To appear in *Advanced Robotics*
Vol. 00, No. 00, — 2020, 1–26

Article

An inverse dynamics-based multi-contact locomotion control framework without joint torque feedback

Rafael CISNEROS-LIMON † ‡ * , Mitsuharu MORISAWA † ‡, Mehdi BENALLEGUE † ‡, Adrien ESCANDE † ‡ and Fumio KANEHIRO † ‡

† *CNRS-AIST JRL (Joint Robotics Laboratory), IRL*;
‡ *National Institute of Advanced Industrial Science and Technology (AIST), Department of Information Technology and Human Factors, Tsukuba, Japan*

(*Received December 2019; Revised May 2020; Revised September 2020*)

Humanoid robots are expected to evolve in complex environments performing parallel tasks including balance in multi-contact motion. Balance requires precise tracking of the contact forces, even in the presence of external disturbances. In this paper, we propose a framework to perform stabilization, force tracking, kinematic tasks, and disturbance-rejecting compliance with robots without joint torque feedback. The solution uses a QP with concurrent tasks to produce an inverse dynamics-based feed-forward torque together with kinematic feedback to achieve feasible Lyapunov-stable motions. The framework offers a range of task formulations and parameters as tools for fine force tracking, including an admittance-like task. This framework is tested in dynamic simulations with several locomotion scenarios in complex environments with continuous non-modeled disturbances.

Keywords: Inverse dynamics; QP-based control; Multi-contact locomotion; Balance control

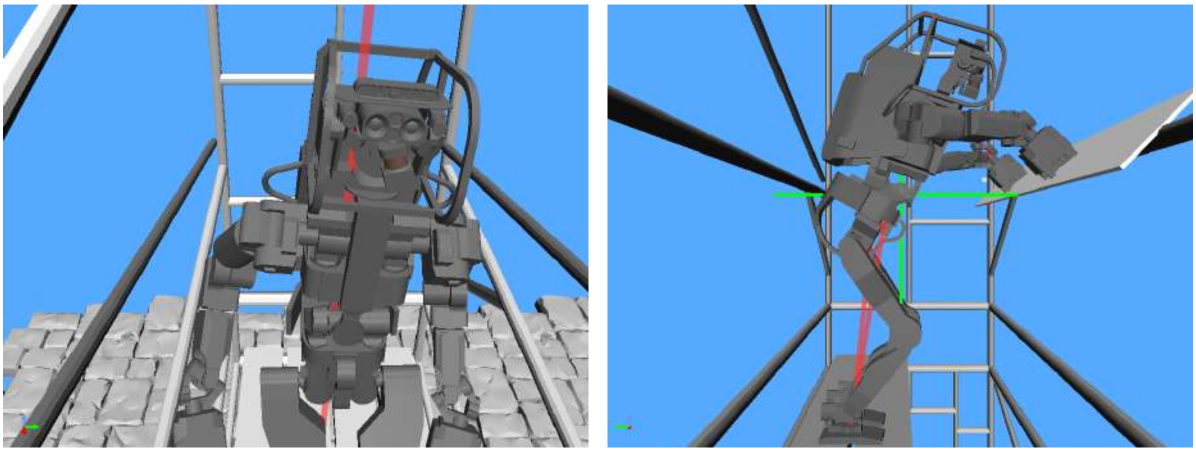
1. Introduction

For a humanoid robot to be useful, it must be capable of traversing complex environments by using its whole body while performing multiple tasks. Proper use of the human-like morphology should allow humanoid robots to work in unstructured environments comprising uneven and discontinuous surfaces, obstacles, and/or narrow spaces. To evolve in these environments, it is necessary to simultaneously achieve multiple objectives, like keeping balance by using multiple contacts and controlling its end-effectors to perform locomotion. These tasks often go beyond kinematic operations and deal with the dynamics of the environment as well as interaction forces. These forces are critical to ensure balance and to respect dynamical constraints imposed by the environment. It is then necessary to ensure that the actual forces match the desired ones, despite the presence of many modeling errors and disturbances. These include non-modeled dynamics, geometric discrepancies, and unexpected external forces due to collisions, which are likely to happen in realistic environments (see Figure 1).

A whole-body motion control framework for a humanoid robot with multiple task layers was first proposed in [1] and [2]. Focusing on the ground reaction force, an optimal force distribution for torque-controlled humanoid robots was proposed in [3]. It was later extended to consider the joint torques as a decision variables in [4]. A constrained optimization method was introduced for whole-body control in [5] and [6]. Centroidal dynamics were considered as constraints in [7].

For a long time, optimization-based whole-body controllers for torque-controlled humanoid

*Corresponding author. Email: rafael.cisneros@aist.go.jp



(a) Bipedal locomotion

(b) Multi-contact locomotion

Figure 1. Stable locomotion in narrow spaces that are typical in the large-scale construction industry where unexpected collisions are likely.

robots were mainly able to maintain the robot balance, but it was still difficult for a real humanoid robot to walk. Since the DARPA Robotics Challenge, optimization-based whole-body controllers using centroidal dynamics constraints were widely used in torque-controlled humanoid robots, and biped locomotion was finally realized with real robots [8] [9] [10]. Passivity-based whole-body control with multiple objectives was proposed in [6] and succeeded to walk on rough and deformable terrain by using a Divergent Component of Motion (DCM) tracking control [11].

However, while these controllers consider contact force constraints and joint torque limits, it is difficult to use these controllers on humanoid robots with joints that have high reduction gears without a torque sensor. Instead of using joint torque control, a whole-body compliant motion was realized by calculating joint PD gains equivalent to a desired stiffness and viscosity in task space by using Resolved Viscoelasticity Control [12] and considering highly back-drivable actuators. For position-controlled humanoid robots, an admittance control of the end-effectors based on an optimization-based whole-body controller was proposed in [13]. By using this method, the robot cannot realize locomotion. Caron et al [14] also formulated an optimal force distribution approach. This framework uses inverse kinematics instead of inverse dynamics. Because of that, the contact force is controlled within the outer loop of joint position control. This, however, prevents to increase the bandwidth of the force control.

We propose a unified framework that allows controlling whole-body motions for humanoid robots without joint torque sensors, especially in multi-contact scenarios. The underlying controller is based on inverse dynamics and uses kinematic feedback to achieve passivity-based Lyapunov-stable control [15]. To achieve stable locomotion, we introduce a set of QP tasks that aim to control the force distribution to realize a DCM-based balance control. The force distribution is realized thanks to an admittance-like control [16] that exploits the internal forces [17]. This force distribution-oriented framework aligns with the multi-contact motion generation introduced in [18]. The compliant behavior of the inverse dynamics-based control and the balance control work synergistically to provide robustness against unexpected external forces during locomotion.

