A Balance-Force Controller for Multi-Flexible-Contacts Legged Robot

Marwan Hamze, Mehdi Benallegue, Rafael Cisneros, Abdelaziz Benallegue

A Balance-Force Controller for Multi-Flexible-Contacts Legged Robot

Marwan Hamze[1], Mehdi Benallegue[2], Rafael Cisneros[2], Abdelaziz Benallegue[1,2]

***Abstract*— In this paper, we propose a controller for multi-contact legged robots that takes into account the contact compliance. This controller is able to achieve both balance stabilization and force control in the same loop in task space thanks to the use of an explicit contact flexibility model and simplified centroidal dynamics that allow exploiting the redundancy of the kinematic and force feedback. The control problem is formulated as an LQR based on a linearized model of the reduced non-linear model. The performance of the controller with regard to robustness to modeling error and external perturbations has been tested in simulation, and compared to a stabilizer based on the zero-moment-point feedback within a control that generates the contact forces with the environment.**

I. Introduction

Legged robots are considered under-actuated because they lack actuation to generate translations and rotations of their body in their environment. A robot's unactuated dynamics and its balance in particular depend only on external forces, mostly reduced to contact forces and gravity [1]. In particular, the bipedal walking of humanoid robots requires fine control of the contact forces to ensure stability despite the small contact surface. It is common to exploit additional contact points allowing the robot to achieve complex transportation motions with increased stability [2], [3], [4], [5]. However, this usually makes the dynamics more complex when the contacts are non-coplanar and of different nature (unilateral, bilateral, point contact, etc.) [6], which makes it difficult to ensure all the reference contact forces are respected, even when the robot is equipped with force sensors. This is particularly true when dealing with uncertain environments in terms of geometry and compliance.

Indeed, the inaccuracy in force tracking comes from the fact that the contacts that the robot makes with the environment aren't fully rigid. That is not only because the environment is never perfectly stiff, but also because the robot itself has a certain level of compliance. The robot HRP-2 which has a flexible bush, placed in the ankle in order to absorb foot impacts [7] is a good example of a compliance by design, but it is common to see robots with a high level of structural compliance, such as COMAN [8] or Sarcos Primus [9]. The presence of such compliance has two consequences. The first one is that forces modify the shape of the environment, and displaces the contact force application point, which alters the robot's dynamics. The second consequence is that, contrarily to the rigid contact model, the robot is not able to instantly generate the desired force since it depends on its state, which needs then to be modified accordingly. This second consequence is the reason why the dynamics of the Zero Moment Point (ZMP) is often modeled with a first order system [10] as an approximation, but this model is also inaccurate since the simplest physics-based model has at least a second order dynamics. Some works explicitly take into account the flexibility model [11], [12], [13], [14], [15], but most of them consider only point contact and mostly don't include force feedback in the control loop, or consider the compliance concentrated on one support [16], [17].

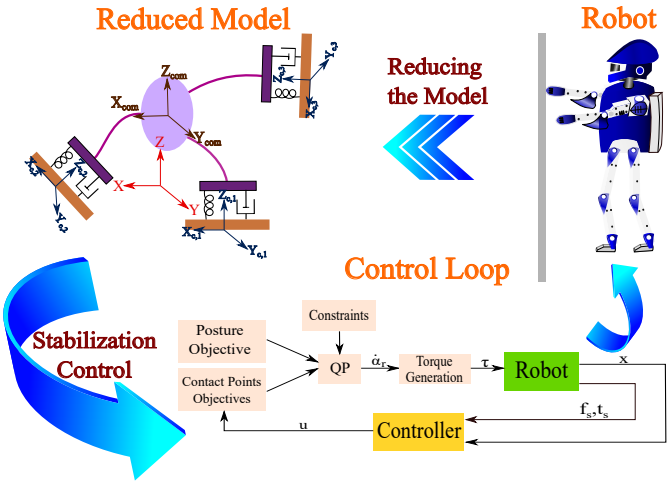

Fig. 1. Diagram showing the robot, its reduced model and the control loop. The world, CoM and contact frames are highlighted for the reduced model. The yellow block is our proposed controller.

Classically, to overcome these issues, a second loop of force control is often exploited, usually as admittance control [9], [18], [19]. These controllers are usually set up in series with the balance control. This means that the kinematic-feedback-based stabilization sends force references to force-feedback-based admittance control loop. This is based on the assumption that the dynamics of the force control are fast enough to converge within the time required to maintain balance. However, not only both dynamics remain always coupled, the admittance control becomes significantly slower with more compliant and uncertain environments. For this reason, there is a need for a controller able to take both dynamics into account in a single loop.

Finally, some parameters in the robot's model or in the environment, such as the stiffness of the contacts, can be very

[1] M. Hamze and A. Benallegue are with Université Paris-Saclay, UVSQ, Laboratoire dIngénierie des Systèmes de Versailles, France. e-mail: marwan.hamze@uvsq.fr, abdelaziz.benallegue@uvsq.fr

[2] M. Benallegue, R. Cisneros and A. Benallegue are with CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki, Japan. e-mail: mehdi.benallegue@aist.go.jp, rafael.cisneros@aist.go.jp.

hard to measure correctly. Furthermore, we usually resort to simplified models such as viscoelastic contact pattern, which does not correspond to the physical reality which is nonlinear and even time-varying. Some methods intend to estimate them online [11], [20], [21] but this estimation requires to gather dynamic data and may take a lot of time to converge. Therefore we need a controller that is robust to this kind of modeling errors.

We propose in this paper a controller with explicit multi-contact support allowing to perform both stabilization and force control in a single loop. The paper will start by presenting the dynamics of the robot, then the selected reduced model, before proceeding to the synthesis of the proposed controller based on the linearized model, and finally to the simulations and results.

