Whole-Body Control and Estimation of Humanoid Robots with Link Flexibility

Giulio Romualdi¹, Nahuel A. Villa², Stefano Dafarra¹, Daniele Pucci¹,³ and Olivier Stasse²,⁴

Abstract—This article presents a whole-body controller for humanoid robots affected by concentrated link flexibility. We characterize the link flexibility by introducing passive joints at the concentration of deflections, which separate the flexible links into two or more rigid bodies. In this way, we extend our robot model to take link deflections into account as underactuated extra degrees of freedom, allowing us to design a whole-body controller capable to anticipate deformations. Since in a real scenario, the deflection is not directly measurable, we present an observer aiming at estimating the flexible joint state, namely position, velocity, and torque, only considering the measured contact force and the state of actuated joint. We validate the overall approach in simulations with the humanoid robot TALOS, whose hip is mechanically flexible due to a localized mechanical weakness. Furthermore, the paper compares the proposed whole-body control strategy with state-of-the-art approaches. Finally, we analyze the performance of the estimator in the case of different values of hip elasticity.

I. INTRODUCTION

The standard state-of-the-art controllers for bipedal locomotion are designed on several layers [1], [2], [3], with the higher layers, closer to the user, producing instructions for the lower layers which work closer to the robot. In the lowest level, the motor drivers work at high frequency to guarantee either certain torque or acceleration on the robot joints, according to the whole-body controller instructions. In turn, the whole-body controller uses a model of the whole-body dynamics to compute the needed joint actuation for the robot to follow higher level instructions while keeping feasible contact forces on the environment.

The whole-body control layer typically assumes that all robot links are rigid bodies, which models correctly the whole-body motion in most cases [4], [5], [6], but becomes problematic with even small link deflections. This problem is common in heavy robot with mechanical weakness in their structure, such as the robot Talos (≈ 100 kg), whose hips bend while walking. At the modeling level, several approaches have been proposed to describe multi-link flexible robots [7]. The lumped-parameter approach separates flexible links into two or more rigid bodies by defining fictitious passive elastic joints [8], [9], [10], [11]. Using this approach, we extend the robot model to incorporate the flexible degrees of freedom as passive joints. Still, it is important to mention that the identification of the correct passive joint placement as well as the elastic parameters is not straightforward and has been the object of several studies [10], [12]. Alternatively, the finite element method (FEM) can also be used to model the flexibility of links [13], [14]. However, while providing enhanced modeling capabilities, FEM methods are computationally greedy, which generally forbids their use in time-critical applications.

From the control perspective, link flexibility is often treated with passive elastic joints [9], [10], [11], [15]. To address the elasticity of the robot link, the authors of [9], [10], [11], [15] locally compensate for the effect of deflections by modifying the measured position and velocity of some actuated joints considered in the whole-body controller.

This paper contributes to the design of a control architecture for humanoid robots affected by link elasticity. More precisely, the main contributions follow. i) A whole-body controller allowing for the locomotion of humanoid robots with flexible links. Our controller incorporates a model of flexible links following an approach similar to [9], but differently from the state of the art, our design does not perform any local compensation of deflections. ii) As link deflections are not directly measurable, we propose an observer that uses only the measured contact forces and information from actuated joints to estimate the position, velocity and torques of the elastic joints. iii) We validate our approach on the simulated torque-controlled humanoid robot TALOS [16], and we show how it performs different from the standard rigid-body-based controller.

The paper is organized as follows: Sec. II introduces our notation and recalls some concepts of floating base system, Sec. III details the whole-body controller, Sec. IV discusses the design of our elastic-joint observer, Section V presents the simulation results on the TALOS humanoid robot, and finally, Sec. VI concludes the paper.
II. BACKGROUND

