2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Improving Low-Level Control of the Exoskeleton Atalante in Single Support by Compensating Joint Flexibility

Matthieu Vigne[1,2], Antonio El Khoury[1], Florent Di Meglio[2] and Nicolas Petit[2]

Abstract— **This paper describes a novel low-level controller for the lower-limb exoskeleton Atalante. The controller implemented on the commercialized product Atalante works under the assumption of full rigidity, performing position control through decentralized joint PIDs. However, this controller is unable to tackle the presence of flexibilities in the system, which cause static errors and undesired oscillations. We modify this controller by leveraging estimations of the position and velocity of the flexibilities, readily available on Atalante through the use of strapdown IMUs. Instead of considering feedback on the motor position only, we perform feedback on both the joint position and the flexibility angle, keeping a decentralized approach. This enables compensation of both the static error present at rest, and rapid damping of the oscillations. To tune the gains of the proposed controller, we use a linearized model of an elastic joint to which we apply a steady-state LQR, which creates desirable robustness to the flexible model. The proposed controller is experimentally validated through various single support experiments on Atalante, either empty or with a user. In all cases, the proposed controller outperforms the state-of-the-art controller, providing improved trajectory tracking and disturbance rejection.**

I. Introduction

The control of a bipedal robot is a challenging task: for instance, in single support, the center of pressure has to be kept precisely under the stance foot to prevent the robot from tipping over. In this context, having an accurate kinematic model for estimation and control is critical to remain stable [1]. One source of vast uncertainty is the presence of mechanical flexibilities [2], typically modeled as linear springs. Not only do they modify the kinematics, but they also introduce undesired oscillations that may in turn destabilize the system.

Compensating joint flexibilities has been broadly studied on robotic manipulators since the 1980s [3]. One of the first proposed approaches is feedback linearization: in [4], the dynamics of a robotic manipulator with linear joint elasticity is proven to be feedback linearizable. However, this control method proves to be quite sensitive to model parameters and uncertainties [5], and is thus difficult to implement on real hardware, requiring accurate system identification and modeling. To alleviate this problem, other controllers have been proposed as a modification of the classical linear controller for rigid systems: a feedforward term and an independent joint feedback in the form of a proportional-derivative (PD) controller. This controller is well-known for its robustness [6], and is often used as a low-level controller for a position-control robot. Such a modification that compensates the static error due to the deflection of the flexibilities is presented in [7]: a gravity-compensation term is used to generate an offset on the feedback PD target. While this additional term yields the correct steady-state position, it does not counteract the effects of the dynamics of the flexibilities. This approach is refined in [8], replacing the feedforward term by a feedback gravity compensation term, improving trajectory tracking. Interestingly, these controllers are designed to work with only motor measurements, without explicit measurements of the flexibility. The interested reader may refer to [9] for a comprehensive survey of controlling manipulators in the presence of flexibilities.

In the context of legged robots, the gravity compensation approach of [7] has been applied to compensate flexibilities, notably on DURUS in [1] and on THORMANG in [10]. This strategy has the advantage of being compatible with existing joint position controllers, common on humanoid robots, but again does not prevent the robot from oscillating due to the dynamics of the flexibilities. One approach to attenuate the oscillations without modifying the robot joint controllers is presented in [11], where the target position for each joint is computed through inverse kinematics from a desired acceleration of the center of mass. The latter, in turn, is determined by considering it as the actuator of a simplified linear model of a single flexibility, and applying LQR design. This cascade approach however assumes that effective low-level controllers are available for trajectory tracking: when this is not the case, the performance of the high-level controller will be severely limited by the low-level loops. More recently, [12] proposed inverse dynamics as feedback linearization to compensate joint elasticity on a torque-controlled humanoid robot.

However, none of the above methods are directly applicable to the lower limb exoskeleton Atalante, depicted in Figure 1, which contains flexibilities that are currently unaccounted for in the control design. Indeed, when working on an exoskeleton, the presence of a user introduces large uncertainties in the dynamics. For instance, by moving his upper body, the user can shift the center of mass of the system, in a manner that is very difficult to capture using a model. This model uncertainty severely limits the performance of an inverse dynamics based framework. It also hinders the previously presented gravity compensation technique, since it relies on knowledge of the position of the center of mass for correcting joint angles.