II. Dynamics of the Robot

In this section we present the Lagrangian dynamics of the robot in contact with a compliant surface. Let us consider a multi-body humanoid robot with $n+6$ degrees of freedom (d.o.f), with a configuration described as $\Psi = (p_B, R, q)$, where $p_B \in \mathbb{R}^3$ and $R \in SO(3)$ represent the position and orientation matrix of the non-actuated floating base, and $q \in \mathbb{R}^n$ the joint angles vector. We define $\alpha \in \mathbb{R}^{n+6}$ as the robot's configuration velocity vector given by

$$\alpha = \begin{pmatrix} \dot{p}_B^\top & \omega^\top & \dot{q}^\top \end{pmatrix}^\top, \tag{1}$$

where $\dot{p}_B$ and $\omega \in \mathbb{R}^3$ are the linear and angular velocities of the base. The angular velocities are such that the derivative with respect to time of the rotation matrix is given by: $\dot{R} = S(\omega)R$, with $S(\omega)$ being the skew-symmetric matrix operator allowing to perform cross-product. The robot's acceleration vector $\dot{\alpha}$ is the time derivative of α.

The Lagrangian dynamics of the robot are written as

$$H(\Psi)\dot{\alpha} + C(\Psi, \alpha)\alpha + G(\Psi) - F = \tau, \tag{2}$$

where $H(\Psi) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the robot's inertia matrix, $C(\Psi, \alpha) \in \mathbb{R}^{(n+6)\times(n+6)}$ is a matrix accounting for the Coriolis and centrifugal effects, $G(\Psi) \in \mathbb{R}^{n+6}$ is the vector of gravitational effects, $\tau \in \mathbb{R}^{n+6}$ is the input torque vector corresponding to both the under-actuated and actuated d.o.f, and $F \in \mathbb{R}^{n+6}$ is the vector of external forces acting through the contacts with the environment, which is calculated using

$$F = \begin{pmatrix} J_{c,1}^\top & \dots & J_{c,n_c}^\top \end{pmatrix} \begin{pmatrix} f_{c,l,1} \\ t_{c,l,1} \\ \vdots \\ f_{c,l,n_c} \\ t_{c,l,n_c} \end{pmatrix}, \tag{3}$$

where $J_{c,i} \in \mathbb{R}^{6\times(n+6)}$ is the Jacobian matrix of the contact i, $f_{c,l,i}$ and $t_{c,l,i}$ are respectively the force and torque applied at contact i ($i = 1, 2, ..., n_c$; n_c is the number of contacts), expressed in the local frame of the contact body.

The classic use of this equation is to consider that the environment is rigid. In this case these forces depend mostly on the torque τ, in a way that any force that is feasible, meaning within unilaterality, friction, and torque limit constraints, can be generated *instantly*. However, a perfectly rigid contact cannot exist, and the compliance means that this assumption is wrong. The contact forces at a given instant depend on many factors and joint torque is virtually not one of them. In particular, if the forces depend only on the local deformation caused by the interaction with the contact body, we can write the forces as a function of the robot state $F(\Psi, \alpha)$. In this case we can control the forces indirectly by modifying this deformation through the state of the robot. This requires to explicitly consider the coupled dynamics between the kinematics and the contact forces to ensure the convergence to a given reference.

III. Dynamic Model for the Controller

To simply parameterize the contact wrenches and formulate how they affect the position of the Center of Mass (CoM) of the robot, the dynamics are reduced to consider the robot as a single rigid body with viscoelastic contacts. Hereinafter is the description of this model

A. Viscoelastic Contact Model

A common model for compliant contact is the viscoelastic approximation allowing to emulate a linear passive interaction. Thus, we write

$$f_{c,l,i} = K_{f,p,i}(p_{c,i} - p_{c,i,r}) + K_{f,d,i}\dot{p}_{c,i}, \tag{4}$$

$$t_{c,l,i} = K_{t,p,i}\Theta\left(R_{c,i,r}R_{c,i}^\top\right) + K_{t,d,i}\omega_{c,i}, \tag{5}$$

where $p_{c,i} \in \mathbb{R}^3$, $\dot{p}_{c,i} \in \mathbb{R}^3$ and $\omega_{c,i} \in \mathbb{R}^3$ are respectively the position, linear and angular velocities of the contact i written in the contact frame and $R_{c,i} \in SO(3)$ is the orientation matrix of the contact i. The variables $p_{c,i,r} \in \mathbb{R}^3$ and $R_{c,i,r} \in SO(3)$ are respectively the rest position and orientation of the contact i, i.e. when the forces and torques are zero. The matrices $K_{f,p,i}$ and $K_{f,d,i}$ are the linear stiffness and damping at contact i, while $K_{t,p,i}$ and $K_{t,d,i}$ are the angular stiffness and damping at contact i. The difference in orientation represented by $R_{c,i}R_{c,i,r}^\top$ is considered so small that the approximation $sin\theta \approx \theta$ is used, thus we define the function Θ as the axis-sine of angle representation. The values of stiffness and damping can be set independently from a contact to another. Note that to emulate an edge-contact i we set semi-definite values for $K_{t,p,i}$ and $K_{t,d,i}$ and to emulate a point-contact i we can set them simply to zero.

B. State and Control Vectors Definition

We define here a reduced state vector focusing on the contacts with the environment, the state variables are the positions, orientations, and velocities of these points, in addition to those of the CoM. The contact positions and orientations are going to be defined with respect to the CoM frame, which is the frame centered at the CoM and having the same orientation as the floating base. The state vector is defined by

$$x \triangleq \begin{pmatrix} x_{com} & x_1 & \cdots & x_n \end{pmatrix}^\top, \tag{6}$$

where $x_{com} = \begin{pmatrix} p_{com}^\top & \dot{p}_{com}^\top & \Omega^\top & \omega^\top \end{pmatrix}^\top$ and $x_i = \begin{pmatrix} p_{c,i,com}^\top & \Omega_{c,i,b}^\top & \dot{p}_{c,i,com}^\top & \omega_{c,i,b}^\top \end{pmatrix}^\top$. The vectors $p_{com} \in \mathbb{R}^3$ and $\dot{p}_{com} \in \mathbb{R}^3$ are respectively the linear position and velocity of the CoM, Ω and ω are respectively the orientation and the angular velocity of the floating base, all written in the world frame. Similarly, $p_{c,i,com} \in \mathbb{R}^3$ and $\dot{p}_{c,i,com} \in \mathbb{R}^3$, $\omega_{c,i,com} \in \mathbb{R}^3$ and $\Omega_{c,i,com}$ are respectively the linear position, velocity, angular velocity and orientation of the contact i, all written in the CoM frame. It should be noted that the orientations (Ω and $\Omega_{c,i,com}$) can be written by using any representation of the orientation, such as the quaternion, the axis-angle, etc. Each Ω has an associated rotation matrix $R \in SO(3)$.