A. Notation
- \( I_{m \times n} \) and \( 0_{m \times n} \) denote the \( m \times n \) identity and zero matrices. When \( m = n \), we use simply \( I_n, 0_n \).
- \( I \) is the inertial frame, \( p_I \) is the origin and \( [I] \) its orientation.
- \( A = (p_A, [A]) \) and \( B = (p_B, [B]) \), then \( B[A] = (p_B, [A]) \) has its origin in \( p_B \) and is oriented as \( A \).
- Given \( A = p_c \in \mathbb{R}^3 \) and \( B = p_{c'} \in \mathbb{R}^3 \), representations of \( p_c \) in \( A \) and \( B \) respectively are \( A^B p_c = A^B p_{c'} + A^B = A^B [s_{pc'}^T 1]^T \). \( A^B \in SE(3) \) is the homogeneous transformation and \( A^B R_{B} \in SO(3) \) is the rotation matrix.
- the hat operator is \( \hat{} : \mathbb{R}^3 \to so(3) \), where \( so(3) \) is the set of skew-symmetric matrices and \( x^\wedge y = x \times y \).
- the vee operator is \( \vee : so(3) \to \mathbb{R}^3 \).
- given a rotation matrix, \( \log \) operator returns the associated element of \( so(3) \). \( \log : SO(3) \to so(3) \).
- \( ^A v_{c,B} \in \mathbb{R}^3 \) denotes the relative linear velocity of the frame \( B \) with respect to \( C \), expressed in \( A \).
- \( ^A \omega_{c,B} \in \mathbb{R}^3 \) denotes the relative angular velocity of the frame \( B \) with respect to \( C \), expressed in \( A \).
- the velocity of a frame \( B \) w.r.t. the frame \( A \) is uniquely identified by the twist \( A_{v,B} = [ ^A \omega_{A,B} \ ^A v_{A,B}] \).
- \( \times^* \) is the dual spatial cross product on \( \mathbb{R}^6 \) such that \( A_{v,A,B} \times^* = \begin{bmatrix} A_{v,B} \ 0_3 \ A_{\omega,A,B} \end{bmatrix} \).
- we drop the suffix \( I \) to identify the velocity of a frame with respect to \( I \), i.e., \( A_{v,B} = A_{v,B} = A_{\omega,B} \).

B. Humanoid Robot Model

A humanoid robot is a floating base multi-body system made up of \( n + 1 \) links joined by \( n \) joints each with one degree of freedom. The robot configuration is fully described by the triplet \( q = (q_B, \tau_B, s) \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n \) where \( B \) is the base frame and \( s \) the joint positions. The velocity of the floating system is represented by the triplet \( \nu = (\nu_B, \nu_{\omega}, s) \in \mathbb{R}^3 \times so(3) \times \mathbb{R}^n \cong \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^2 \).

The position and orientation of any frame \( A \), attached to some link, is identified by \( \hat{} : \mathbb{R}^3 \to SE(3) \), and its twist \( v_A = J_A(q) \nu \) is linearly related to the triplet \( \nu \) by the Jacobian matrix \( J_A(q) \).

Each link of the robot follows the rigid-body dynamics described by the Euler-Poincaré Equations [17]

\[
M_L \dot{L} v_L + L v_L \times \hat{M}_L L v_L = M_L \dot{L} R_L^T g + \sum_{k=1}^{n_f} L \hat{f}_k,
\]  

where \( L \) stands for link frame, \( M_L \in \mathbb{R}^{6 \times 6} \) is the constant inertia tensor, \( L v_L \in \mathbb{R}^6 \) is the link spatial velocity, \( \sum_{k=1}^{n_f} \hat{f}_k \in \mathbb{R}^6 \) are the \( n_f \) wrenches exerted on the link. The proper sensor acceleration is given as [18, Sec. 2.4.4]

\[
\alpha^q_L := L X_{L[I]} \dot{L} \dot{[I]} = \begin{bmatrix} \dot{L} R_L^T g \\ 0_{3 \times 1} \end{bmatrix},
\]  

where \( L X_{L[I]} \) is the adjoint matrix, \( \alpha^q_L \) is the acceleration obtained by an inertial measurement unit aligned to the link. Combining (1) with (2), we obtain [18, Eq. (2.80)]

\[
\begin{align*}
\phi_L &= \sum_{k=1}^{n_f} L \hat{f}_k, \\
\phi_L &= \mathbb{M}_L A^q_L + \begin{bmatrix} 0_{3 \times 1} \end{bmatrix} \times^* \mathbb{M}_L \begin{bmatrix} 0_{3 \times 1} \end{bmatrix}.
\end{align*}
\]

The complete robot dynamics can be described as [19]

\[
M(q) \dot{\nu} + \eta(q, \nu) = B \tau + \sum_k J_{Ca}(q) \hat{f}_k.
\]

The Coriolis, centrifugal and gravitational effects. \( \tau \in \mathbb{R}^n \) are the joint torques, acting only on joints as indicated by the selection matrix \( B \in \mathbb{R}^{(n+6) \times n} \). We express each contact wrench \( \hat{f}_k \in \mathbb{R}^6 \) in the frame \( C_k \in \mathbb{R}^{6 \times 6} \) located at the contact point and aligned to the inertial frame, \( J_{Ca}(q) \) is its associated Jacobian.