This paper is organized as follows:

- Section 2 introduces notation and provides a summary of the concept of force distribution ratio [18], extensively used through this paper. It also explains how to estimate the pose of the floating-base.
- Section 3 summarizes the multi-contact motion control framework introduced in [15]. Its purpose is to introduce notation and concepts and to provide an improvement.
- Section 4 describes the tasks that we introduce in this paper to implement the stabilization framework within the inverse dynamics-based control framework. These tasks represent the

main contribution of this paper concerning our previous works [15] [16] [17][1].
- Section 5 presents some biped and multi-contact simulation results to assess the proposed framework.
- Finally, Section 6 concludes the paper.

2. Humanoid Robot Model

2.1 *Robot Dynamics*

Let us consider a humanoid robot having $n+6$ degrees of freedom (dof) and describe its configuration as $\boldsymbol{q} = (\boldsymbol{p_B}, \boldsymbol{R_B}, \boldsymbol{q_\theta})$, where $\boldsymbol{p_B} \in \mathrm{R}^3$ is the position of the floating-base, $\boldsymbol{R_B} \in SO(3)$ represents its orientation, and $\boldsymbol{q_\theta} \in \mathrm{R}^n$ comprises the joint angles. The *configuration velocity*, $\boldsymbol{\alpha} \in \mathrm{R}^{n+6}$, is given by

$$\boldsymbol{\alpha} = \left[\boldsymbol{v_B^T} \ \boldsymbol{\omega_B^T} \ \boldsymbol{\dot{q}_\theta^T} \right]^T. \tag{1}$$

Here, $\boldsymbol{v_B}$ and $\boldsymbol{\omega}$ are the linear and angular velocities of the floating-base. The time derivative of the configuration velocity, $\dot{\boldsymbol{\alpha}} \in \mathrm{R}^{n+6}$, is the *configuration acceleration.*

The dynamical model of the humanoid robot is written

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\alpha}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\alpha})\boldsymbol{\alpha} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} + \boldsymbol{u_e}. \tag{2}$$

Here, $\boldsymbol{M}(\boldsymbol{q}) \in \mathrm{R}^{(n+6)\times(n+6)}$ is the mass matrix. $\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\alpha}) \in \mathrm{R}^{(n+6)\times(n+6)}$ is a Coriolis and centripetal matrix such that $\dot{\boldsymbol{M}}(\boldsymbol{q},\boldsymbol{\alpha}) - 2\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\alpha})$ is skew-symmetric, and $\boldsymbol{g}(\boldsymbol{q}) \in \mathrm{R}^{n+6}$ is the vector of gravitational effects. Additionally, $\boldsymbol{u} = \left[\boldsymbol{0_6^T} \ \boldsymbol{u_\theta} \right]^T$ corresponds to the vector of input generalized forces. The zero vector $\boldsymbol{0_m} \in \mathrm{R}^m$ with $m = 6$ represents the 6 unactuated dof corresponding to the position and orientation (later on referred to as *pose*) of the floating-base, whereas $\boldsymbol{u_\theta} \in \mathrm{R}^n$ corresponds to the actuated dof. Finally, $\boldsymbol{u_e} \in \mathrm{R}^6$ corresponds to the vector of external generalized forces induced by the environment.

2.2 *Unilateral Contact Model*

Let us consider that the humanoid robot establishes contact with the environment by using L links, as shown in Figure 2. The contact surface of each link can be described with a k-tuple of xy-points defined with respect to the surface frame with origin in $\boldsymbol{p_l}$. These points represent the vertices of a convex polygon that approximates the contact region.

We assume that the contact force distribution on the surface of link l can be approximated by K_l lumped reaction force vectors placed at the vertices of the contact region. These forces are directed *towards* the contact surface as they represent reaction forces. Each one of these forces can be denoted as $\boldsymbol{f_{l,k}}$, $l \in \{1,\ldots,L\}$, $k \in \{1,\ldots,K_l\}$.

To hold a unilateral contact: (a) the normal component of each force vector must be positive and (b) each force vector must remain inside the corresponding friction cone, i.e. $\boldsymbol{f_{l,k}} \in \mathcal{C}_{l,k}$, to avoid tipping and slipping.

To fulfill both requirements, we employ pyramidal approximations of each friction cone, $\mathcal{P}_{l,k} \subset \mathcal{C}_{l,k}$, described by 4 unitary bases denoted as $\boldsymbol{\beta_{l,k,j}}$, $j \in \{1,\ldots,4\}$ and arranged as columns of a matrix $\boldsymbol{\beta_{l,k}} \in \mathrm{R}^{3\times J}$; that is,

[1]In [15], although we introduced the passivity-based Lyapunov-stable control, the inverse dynamics-based framework was limited to perform a statically stable motion. In [16] we introduced the force control framework and demonstrated locomotion on an irregular floor but without properly exerting balance control. In [17] we introduced the concept of force distribution ratio and achieved stabilization using stiff position-based control, which is not robust against unexpected contacts.

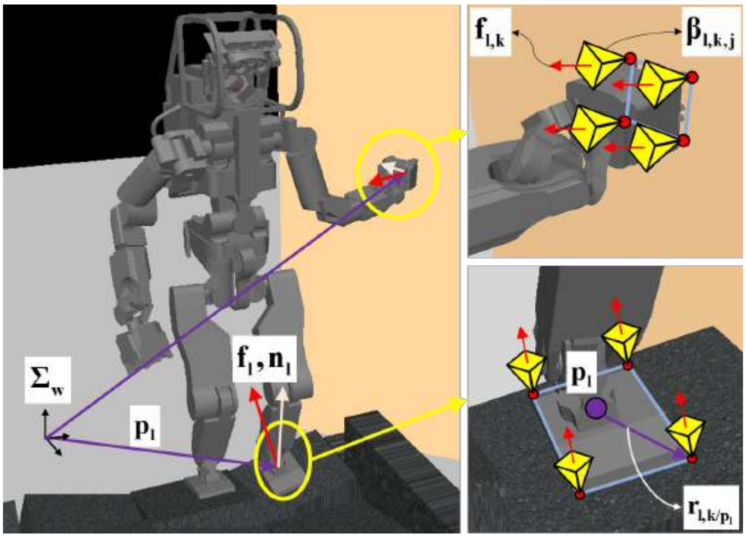


Figure 2. Lumped reaction forces acting on contact surfaces of HRP-5P, constrained inside of pyramidal approximations of the friction cones [17].

$$\boldsymbol{f_{l,k}} = \begin{bmatrix} \boldsymbol{\beta_{l,k,1}} \cdots \boldsymbol{\beta_{l,k,J}} \end{bmatrix} \boldsymbol{\rho_{l,k}} = \boldsymbol{\beta_{l,k}} \boldsymbol{\rho_{l,k}}, \tag{3}$$

where $\boldsymbol{\rho_{l,k}} \in \mathrm{R}^4$ is a vector of non-negative coefficients that constrains each force $\boldsymbol{f_{l,k}}$ to be inside of the friction pyramid.