As for the control vector, and as we are aiming to control the contact points, the control vector is defined as

$$u \triangleq \begin{pmatrix} \ddot{p}_{c,1,com}^\top & \dot{\omega}_{c,1,com}^\top & \cdots & \ddot{p}_{c,n_c,com}^\top & \dot{\omega}_{c,n_c,com}^\top \end{pmatrix}^\top, \tag{7}$$

where $\ddot{p}_{c,i,com} \in \mathbb{R}^3$ and $\dot{\omega}_{c,i,com} \in \mathbb{R}^3$ are respectively the linear and angular accelerations of the contact i, written in the CoM frame. Note that these are immediate time-derivatives of state variables. The dynamics of this state is described hereinafter.

C. Reduced Dynamics

The reduced model is a rigid body with massless legs, which gives a constant inertia tensor $I \in \mathbb{R}^{3\times3}$ in the base frame, expressed as $RIR^\top$ in the world frame (R here is the orientation matrix of the robot's base). Thus, the angular momentum of the robot is $RIR^\top\omega$.

Using Euler's second law, the relation between angular momentum and the total external torque, is expressed by

$$\sum_{i=1}^{n_c} \left(S\left(Rp_{c,i,com}\right) f_{c,i} + t_{c,i}\right) = S\left(\omega\right) RIR^\top\omega + RIR^\top\dot{\omega}, \tag{8}$$

where $f_{c,i}$ and $t_{c,i}$ are respectively the force and torque at the contact i written in the world frame, which can be obtained from equations (4) and (5) using

$$f_{c,i} = R_{c,i} f_{c,l,i}, \tag{9}$$

$$t_{c,i} = R_{c,i} t_{c,l,i}. \tag{10}$$

Using Newton's second law, and equation (8), the linear and angular accelerations of the CoM and of the floating base of the robot can be expressed as

$$\ddot{p}_{com} = \frac{1}{m}\sum_{i=1}^{n_c} f_{c,i} + g, \tag{11}$$

$$\begin{aligned}\dot{\omega} = {} & RI^{-1}R^\top \sum_{i=1}^{n_c} \left(S\left(Rp_{c,i,com}\right) f_{c,i} + t_{c,i}\right) \\ & - RI^{-1}R^\top S\left(\omega\right) RIR^\top\omega,\end{aligned} \tag{12}$$

where $g = \begin{bmatrix} 0 & 0 & -g_0 \end{bmatrix}^\top$ and g_0 is the gravity constant.

Using equations (7), (11) and (12), we can finally write the non-linear model of the robot as

$$\dot{x} = f\left(x, u\right). \tag{13}$$

IV. Proposed Balance-Force Controller

A. Linearized Dynamics

The dynamics given by equation (13) can be used to derive different types of control, in this paper we propose to stabilize the robot around a fixed, stationary desired equilibrium state x^*. But even in that case, the non-linear dynamics is complex to tackle directly, therefore to simplify the control synthesis, the local dynamics are approximated by linearization around this desired state. To simplify the notation, we represent the state error x^Δ between the actual state and the desired one by using an operator noted Δ. For the positions and velocities in the state vector, Δ represents the Euclidean difference. As for the orientations, it is the axis-sine of angle representation of the difference between the orientation matrices. The linearization is achieved as detailed in the Appendix.

It is important to note that this linearization is very different from the one commonly performed with the inverted pendulum. Indeed, no assumptions are made either on the kinematics of the CoM, on the position nor on the orientation of the contacts. So this linearization is not less precise in the case of multiple non-coplanar contacts with different stiffness and damping (including point and edge contacts).

At the end of the linearization process of the reduced model (13), we define the matrices A and B (given in the appendix) such that

$$\dot{x}^\Delta = Ax^\Delta + Bu^\Delta. \tag{14}$$

This model is well suited for a state feedback control law. However, it would be purely based on kinematics and cannot track reference forces except through the viscoelastic model. To add more reliability to the knowledge of the contacts, the forces measured by force sensors are going to be appended to the positions as explained in the next section.

B. Balance-Force Control Law

One main contribution of the proposed controller is to include force tracking in the same control loop as CoM and kinematic tracking. However, it's not possible for the controller to use all the available actuation to kinematics and force control at the same time, thus the need to establish a trade-off between the conflicting kinematics and force control, which is common to have. Therefore we build a new state vector where the contact forces and torques are combined with the positions and orientations of the corresponding contact bodies. Hence, the force vector z is defined as

$$z \triangleq \begin{pmatrix} 0_{1\times12} & z_1 & \cdots & z_n \end{pmatrix}^\top, \tag{15}$$

where $z_i = \begin{pmatrix} f_{s,i}^\top & t_{s,i}^\top & 0_{1\times3} & 0_{1\times3} \end{pmatrix}^\top$. The vectors $f_{s,i}$ and $t_{s,i} \in \mathbb{R}^3$ are respectively the force and torque at the contact point i, scaled to the positions and orientations by dividing by the stiffness of the contact, and written in the base frame to make their appendage to the positions and orientations possible. The actual forces and torques are measured with force sensors at the contact points.

