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

Marwan Hamze, Mehdi Benallegue, Rafael Cisneros, Abdelaziz Benallegue

A Balance-Admittance Stabilizer 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 stabilizer for multi-contact legged robots that takes into account the environment compliance. This stabilizer is able to achieve both balance and admittance control in the same loop through the use of an explicit contact flexibility model and simplified centroidal dynamics that allows 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 stabilizer with regard to robustness to modeling error and external perturbations have been tested in simulations and compared to a rigid-contact-model regulator.**

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 gives 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

[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.

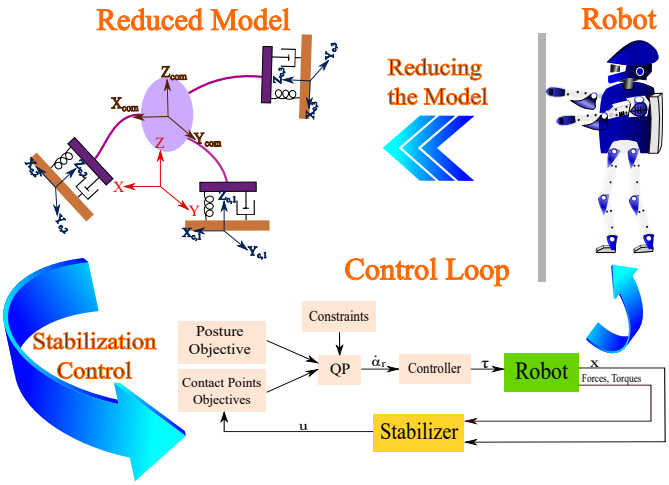

Fig. 1. Diagram showing the robot, its reduced model and the control loop. The world, CoM and contact frames are highlighted for the reduced model.

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].

Classically, to overcome these issues, a second loop of force control is usually 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 is 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

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 $R \in \mathbb{R}^{3\times3}$ represent the position and orientation matrix of the non-actuated floating base, and $q \in \mathbb{R}^n$ being 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:

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

where $M(q) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the robot's mass matrix, $C(q,\alpha) \in \mathbb{R}^{(n+6)\times(n+6)}$ is a matrix accounting for the Coriolis and centrifugal effects, $G(q) \in \mathbb{R}^{n+6}$ 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 being 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 Stabilizer

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:

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

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

$p_{c,i} \in \mathbb{R}^3$, $\dot{p}_{c,i} \in \mathbb{R}^3$ 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 $R_{c,i} \in \mathbb{R}^{3\times3}$ is the orientation matrix of the contact i. $p_{c,i,r} \in \mathbb{R}^3$ and $R_{c,i,r} \in \mathbb{R}^{3\times3}$ are respectively the rest position and orientation of the contact i, i.e. when the forces and torques are zero. $K_{f,p,i}$ 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 relatively small that the approximation $sin\theta \approx \theta$ is used, thus we define the function Θ as the axis-sine of angle representation. Note that a point contact i consist simply in setting $K_{t,p,i}$ and $K_{t,d,i}$ to zero, and edge contact by setting semi definite values for these matrices. These values can be set independently from a contact to another.

B. State and Command Vectors

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 contacts 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 base. The state vector is defined by:

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

where $x_{com} = \begin{pmatrix} p_{com}^\top & \dot{p}_{com}^\top & \Omega^\top & \omega^\top \end{pmatrix}^\top$ and $x_i = \begin{pmatrix} p_{c,i,com}^\top & \Omega_{c,i,b}^\top & \dot{p}_{c,i,com}^\top & \omega_{c,i,b}^\top \end{pmatrix}^\top$. $p_{com} \in \mathbb{R}^3$ 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 $\dot{p}_{c,i,com} \in \mathbb{R}^3$, $\Omega_{c,i,com} \in SO(3)$ and $\omega_{c,i,com} \in \mathbb{R}^3$ are respectively the linear position, velocity, orientation and angular velocity 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 any representation of the orientation, such as the quaternion, the axis-angle, etc. Each Ω has an associated rotation matrix $R \in \mathbb{R}^{3\times3}$.