From this representation, the external wrench $\boldsymbol{F_l}$ acting on link l is calculated with a wrench matrix $\boldsymbol{W_l} \in \mathrm{R}^{6\times 4K_l}$ and a concatenated vector of coefficients $\boldsymbol{\rho_l} \in \mathrm{R}^{4K_l}$, as

$$\begin{aligned} \boldsymbol{F_l} &= \begin{bmatrix} \boldsymbol{f_l} \\ \boldsymbol{n_l} \end{bmatrix} = \sum_{k=1}^{K_l} \begin{bmatrix} \boldsymbol{\beta_{l,k}} \\ \boldsymbol{S}(\boldsymbol{r_{k/p_l}})\boldsymbol{\beta_{l,k}} \end{bmatrix} \boldsymbol{\rho_{l,k}} = \begin{bmatrix} \boldsymbol{\beta_{l,1}} & \cdots & \boldsymbol{\beta_{l,K_l}} \\ \boldsymbol{S}(\boldsymbol{r_{1/p_l}})\boldsymbol{\beta_{1,k}} & \cdots & \boldsymbol{S}(\boldsymbol{r_{K_l/p_l}})\boldsymbol{\beta_{l,K_l}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho_{l,1}} \\ \vdots \\ \boldsymbol{\rho_{l,K_l}} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{W_{l,f}} \\ \boldsymbol{W_{l,n}} \end{bmatrix} \boldsymbol{\rho_l} = \boldsymbol{W_l} \boldsymbol{\rho_l} \end{aligned} \tag{4}$$

Here, $\boldsymbol{f_l}$ and $\boldsymbol{n_l}$ are the contact force and couple moment acting at the anchor point $\boldsymbol{p_l}$ of link l. Also, $\boldsymbol{r_{k/p_l}}$ represents the relative position from $\boldsymbol{p_l}$ to the point of application of $\boldsymbol{f_{l,k}}$ and $\boldsymbol{S}(\cdot) : \mathrm{R}^3 \to \mathrm{R}^{3\times 3}$ is the skew-symmetric operator. These vectors are expressed in world coordinates.

Then, $\boldsymbol{u_e}$ in (2) can be expressed as

$$\boldsymbol{u_e} = \begin{bmatrix} \boldsymbol{u_{e,B}} \\ \boldsymbol{u_{e,\theta}} \end{bmatrix} = \sum_{l=1}^{L} \boldsymbol{J_l^T F_l} = \begin{bmatrix} \boldsymbol{J_1^T W_1} \cdots \boldsymbol{J_L^T W_L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho_1} \\ \vdots \\ \boldsymbol{\rho_L} \end{bmatrix} = \boldsymbol{D\rho}, \tag{5}$$

where $\boldsymbol{u_{e,B}} \in \mathrm{R}^6$ and $\boldsymbol{u_{e,\theta}} \in \mathrm{R}^n$. Also, $\boldsymbol{J_l} \in 6 \times (6+n)$ is the Jacobian of point $\boldsymbol{p_l}$.

Note that $\boldsymbol{u_{e,B}} = \begin{bmatrix} \boldsymbol{f_B^T} \ \boldsymbol{n_B^T} \end{bmatrix}^T$, where $\boldsymbol{f_B}$ and $\boldsymbol{n_B}$ are the resultant force and couple moment acting at $\boldsymbol{p_B}$.

2.3 *Force Distribution Ratio*

We consider the concept of *force distribution ratio* among all the contact links. This one was originally proposed in [18] as an intuitive way to generate the Center of Mass (CoM) motion and later used in [17] to perform multi-contact motion control on a position-controlled robot. Here we use it within our inverse dynamics-based control framework. A summary is given below.

In addition to the dynamics of the robot in (2) we write the centroidal dynamics as

$$\begin{bmatrix} m\boldsymbol{I_3} & \boldsymbol{0} \\ m\boldsymbol{S}(\boldsymbol{p_G}) & \boldsymbol{I_3} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{p}}_{\boldsymbol{G}} \\ \dot{\boldsymbol{k}}_{\boldsymbol{G}} \end{bmatrix} - \begin{bmatrix} m\boldsymbol{g} \\ m\boldsymbol{S}(\boldsymbol{p_G})\boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_0} \\ \boldsymbol{n_0} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{p_G} \in \mathrm{R}^3$ is the CoM position, m is the total mass of the robot, $\boldsymbol{g} = \begin{bmatrix} 0 \; 0 \; -g \end{bmatrix}^T$ is the gravity vector, and $\boldsymbol{k_G} \in \mathrm{R}^3$ is the angular momentum around the CoM. Finally, $\boldsymbol{f_0} \in \mathrm{R}^3$ and $\boldsymbol{n_0} \in \mathrm{R}^3$ represent the reaction force and moment with respect to the origin of the world frame.

We can also express $\boldsymbol{f_0}, \boldsymbol{n_0}$ as

$$\begin{bmatrix} \boldsymbol{f_0} \\ \boldsymbol{n_0} \end{bmatrix} = \sum_{l=1}^{L} \begin{bmatrix} \boldsymbol{f_l} \\ \boldsymbol{S}(\boldsymbol{p_l})\boldsymbol{f_l} + \boldsymbol{n_l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J_t} \\ \boldsymbol{J_r} \end{bmatrix} \boldsymbol{f_c} + \begin{bmatrix} \boldsymbol{0_{3\times 3L}} \\ \boldsymbol{J_t} \end{bmatrix} \boldsymbol{n_c}, \tag{7}$$

where $\boldsymbol{f_c}, \boldsymbol{n_c} \in \mathrm{R}^{3L}$ are the sets of contact forces and moments acting on the links, arranged as

$$\boldsymbol{f_c} = \begin{bmatrix} \boldsymbol{f_1^T} \cdots \boldsymbol{f_L^T} \end{bmatrix}^T, \quad \boldsymbol{n_c} = \begin{bmatrix} \boldsymbol{n_1^T} \cdots \boldsymbol{n_L^T} \end{bmatrix}^T, \tag{8}$$

whereas $\boldsymbol{J_t}, \boldsymbol{J_n} \in \mathrm{R}^{3\times 3L}$ represent contact Jacobians. These are given by

$$\boldsymbol{J_t} = \begin{bmatrix} \boldsymbol{I_3} \cdots \boldsymbol{I_3} \end{bmatrix}, \quad \boldsymbol{J_r} = \begin{bmatrix} \boldsymbol{S}(\boldsymbol{p_1}) \cdots \boldsymbol{S}(\boldsymbol{p_L}) \end{bmatrix}, \tag{9}$$

where $\boldsymbol{I_m} \in \mathrm{R}^{m\times m}$ represents the identity.