Now, we can combine both vectors x and z in what we define as the new state error y^{Δ}

$$y^{\Delta} \triangleq (\mathbb{I} - W)\, x^{\Delta} + W z^{\Delta}. \tag{16}$$

where $\mathbb{I}$ is an identity matrix, W is a weight matrix, used to set the trade-off between force tracking and position tracking. One way of defining it is

$$W \triangleq \text{diag}\,(0_{1\times 12}, w_1, \cdots, w_{n_c}), \tag{17}$$

where diag(.) is an operator that gives a square matrix, having on its diagonal the values given in between the parenthesis and zeros elsewhere, and

$$w_i \triangleq \begin{pmatrix} w_{fi} & w_{ti} & 0_{1\times 3} & 0_{1\times 3} \end{pmatrix}, \tag{18}$$

where $w_{fi} \in \mathbb{R}^3$ and $w_{ti} \in \mathbb{R}^3$ are vectors having values between 0 and 1 that respectively multiply $f_{s,i}$ and $t_{s,i}$ in the vector z.

Considering that the forces and torques in z can be written as functions of the variables in x using (4) and (5), we can write, after linearization, the vector z^{Δ} as a matrix M (given in the appendix) multiplying the vector x^{Δ}

$$z^{\Delta} = M x^{\Delta}. \tag{19}$$

Using equation (19) in (16), we can write y^{Δ} as:

$$y^{\Delta} = N x^{\Delta}, \tag{20}$$

with $N = \mathbb{I} - W + WM$ and W is chosen so that the matrix N is non singular.

We can now write the dynamics of y^{Δ} as

$$\dot{y}^{\Delta} \simeq A_y y^{\Delta} + B_y u^{\Delta}, \tag{21}$$

with $A_y = NAN^{-1}$ and $B_y = NB$.

With these new dynamics, we use a linear quadratic regulator (LQR) to minimize over the control space the following quadratic cost L such that

$$L = \int_{t_0}^{\infty} \left(y^{\Delta\top} Q y^{\Delta} + u^{\Delta\top} P u^{\Delta} \right) dt, \tag{22}$$

where Q is the weight matrix for the state, and P is the weight matrix for the control. The problem then boils down to solving a Riccati Equation, which provides us with the optimal gain matrix K such that $u^{\Delta} = -K y^{\Delta}$ induces the minimum cost L. The calibration of the cost matrices Q and P allows to modify the behavior of the controller.

This feedback takes into account the balance, contact kinematics, and forces in the same loop.

C. Integration Within a Multi-Objective Motion Generator

The proposed control is using $6 \times n_c$ variables of control. Many robots, especially humanoid ones, are equipped with more d.o.f. and are able to perform other concurrent tasks. To deal with the redundancy and with additional objectives, we propose to use a whole-body motion solver based on a quadratic program (QP). This optimization problem minimizes the tracking error of the different weighted objectives. It is convenient to have the acceleration $\dot{\alpha}$ as a decision variable to take into account dynamical constraints. The QP calculates the optimal reference acceleration $\dot{\alpha}_r$, by solving

$$\begin{aligned} \dot{\alpha}_r = \underset{\xi}{argmin}\, &\left\| W_{task} \left(A_{ob}\xi - b_{ob} \right) \right\|^2, \\ s.t.\ &A_{eq}\xi = b_{eq},\ A\xi \leq b,\ l_b \leq \xi \leq u_b \end{aligned} \tag{23}$$

W_{task} is a positive diagonal matrix made up of diagonal weighting matrices for each objective. The matrices A_{ob}, A_{eq}, and A and the vectors b_{ob}, b_{eq},b, l_b, and u_b contain the corresponding objectives and constraints. Note that our controller naturally fits into this motion solver since it directly provides the desired Cartesian accelerations of the contact bodies and needs only simple Jacobians. For details about how the tasks and constraints for the QP are formulated see [22] and [23], the difference is that in our case the QP generates the reference joint accelerations only, and that the tasks for the contact points get their accelerations from the controller.

Figure 1 gives an overall view of the robot and the control structure. The robot's model is reduced and used by the controller to generate the accelerations for the contact points, whose objectives, alongside the posture objective and constraints, are minimized using the QP. The latter generates the reference accelerations for the torque controller to produce the joint torques using (2) with a passivity based term $\delta\tau$, which is calculated using

$$\delta\tau = (C + \lambda H)\,(\alpha_r - \alpha), \tag{24}$$

where α_r is the reference velocity vector, obtained by integrating $\dot{\alpha_r}$, λ is a constant, H is the Inertia matrix, and C the Coriolis matrix. For more details about the passivity-term, see [23].

V. Simulations and Results

The control is tested in simulation on a biped described in [22], designed using Matlab Simscape Multibody. The robot has 26 d.o.f in total: 6 d.o.f for each leg and 7 for each arm, having a total mass of 77 Kg. The Simscape Multibody Contact Forces Library was used to generate the contact models. Each corner of the base of a foot or hand has a sphere attached to it, generating linear and angular stiffness, plus damping. The simulations were run while allowing the simulator to automatically choose the max and min step size with a variable step solver. This is considered as higher precision simulations that are able to break down highly dynamical effects and is computationally very costly [24].

The control is compared to the stabilizer of [25], where the effect of the compliance in the environment is modeled as a first order lag system on the zero-moment point. This stabilizer replaces our controller in Figure 1, generating feet damping control, and the QP in this case generates the contact forces alongside the reference accelerations, just like in [23]. We're going to refer to this stabilizer as the feet damping controller.

The tests were done on different starting postures labeled scenarios, as illustrated in Figure 2. We start the simulation with the robot in contact with the environment, slightly above

the equilibrium state, to avoid introducing too much initial energy, and the robot is perturbed in order to assess control robustness. The robot is expected to maintain its balance and to keep the contacts with the environment, which is why we focus on the CoM position of the robot, and on the center of pressure (CoP) of each contact (in the contact frame) while respecting the friction constraint, hence we set the weight matrix W such as $w_{fi} = \begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix}$ and $w_{ti} = \begin{bmatrix} 0.5 & 0.5 & 0 \end{bmatrix}$ in all scenarios. Furthermore, to challenge the controller, white noise is added to the measured forces, and modeling error is introduced by reducing the stiffness by 20% and the damping by 10%.