Given a frame \( G = (x_{CoM}, [I]) \) placed at the robot’s center of mass (CoM) and aligned to the inertial frame, and applying a given change of coordinates as in [20], the dynamics (4) can be expressed separating the first 6 rows, referring to the centroidal momentum of the robot \( \bar{\mathbf{h}} \) [21], from the last \( n \) rows referring to the actuated joints.

The rate of change of the centroidal momentum of the robot can be expressed by means of the external contact wrenches acting on the system [17], [18]

\[
\bar{\mathbf{h}} = m \ddot{\mathbf{h}} + \sum_k \mathbf{g} X_{C_k[I]} \hat{f}_k,
\]

where \( m \ddot{\mathbf{h}} = \begin{bmatrix} 0 & 0 & -mg \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^6 \) is the spatial robot weight and the adjoint matrix \( \mathbf{g} X_{C_k[I]} \in \mathbb{R}^{6 \times 6} \) changes the coordinates of a 6D force from \( C_k \) to \( G \).

C. Model of the link flexibility

Let us consider that some robot links are affected by punctual mechanical weakness allowing the apparition of concentrated deformations. We model the extra degrees of freedom introduced by this flexible behavior by splitting each flexible body into two or more rigid links connected by virtual joints [11], [10]. So, the \( i \)-th virtual joint exerts a torque \( \tau_i \) that depends on the deflection \( s_i \) and its velocity \( \dot{s}_i \) i.e. \( \gamma_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \)

\[
\tau_i = \gamma_i(s_i, \dot{s}_i).
\]

By numerically differentiating the position of the elastic joint \( s_i \) over a sampling period \( \Delta t \), we discretize (6) as

\[
\tau_i[k] = \gamma_i(s_i[k], s_i[k - 1]),
\]

where \( k \in \mathbb{Z} \) is the discretized time.

Following [10], [11], we assume that \( \gamma_i \) models the elastic behavior as

\[
\gamma_i(s_i, \dot{s}_i) = -k_i s_i - d_i \dot{s}_i,
\]

where \( k_i \) and \( d_i \) are, respectively, the stiffness and damping of the \( i \)-th elastic joint. Discretizing the joint velocity with
where we have separated the base dynamics (subscript \( n \)) from the tasks considered in the optimal control problem.

A. Costs and Constraints

This section introduces the list of low and high priority tasks considered in the optimal control problem.

1) Centroidal momentum task: To ask for a desired centroidal momentum trajectory, we introduce the task as

\[
\Psi_{\text{h}} = \tilde{\mathbf{h}}^* - A_c \mathbf{f} - m\mathbf{g}. \tag{12}
\]

where \( A_c = \begin{bmatrix} \mathbf{G}^T \mathbf{C} \eta_1 \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \mathbf{X} \mathbf{C} \mathbf{1} \end{bmatrix} \) gathers the adjoint matrix for each contact, and the desired value \( \tilde{\mathbf{h}}^* \) is chosen from a feedback law to guarantee asymptotic convergence to the desired trajectory

\[
\tilde{\mathbf{h}}^* = \underbrace{\begin{bmatrix} m \bar{\mathbf{h}}_{\text{CoM}}^\text{ref} \\ m \bar{\mathbf{h}}_{\text{CoM}}^\text{ref} \end{bmatrix}}_{\mathbf{q}_{\text{ref}}} + \begin{bmatrix} \bar{\mathbf{h}}_{\text{CoM}}^\text{ref} \\ \bar{\mathbf{h}}_{\text{CoM}}^\text{ref} \end{bmatrix} \begin{bmatrix} 0_3 \\ k_{\text{h}} \end{bmatrix}, \tag{13}
\]

with positive definite matrices \( k_{\text{h}} \).