The objective is to find a non-unique solution for $\boldsymbol{f_c}$ in (7) satisfying $\boldsymbol{f_0}$ in (6). One way to find that solution is by defining a *force distribution matrix*, $\boldsymbol{G_\sigma}$, as

$$\boldsymbol{G_\sigma} = \begin{bmatrix} \boldsymbol{\sigma_1} \cdots \boldsymbol{\sigma_L} \end{bmatrix}^T \tag{10}$$

where $\boldsymbol{\sigma_l} = \mathrm{diag}\left(\sigma_{l,x}, \sigma_{l,y}, \sigma_{l,z}\right)$ and $\sum_{l=1}^{L} \sigma_{l,x} = \sum_{l=1}^{L} \sigma_{l,y} = \sum_{l=1}^{L} \sigma_{l,z} = 1$. Here, $\sigma_{l,\circ}$ is a *force distribution ratio* and $\boldsymbol{G_\sigma}$ corresponds to a weighted pseudo-inverse of J_t, such that from the upper part of (7) we can get

$$\begin{aligned} \boldsymbol{f_c} &= \boldsymbol{G_\sigma} \boldsymbol{f_0} + \boldsymbol{\Phi} \boldsymbol{f}_{\mathrm{int}} \\ &= m\boldsymbol{G_\sigma}\left(\ddot{\boldsymbol{p}}_{\boldsymbol{G}} - \boldsymbol{g}\right) + \boldsymbol{\Phi} \boldsymbol{f}_{\mathrm{int}}, \end{aligned} \tag{11}$$

where $\boldsymbol{f}_{\mathrm{int}} \in \mathrm{R}^{3L}$ represents the internal forces and $\boldsymbol{\Phi} \in \mathrm{R}^{3L\times 3L}$ is a projector on the null-space of $\boldsymbol{J_t}$.

Once $\boldsymbol{f_c}$ is found in (11), we use the lower part of (6) and (7) (corresponding to $\boldsymbol{n_0}$) to extract an expression for the CoM motion that is equivalent to the linear pendulum. In the x-axis (and similarly in the y-axis), it is written as

$$\begin{aligned} \ddot{p}_{G,x} &= \frac{g + \ddot{p}_{G,z}}{p_{G,z} - \sum_{l=1}^{L} \sigma_{l,x} p_{l,z}} \left(p_{G,x} - \sum_{l=1}^{L} \sigma_{l,z} p_{l,x} + \frac{\eta_y}{m\left(g + \ddot{p}_{G,z}\right)} \right) \\ &= \frac{g + \ddot{p}_{G,z}}{h_G} \left(p_{G,x} - p_{cmp,x}\right), \end{aligned} \tag{12}$$

where $\boldsymbol{\eta} = -\dot{\boldsymbol{k}}_{\boldsymbol{G}} + \boldsymbol{J_t n_c}$, while h_G represents the pendulum height and $\sum_{l=1}^{L} \sigma_{l,x} p_{l,z}$, a *virtual height*. The CoM behaves like a Linear Inverted Pendulum (LIP) if $h_G > 0$. In this equation, $\boldsymbol{p_{cmp}}$ computes a position that is equivalent to the Centroidal Momentum Pivot (CMP) [19]. However, if we assume that $\dot{\boldsymbol{k}}_{\boldsymbol{G}} = \boldsymbol{0}$ then $\boldsymbol{p_{cmp}}$ is equivalent to the Zero Moment Point (ZMP), $\boldsymbol{p_{zmp}}$; that is, the multi-contact equivalent to the desired ZMP can be determined by specifying the desired force distribution and the contact moments. See [17] for more details.

2.4 *Estimation of the floating-base*

The position and orientation of the floating-base, $\boldsymbol{p_B}$ and $\boldsymbol{R_B}$ (included in $\boldsymbol{q}$), are not directly measured. They must be estimated. To estimate the position of the floating-base we use the approach of [17]. We compute it by merging the difference between estimated and desired positions of the contact links according to the desired force distribution:

$$\hat{\boldsymbol{p}}_{\boldsymbol{B}} = \boldsymbol{p}_{\boldsymbol{B}}^{\boldsymbol{d}} + \sum_{l=1}^{L} \boldsymbol{\sigma_l} \left(\boldsymbol{R}_{\boldsymbol{B}}^{\boldsymbol{d}}\, {}^{\boldsymbol{B}}\boldsymbol{p}_{\boldsymbol{l}}^{\boldsymbol{d}} - \hat{\boldsymbol{R}}_{\boldsymbol{B}}\, {}^{\boldsymbol{B}}\hat{\boldsymbol{p}}_{\boldsymbol{l}} \right), \tag{13}$$

where ${}^{\boldsymbol{B}}\boldsymbol{p_l}$ is the position of the l-th contact link in the frame of the floating-base, $\{B\}$. We use the superscript d to indicate the desired value, and $\hat{\cdot}$ to indicate an estimated one. Note that ${}^{\boldsymbol{B}}\hat{\boldsymbol{p}}_{\boldsymbol{l}}$ is computed using forward-kinematics on the current joint values.

The estimated orientation of the base is calculated by using the tilt observer reported in [20]. This one estimates roll and pitch angles only. As the yaw angle is not observable, we merge the desired yaw with the estimated roll and pitch to calculate $\hat{\boldsymbol{R}}_{\boldsymbol{B}}$.