A. Two Contacts Case

The robot, in this case, is in a half-sitting configuration, making contacts with its feet on the ground, as shown in Figure 2 (a). The stiffness and damping constants at the contacts are given in Table I for the Right Foot (RF) and Left Foot (LF). We choose the LQR control matrices such that $Q = diag(\ Q_{com} \quad Q_c \quad Q_c\ )$, with $Q_{com} = (\ 1e4\mathbb{I}_3 \quad \mathbb{I}_3 \quad 3e4\mathbb{I}_3 \quad \mathbb{I}_3\ )$, $Q_c = (\ 3e4\mathbb{I}_3 \quad \mathbb{I}_3 \quad 3e4\mathbb{I}_3 \quad \mathbb{I}_3\ )$ and $P = diag(\mathbb{I}_6)$. The robot is hit with a perturbation of $250N$ at $t = 0.5s$ for $0.1s$.

Contact	RF	LF
Stiffness KFP (N/m)	3e4	3e4
Damping KFD (N.s/m)	1e3	1e3
Stiffness KTP (N/rad)	400	400
Damping KTD (N.s/rad)	15	15

TABLE I

STIFFNESS AND DAMPING FOR EACH FOOT

Our proposed controller manages to keep the robot's balance, while the feet damping controller struggles until the robot falls down. This is clear when looking at the CoM position in Figure 3 and the CoP of the right foot in Figure 4. The feet damping controller struggles to generate the forces in the non-rigid environment, which explains the large fluctuations in foot CoP and the inability of the controller to stabilize the robot after the perturbation. On the other hand, our proposed controller manages to minimize the error of the CoM and to keep the CoP of each foot within its support zone (each foot has a dimension of 0.23×0.13). The maximum value of the friction ratio at the feet does not exceed 0.176, and no slippage occurs.

B. Four Contacts Case

Here we show two different scenarios while the robot is making four contacts with the environment, illustrated in Figure 2, the robot is putting its hands on a table-like surface (scenario (b)), and the robot is making non-coplanar contacts with its hands (scenario (c)). The stiffness and damping constants at the contacts for the Right Foot (RF), Left Foot (LF), Right Hand (RH) and Left Hand (LH) are given in Table II. In these multi-contact scenarios, we are not going to represent the data from the feet damping controller, as the robot was struggling to generate the contact forces through the QP even before any perturbation due to the non-rigidity of the contacts, and the fluctuation in the CoP of each contact was very strong. The contacts broke quickly and the robot fell.

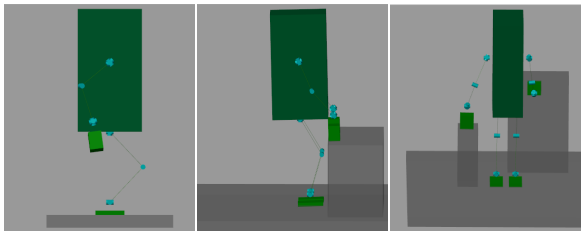

Fig. 2. Starting postures of scenario (a) (left), (b) (middle) and (c) (right)

Contact	RF	LF	RH	LH
KFP (N/m)	3e4	25e3	2e4	1e4
KFD (N.s/m)	4e3	3e3	4e3	4e3
KTP (N/rad)	400	350	180	90
KTD (N.s/rad)	60	60	30	30

TABLE II

CONTACTS STIFFNESS AND DAMPING FOR SCENARIOS (B) AND (C)

With our proposed approach, the robot is hit with a perturbation of $200N$ at $t = 0.5s$ and then again with $250N$ at $t = 2.5s$ for $0.1s$. In the case of scenario (b), a bias of $5°$ to the orientation of the base is added at the start of the simulation to test the robustness of the control due to estimation errors. In this case, we set the control matrices such that $Q = diag(\ Q_{com} \quad Q_c \quad Q_c \quad Q_c \quad Q_c\ )$, with $Q_{com} = (\ 1e4\mathbb{I}_3 \quad 100\mathbb{I}_3 \quad 3e4\mathbb{I}_3 \quad 100\mathbb{I}_3\ )$, $Q_c = (\ 3e4\mathbb{I}_3 \quad 100\mathbb{I}_3 \quad 3e4\mathbb{I}_3 \quad 100\mathbb{I}_3\ )$ and $P = diag(\mathbb{I}_6)$. By setting the angular positions and velocities gains at 100, we were able to minimize the CoM error while regaining the default orientation of the base. We noticed that giving too much weight to the orientations results in worse behavior. In scenario (c), we set $Q_{com} = (\ 1e5\mathbb{I}_3 \quad \mathbb{I}_3 \quad 1e4\mathbb{I}_3 \quad \mathbb{I}_3\ )$, $Q_c = (\ 1e4\mathbb{I}_3 \quad \mathbb{I}_3 \quad 1000\mathbb{I}_3 \quad \mathbb{I}_3\ )$ and $P = diag(\mathbb{I}_6)$.

We see in Figures 5, 6 and 7 that the proposed controller is able to maintain the balance of the robot, minimizing the CoM and CoP errors. No slippage of contacts occurred, and the maximum value of the friction ratio at each contact in each scenario is given in Table III.

Contact	RF	LF	RH	LH
Scenario b	0.31	0.354	0.372	0.355
Scenario c	0.278	0.193	0.361	0.359

TABLE III

MAXIMUM ABSOLUTE VALUE OF THE FRICTION AT EACH CONTACT

VI. CONCLUSION AND FUTURE WORK

In this paper a novel controller is proposed to manage balance and force control in the same loop in task space

for multiple flexible-contacts configurations. A trade-off between force feedback and the kinematics of the system using a flexibility model of the contacts with centroidal dynamics is established to create an LQR control that generates the required accelerations of the contact points for the QP solver. The performance and robustness of the controller has been demonstrated in simulations.

The next step would be to implement this control on a real legged robot, where modeling errors of the stiffness of the environment, estimation errors and noisy force signals could pose a challenge. To overcome these challenges, the trade-off weighting of the positions and the forces will be crucial, by giving more weight to the more reliable signals. Further extension of this work might include an integration of the controller with a motion planner to achieve dynamic locomotion in the context of compliant multi-contacts.