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

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

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 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) above, the linear and angular accelerations of the CoM and of the base of the robot can be expressed as:

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

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

where $g = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\top$ is the gravity field vector.

Using equations (7), (11) and (12), we can finally write the non-linear model of the robot as $\dot{x} = f\left(x\right) + g\left(x\right) u$.

IV. Proposed balance-admittance stabilizer

The dynamics given by equations (11) and (12) can be used to derive different types of control, in this paper we propose to stabilize the robot around a 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 the desired state. To simplify the notation, we represent the error between the state value and the corresponding desired value using an operator noted Δ. For the positions and velocities in the state vector, Δ represents the euclidean difference, for example $p_{c,i,com}^\Delta = p_{c,i,com} - p_{c,i,com}^*$. As for the orientations, it is the axis-sine of angle representation of the difference between the orientation matrices: $\Omega^\Delta = \Theta(RR^{*\top})$. The linearization of the dynamic equations was mostly done by applying the formulas (34) and (35) given in the Appendix, where we also give the matrices of the linear model (36) at the end of the linearization process.

One important thing to note is that this linearization is very different from the one commonly performed with the inverted pendulum. Here, no assumption 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 co-planar contacts with different stiffness and damping (including point and edge contact).

A. State Feedback Control

At the end of the linearization process of the reduced model (11) and (12), we define the matrices A and B such that:

$$\dot{x}^\Delta = Ax^\Delta + Bu^\Delta. \tag{13}$$

We use a linear quadratic regulator (LQR) to minimize over the control space the following quadratic cost L such that:

$$L = \int_{t_0}^{\infty} \left(x^{\Delta\top} Q x^\Delta + u^{\Delta\top} P u^\Delta \right) dt, \tag{14}$$

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

This controller takes only the feedback on the state x into account. Therefore, it is purely based on kinematics and cannot track reference forces except through the viscoelastic model. This feedback can be used for stabilization in the case of the absence of contact force and torque sensors.

B. Output Feedback Control

One main target of the proposed stabilizer is to include force tracking in the same control loop as CoM and kinematic tracking. However, the state feedback controller is already using all the available actuation of the system. It means that the control of the forces and kinematics are conflicting and require to establish a trade off. It is common

to have this trade off between the tracking of force at a contact body and the tracking of the position of the same body. Therefore we build an "output vector" where the contact forces and torques are combined with the positions and orientations of the corresponding contact bodies. To make the trade off parameterization easier, the forces are scaled by dividing them by the stiffness of the contacts' springs. Furthermore, the equations (4) and (5) give the forces and torques in the contact frame, while the positions and orientations of contacts are in the CoM and base frames, which is why a transformation of frame is done to write the forces and torques in the base frame. Hence, the force vector z is defined as:

$$z = \begin{pmatrix} 0_{1\times 12} & z_1 & \cdots & z_n \end{pmatrix}^\top, \tag{15}$$

where $z_i = \begin{pmatrix} kf_{c,i}^\top & kt_{c,i}^\top & 0_{1\times 3} & 0_{1\times 3} \end{pmatrix}^\top$. $kf_{c,i}$ and $kt_{c,i} \in \mathbb{R}^3$ are respectively the scaled contact force and torque, written in the base frame, and calculated using (16) and (17):

$$kf_{c,i} = -R^\top K_{f,p,i}^{-1} R_{c,i} f_{c,l,i}, \tag{16}$$

$$kt_{c,i} = -R^\top K_{t,p,i}^{-1} R_{c,i} t_{c,l,i}. \tag{17}$$

Now, we can combine both vectors in what we define as the output vector y:

$$y = (\mathbb{I} - W)\, x + W z. \tag{18}$$

$\mathbb{I}$ 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

$$W = \text{diag}\left(0_{1\times 12}, w_1, \cdots, w_{n_c}\right), \tag{19}$$

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:

$$w_i = \begin{pmatrix} w_{fi} & w_{ti} & 0_{1\times 3} & 0_{1\times 3} \end{pmatrix}, \tag{20}$$