3. Multi-contact motion control

3.1 *Motion Solver*

A humanoid robot can simultaneously achieve multiple objectives (tasks) while satisfying kinematic and dynamic constraints. To solve conflicts among these tasks while fulfilling the constraints, we employ a Quadratic Programming (QP) solver. This solver computes an optimal reference configuration acceleration, $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$, and a feasible reference of external forces $\boldsymbol{u_{e,r}}$ parametrized by $\boldsymbol{\rho_r}$ (see (5)) subject to linear equality, linear inequality, and bounding constraints. The problem is stated as

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} \\ \boldsymbol{\rho_r} \end{bmatrix} &= \arg\min_{\boldsymbol{x}} \ \frac{1}{2} \left\| \boldsymbol{\mathcal{W}} \left(\boldsymbol{A}_{\text{ob}} \boldsymbol{x} - \boldsymbol{b}_{\text{ob}} \right) \right\|^2 + \frac{1}{2} \gamma \left\| \boldsymbol{x} \right\|^2, \\ &\text{s.t.} \quad \boldsymbol{A}_{\text{eq}} \boldsymbol{x} = \boldsymbol{b}_{\text{eq}}, \ \boldsymbol{A x} \leq \boldsymbol{b}, \ \boldsymbol{l_b} \leq \boldsymbol{x} \leq \boldsymbol{u_b}, \end{aligned} \tag{14}$$

where $\boldsymbol{x}$ is the decision variable vector of the optimization problem comprising $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ and $\boldsymbol{\rho_r}$. Also, $\boldsymbol{\mathcal{W}} = \text{blkdiag}\left(\boldsymbol{\mathcal{W}_1}, \ldots, \boldsymbol{\mathcal{W}_{g,k}}\right)$ is a block diagonal matrix made up of weight matrices for k tasks [21] [9] and γ is a small weight that minimizes $\boldsymbol{x}$.

Tasks are formulated through the linear system $(\boldsymbol{A}_{\text{ob}}, \boldsymbol{b}_{\text{ob}})$, which vertically concatenates the matrices and vectors for k tasks; that is,

$$\boldsymbol{A}_{\text{ob}} = \begin{bmatrix} \boldsymbol{A}_{\text{ob},\boldsymbol{1}} \\ \vdots \\ \boldsymbol{A}_{\text{ob},\boldsymbol{k}} \end{bmatrix}, \qquad \boldsymbol{b}_{\text{ob}} \begin{bmatrix} \boldsymbol{b}_{\text{ob},\boldsymbol{1}} \\ \vdots \\ \boldsymbol{b}_{\text{ob},\boldsymbol{k}} \end{bmatrix}. \tag{15}$$

Constraints are formulated similarly, by vertically concatenating matrices and vectors. Equality constraints are formulated through $(\boldsymbol{A}_{\text{eq}}, \boldsymbol{b}_{\text{eq}})$, inequality constraints through $(\boldsymbol{A}, \boldsymbol{b})$ and boundary constraints through vectors $\boldsymbol{l_b}$ and $\boldsymbol{u_b}$.

3.2 *Tasks*

Motion-related tasks (in joint or Cartesian space) are specified with acceleration objectives, $\ddot{\boldsymbol{g}}_{\text{ob},\boldsymbol{t}}$. These are implemented with PD tracking and a feed-forward term. For example, the posture task (in joint space) is defined as $\ddot{\boldsymbol{g}}_{\text{ob},\boldsymbol{t}} = \ddot{\boldsymbol{q}}_{\boldsymbol{\theta},\text{ob}}$, while the position and orientation tasks of a link l in Cartesian space are defined as $\ddot{\boldsymbol{g}}_{\text{ob},\boldsymbol{t}} = \dot{\boldsymbol{v}}_{\boldsymbol{l},\text{ob}}$ and $\ddot{\boldsymbol{g}}_{\text{ob},\boldsymbol{t}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{l},\text{ob}}$, such that

$$\ddot{\boldsymbol{q}}_{\boldsymbol{\theta},\text{ob}} = \boldsymbol{K_p}\left(\boldsymbol{q}_{\boldsymbol{\theta}}^{\boldsymbol{d}} - \boldsymbol{q}_{\boldsymbol{\theta}}\right) + \boldsymbol{K_v}\left(\dot{\boldsymbol{q}}_{\boldsymbol{\theta}}^{\boldsymbol{d}} - \dot{\boldsymbol{q}}_{\boldsymbol{\theta}}\right) + \ddot{\boldsymbol{q}}_{\boldsymbol{\theta}}^{\boldsymbol{d}}, \tag{16}$$

$$\dot{\boldsymbol{v}}_{\boldsymbol{l},\text{ob}} = \boldsymbol{K_p}\left(\boldsymbol{p}_{\boldsymbol{l}}^{\boldsymbol{d}} - \boldsymbol{p}_{\boldsymbol{l}}\right) + \boldsymbol{K_v}\left(\boldsymbol{v}_{\boldsymbol{l}}^{\boldsymbol{d}} - \boldsymbol{v}_{\boldsymbol{l}}\right) + \dot{\boldsymbol{v}}_{\boldsymbol{l}}^{\boldsymbol{d}}, \tag{17}$$

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{l},\text{ob}} = \boldsymbol{K_p}\tilde{\boldsymbol{\Omega}} + \boldsymbol{K_v}\left(\boldsymbol{\omega}_{\boldsymbol{l}}^{\boldsymbol{d}} - \boldsymbol{\omega}_{l}\right) + \dot{\boldsymbol{\omega}}_{\boldsymbol{l}}^{\boldsymbol{d}}, \tag{18}$$

where $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$ are diagonal matrices of PD gains and $\tilde{\boldsymbol{\Omega}} = \boldsymbol{S^{-1}}(\log\left\{\boldsymbol{R}_{\boldsymbol{l}}^{\boldsymbol{d}}\boldsymbol{R}_{\boldsymbol{l}}^{\boldsymbol{T}}\right\})$ calculates the error vector in orientation. The super-script d stands for desired values, terms without subscript indicate current values and $\boldsymbol{S^{-1}}(\cdot) : \mathrm{R}^{3\times 3} \to \mathrm{R}^3$ is the inverse of the skew-symmetric operator.

Then, for task t, $\boldsymbol{A}_{\text{ob},\boldsymbol{t}}$ and $\boldsymbol{b}_{\text{ob},\boldsymbol{t}}$ are given by

$$\boldsymbol{A}_{\text{ob},\boldsymbol{t}} = \boldsymbol{J}_{\boldsymbol{g},\boldsymbol{t}}(\boldsymbol{q}), \qquad \boldsymbol{b}_{\text{ob},\boldsymbol{t}} = \ddot{\boldsymbol{g}}_{\text{ob},\boldsymbol{t}} - \dot{\boldsymbol{J}}_{\boldsymbol{g},\boldsymbol{t}}(\boldsymbol{q},\boldsymbol{\alpha})\boldsymbol{\alpha}, \tag{19}$$

where $\boldsymbol{J}_{\boldsymbol{g},\boldsymbol{t}}(\boldsymbol{q})$ and $\dot{\boldsymbol{J}}_{\boldsymbol{g},\boldsymbol{t}}(\boldsymbol{q},\boldsymbol{\alpha})$ are the task Jacobian and its time derivative.

3.3 *Constraints*

Here we give a summary of the considered constraints. For more details, see [15].

3.3.1 Under-actuation / torque constraint

The under-actuation constraint ensures the generation of a feasible motion for the floating-base. The torque constraint ensures that the required torques are within the limitations of the actuators (minimum and maximum torques: $\underline{\boldsymbol{\tau}}$ and $\overline{\boldsymbol{\tau}}$).