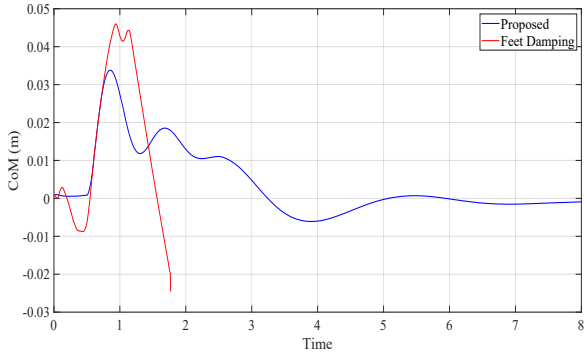

Fig. 3. CoM position error (x component) in scenario (a)

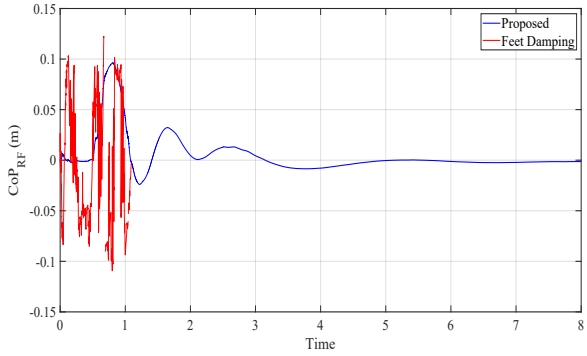

Fig. 4. Right foot CoP (x component) in scenario (a)

APPENDIX

The operator Δ represents the Euclidean difference for the positions and velocities in the state vector for instance $v^{\Delta} = v - v^{*}$. As for the orientations, it is the axis-sine of angle representation of the difference between the orientation matrices, for instance $\Omega^{\Delta} = \Theta(RR^{*\top})$.

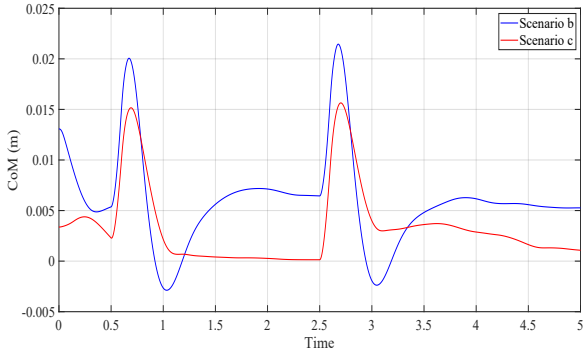

Fig. 5. CoM position error (x component) in scenarios (b) and (c)

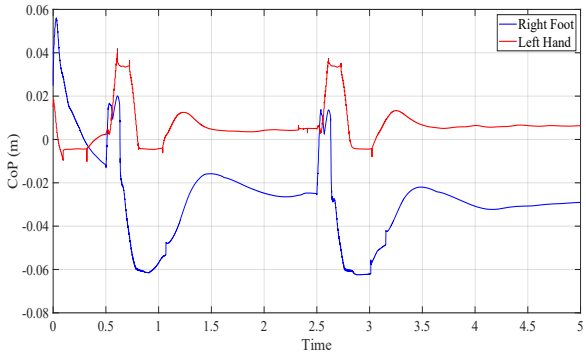

Fig. 6. Right foot and left hand CoP (x component) in scenario (b)

The formula used in the linearization process is:

$$(XY + Z + S(W))^{\Delta} \simeq X^{\Delta}Y^{*} + X^{*}Y^{\Delta} + Z^{\Delta} + S(W^{\Delta}) \tag{25}$$

where X, Y and Z all $\in \mathbb{R}^{3\times3}$, and $W \in \mathbb{R}^{3}$. X^{Δ} is the error of X, X^{*} is its reference value, same for the other variables. $S(W)$ is the skew-symmetric operator of the vector W. As we can notice, the operator Δ functions similarly to the derivative operator. As for the term $\Theta\left(R_{c,i,r}R_{c,i}^{\top}\right) = \Omega\left(R_{c,i,r}R_{c,i,b}^{\top}R^{\top}\right)$, the linearization was done in the limits where the estimation $sin\theta \approx \theta$ can be used, as the following:

$$\begin{aligned}\Omega\left(R_M\right)^{\Delta} &\simeq \frac{1}{2}Vec\left(R_M - R_M^{\top}\right)^{\Delta} \\ &= C_b\left(\Omega^{\Delta} + R^{*}\Omega_{c,i,b}^{\Delta}\right)\end{aligned} \tag{26}$$

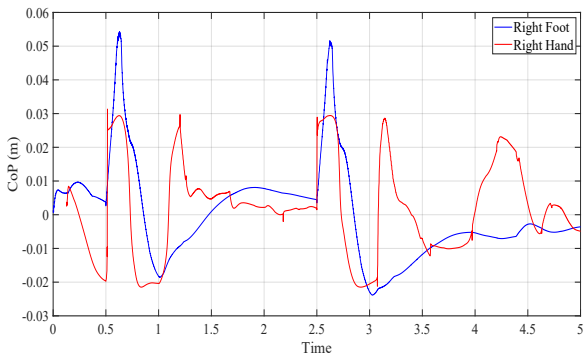

Fig. 7. Right foot and right hand CoP (x component) in scenario (c)

where $C_b = \frac{1}{2}\sum_{i=1}^{3} S(e_i) R_M^* e_i^\top$, $R_M = R_{c,i,r} R_{c,i,b}^\top R^\top$, $Vec(S(\omega)) = \omega$ and $e_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, $e_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$, and $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$.