2) Cartesian task: While walking, we often require some link frames to have a specific position and orientation with respect to the inertial frame \( \mathbf{Z}_L^\text{ref} = (\mathbf{p}_L^\text{ref}, \mathbf{R}_L^\text{ref}) \in \mathbb{R}^3 \times \text{SO}(3) \), by specifying the following task

\[
\Psi_{L_{\text{SE}(3)}} = \left[ \begin{array}{c} \bar{\mathbf{v}}_L^\text{ref} \\ \bar{\mathbf{w}}_L^\text{ref} \end{array} \right] - \mathbf{J}_L \mathbf{\nu} - \mathbf{J}_L \mathbf{\nu}, \tag{14}
\]

\[
\bar{\mathbf{v}}_L^\text{ref} = \bar{\mathbf{v}}_L^\text{ref} + k_{\text{v}}^L (\mathbf{v}_L - \bar{\mathbf{v}}_L) + k_{p}^L (\mathbf{p}_L^\text{ref} - \mathbf{p}_L) \quad \text{and} \quad \bar{\mathbf{w}}_L^\text{ref} = \mathbf{J}_L \mathbf{\omega}_L^\text{ref} - \mathbf{J}_L \mathbf{\omega}_L
\]

\[
\mathbf{J}_L = \begin{bmatrix} 0_3 & \mathbf{I}_3 \end{bmatrix}. \tag{16}
\]

3) Floating base dynamics task: The whole-body controller considers the base (11a) and joint dynamics (11b) as a set of high-priority tasks. We define the base dynamics task as

\[
\Psi_{\text{dyn}} = \eta_b + M_a \mathbf{\nu} - \sum_k J_{\text{Cl}}^T(a_k(\mathbf{q}) \mathbf{h}_k. \tag{17}
\]

If the robot has under-actuated elastic joints, the joint dynamics should consider the measured (or estimated) joint torques acting on the system as

\[
\Psi_{\text{dyn}} = \eta_a + M_a (\mathbf{q}) \mathbf{\nu} - \begin{bmatrix} I_{n_a} \\ 0_{n_a \times n_f} \end{bmatrix} \mathbf{\tau}^a - \begin{bmatrix} 0_{n_a \times n_f} \\ I_{n_f} \end{bmatrix} \mathbf{T}^f - \mathbf{J}_{\text{Cl}}^T \mathbf{f}. \tag{18}
\]

Notice that the last \( n_f \) rows of (18) represent the underactuated flexible behavior of the robot.

4) Joint position regularization task: To prevent the controller from computing solutions that generate a huge variation in joint acceleration, we introduce a joint regularization task for both the actuated and elastic joints, as

\[
\Psi_{\text{sa}} = s_a - \begin{bmatrix} I_{n_a} \\ 0_{n_f \times n_f} \end{bmatrix} \mathbf{\nu}, \tag{19a}
\]

\[
\Psi_{\text{sf}} = s_f - \begin{bmatrix} I_{n_f} \\ 0_{n_f \times n_f} \end{bmatrix} \mathbf{\nu}, \tag{19b}
\]

the forward Euler method, we approximate the elastic joint state, namely its position and velocity, by

\[
s_i[k] = s_i[k - 1] - \tau_i[k] \frac{dt}{d_1}, \tag{9a}
\]

\[
\dot{s}_i[k] = \frac{s_i[k] - s_i[k - 1]}{dt}, \tag{9b}
\]

D. Modeling of a floating base system with elastic joints

Let us redefine the robot configuration as:

\[
\mathbf{q} = (\mathbf{p}_B, \mathbf{R}_B, s^a, s^f), \quad \mathbf{\nu} = (\mathbf{\nu}_B, \mathbf{\nu}_B, s^a, s^f) \tag{10}
\]

with \( n_a \) actuated joints \( s^a \) and \( n_f \) elastic joints \( s^f \), so that the robot posture has \( n = n_a + n_f \) degrees of freedom, and the robot dynamics (4) becomes:

\[
\begin{align}
M_b \mathbf{\ddot{\mathbf{v}}} + \mathbf{\eta}_b &= \sum_k J^T_{\text{Cl}} \mathbf{\dot{b}}_k, \tag{11a}

M_a \mathbf{\ddot{\mathbf{v}}} + \mathbf{\eta}_a &= \begin{bmatrix} I_{n_a} \\ 0_{n_f \times n_a} \end{bmatrix} \mathbf{\tau}^a + \begin{bmatrix} 0_{n_a \times n_f} \\ I_{n_f} \end{bmatrix} \mathbf{\tau}^f + \sum_k J^T_{\text{Cl}} \mathbf{\dot{f}}_k. \tag{11b}
\end{align}
\]

where we have separated the base dynamics (subscript \( b \) corresponding to the 6 first rows) from the postural dynamics (subscript \( a \) corresponding to the \( n \) final rows).