The main contribution of this paper is the design of a

[1]Control-Command team, Wandercraft, Paris, France matthieu.vigne@mines-paristech.fr
[2]Centre Automatique et Systèmes, MINES ParisTech, PSL Research University, France

specific flexibility compensation controller for Atalante. We use a decentralized control approach to obtain robustness to uncertainties in the dynamics. To improve performance, we leverage estimates of the angle and velocity of the flexibilities, provided by inertial measurement units (IMUs). We formulate a local feedback controller on a fourth-order linear model of an elastic joint, and use LQR design to compute the feedback gains. This controller is experimentally benchmarked in single support against a joint PID controller that ignores the flexibilities, and is shown to provide both better disturbance rejection and trajectory tracking.

The paper is organized as follows. In Section II we give a description of Atalante, and model its dynamics. In Section III, we analyze existing state-of-the-art controllers and present our flexibility compensation controller. Finally, Section IV reports experimental results which stress the performance of this controller, both on an empty exoskeleton and with a user.

II. Description of Atalante

Atalante contains 12 actuated degrees of freedom, each instrumented by a joint encoder. In the context of this paper, we consider that the robot is standing in single support, with a stance foot flat on the ground: then, the system becomes similar to a fixed-base robotic manipulator. The system is position controlled using independent joint PID controllers [13]. This is a classical choice for controlling rigid robots [6], as it proves robust to uncertainties and changes in the dynamics. It is also easier to develop and tune than a whole-body controller [14], as each joint can be individually tuned on a test bench. Indeed, the current PID-based controller has been successfully tuned to enable stable dynamic walking of Atalante.

However, like many humanoid robots, Atalante is not fully rigid. This can be seen in Figure 1: while each motor reaches its target position $\boldsymbol{\theta}^*$, the resulting exoskeleton position (in orange, measured by motion capture) does not match the expected position (in blue, predicted by a fully rigid model): there is a mismatch of about 4 cm in the position of the flying foot. Furthermore, when a disturbance is applied - for instance by the user moving inside the exoskeleton - the robot oscillates, indicating the presence of an elastic behavior. To enhance user experience, we need to improve the tracking performance of Atalante, and in particular, quickly damp the oscillating behavior. Thus, we are interested not only in compensating the static effect of a deformation, but also the induced dynamics - while providing tracking performance at least equivalent to that the current PID controller.

Motion capture analysis shows that the deformations happen not in the structure of the exoskeleton, but is rather localized at the robot joints. Thus, we model Atalante as a set of rigid bodies having flexible revolute joints. Following [16], we denote $\boldsymbol{q}$ the link angle (i.e. the angle taken after the flexibility), and $\boldsymbol{\theta}$ the motor angle: the deformation induced by the flexibility is thus defined as

$$\boldsymbol{\alpha} \triangleq \boldsymbol{q} - \boldsymbol{\theta} \tag{1}$$

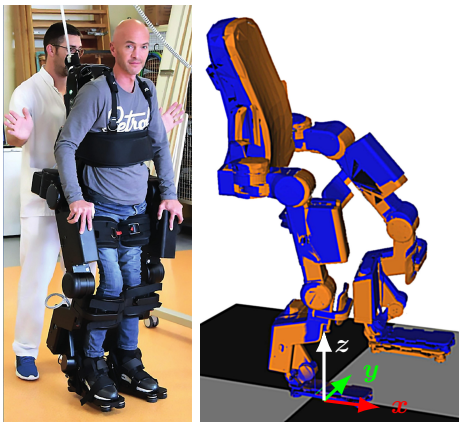

Fig. 1. From [15]: picture of the exoskeleton Atalante. On the right, CAD reconstruction under the assumption of full rigidity (blue), and ground truth from motion capture (orange).

The angles $\boldsymbol{\theta}$ are accurately measured by joint encoders. Since we desire to control the dynamics of the flexiblity $\boldsymbol{\alpha}$, we further instrument the system by using several IMUs to provide an estimate of the link variables $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. Indeed, IMU sensors are now cheap (those used on Atalante are low-grade MEMS sensors costing no more than a few dollars), and easy to integrate in a robot hardware. More details on performing flexibility estimation using IMUs are given in Section IV, the important point to keep in mind for controller design being that the full state ($\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$) is available for feedback.

