Hua, P. Martin, and T. Hamel, “Stability analysis of velocity-aided attitude observers for accelerated vehicles,” *Automatica*, vol. 63, pp. 11–15, Jan. 2016.
- [13] J. La Salle and S. Lefschetz, *Stability by Liapunov’s direct method: with applications*, 5th ed., ser. Mathematics in science and engineering. New York [u.a.]: Acad. Press, 1973, no. 4.
- [14] M. Vigne, A. El Khoury, M. Pétriaux, *et al.* Video of the experiments on Atalante. [Online]. Available: <https://youtu.be/nYig0QxoVzk>
- [15] M. Vigne, A. E. Khoury, F. Di Meglio, *et al.*, “Improving low-level control of the exoskeleton atalante in single support by compensating joint flexibility,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3437–3443.
- [16] M. W. Spong, “Modeling and Control of Elastic Joint Robots,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 4, pp. 310–318, Dec. 1987.
- [17] 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)*, vol. 2. Takamatsu, Japan: IEEE, 2000, pp. 1087–1093.