III. Whole-body Controller

The whole-body controller guarantees the tracking of desired kinematic quantities while ensuring the feasibility of the contact forces. The proposed controller computes the desired actuated joint torques \( \mathbf{\tau}^a \), considering the robot dynamics (11) and the elastic joint model (8). The control objective is met by framing the controller as a constrained optimization problem in which low-priority tasks are embedded in the cost function, and high-priority tasks are handled as constraints. In the following equations, the superscript \( \text{ref} \) denotes the quantities provided by a high-level planner that assumes the robot made of rigid links only. Fig 2 presents the input and output quantities of the Whole-body controller.

A. Costs and Constraints

This section introduces the list of low and high priority tasks considered in the optimal control problem.

...
where \( s_a^{\text{ref}} \) is the desired joint trajectory. \( k_d^s \) and \( k_p^s \) are two positive-defined diagonal matrices. On the other hand, we ask for \( s_f^s \) equal to

\[
s_f^s = -k_d^s \dot{s}_f - k_p^s s_f.
\]

(21)

\( k_d^s \) and \( k_p^s \) are defined positive matrices. Thanks to (21) the controller tries to stabilize the flexible joint position to zero.

5) Joint torque regularization task: To prevent the controller from providing solutions with large actuated joint torques, we introduce the following task

\[
\Psi_f = \tau_a^{\text{ref}} - \tau_a.
\]

(22)

6) Feasible contact force task: The feasibility of the contact 6D force \( C_i \) is guaranteed by the set of inequalities

\[
\Phi_{t,i} = \Gamma_{C_i} C_i - b_i \quad \Phi_{t,i} \leq 0,
\]

(23)

where \( \Gamma_{C_i} \) depends on the robot state \( q \) as in [23], and the inequalities \( \leq \) hold elementwise.

B. Quadratic programming formulation

We achieve the control objective by designing a constrained quadratic programming (QP) problem of the form

\[
\begin{align*}
\text{minimize} & \quad \|\Psi_{SO(3)}\|^2 + \|\Psi_{R(0)}\|^2 + \|\Psi_s\|^2 + \|\Psi_f\|^2, \\
\text{subject to} & \quad \Psi_{SE(3)} = 0, \quad \circ = \{L, R\}, \\
& \quad \Psi_h = 0, \\
& \quad \Psi_{\text{gyr}} = 0, \quad \circ = \{s, \nu\}, \\
& \quad \Phi_{t,i} \leq 0, \quad \circ = \{L, R\}.
\end{align*}
\]

(24)

Here, contact forces \( f_i \), the base acceleration \( B \otimes \dot{V}_{B} \), the actuated joint acceleration \( \dot{s}_a \), and the actuated joint torques \( \tau_a \) are considered as optimization variables.

In (24), the tracking of the left \( \Psi_{L(0)} \) and right \( \Psi_{R(0)} \) foot poses is treated as high-priority SE(3) tasks (14). The Centroidal momentum tracking is considered as a high-priority task (12). We also consider the base (17) and joints dynamics (18) as high-priority tasks. To prevent the controller from asking for a high motion of the upper body, we introduce two SO(3) tasks, one associated with the chest and the other with the waist orientations, respectively, denoted \( \Psi_{SO(3)} \) and \( \Psi_{RSO(3)} \). In both cases, we ask the z coordinates of the link frames to remain parallel to the gravity vector. The postural conditions of the actuated and elastic joints (19) are considered as low-priority tasks. We regularize the desired actuated joint torques as a low priority task \( \Psi_f \) (22). Finally, to guarantee feasible contact forces for the feet, we add the task (23), denoted respectively as \( \Phi_{t,L} \) and \( \Phi_{t,R} \).