To derive the dynamics of the flexibilities, we utilize the fact that each rotor is symmetric about its rotation axis. Then, the dynamics of the robot in single support writes [3]

$$\begin{pmatrix} M(\boldsymbol{q}) & 0 \\ 0 & I_m \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{\theta}} \end{pmatrix} + \begin{pmatrix} C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \\ 0 \end{pmatrix} + \begin{pmatrix} -\boldsymbol{\tau}_f(\boldsymbol{\alpha},\dot{\boldsymbol{\alpha}}) \\ \boldsymbol{\tau}_f(\boldsymbol{\alpha},\dot{\boldsymbol{\alpha}}) \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{u} \end{pmatrix} \tag{2}$$

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, located between the rotor and the robot links. It results from the deformation of the transmission elements, classically modeled as a pure elasticity, or a spring-damper system, and writes

$$\boldsymbol{\tau}_f(\boldsymbol{\alpha},\dot{\boldsymbol{\alpha}}) = -K\boldsymbol{\alpha} - \nu\dot{\boldsymbol{\alpha}} \tag{3}$$

Note that on Atalante, this naive linear model is inaccurate: the real dynamics of the flexibilities is poorly known. This limits its relevance for the control design, as explained in the following section.

III. Controller design

A. Discussion on state-of-the-art controllers

As mentioned earlier, several solutions for controlling a system characterized by (2) and (3) have already been proposed. A first family of solutions uses feedback linearization.

In the absence of viscous damping in (3), a feedback linearization controller is proposed in [9, Equation (13.45)] as

$$\begin{aligned}\boldsymbol{u} = I_m K^{-1} \Big(& M(\boldsymbol{q})\boldsymbol{v} + \ddot{M}(\boldsymbol{q})\ddot{q} + 2\dot{M}(\boldsymbol{q})\boldsymbol{q}^{(3)} \\ & + \frac{d^2}{dt^2}\left(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\right) \Big) \\ & + \left(M(\boldsymbol{q}) + I_m\right)\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\end{aligned} \tag{4}$$

where $\boldsymbol{v}$ is the new control input.

However, feedback linearization controllers are known to be quite sensitive to modeling errors [5]: for instance, computing the control law (4) requires the second derivative of the terms of (2). This computation is not suitable in the case of an exoskeleton, where motion of the user introduces large unknown variations in the dynamics. Furthermore, part of the feedback term is multiplied by the inverse of the joint torque model, K^{-1}. Yet on Atalante, modeling a flexibility as a linear spring is vastly erroneous. When trying to identify a stiffness parameter based on the frequency of the oscillations, the identified value changes based on the configuration of the robot, and also from one prototype to the other. This uncertainty makes this method difficult to implement even on an empty exoskeleton.

Another widely used approach, mentioned earlier, is gravity compensation, which aims at correcting only the static effect of the flexibilities when the system is at rest. Considering an existing, stabilizing decentralized PD controllers on each joint, the idea is to modify the feedforward term to reach the correct equilibrium position. Indeed, at rest, (2) yields

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_f(\boldsymbol{\alpha}_e, 0) \tag{5}$$

and thus using the linear spring model (3) yields the following deformation at rest

$$\boldsymbol{\alpha}_e(\boldsymbol{q}) = -K^{-1}\boldsymbol{g}(\boldsymbol{q}) \tag{6}$$

Thus, considering a target link position $\boldsymbol{q}^*$, one can simply give as target to a motor PD controller the gravity-biased value $\boldsymbol{q}^* - \boldsymbol{\alpha}_e$ instead of $\boldsymbol{q}^*$. This is originally proposed in [7, Equation (21)] as

$$\boldsymbol{u} = -K_p(\boldsymbol{\theta} - (\boldsymbol{q}^* - \boldsymbol{\alpha}_e(\boldsymbol{q}^*))) - K_d\dot{\boldsymbol{\theta}} + \boldsymbol{g}(\boldsymbol{q}^* - \boldsymbol{\alpha}_e(\boldsymbol{q}^*)) \tag{7}$$