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

Since the forces and torques represented by the vector z are written as functions of the variables in x, we can write, after linearization, the vector z^Δ as a matrix M (obtained in the linearization process and given in (37)) multiplying the vector x^Δ:

$$z^\Delta = M x^\Delta. \tag{21}$$

Using equation (21) in (18), we can write the output error as:

$$y^\Delta = N x^\Delta, \tag{22}$$

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

The dynamics of our system are as the following:

$$\begin{cases} \dot{x}^\Delta \simeq A x^\Delta + B u^\Delta, \\ y^\Delta = N x^\Delta. \end{cases}$$

We can also write the dynamics of the output error as:

$$\dot{y}^\Delta \simeq A_y y^\Delta + B_y u^\Delta, \tag{23}$$

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

With these new dynamics, a new LQR gain is calculated, minimizing:

$$L_y = \int \left(y^{\Delta\top} Q_y y^\Delta + u^{\Delta\top} P u^\Delta\right) dt. \tag{24}$$

We can write Q_y in terms of Q and N. Using equation (22) in (24)

$$L_y = \int \left(x^{\Delta\top} N^\top Q_y N x^\Delta + u^{\Delta\top} P u^\Delta\right) dt. \tag{25}$$

In order to compare the performance of the state feedback and the output feedback controllers, it is better to give them comparable cost to minimize. Thus, doing a correspondence between (14) and (25) we can take:

$$Q_y = N^{-\top} Q N^{-1}. \tag{26}$$

The new LQR gain K_y is calculated using the matrices A_y, B_y, Q_y, P such that $u^\Delta = -K_y y^\Delta$.

This is what we define as the output feedback control, where the feedback is an output formed by a combination of the state and force vectors. 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 at most $6 \times n_c$ d.o.f of actuation. 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:

$$\begin{aligned} \dot{\alpha}_r = \underset{\xi}{argmin}\, &\left\| W_{task} \left(A_{ob}\xi - b_{ob}\right)\right\|^2, \\ s.t. A_{eq}\xi = b_{eq}, &A\xi \leq b, l_b \leq \xi \leq u_b \end{aligned} \tag{27}$$

W_{task} is a positive diagonal matrix made up of diagonal weighting matrices for each objective. The matrices A_{ob}, A_{eq}, A and vectors $b_{ob}, b_{eq}, b, l_b, u_b$ contain the corresponding objectives and constraints, as detailed in the sections below. Note that our stabilizer naturally fits into this motion solver since it provides directly the desired Cartesian accelerations of the contact bodies and needs only simple Jacobians.

1) Objectives: Two different objectives are considered:

- Posture objective: an acceleration objective, tracked with a PD of scalar gains k_p and k_v, is used to track the posture of the robot. The objective is calculated using $A_{ob} = \begin{bmatrix} 0_{n\times 6} & \mathbb{I}_n \end{bmatrix}$ and $b_{ob} = \ddot{g}_{ob}$, where in this case $\ddot{g}_{ob} = k_p\left(q_d - q\right) + k_v\left(\dot{q}_d - \dot{q}\right) + \ddot{q}_d$.
- Contacts objectives: Acceleration objectives are defined for the contact points, labeled as hands and feet objectives. Here we define 2 different cases, according to the control scheme used:
  - Case of the addition of the stabilizer: calculated using $A_{ob} = J_{c,i,com}(q)$ and $b_{ob} = \ddot{g}_{ob,i} - \dot{J}_{c,i,com}(q,\alpha)\alpha$, where $\ddot{g}_{ob,i} = \begin{pmatrix} \ddot{p}^{\top}_{c,i,com} & \dot{\omega}^{\top}_{c,i,b} \end{pmatrix}^{\top}$ are the accelerations of the contacts given by the stabilizer, and $J_{c,i,com}$ is the Jacobian of the contact i, written in the CoM frame, with $\dot{J}_{c,i,com}$ being its time derivative.
  - Case of a default PD of gains k_p and k_v: calculated using $A_{ob} = J_{c,i,com}(q)$ and $b_{ob} = \ddot{g}_{ob,i} - \dot{J}_{c,i,com}(q,\alpha)\alpha$, where $\ddot{g}_{ob,i} = k_p\left(p_{c,i,com,d} - p_{c,i,com}\right) + k_v\left(\dot{p}_{c,i,com,d} - \dot{p}_{c,i,com}\right) + \ddot{p}_{c,i,com,d}$.