Let us consider (2) and (5), as well as the decision variable $\boldsymbol{x}$ in (14). Then, both constraints can be written as

$$\begin{bmatrix}\boldsymbol{M_B} & -\boldsymbol{D_B}\end{bmatrix}\boldsymbol{x} = -\boldsymbol{C_B}\boldsymbol{\alpha} - \boldsymbol{g_B}, \tag{20}$$

$$\underline{\boldsymbol{\tau}} - \boldsymbol{C_j}\boldsymbol{\alpha} - \boldsymbol{g_j} \leq \begin{bmatrix}\boldsymbol{M_j} & -\boldsymbol{D_j}\end{bmatrix}\boldsymbol{x} \leq \overline{\boldsymbol{\tau}} - \boldsymbol{C_j}\boldsymbol{\alpha} - \boldsymbol{g_j}, \tag{21}$$

where the subscript B stands for the first 6 rows of $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{D}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\boldsymbol{\alpha})$ and $\boldsymbol{g}(\boldsymbol{q})$, while the subscript j stands for the remaining rows.

3.3.2 Contact unilaterality / friction constraint

It is used to ensure that the external forces are repelling and contained inside of the friction pyramid. It is implemented by using the following bounding constraint: $\boldsymbol{\rho_r} \geq \boldsymbol{0}$ (see Section 2.2).

3.3.3 Joint limits constraints

Joint range and speed limits are implemented as inequality constraints, as done in [22].

3.3.4 Self-collision constraint

It implements collision avoidance between relevant links. It is based on the method proposed in [23] and implemented as in [22].

3.4 *Low-Level Torque Control*

The reference acceleration $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ and the reference of external forces $\boldsymbol{u}_{\boldsymbol{e},\boldsymbol{r}}$ (parameterized by $\boldsymbol{\rho}_{\boldsymbol{r}}$) produced by the QP feed an inverse dynamics-like torque control scheme based on the introduction of integral gains. Here we use the Lyapunov-stable control proposed in [15] and improve it by including an anti-windup correction. We first give a summary of the control scheme and how to use it for under-actuated systems. Then, we explain the anti-windup correction method.

3.4.1 Passivity-based Integral Term

An inverse dynamics control seeks a control law $\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{r}}$ (reference torque) for (2) given by

$$\boldsymbol{u}_{\boldsymbol{r}} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\alpha})\boldsymbol{\alpha} + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{u}_{\boldsymbol{e},\boldsymbol{r}}. \tag{22}$$

Let us add an integral term to (22) to get the torque control law $\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{p}}$ with

$$\boldsymbol{u}_{\boldsymbol{p}} = \boldsymbol{u}_{\boldsymbol{r}} + \boldsymbol{L}\boldsymbol{s} \tag{23}$$

where $\boldsymbol{L} \in \mathrm{R}^{n\times n}$ is an integral gain, $\boldsymbol{s} = \boldsymbol{\alpha}_{\boldsymbol{r}} - \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}_{\boldsymbol{r}}(t) = \int_{t_0}^{t} \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}(\iota)\, d\iota$.

A passivity-based controller is achieved by choosing

$$\boldsymbol{L} = \boldsymbol{C}(\boldsymbol{q},\boldsymbol{\alpha}) + \boldsymbol{K}, \tag{24}$$

where $\boldsymbol{K} \in \mathrm{R}^{n\times n}$ is any positive definite matrix ($\boldsymbol{K} > 0$).

The control law (23) with (24) achieves exponential stability for $\boldsymbol{s}$ and $\dot{\boldsymbol{s}}$. For the proof see [15].

The gain matrix can be chosen as time-varying $\boldsymbol{K} = \lambda\boldsymbol{M}$ or $\boldsymbol{K} = \lambda\mathrm{diag}(\boldsymbol{M})$, where $\boldsymbol{M}$ is the mass matrix, $\mathrm{diag}(\cdot)$ returns a diagonal matrix of the diagonal elements of the input matrix and $\lambda > 0$. This gives a weighting factor related to the inertia driven by each joint.

3.4.2 Under-Actuated Systems

Let us consider the first 6 rows of (23) (indicated by the subscript B):

$$\boldsymbol{u}_{\boldsymbol{p},\boldsymbol{B}} = \boldsymbol{u}_{\boldsymbol{r},\boldsymbol{B}} + (\boldsymbol{C}_{\boldsymbol{B}}(\boldsymbol{q},\boldsymbol{\alpha}) + \boldsymbol{K}_{\boldsymbol{B}})\,\boldsymbol{s}. \tag{25}$$

We know that $\boldsymbol{u}_{\boldsymbol{r},\boldsymbol{B}} = \boldsymbol{0}_{\boldsymbol{6}}$ and that $\boldsymbol{u}_{\boldsymbol{p},\boldsymbol{B}} = \boldsymbol{0}_{\boldsymbol{6}}$ must hold, as wrenches cannot be directly exerted on the floating-base. Also, to achieve exponential stability $\boldsymbol{K} > 0$ must hold. However, there is no $\boldsymbol{K} > 0$ such that $(\boldsymbol{C}_{\boldsymbol{B}}(\boldsymbol{q},\boldsymbol{\alpha}) + \boldsymbol{K}_{\boldsymbol{B}})\,\boldsymbol{s} = \boldsymbol{0}_{\boldsymbol{6}}$. See the proof in [15].

To solve this problem, we modify the under-actuation constraint of (20) by including the integral term as

$$\begin{bmatrix} \boldsymbol{M}_{\boldsymbol{B}} & -\boldsymbol{D}_{\boldsymbol{B}} \end{bmatrix} \boldsymbol{x} = -\boldsymbol{C}_{\boldsymbol{B}}\boldsymbol{\alpha} - \boldsymbol{g}_{\boldsymbol{B}} - \boldsymbol{L}_{\boldsymbol{B}}\boldsymbol{s}. \tag{26}$$

This modification artificially produces $\boldsymbol{u}_{\boldsymbol{r},\boldsymbol{B}} = -\boldsymbol{L}_{\boldsymbol{B}}\boldsymbol{s}$ and $\boldsymbol{u}_{\boldsymbol{p},\boldsymbol{B}} = \boldsymbol{0}_{\boldsymbol{6}}$. Figure 3 shows the proposed control framework. With this formulation, we have two guarantees. The first one is that the convergence proof is still valid while including the under-actuated degrees of freedom. The second one is that all the feasibility constraints that are embedded in the QP will always be respected. However, there can be physical constraints that are not considered in the QP. These constraints can produce a windup of the integral term that leads to the eventual failure of the control. Therefore, we implement the following anti-windup solution.