A similar approach has been used on humanoid robots in [1] and [10]. Note however that this controller (7) also relies on the inverse K^{-1}: although this is less critical than for the feedback linearization controller (4), as it does not impact the stability, an incorrect value of K introduces a bias in the feedforward term $\boldsymbol{\alpha}_e(\boldsymbol{q}^*)$. A similar bias is obtained if the value of the gravity torque $\boldsymbol{g}(\boldsymbol{q}^*)$ is incorrect. This is often the case on an exoskeleton, as by moving his torso the user displaces the center of mass of the system by several centimeters, drastically changing the value and even the sign of the gravity torque. Thus, computation of an accurate bias angle $\boldsymbol{\alpha}_e(\boldsymbol{q}^*)$ would at least require instrumentation of the user. More importantly, this methodology only works to correct the static bias due to the flexibilities. It does not handle the dynamic oscillation that may occur in motion, or when applying a disturbance. A reason for this limitation is that the controller (7) does not use any measurement of the flexible state $\boldsymbol{\alpha}$: it is designed to work with existing hardware where no instrumentation of the flexibility is available. By contrast, we have assumed that this state can be estimated by IMUs, which allows for an alternative controller design.

B. Proposed control structure and feedforward term

The above discussion stresses the need to design a new controller taking into account the specificities of our problem, which are: uncertainty in the stiffness, and parameters of the dynamics in general, availability of extra measurements, and the need to compensate the dynamic effects of the flexibility as well as the static error. To solve this problem, we decide to keep a decentralized control approach, both for robustness and ease of design. Thus, we propose a low-level controller consisting of a feedforward term and a decoupled feedback term. Instead of doing feedback on $\boldsymbol{\theta}$ only, we propose to feedback both $\boldsymbol{q}$ and $\boldsymbol{\alpha}$, and their first derivatives, thereby doing full state feedback. An integral term on $\boldsymbol{q}$ is added, to cancel the static error that would arise due to modeling errors in the feedforward term. This controller writes

$$\begin{aligned}\boldsymbol{u} = & -K_q(\boldsymbol{q} - \boldsymbol{q}^*) - K_{dq}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^*) - K_i \int (\boldsymbol{q} - \boldsymbol{q}^*) \\ & - K_\alpha(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - K_{d\alpha}(\dot{\boldsymbol{\alpha}} - \dot{\boldsymbol{\alpha}}^*) \\ & + \boldsymbol{u}_{ff}\end{aligned} \tag{8}$$

where K_q, K_{dq}, K_α, $K_{d\alpha}$ and K_i are diagonal gain matrices, $\boldsymbol{q}^*$ is the target link position, $\boldsymbol{\alpha}^*$ the target spring angle, and $\boldsymbol{u}_{ff}$ the feedforward term.

This feedforward term is computed using the full dynamics of the system (2). Neglecting the torque due to the rotor acceleration $I_m\ddot{\boldsymbol{\theta}}$ as compared to the torque due to the much heavier links, the feedforward term writes

$$\boldsymbol{u}_{ff} \triangleq M(\boldsymbol{q}^*)\ddot{\boldsymbol{q}}^* + C(\boldsymbol{q}^*, \dot{\boldsymbol{q}}^*)\dot{\boldsymbol{q}}^* + \boldsymbol{g}(\boldsymbol{q}^*) \tag{9}$$

One challenge in implementing (8) lies in computing the value of $\boldsymbol{\alpha}^*$. Using the linear spring model (3), a natural choice would be to use the equilibrium value (6). However, as already mentioned, this value is erroneous on Atalante, or generally on an exoskeleton with a user. Instead, we make a simpler approximation by setting $\boldsymbol{\alpha}^*$ to zero. While this provides the wrong feedforward term in our controller, this effect is compensated by the integrator. Thus, our proposed flexibility compensation controller writes

$$\begin{aligned}\boldsymbol{u} = & -K_q(\boldsymbol{q} - \boldsymbol{q}^*) - K_{dq}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^*) - K_i \int (\boldsymbol{q} - \boldsymbol{q}^*) \\ & - K_\alpha\boldsymbol{\alpha} - K_{d\alpha}\dot{\boldsymbol{\alpha}} \\ & + \boldsymbol{u}_{ff}\end{aligned} \tag{10}$$

C. Feedback gain design

To compute the gain matrices of (10), we use a simplified model for each joint. We consider the robot standing still in single support, and we model the dynamics of joint i to compute the corresponding gain. Given that the controller

structure (10) 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 2.