2) Constraints: Two different constraints are considered:

- Joint limits constraints: The range and speed limit of the joints can be specified as done in [22].
- Under-actuation and Torque constraint: This constraint ensures that the torques to be calculated are within the limitations of the actuators. An equality constraint is used for the under-actuation, where the index B is used to refer to the first 6 rows of the matrices, while upper and lower bounds are used for the torques, where the index j is used to refer to the remaining rows of the matrices. These constraints are calculated using:

$$A_{eq} = M_B, b_{eq} = -C_B\alpha - G_B + F_B, \tag{28}$$

$$A = M_j, l_b = \tau_{min} - C_j\alpha - G_j + F_j,$$
$$u_b = \tau_{max} - C_j\alpha - G_j + F_j. \tag{29}$$

Figure 1 gives an overall view of the robot and the control structure. The robot's model is reduced and used by the stabilizer to generate the accelerations for the contact points, whose objectives, alongside the posture objective and constraints, are minimized using the QP, which generates the reference accelerations for the controller. The controller uses (2) to produce the joint torques, and adds a passivity based term P before sending the torques to the joints, which is calculated using:

$$P = \left(C + \lambda M\right)\left(\alpha_r - \alpha\right), \tag{30}$$

where α_r is the reference velocity vector, obtained by integrating $\dot{\alpha_r}$, and λ is a constant. For more details about the passivity-term, we suggest to check [22].

V. Simulations and Results

A. Simulation environment and details

The control is tested in simulation on a biped described in [23], 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, having its own stiffness and damping constants, to generate a contact with the environment. The contact forces are generated by the spheres, and the contact torques are be calculated with respect to the hand or foot. The stiffness and damping constants of the contact spheres are shown in table I. The simulations were run while allowing the simulator to choose automatically 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].

Contact	Right foot	Left foot	Right hand	Left hand
Stiffness KFP (N/m)	7500	7500	5000	2500
Damping KFD (N.s/m)	1000	1000	500	250

TABLE I

Constants for each contact sphere

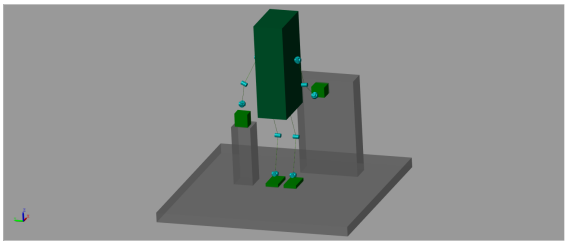

Fig. 2. Starting posture of the robot, making 4 contacts with the environment.

The robot is torque-controlled, which means that the acceleration references are tracked using inverse dynamics and a passivity-based integral term similarly to [22].

B. Stabilization tests

Since our aim is to test a stabilization control, we start the simulation with the robot in contact with the environment (as shown in figure 2) slightly above the equilibrium state, to avoid introducing too much initial energy. After the robot reaches equilibrium, it is perturbed in order to test the stabilization control.

We test our proposed stabilizer against a controller for the CoM and contact bodies assuming rigid contacts, it uses this assumption together with inverse dynamics to produce torques to bring the robot to balance. The CoM is controlled in three dimensions to produce a PD reference acceleration of the CoM, i.e $\ddot{p}^r_{com} = -K_{p,com}p_{com} - K_{v,com}\dot{p}_{com}$ and is subject to ZMP and friction constraints. Since it is perfectly torque controlled, this controller would be a good stabilizer on rigid floor. The hands and feet have their objectives for the QP defined also as PD reference accelerations for this case. The PD gains were set as $K_p = 100$ and $K_p = 20$ for all objectives.