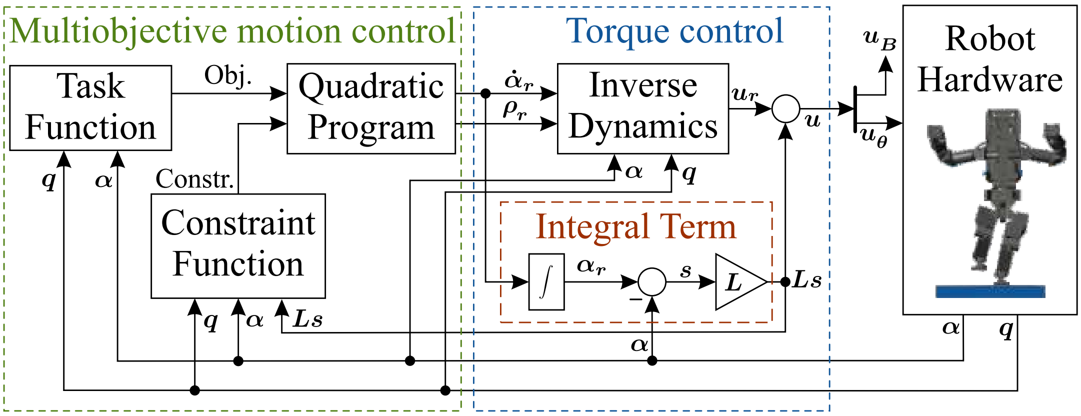


Figure 3. Framework architecture [15].

3.4.3 *Anti-Windup Solution*

The integral term added to the control law in (23) can have a side effect. If the difference between the reference and current configuration cannot be compensated for then the velocity error $\boldsymbol{s}$ will keep growing without bounds. This happens, for example, when there is a non-considered contact with the environment that prevents a link to move in the desired direction.

A growing integral term increases the applied torque until reaching the torque limits. Since there is a constraint on the torque limits, the applied torque does not increase anymore. Instead, the effect gets shifted to the floating-base due to the modification described in Section 3.4.2. This causes the QP to find a feasible but unbalanced solution leading to failure.

In the literature, there are several methods to avoid this problem, caused by the *windup* effect. Some can be found in [24]. However, they are not consistent with the proof mentioned in Section 3.4.1. What we needed is a method that preserves the positive definiteness of the integral gain. This can be achieved by scaling the integral term to keep it at a reasonable value.

Let us construct a vector of maximum generalized forces that can be safely generated by the integral term:

$$\bar{\boldsymbol{u}} = \left[(m\dot{\boldsymbol{v}}_{\boldsymbol{B},\max})^T \; (\mathcal{I}\dot{\boldsymbol{\omega}}_{\boldsymbol{B},\max})^T \; p \cdot \bar{\boldsymbol{\tau}}^T \right]^T . \tag{27}$$

Where, m is the mass of the robot, $\mathcal{I}$ is its tensor of inertia, $\dot{\boldsymbol{v}}_{\boldsymbol{B},\max}$ and $\dot{\boldsymbol{\omega}}_{\boldsymbol{B},\max}$ are the maximum allowed linear and angular accelerations of the floating-base, $\bar{\boldsymbol{\tau}}$ holds the joint torque limits and $p \in (0, 1]$ indicates how much percentage of that maximum torque can be allowed to the integral term. Then, we compute a scaling factor ϵ,

$$\epsilon = \max_{i \in 1,\ldots,n} |u_{c,i}/\bar{u}_i| , \tag{28}$$

where $\boldsymbol{u_c} = [u_{c,i}] = \boldsymbol{Ls}$ is the current integral term.

If $\epsilon > 1$ then the anti-windup correction is applied with

$$\boldsymbol{u_c} \leftarrow \boldsymbol{u_c}/\epsilon \tag{29}$$

This solution reduces the integral term on the whole vector, but it corresponds temporarily to having a small scaling factor on the gain without changing its positive definiteness. Note that this will cause non-smooth but continuous torque, which is sufficient for electrically-actuated robots.

4. QP-Based Stabilization Framework

In our previous works [15] [16] we defined a task for the CoM in the same way as in (17). This approach can behave robustly in case of discrepancies regarding the geometric model of the environment due to the admittance control introduced in [16]. However, its robustness is limited when considering larger disturbances. In that case, it is necessary to consider the implementation of a proper balance controller.

In what follows, we assume that the desired ZMP trajectory has been calculated from the sequence of contacts with the environment, the corresponding force distribution and a desired local Center-of-Pressure (CoP) at each contact surface. We also assume that a proper 3D trajectory of the CoM has been calculated to realize the desired ZMP by using the method described in [18].

4.1 *Balance Control*

In this Section, a balance controller is implemented as a set of tasks within the multi-contact motion control framework of Section 3. This balance controller is based on the concept of the Divergent Component of Motion (DCM), $\boldsymbol{\xi}$ [25] [26], defined as

$$\boldsymbol{\xi} = \boldsymbol{p_G} + \frac{\dot{\boldsymbol{p}}_{\boldsymbol{G}}}{\omega}, \tag{30}$$

where

$$\omega \equiv \sqrt{\frac{g + \ddot{p}_{G,z}}{p_{G,z} - p_{zmp,z}}}. \tag{31}$$

Here, $\boldsymbol{p_G}$ is the CoM, whereas ω is the natural frequency of the equivalent LIP dynamics of (12).

We use a PID controller applied to the DCM error, as in [26]. This one calculates a required *modification* of the ZMP based on the difference between the desired DCM (calculated from the desired trajectory of the CoM) and the estimated DCM (calculated from the observed state). In the x-axis (and similarly in the y-axis), this modification is given by

$$p^{mod}_{zmp,x} = \kappa_1 \int \left(\xi^d_x - \xi_x\right) dt + \kappa_2 \left(\xi^d_x - \xi_x\right) + \kappa_3 \left(\dot{\xi}^d_x - \dot{\xi}_x\right), \tag{32}$$

where κ_1, κ_2 and κ_3 are feedback gains calculated by pole assignment with

$$\begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{bmatrix} = \frac{-1}{\omega g_p} \begin{bmatrix} \alpha\beta\gamma \\ \alpha\beta + \beta\gamma + \gamma\alpha + \omega g_p \\ \alpha + \beta + \gamma + \omega - g_p \end{bmatrix}. \tag{33}$$

Here, α, β and γ are the poles of the system, whereas g_p represents the reciprocal of a time constant resulting from representing the ZMP dynamics as a first-order delay system.