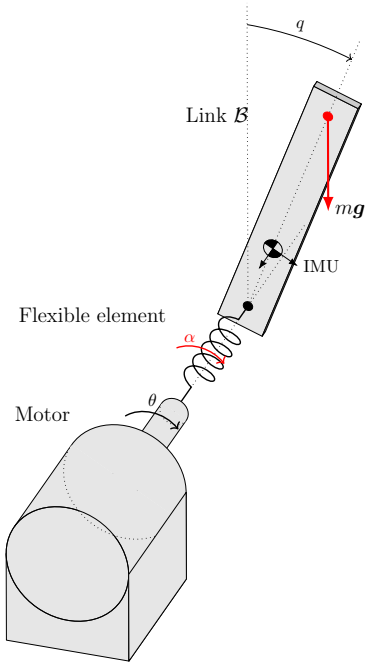

Fig. 2. Model of a single joint of the robot: a flexible element links a body of mass m to the rotor shaft.

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

$$\begin{cases} J\ddot{\alpha} = -u + \nu_m(\dot{q} - \dot{\alpha}) - k(1 + \frac{J}{I})\alpha \\ \qquad\quad - \nu(1 + \frac{J}{I})\dot{\alpha} + \frac{J}{I}mlg\sin(q) \\ I\ddot{q} = -k\alpha - \nu\dot{\alpha} + mlg\sin(q) \end{cases} \tag{11}$$

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 our simplified model (q_0 is in general different from 0). The associated equilibrium state of (11) is

$$\begin{cases} \alpha_0 = \dfrac{mlg\sin q_0}{k} \\ u_0 = -mlg\sin q_0 \end{cases} \tag{12}$$

Linearizing (11) about this equilibrium yields a linear time-invariant system

$$\delta\dot{\boldsymbol{x}} = A\delta\boldsymbol{x} + B\delta u \tag{13}$$

where

$$\boldsymbol{x} \triangleq \begin{pmatrix} \delta q & \dot{\delta q} & \delta\alpha & \dot{\delta\alpha} \end{pmatrix}^T \tag{14}$$

$$A \triangleq \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{mlg\cos q_0}{I} & 0 & -\frac{k}{I} & -\frac{\nu}{I} \\ 0 & 0 & 0 & 1 \\ \frac{mlg\cos q_0}{I} & \frac{\nu_m}{J} & -k\left(\frac{1}{I}+\frac{1}{J}\right) & -\nu\left(\frac{1}{I}+\frac{1}{J}\right) - \frac{\nu_m}{J} \end{pmatrix} \tag{15}$$

$$B \triangleq \begin{pmatrix} 0 & 0 & 0 & -\frac{1}{J} \end{pmatrix}^T \tag{16}$$

The given linear system is controllable, except in the singular case $Ik^2 \neq mlg\cos q_0\nu^2$. Since the state $\delta\boldsymbol{x}$ can be fully estimated, as detailed in the next section, a classical solution to design robust controller is using a steady-state LQR [17]. We use diagonal weighing matrices, which are manually tuned to obtain the desired response for each axis.

IV. Experimental results

The proposed flexibility compensation controller (10) is experimentally tested on Atalante, in various single support task, and compared with a simple motor PID controller. This controller assumes a rigid system, and thus simply uses the target link position $\boldsymbol{q}^*$ as target for the motor variable $\boldsymbol{\theta}$. This writes

$$\boldsymbol{u} = -K_p(\boldsymbol{\theta} - \boldsymbol{q}^*) - K_d(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{q}}^*) - K_i\int(\boldsymbol{\theta} - \boldsymbol{q}^*) \tag{17}$$

We call (17) the rigid controller, and (10) the flexibility compensation controller. Recall that (17) is the controller currently implemented on the commercialized product Atalante[1], and thus is tuned to provide a sufficiently accurate tracking to perform complex motion such as walking or turning.

For all the experimental curves displayed below, the link positions plotted are the one estimated by the onboard IMUs, as detailed in Section IV-A. Plots are done in the fixed world frame, with z the upward vertical axis and x the forward-pointing axis, as depicted in Figure 1. Videos of the experiments are available online at [18].

A. Controller implementation

Our flexibility compensation controller relies on measurements or estimates of the full state of the system, that is $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$. As classically done in robotics, motor angles $\boldsymbol{\theta}$ are measured by joint encoders, and their derivative $\dot{\boldsymbol{\theta}}$ computed through numerical derivation. Instrumentation of the flexibilities is done using several IMUs, composed each of a triaxial gyroscope and accelerometer. Details of the state estimation procedure, and validation against motion capture, can be found in [15]. Since the three hip joints are orthogonal, there exist a unique decomposition of the spherical deformation measured by the IMU into joint deformation, performed using Euler angles. Likewise, a single IMU placed

[1]Note that on the commercial product, for walking gaits, which are outside of the scope of this paper, other balance-control strategies are implemented on top of this low-level controller to prevent the robot from falling.