From our stabilizer side, we set the weight matrices as the

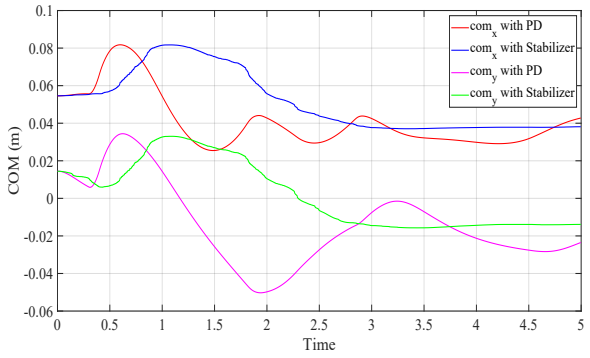

Fig. 3. The x and y components of the CoM position, when using the Stabilizer or the PD gains

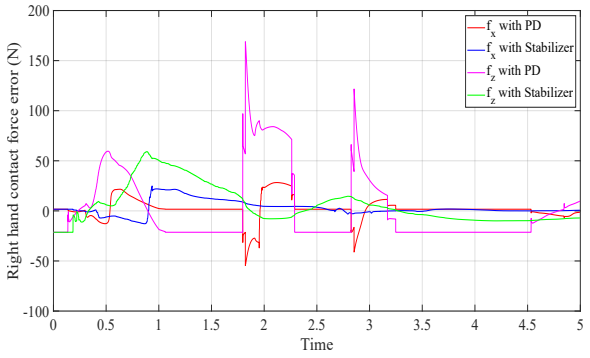

Fig. 4. The x and z components of the contact force error at the right hand, when using the Stabilizer or the PD gains

following:

$$Q = diag\left(100\mathbb{I}_6, \mathbb{I}_{54}\right), \tag{31}$$

$$P = diag\left(P_{lw}, \cdots, P_{lw}\right), P_{lw} = diag\left(10\mathbb{I}_3, \mathbb{I}_3\right), \tag{32}$$

$$w_{fi} = \begin{bmatrix} 0.5 & 0 & 0.5 \end{bmatrix}, w_{ti} = \begin{bmatrix} 0 & 0.5 & 0 \end{bmatrix}, \tag{33}$$

We did not provide the stabilizer with a perfect model of the environment, first the contact model in the simulation is non linear and is approximated by linearization, then we included also modeling errors, underestimating the environment stiffness by 25%.

Both controllers were subjected to external disturbances by simulating pushes on the robot as following:

1) An external push of $f_x = 200N$ and $f_y = 200N$ is applied at t=0.3s for 0.1s. Both controllers were able to stabilize the robot, as shown from the CoM curves in Figure 3. However, the proposed controller is able to stabilize the robot much less oscillations compared to the PD controller, where oscillations are still present even after 5 seconds. Furthermore, Figure 4 show the errors of the x and z component of the contact force at the right hand. The proposed stabilizer is clearly superior in force tracking, as the error of the contact forces is lower throughout the simulation, and can even reach 0 for the x component, even at the presence of modeling errors. On the other hand when using the PD gains, the robot struggles to keep the hand in contact, which is observable via the peaks present at $t = 1.8s$ and $t = 2.8s$.
2) Another simulation has been conducted considering an external push $f_x = 300N$ applied at $t = 0.3s$ for 0.1s, then another force having $f_x = -50N$ and $f_y = -50N$ which hit the robot again at $t = 1s$ for 0.1s. The robot when controlled with the PD gains couldn't maintain its equilibrium, and falls after the second push, while the proposed stabilizer is able to maintain the balance of the robot, as seen from the CoM position in Figure 5. The error of the force at the right hand shows that force tracking is still effective, as seen in Figure 6.