IV. ELASTIC JOINT STATE OBSERVER

Solving the optimal control problem (24) requires a prior-knowledge of the position, velocity and torque of all elastic joints. These values are fully known in simulation. However, in the real scenario, they must be estimated. We present an algorithm that computes the state of elastic joints considering the contact wrenches acting on the robot feet and the actuated joint state. Let us assume:

- the link flexibility is located on the robot leg;
- the robot leg is described by a kinematic chain where the first node represents the robot base;
- all the flexible links are modeled with \( m_f \) elastic joints described by (8) and (9);
- each elastic joint is connected to its child joint with a fake link having zero mass and inertia;
- there exists \( m_a \) actuated links and joints between the latest elastic joint and the robot foot.

Let us consider a single leg: we denote the waist of the robot by \( L_0 \), all virtual links and elastic joints by \( L_i \) and \( s_i \) with \( 1 \leq i \leq m_f \); and all actual robot links and actuated joints by \( L_i \) and \( s_i^a \) with \( m_f + 1 \leq i \leq m_a + m_f \), respectively. We also name the frame associated with each link as the link itself. We assume that each link is subject to only two wrenches, exerted by the parent and child links. With an abuse of notation, we consider the environment as the child of the foot \( L_{m_a+m_f} \). For the dynamics of the links (3), we denote by \( \phi \) the wrenches exerted on the actual robot links and on the fictitious flexible links.

A. Forward kinematics

Given the angular velocity and the proper sensor acceleration of the foot \( L_{m_a+m_f} \), we can compute \( \phi \) with \( m_f + 1 \leq i \leq m_a + m_f \), recursively. Given the link \( L_i^a \), its body angular velocity writes as

\[
i \omega_i = i R_i^{1+1} \omega_{i+1} + i \omega_{i+1,1}, \quad m_f + 1 \leq i \leq m_a + m_f
\]

(25)

where \( i \omega_{i+1} \) is the angular velocity of the child link, \( i \omega_{i+1,1} \) is the velocity of \( L_i^a \) with respect to \( L_{i+1}^a \) written in \( L_i^a \). Similarly, we recall that \( \alpha_0 \) can be computed recursively by considering the joint position \( s_{i+1}^a \) velocity \( s_{i+1}^a \) and acceleration \( s_{i+1}^{aT} \), as in [18].

We want to underline that if an IMU is mounted on the robot feet, \( \phi_{m_a+m_f} \), can be derived from the sensor readouts. Otherwise, we propose two possible solutions: i) In the case of low swing foot velocity and acceleration, we suggest considering \( m_a+m_f \) \( \omega_{m_a+m_f} \) and \( \alpha_0 \) equal to zero. ii) Assuming that an IMU is mounted on the robot waist. We suggest setting \( \alpha_0 \) at time \( t \) equal to the one predicted by the forward kinematics at the previous step \( t - d t \). To guarantee the convergence of the algorithm, the choice of \( d t \) becomes crucial. Here, we suggest setting \( d t \) small enough to capture the evolution of the system dynamics.
B. Inverse dynamics propagation

Assuming that \( \phi_{L,m}^{f} \) has been computed as in Sec. IV-A and considering (13) with \( n_f = 2 \), the dynamics of \( L_m^{a} \) holds

\[
\dot{\phi}_i = i\dot{f}_m(i,i) + \dot{f}_{i+1,i}, \quad m_f + 1 \leq i < m_a + m_f,
\]

(26)

where \( \lambda(i) \) gives the parent link of \( L_m^{a} \), \( i \dot{f}_m(i,i) \) is the spatial force exerted by the parent link \( \lambda(i) \) to \( L_m^{a} \) whose coordinates are expressed in \( L_m^{a} \), \( i \dot{f}_{i+1,i} \) is the spatial force exerted by \( L_{i+1}^{a} \) to \( L_m^{a} \) expressed in \( L_m^{a} \). We now reorganize (26) to reveal its recursive structure:

\[
i\dot{f}_m(i,i) = \phi_i - i\dot{f}_{i+1,i} \\
= \phi_i + iX^{i+1-1}\dot{f}_{i,i+1}
\]

(27a)

(27b)