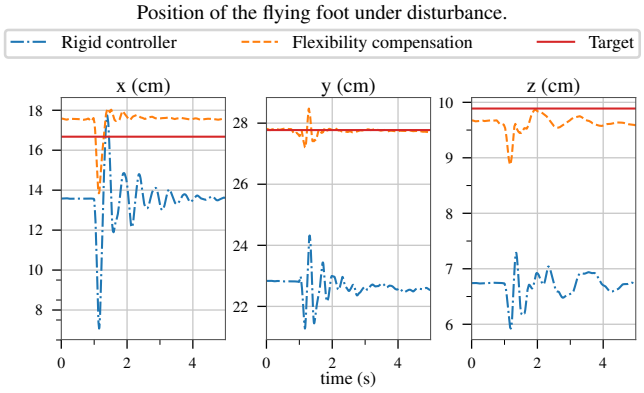

Fig. 3. Position of the flying foot, when using the rigd controller (first experiment, blue) or the flexibility compensation controller (second experiment, orange). The target position is the same, and in both experiment an external force is applied to the flying foot at t=1s.

in the tibia is sufficient to estimate the deformations around both ankle axis. Finally, as shown in [15], the deformation at the knee joint is negligible, as reconstruction using only hips and ankle deformation is enough to match motion capture ground truth. Thus, only three IMUs (one for the support ankle, and one for each hip) are sufficient to provide an estimate of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$.

Gain tuning for the flexibility compensation controller is done using the method presented in Section III-C, taking as model the empty exoskeleton in a standstill single support position. We use the theoretical values of mass and inertia m, I, J given by the CAD model, and identified values of stiffness and damping parameters k, ν and ν_m. This identification is performed on a single experiment, by pushing the exoskeleton and measuring at the resulting oscillation frequency and damping. For the knee joints and the flying ankle, where the deformations are negligible, the rigid controller is kept as-is.

When tuning the gains, two unmodeled phenomena have to be taken into account. First, some of the exoskeleton joint feature a large amount of dry friction. In order to prevent stick-slip from causing oscillations, a deadzone is implemented on the integral term. Second, delays in the control loop 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 implemented on the hardware safely.

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 (10), applied on the whole system, remains stable. To numerically validate our controller, we linearize the dynamics (2) and check that the resulting close-loop dynamics is indeed stable, despite having neglected joint coupling for gain tuning.

B. 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:

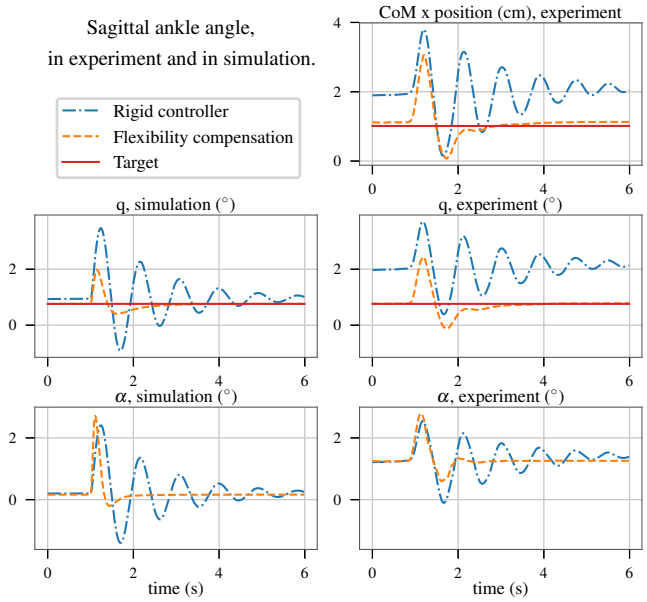

Fig. 4. Sagittal ankle joint position (top) and flexibity angle (bottom). The right column shows experimental results, the left simulation results of the proposed linear model. In both cases, an external force is applied at t=1s.

pushing the flying foot downward and pushing the back of the exoskeleton forward. A similar experiment is carried out for both the rigid controller (17) and the flexibility compensation controller (10).