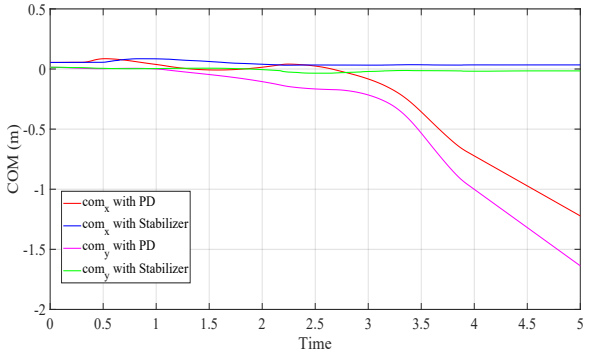

Fig. 5. The x and y components of the CoM position in case of multiple perturbations, when using the Stabilizer or the PD gains

VI. Conclusion and Future Works

We demonstrated in this paper the performance and robustness of the proposed stabilizer when the robot is making multiple non-rigid contacts with the environment. The stabilizer was able to maintain the balance of the robot, even against successive pushes, where the default PD fell short. Additionally, the forces, which are included in the same loop as the balance feedback, ensure the contact forces to track the reference values, even when modeling errors were considered.

We believe this work is only the first controller using this model which can be extended to track dynamic trajectories

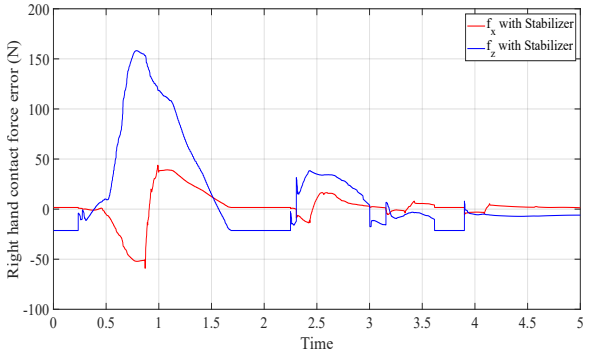

Fig. 6. The x and z components of the contact force error at the right hand when using the Stabilizer, in the case of multiple perturbations

and even exploit the paradigm of divergent component of motion.

Finally, testing on a real humanoid robot is different from testing on a model created in Simulink, where the feedback from the real robot isn't free from noise and error, which wasn't considered in our simulations, in addition to a few other differences. Future works will start by testing the stabilizer on an actual model of a robot like HRP-4, before applying it on the real robot.

APPENDIX

The formula used in the linearization process is the following:

$$(AB + C + S(D))^{\Delta} \simeq A^{\Delta}B^{*} + A^{*}B^{\Delta} + C^{\Delta} + S(D^{\Delta}) \tag{34}$$

where A, B and C all $\in \mathbb{R}^{3\times3}$, and $D \in \mathbb{R}^{3}$. A^{Δ} is the error of A, A^{*} is its reference value, same for the other variables. S is the skew-symmetric operator of the vector D. 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:

$$\begin{aligned}\Omega\left(R_M\right)^{\Delta} &\simeq 0.5Vec\left(R_M - R_M^{\top}\right)^{\Delta} \\ &= C_b\left(\Omega^{\Delta} + R^{*}\Omega_{c,i,b}^{\Delta}\right)\end{aligned} \tag{35}$$

where

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

At the end of the linearization process, we can write:

$$\dot{x}^{\Delta} = Ax^{\Delta} + Bu^{\Delta}, \tag{36}$$

where $A = \begin{pmatrix} F_0 & F_1 & F_2 & \dots & F_n \\ 0 & D_1 & 0 & \dots & 0 \\ 0 & 0 & D_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & D_n \end{pmatrix}, B = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ G_1 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & G_n \end{pmatrix}$, with:

$$F_0 = \begin{pmatrix} 0 & \mathbb{I} & 0 & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & 0 & 0 & \mathbb{I} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix}$$

and

$$\begin{aligned}A_{21} &= -\frac{1}{m}\sum_{i=1}^{n} K_{f,p,i} \\ A_{22} &= -\frac{1}{m}\sum_{i=1}^{n} K_{f,d,i} \\ A_{23} &= \frac{1}{m}\sum_{i=1}^{n}\left(A_{231} + A_{232}\right) \\ A_{24} &= \frac{1}{m}\sum_{i=1}^{n}\left(K_{f,d,i}S\left(R^{*}\mathbf{p}_{c,i,b}^{*}\right)\right) \\ A_{231} &= K_{f,p,i}S\left(R^{*}\mathbf{p}_{c,i,b}^{*}\right), \\ A_{232} &= K_{f,d,i}\left(S\left(R^{*}\dot{\mathbf{p}}_{c,i,b}^{*}\right) + S\left(\omega^{*}\right)S\left(R^{*}\mathbf{p}_{c,i,b}^{*}\right)\right)\end{aligned}$$