We notice that by projecting (27) into the joint motion axis \( \mathbf{s} \in \mathbb{R}^6 \) [17], [18], the torque acting on the joint \( a^m \) writes as \( \tau^a_m = i^s_i\dot{f}_m(i,i) \in \mathbb{R}^6 \). If the system is equipped with joint torque sensors, \( \tau^a_m \) can be directly measured. Using this information, we attempt to improve the estimation of the 6D force \( i\dot{f}_m(i,i) \) as follows:

\[
i\dot{f}_m(i,i) = \left[ (1 - \beta)\tau^a_m + \beta i^s_i\dot{f}_m(i,i) \right] \mathbf{s}
+ \left( I_6 - i\mathbf{s}^s_i \mathbf{s}^s_i \right) i\dot{f}_m(i,i).
\]

(28)

Here, \( i\dot{f}_m(i,i) \) denotes the corrected 6D force and \( \tau^a_m \) is the measured joint torque. We choose the parameter \( \beta \in [0, 1] \) according to our trust among measurement and calculations. At its extremes, \( \beta = 0 \) considers only the measured torques, and \( \beta = 1 \) just takes the computed torques.

C. Elastic joint state estimation

Propagating the inverse dynamics in Sec. IV-B, we recursively compute the wrench acting on the joint \( s_f^i \). Assuming a neglegted mass and inertia for the flexible link \( L_m^{f} \), i.e., \( M_{L_m^{f}} \approx 0_{6 \times 6} \), we write the flexible link dynamics as

\[
m_f m_f m_f - 1, m_f = -m_f m_f m_f +1, m_f \\
= m_f X_{m_f +1} m_f +1, m_f +1, m_f +1,
\]

(29)

where \( m_f X_{m_f +1} m_f \) depends on the position of the joint \( a^m \) obtained in Sec. IV-B.

Given (29), we can compute the elastic joint torque \( \tau^f_{m_f} \) by projecting \( m_f m_f m_f - 1, m_f \) into the joint motion axis \( m_f \mathbf{s} \)

\[
\tau^f_{m_f} = m_f \mathbf{s}^s m_f m_f m_f - 1, m_f.
\]

(30)

Combining (30) with the discretized elastic joint position (9a) we estimate the elastic joint position \( s_f^m \). Applying the same approach, we can estimate the elastic joint torque \( \tau^a_i \) and position \( s_f^i \), with \( 1 \leq i \leq m_f + 1 \).

At each time step, applying the procedure of Sec. IV-A, IV-B, Eq. (29) and (30), we estimate the elastic joint state. The result is finally considered by the whole-body controller to compute the desired actuated joint torques.

V. Results

In this section, we test the control strategy presented in Sec III on a simulated version of the TALOS humanoid robot [16] – see Fig. 1. TALOS’s hip flexibility has a significant impact on its leg control and, as a result, its balance and locomotion [6]. Since the hip flexibility is due to the vertical linkage following previous work [10], we model the flexibility along the pitch and roll axis for each leg. As a result, we have \( n_f = 4 \) while \( n_a = 32 \). The architecture takes (on average) less than 1 ms to evaluate its output. The OSQP [24] library is used to solve the optimization problem. The simulations are obtained by integrating the robot dynamics and the joint elasticity as in Fig. 2.

To validate the performance of the proposed architecture, we present two main experiments. First, the proposed strategy is compared with a state-of-the-art task-based inverse dynamics algorithm that considers rigid links [25] in the case of different stiffness parameters \( k \). From now on, the proposed control approach is called TSID-Flex and the state-of-the-art controller TSID-Rigid. Second, we analyze the performance of the TSID-Flex in the case of different stiffness. In both scenarios, the desired robot CoM, footsteps, and actuated joint trajectories are computed offline1. The robot walks 1 meter forward with a step length of 20 cm, starting with the right foot. The first and last steps are 10 cm long. The double support lasts 0.2 s, while the single support lasts 1.2 s.

A. Comparison between TSID-Flex and TSID-Rigid

In Table I, we summarize the results of the control strategies for different stiffness parameters \( k \). Labels success and failure mean that the associated controller is either able or not to ensure the robot’s balance while walking. In all the experiments presented in this section, the damping parameter is arbitrarily set to \( b = 2\sqrt{k} \).

To compare the two controllers, we decided to perform two main experiments. In the former, we choose a set of stiffness parameters such that both whole-body controllers guarantee the balance while walking. In the latter, we decrease the value of the stiffness parameter. Namely:

- **Experiment 1** \( k = 1 \times 10^4 \text{ N m rad}^{-1} \);

- **Experiment 2** \( k = 3 \times 10^3 \text{ N m rad}^{-1} \).