At the end of the linearization process, we get equation (14), with the matrices A and B given by

$$A = \begin{pmatrix} F_0 & F_1 & F_2 & \dots & F_n \\ 0 & D_1 & 0 & \dots & 0 \\ 0 & 0 & D_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & D_n \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ G_1 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & G_n \end{pmatrix}, \text{ and}$$

$$F_0 = \begin{pmatrix} 0 & \mathbb{I} & 0 & 0 \\ 0 & 0 & 0 & \mathbb{I} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix}, \text{ with}$$

$$\begin{aligned}
A_{31} &= -\frac{1}{m}\sum_{i=1}^{n} K_{f,p,i} \\
A_{32} &= \frac{1}{m}\sum_{i=1}^{n} (A_{321} + A_{322}) \\
A_{33} &= -\frac{1}{m}\sum_{i=1}^{n} K_{f,d,i} \\
A_{34} &= \frac{1}{m}\sum_{i=1}^{n} \left(K_{f,d,i} S\left(R^* \mathbf{p}_{c,i,b}^*\right)\right) \\
A_{321} &= K_{f,p,i} S\left(R^* \mathbf{p}_{c,i,b}^*\right), \\
A_{322} &= K_{f,d,i}\left(S\left(R^* \dot{\mathbf{p}}_{c,i,b}^*\right) + S(\omega^*) S\left(R^* \mathbf{p}_{c,i,b}^*\right)\right) \\
A_{41} &= -R_I \sum_{i=1}^{n} S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{f,p,i} \\
A_{42} &= -R_I \sum_{i=1}^{n} S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{f,d,i} \\
A_{43} &= R_I S(\boldsymbol{\omega}^*) R_I^{-1} S(\boldsymbol{\omega}^*) \\
&\quad + R_I \sum_{i=1}^{n} (A_{431} + A_{432} + A_{433}) \\
A_{44} &= R_I \left(S\left(R_I^{-1}\boldsymbol{\omega}^*\right) - S(\boldsymbol{\omega}^*) R_I^{-1} - \sum_{i=1}^{n} A_{441} \right) \\
R_I &= R^* I^{-1} R^{*\top} \\
A_{431} &= S\left(\mathbf{f}_{c,i}^*\right) S\left(R^* \mathbf{p}_{c,i,b}^*\right) \\
A_{432} &= S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{f,p,i} S\left(R^* \mathbf{p}_{c,i,b}^*\right) \\
&\quad + S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{f,d,i}\left(S\left(R^* \dot{\mathbf{p}}_{c,i,b}^*\right) + S(\omega^*) S\left(R^* \mathbf{p}_{c,i,b}^*\right)\right) \\
A_{433} &= K_{t,p,i} C_{b,i} + K_{t,d,i} S\left(R^* \boldsymbol{\omega}_{c,i,b}^*\right) \\
A_{441} &= K_{t,d,i} + S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{f,d,i} S\left(R^* \mathbf{p}_{c,i,b}^*\right)
\end{aligned}$$

The matrices F_i, D_i and G_i $(i \neq 0)$ are given by

$$F_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ F_{i31} & 0 & -\frac{1}{m} K_{f,d,i} R^* & 0 \\ F_{i41} & F_{i42} & F_{i43} & F_{i44} \end{pmatrix},$$

$$D_i = \begin{pmatrix} 0 & 0 & \mathbb{I} & 0 \\ 0 & 0 & 0 & \mathbb{I} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, G_i = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \mathbb{I} & 0 \\ 0 & \mathbb{I} \end{pmatrix}$$

with

$$\begin{aligned}
F_{i31} &= -\frac{1}{m}\left(K_{f,p,i} + K_{f,d,i} S(\boldsymbol{\omega}^*)\right) R^* \\
F_{i41} &= R_I\left(-S\left(\mathbf{f}_{c,i}^*\right) - S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{dp}\right) R^* \\
K_{dp} &= K_{f,d,i} S(\boldsymbol{\omega}^*) + K_{f,p,i} \\
F_{i42} &= R_I K_{t,p,i} C_{b,i} R \\
F_{i43} &= -R_I S\left(R^* \mathbf{p}_{c,i,b}^*\right) K_{f,d,i} R^* \\
F_{i44} &= -R_I K_{t,d,i} R^*
\end{aligned}$$

The matrix M in (19) is written as:

$$M = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ T_1 & V_1 & 0 & \dots & 0 \\ T_2 & 0 & V_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ T_n & 0 & \dots & 0 & V_n \end{pmatrix}, \tag{27}$$

with

$$T_i = \begin{pmatrix} R^{*\top} & T_{i12} & T_{i13} & -T_{i12} S\left(R^* \mathbf{p}_{c,i}^*\right) \\ 0 & T_{i22} & 0 & R^{*\top} K_{t,p,i}^{-1} K_{t,d,i} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$V_i = \begin{pmatrix} V_{i11} & 0 & V_{i13} & 0 \\ 0 & -R^{*\top} C_{b,i} R & 0 & V_{i24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{aligned}
T_{i12} &= -R^{*\top} S\left(R^* \mathbf{p}_{c,i}^*\right) \\
&\quad - T_{i13}\left(S(R^* \dot{\mathbf{p}}_{c,i}^*) + S(\boldsymbol{\omega}^*) S\left(R^* \mathbf{p}_{c,i}^*\right)\right) \\
T_{i13} &= R^{*\top} K_{f,p,i}^{-1} K_{f,d,i} \\
T_{i22} &= -R^{*\top}\left(C_{b,i} + K_{t,p,i}^{-1} K_{t,d,i} S\left(R^* \boldsymbol{\omega}_{c,i}^*\right)\right) \\
V_{i11} &= R^{*\top}\left(\mathbb{I} + K_{f,p,i}^{-1} K_{f,d,i} S(\omega^*)\right) R^* \\
V_{i13} &= R^{*\top} K_{f,p,i}^{-1} K_{f,d,i} R^* \\
V_{i24} &= R^{*\top} K_{t,p,i}^{-1} K_{t,d,i} R^*
\end{aligned}$$

REFERENCES

[1] Pierre-Brice Wieber. On the stability of walking systems. Proceedings of the International Workshop on Humanoid and Human Friendly Robotics, 2002, Tsukuba, Japan. inria-00390866.