Figure 3 shows the comparison of both of these experiments when a push is applied to the flying 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 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.

Furthermore, our controller provides a much better damping of the oscillates, which are mostly absorbed in a single period, whereas the rigid controller oscillates for several seconds, and slows down mostly thanks to the natural damping of the system.

Since the disturbances applied are pushes in the sagittal plane, the joint with the most impact on the overall response of the system is the sagittal ankle, which bears the largest load. Figure 4 shows the response of the sagittal ankle flexible joint, when pushing the back of the exoskeleton. 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. The response in joint angle and flexibility angle of the sagittal ankle are compared with simulation of both controllers on the linearized joint model (13) used for gain synthesis. 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 rest: while a linear spring models

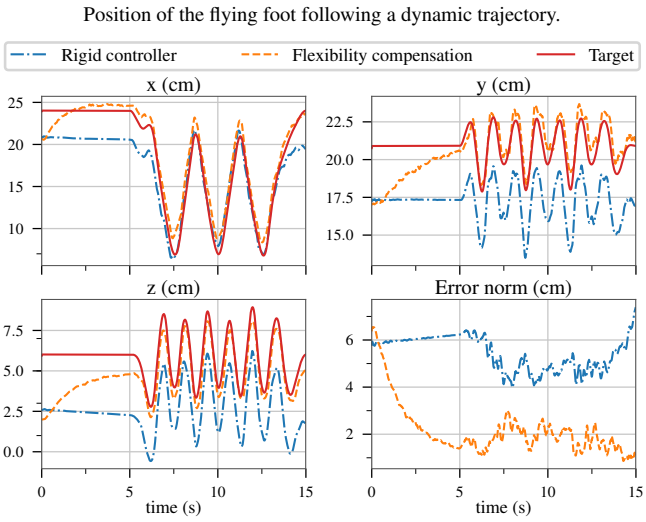

Fig. 5. Tracking of the flying foot in a dynamic single support trajectory.Both controllers are started 5 s before the start of motion.

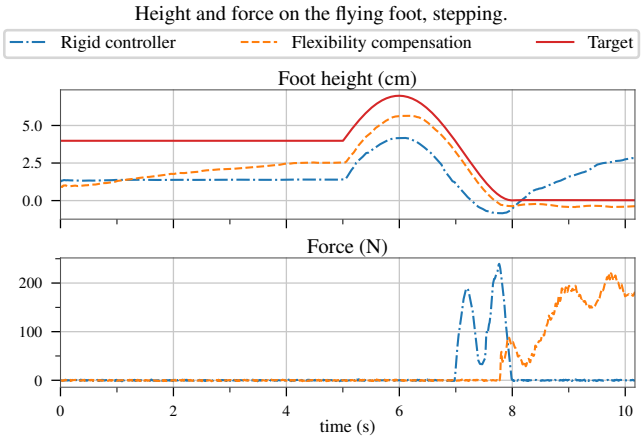

Fig. 6. Height of the flying foot during a stepping trajectory with a dummy, and flying foot force sensor readings showing ground impact. Due to early ground strike, the exoskeleton falls (the force goes back to zero) when using the rigid controller.

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.

C. Trajectory tracking

The trajectory tracking ability of the flexibility compensation controller is also evaluated. For that purpose, we generate offline a dynamically stable trajectory that makes the flying foot oscillate between several set-points, to generate a step-like motion, while keeping the center of pressure within the support foot. In Figure 5, we compare the tracking performance of both controllers. As before, compensating the flexibility leads to a significant reduction of the error, with a average error of 1.7 cm against 5.1 cm for the rigid controller. Recall that a large static error for the PID is to be expected, as the integrator is applied to the rigid state $\boldsymbol{\theta}$ rather than $\boldsymbol{q}$. Both controllers share similar deviation around the reference trajectory: the standard deviation of the error is 0.5 cm for our controller, against 0.7 cm for the rigid controller. Thus, the proposed controller is able to achieve better tracking than the original controller, even on dynamic trajectories.

D. Stepping with a user

Finally, we consider a more realistic scenario: a stepping trajectory with a user inside. Atalante is of course not meant to be used empty, but together with patients with varying morphology. A very convenient property of the rigid controller is the fact that the same controller can be used both on the empty exoskeleton and when bearing any user. This robustness is of great practical importance, 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 6, 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 (see video at [18]). 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 6), 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 (10) 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 enable smooth landing, since our current control framework does not handle impact. 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 push the foot lower, the exoskeleton tips over and falls.