1. Experiment 1: Fig. 3 shows the CoM tracking performance obtained with TSID-Flex and TSID-Rigid in terms of tracking error. The TSID-Flex controller seems to show good tracking performance, and the CoM error is kept below 2 mm. On the other hand, the TSID-Rigid induces a higher error on the CoM tracking. Similar considerations hold also for the tracking of the centroidal angular momentum.

In fact, TSID-Flex ensures a smaller angular momentum error with respect to the TSID-Rigid. One reason for this behavior is that the TSID-Rigid assumes the robot composed only by rigid link, hence all the joints are considered to be actuated. This assumption is generally valid in the case of stiff \( k \), but it does not hold if \( k \) decreases. Fig. 5 presents

1The whole-body reference trajectories are provided by:

[github.com/loco-3d/multicontact-api/tree/v2.1.0](https://github.com/loco-3d/multicontact-api/tree/v2.1.0)
2) Experiment 2: The centroidal quantity tracking problem discussed in Experiment 1 worsens at lower values of the stiffness parameter $k$. Fig. 4 shows the CoM tracking performance of the two controllers. In Fig. 4 we also present the tracking of the angular momentum. The TSID-Flex is still capable of ensuring good performance. On the other hand, the TSID-Rigid does not consider the elastic joint state. As a consequence, this leads to a non-negligible error on the robot CoM. In order to keep the balance, the TSID-Rigid controller requires high variations of the robot’s foot orientation and at $t \approx 10\text{ s}$ the robot falls – Fig. 6.
We model the flexibility by introducing equivalent passive joints that simulate the motion caused by the viscoelastic link deformation. We then considered the passive joints position and velocity – see Fig. 8b and 8c. This behavior is caused by the discontinuity of the contact force. To mitigate this effect, we may perform a smoother transition between single and double support.

B. Performance of the TSID-Flex in the case of different $k$

We now benchmark the performance of the TSID-Flex controller for different stiffness parameters $k$, namely $k = 10 \text{ kN m rad}^{-1}, 5 \text{ kN m rad}^{-1}, 3 \text{ kN m rad}^{-1}, 2 \text{ kN m rad}^{-1}, 1 \text{ kN m rad}^{-1}$, and $0.5 \text{ kN m rad}^{-1}$.

Fig. 7 shows the CoM tracking error for different values of $k$. The controller ensure acceptable performance for all values of $k$, i.e., the error is always below 3 mm. However, the lower the stiffness, the higher the tracking error. This increase in tracking error is caused by the elastic joint state estimation error. Fig. 8 presents the estimation error of an elastic joint. Similar considerations hold also for the other three joints. When the robot switches from single support to double support, the estimated torque associated with the elastic joint has a spike – at $t \approx 8.5 \text{ s}$, $11 \text{ s}$ and $13.5 \text{ s}$ in Fig. 8a. As a consequence, the error propagates in the estimation of the elastic joint position and velocity – see Fig. 8b and 8c. This behavior is caused by the discontinuity of the contact force. To mitigate this effect, we may perform a smoother transition between single and double support.

VI. CONCLUSIONS AND FUTURE WORK

This paper contributes to the design of a whole-body QP controller for a humanoid robot affected by link flexibility. We model the flexibility by introducing equivalent passive joints that simulate the motion caused by the viscoelastic link deformation. We then considered the passive joints position and velocity as state of the floating base system dynamics. Thanks to this choice, unlike other state-of-the-art approaches, we develop a whole-body controller that explicitly considers link flexibility in the stabilization problem. The paper also details the design of an estimator that aims at computing the elastic joint state in real-time. The proposed approach is validated in a simulated version of the TALOS humanoid robot, where its hip flexibility has a significant impact while performing locomotion tasks. Moreover, the architecture is then compared with a state-of-the-art whole-body controller that considers all links rigid. In future work, we plan to mitigate the contact forces discontinuity by performing a smoother transition between contigous support phases. We also plan to make a comparison with other state-of-the-art controllers that consider the flexibility of the robot link [10]. Another interesting research direction is to consider the link flexibility in the generation of reference centroidal trajectories [27], [28] and whole-body-motion [29], [30].

REFERENCES