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

The matrices F_i, D_i and G_i are given by

$$F_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ F_{i21} & 0 & -\frac{1}{m}K_{f,d,i}R^{*} & 0 \\ 0 & 0 & 0 & 0 \\ F_{i41} & F_{i42} & F_{i43} & F_{i44} \end{pmatrix},$$

$$\begin{aligned}F_{i21} &= -\frac{1}{m}\left(K_{f,p,i} + K_{f,d,i}S\left(\boldsymbol{\omega}^{*}\right)\right)R^{*} \\ F_{i41} &= R_I\left(-S\left(\mathbf{f}_{c,i}^{*}\right) - S\left(R^{*}\mathbf{p}_{c,i,b}^{*}\right)K_{dp}\right)R^{*} \\ K_{dp} &= K_{f,d,i}S\left(\boldsymbol{\omega}^{*}\right) + K_{f,p,i} \\ F_{i42} &= R_I K_{t,p,i}C_{b,i}R \\ F_{i43} &= -R_I S\left(R^{*}\mathbf{p}_{c,i,b}^{*}\right)K_{f,d,i}R^{*} \\ F_{i44} &= -R_I K_{t,d,i}R^{*}\end{aligned}$$

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

The matrix M in (21) is written as:

$$M = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ T_1 & V_1 & 0 & \dots & 0 \\ T_2 & 0 & V_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ T_n & 0 & \dots & 0 & V_n \end{pmatrix}, \tag{37}$$

with:

$$T_i = \begin{pmatrix} R^{*\top} & T_{i12} & T_{i13} & -T_{i12} S\left(R^* \mathbf{p}^*_{c,i}\right) \\ 0 & 0 & T_{i23} & R^{*\top} K^{-1}_{t,p,i} K_{t,d,i} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

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

$$\begin{aligned} T_{i12} &= R^{*\top} K^{-1}_{f,p,i} K_{f,d,i} \\ T_{i13} &= -R^{*\top} S\left(R^* \mathbf{p}^*_{c,i}\right) \\ &\quad - T_{i12}\left(S(R^* \dot{\mathbf{p}}^*_{c,i}) + S\left(\boldsymbol{\omega}^*\right) S\left(R^* \mathbf{p}^*_{c,i}\right)\right) \\ T_{i23} &= -R^{*\top}\left(C_{b,i} + K^{-1}_{t,p,i} K_{t,d,i} S\left(R^* \boldsymbol{\omega}^*_{c,i}\right)\right) \\ V_{i11} &= R^{*\top}\left(\mathbb{I} + K^{-1}_{f,p,i} K_{f,d,i} S\left(\omega^*\right)\right) R^* \\ V_{i13} &= R^{*\top} K^{-1}_{f,p,i} K_{f,d,i} R^* \\ V_{i24} &= R^{*\top} K^{-1}_{t,p,i} K_{t,d,i} R^* \end{aligned}$$

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. Series Title: Springer Proceedings in Advanced Robotics.

[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. *arXiv:1904.12306 [cs]*, November 2019. arXiv: 1904.12306.

[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. *arXiv:1909.07732 [cs]*, March 2020. arXiv: 1909.07732.

[19] Stephane Caron, Abderrahmane Kheddar, and Olivier Tempier. Stair Climbing Stabilization of the HRP-4 Humanoid Robot using Whole-body Admittance Control. *arXiv:1809.07073 [cs]*, March 2020. arXiv: 1809.07073.

[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, 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.

[23] 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.

[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.