On the contrary, the flexible controller greatly reduces the static error and improves trajectory tracking. Even though the position of the flying 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.

V. Conclusion

This paper presents and experimentally validates an alternative low-level controller for the exoskeleton Atalante. This controller compensates both the static deflection and the dynamic effects of joint flexibilities, using full-state feedback

provided by joint encoders and low-cost MEMS IMUs. Since the presence of a user introduces large uncertainties in the dynamics, we avoid the pitfall of using any model-inversion technique. Robustness is achieved by our controller through the use of decentralized control and LQR design for the gain tuning. This controller is experimentally validated both on an empty exoskeleton and with a user, showing improved trajectory tracking and disturbance rejection when compared to a simple joint PID controller.

Future work includes further study of the theoretical stability of our controller, which has only been numerically evaluated on the nominal system: parametrizing the error in (15) and using LMI could be used to prove its stability even in extreme cases. Furthermore, the proposed controller is designed with single support in mind. In order to generalize this controller, and make it applicable to full walking gaits, double support and phase transition need to be considered, for instance by switching gain values based on the current stance state.

REFERENCES

[1] J. P. Reher, A. Hereid, S. Kolathaya, *et al.*, "Algorithmic Foundations of Realizing Multi-Contact Locomotion on the Humanoid Robot DURUS," *Twelfth InternationalWorkshop on Algorithmic Foundations on Robotics*, 2016.

[2] M. Johnson, B. Shrewsbury, S. Bertrand, *et al.*, "Team IHMC's lessons learned from the DARPA robotics challenge trials," *Journal of Field Robotics*, vol. 32, Mar. 2015.

[3] 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.

[4] A. D. Luca, "Decoupling and feedback linearization of robots with mixed rigid/elastic joints," *Int. J. Robust Nonlinear Control*, p. 13, 1998.

[5] S. Moberg and S. Hanssen, "On Feedback Linearization for Robust Tracking Control of Flexible Joint Robots," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 12 218–12 223, Jan. 2008.

[6] V. Santibañez and R. Kelly, "PD control with feedforward compensation for robot manipulators: analysis and experimentation," *Robotica*, vol. 19, no. 1, pp. 11–19, Jan. 2001.

[7] P. Tomei, "A simple PD controller for robots with elastic joints," *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1208–1213, Oct. 1991.

[8] A. De Luca, B. Siciliano, and L. Zollo, "PD Control with On-line Gravity Compensation for Robots with Elastic Joints: Theory and Experiments," *Automatica*, vol. 41, no. 10, pp. 1809–1819, Oct. 2005.

[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] 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)*, Nov. 2016, pp. 29–34.

[11] 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*, vol. 12, no. 3, Sept. 2015.

[12] J. Jung, D. Kim, and J. Park, "Operational Space Control Framework for Torque Controlled Humanoid Robots with Joint Elasticity," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019, pp. 3063–3069.

[13] O. Harib, A. Hereid, A. Agrawal, *et al.*, "Feedback Control of an Exoskeleton for Paraplegics: Toward Robustly Stable, Hands-Free Dynamic Walking," *IEEE Control Systems Magazine*, vol. 38, no. 6, pp. 61–87, Dec. 2018.

[14] N. Paine, J. S. Mehling, J. Holley, *et al.*, "Actuator Control for the NASA-JSC Valkyrie Humanoid Robot: A Decoupled Dynamics Approach for Torque Control of Series Elastic Robots," *Journal of Field Robotics*, vol. 32, no. 3, pp. 378–396, 2015.

[15] M. Vigne, A. El Khoury, F. Di Meglio, *et al.*, "Imu-based state estimation for a legged robot with multiple flexibilities," in *2019 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Oct. 2019.

[16] M. W. Spong, "Control of Flexible Joint Robots: a Survey," p. 31, 1990.

[17] L. Lublin and M. Athans, "Linear Quadratic Regulator Control," in *The Control Systems Handbook, Second Edition*, ser. Electrical Engineering Handbook, W. Levine, Ed. CRC Press, Dec. 2010, vol. 20103237.

[18] M. Vigne, A. El Khoury, F. D. Meglio, *et al.* Video of the experiments on Atalante. [Online]. Available: https://youtu.be/eg6J04WnutI