[2] Adrien Escande and Abderrahmane Kheddar. Contact planning for acyclic motion with tasks constraints. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 435–440, St. Louis, MO, USA, October 2009. IEEE.

[3] Layale Saab, Oscar E. Ramos, Francois Keith, Nicolas Mansard, Philippe Soueres, and Jean-Yves Fourquet. Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints. *IEEE Transactions on Robotics*, 29(2):346–362, April 2013.

[4] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur. Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 684–689, Seoul, South Korea, November 2015. IEEE.

[5] Joris Vaillant, Abderrahmane Kheddar, Herve Audren, Francois Keith, Stanislas Brossette, Adrien Escande, Karim Bouyarmane, Kenji Kaneko, Mitsuharu Morisawa, Pierre Gergondet, Eiichi Yoshida, Suuji Kajita, and Fumio Kanehiro. Multi-contact vertical ladder climbing with an HRP-2 humanoid. *Autonomous Robots*, 40(3):561–580, March 2016.

[6] Nicolas Perrin, Darwin Lau, and Vincent Padois. Effective Generation of Dynamically Balanced Locomotion with Multiple Non-coplanar Contacts. In Antonio Bicchi and Wolfram Burgard, editors, *Robotics Research*, volume 3, pages 201–216. Springer International Publishing, Cham, 2018.

[7] N. Kanehira, T.U. Kawasaki, S. Ohta, T. Ismumi, T. Kawada, F. Kanehiro, S. Kajita, and K. Kaneko. Design and experiments of advanced leg module (HRP-2L) for humanoid robot (HRP-2) development. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2455–2460, Lausanne, Switzerland, 2002. IEEE.

[8] Zhibin Li, Nikos G. Tsagarakis, and Darwin G. Caldwell. Walking pattern generation for a humanoid robot with compliant joints. *Autonomous Robots*, 35(1):1–14, July 2013.

[9] Benjamin J. Stephens and Christopher G. Atkeson. Push Recovery by stepping for humanoid robots with force controlled joints. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 52–59, Nashville, TN, USA, December 2010. IEEE.

[10] S Kajita, M Morisawa, K Miura, S Nakaoka, K Harada, K Kaneko, F Kanehiro, and K Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4489–4496, Taipei, October 2010. IEEE.

[11] Shamel Fahmi, Michele Focchi, Andreea Radulescu, Geoff Fink, Victor Barasuol, and Claudio Semini. STANCE: Locomotion Adaptation over Soft Terrain. *IEEE Transactions on Robotics*, 36:443–457, 2020.

[12] M. Azad and M. N. Mistry. "Balance control strategy for legged robots with compliant contacts," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4391-4396, doi: 10.1109/ICRA.2015.7139806.

[13] Vasileios Vasilopoulos, Iosif S. Paraskevas, and Evangelos G. Papadopoulos. Monopod hopping on compliant terrains. *Robotics and Autonomous Systems*, 102:13–26, April 2018.

[14] Michael Neunert, Markus Stauble, Markus Giftthaler, Carmine D. Bellicoso, Jan Carius, Christian Gehring, Marco Hutter, and Jonas Buchli. Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds. *IEEE Robotics and Automation Letters*, 3(3):1458–1465, July 2018.

[15] A. H. Chang, C. M. Hubicki, J. J. Aguilar, D. I. Goldman, A. D. Ames, and P. A. Vela. "Learning to jump in granular media: Unifying optimal control synthesis with Gaussian process-based regression," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 2154-2160, doi: 10.1109/ICRA.2017.7989248.

[16] Mehdi Benallegue and Florent Lamiraux. Humanoid flexibility deformation can be efficiently estimated using only inertial measurement units and contact information. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 246–251, Madrid, Spain, November 2014. IEEE.

[17] Alexis Mifsud, Mehdi Benallegue, and Florent Lamiraux. Stabilization of a compliant humanoid robot using only Inertial Measurement Units with a viscoelastic reaction mass pendulum model. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5405–5410, Daejeon, South Korea, October 2016. IEEE.

[18] Stephane Caron. Biped Stabilization by Linear Feedback of the Variable-Height Inverted Pendulum Model. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9782–9788, 2020.

[19] Stephane Caron, Abderrahmane Kheddar, and Olivier Tempier. Stair Climbing Stabilization of the HRP-4 Humanoid Robot using Whole-body Admittance Control. *2019 International Conference on Robotics and Automation (ICRA)*, pages 277–283, 2019.

[20] Janete Alves, Nuno Peixinho, Miguel Tavares da Silva, Paulo Flores, and Hamid M. Lankarani. A comparative study of the viscoelastic constitutive models for frictionless contact interfaces in solids. *Mechanism and Machine Theory*, 85:172–188, March 2015.

[21] Will Bosworth, Jonas Whitney, Sangbae Kim, and Neville Hogan. Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3582–3589, Stockholm, Sweden, May 2016. IEEE.

[22] Rafael Cisneros, Mehdi Benallegue, Mitsuharu Morisawa, Eiichi Yoshida, Kazuhito Yokoi, and Fumio Kanehiro. Partial Yaw Moment Compensation Using an Optimization-Based Multi-Objective Motion Solver. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1017–1024, Beijing, China, November 2018. IEEE.

[23] Rafael Cisneros, Mehdi Benallegue, Abdelaziz Benallegue, Mitsuharu Morisawa, Herve Audren, Pierre Gergondet, Adrien Escande, Abderrahmane Kheddar, and Fumio Kanehiro. Robust Humanoid Control Using a QP Solver with Integral Gains. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7472–7479, Madrid, October 2018. IEEE.

[24] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots. pages 1–7, March 2021.

[25] Masaki Murooka, Kevin Chappellet, Arnaud Tanguy, Mehdi Benallegue, Iori Kumagai, Mitsuharu Morisawa, Fumio Kanehiro, and Abderrahmane Kheddar. Humanoid Loco-Manipulations Pattern Generation and Stabilization Control. *IEEE Robotics and Automation Letters*, 6(3):5597–5604, July 2021.