The calculated modification is added to the desired ZMP to obtain the *reference* ZMP:

$$\boldsymbol{p^{ref}_{zmp}} = \boldsymbol{p^{d}_{zmp}} + \boldsymbol{p^{mod}_{zmp}}. \tag{34}$$

In the following, we define two tasks that use this *reference* ZMP to control the actual ZMP and the CoM.

4.1.1 ZMP Task

To realize the reference ZMP we design a task acting in the space of the wrenches. This task aims to minimize the moment of all the reaction forces $\boldsymbol{f_{l,k}}$ with respect to this reference ZMP.

First, we develop an expression to calculate the moment with respect to a point ($\boldsymbol{p^{ref}_{zmp}}$), by using (7) and (4), as

$$\begin{aligned}
\boldsymbol{n_{zmp}} &= \boldsymbol{n_0} - \boldsymbol{p^{ref}_{zmp}} \times \boldsymbol{f_0} \\
&= \sum_{l=1}^{L} \left(\boldsymbol{S}(\boldsymbol{p_l})\, \boldsymbol{f_l} + \boldsymbol{n_l} - \boldsymbol{S}\left(\boldsymbol{p^{ref}_{zmp}}\right) \boldsymbol{f_l} \right) = \sum_{l=1}^{L} \left(\begin{bmatrix} \boldsymbol{S}(\boldsymbol{r_{l/zmp}}) & \boldsymbol{I_3} \end{bmatrix} \begin{bmatrix} \boldsymbol{f_l} \\ \boldsymbol{n_l} \end{bmatrix} \right) \\
&= \begin{bmatrix} \boldsymbol{S}(\boldsymbol{r_{1/zmp}})\, \boldsymbol{I_3} \cdots \boldsymbol{S}(\boldsymbol{r_{L/zmp}})\, \boldsymbol{I_3} \end{bmatrix} \begin{bmatrix} \boldsymbol{W_1} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{W_L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho_1} \\ \vdots \\ \boldsymbol{\rho_L} \end{bmatrix} \\
&= \boldsymbol{W_{zmp}} \boldsymbol{\rho}.
\end{aligned} \tag{35}$$

Here, $\boldsymbol{r_{l/zmp}}$ represents the relative position vector from the ZMP to each anchor point $\boldsymbol{p_l}$.

Then, we just build a task in the space of the wrenches to minimize $\frac{1}{2}\left\|\boldsymbol{W_{zmp}}\boldsymbol{\rho_r} - \boldsymbol{0_3}\right\|^2$ by setting

$$\boldsymbol{A}_{\mathrm{ob},\boldsymbol{t}} = \boldsymbol{W_{zmp}}, \qquad \boldsymbol{b}_{\mathrm{ob},\boldsymbol{t}} = \boldsymbol{0_3}. \tag{36}$$

4.1.2 ZMP-based CoM Task

The motion of the CoM is not independent of the ZMP as seen in (12). The LIP dynamics are used to generate its desired motion.

If the reference ZMP was equal to the desired one, it would suffice to track the desired CoM as in (17). However, if the ZMP gets modified through $\boldsymbol{p^{mod}_{zmp}}$, it is necessary to reflect this modification on the CoM.

We specify the corresponding task by defining a "natural" acceleration objective of the CoM following the LIP dynamics, as

$$\dot{\boldsymbol{v}}_{\boldsymbol{G},\mathrm{ob}} = \begin{bmatrix} \frac{g+\dot{v}^d_{G,z}}{p^d_{G,z}-p^{ref}_{zmp,z}} \left(p^d_{G,x} - p^{ref}_{zmp,x} \right) \\ \frac{g+\dot{v}^d_{G,z}}{p^d_{G,z}-p^{ref}_{zmp,z}} \left(p^d_{G,y} - p^{ref}_{zmp,y} \right) \\ \dot{v}_{G,z} \end{bmatrix}. \tag{37}$$

Notice that $\boldsymbol{p^{ref}_{zmp}}$ is a function not only of the desired ZMP but also of the desired and actual motion of the CoM (through the DCM definition).

We define the task in the space of the accelerations as written in (19), by setting

$$\boldsymbol{A}_{\mathrm{ob},\boldsymbol{t}} = \boldsymbol{J_G}(\boldsymbol{q}), \qquad \boldsymbol{b}_{\mathrm{ob},\boldsymbol{t}} = \dot{\boldsymbol{v}}_{\boldsymbol{G},\mathrm{ob}} - \dot{\boldsymbol{J}}_{\boldsymbol{G}}(\boldsymbol{q},\boldsymbol{\alpha})\boldsymbol{\alpha}, \tag{38}$$

where $\boldsymbol{J_G}$ is the CoM Jacobian and $\dot{\boldsymbol{J}}_{\boldsymbol{G}}$, its time derivative.

4.2 *Force Distribution*

The ZMP Task does not provide enough conditions in the space of the wrenches to have a unique solution for the external forces. The QP may not be well-conditioned despite the presence of the small weight γ that minimizes $\boldsymbol{\rho_r}$, as seen in (14), especially when having several contacts. In such a case, there are several solutions capable to achieve a reference ZMP.

The problem is that the QP can shift from one solution to another between two consecutive iterations. To avoid this problem, it is a good practice to provide enough conditions leading to a unique solution. Here, we do that by providing two extra tasks in the space of the wrenches: a *Force Distribution Task* and a *Local CoP Task* per link in contact.

4.2.1 Force Distribution Task

Let us substitute $\boldsymbol{f_0}$ of (7) in (11) by omitting the internal forces to get $\boldsymbol{f_c} = \boldsymbol{G_\sigma J_t f_c}$, then

$$\begin{gathered}
\left(\boldsymbol{I_{3L}} - \boldsymbol{G_\sigma J_t}\right)\boldsymbol{f_c} = \boldsymbol{0_{3L}} \\
\left(\boldsymbol{I_{3L}} - \boldsymbol{G_\sigma J_t}\right)\begin{bmatrix} \boldsymbol{W_{1,f}} & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{W_{L,f}} \end{bmatrix}\begin{bmatrix} \boldsymbol{\rho_1} \\ \vdots \\ \boldsymbol{\rho_L} \end{bmatrix} = \boldsymbol{0_{3L}} \\
\left(\boldsymbol{I_{3L}} - \boldsymbol{G_\sigma J_t}\right)\boldsymbol{W_f \rho} = \boldsymbol{0_{3L}},
\end{gathered} \tag{39}$$

where $\boldsymbol{W_{l,f}}$ is as defined in (4). Then, we just build a task in the space of the wrenches to minimize $\frac{1}{2}\left\|\left(\boldsymbol{I_{3L}} - \boldsymbol{G_\sigma J_t}\right)\boldsymbol{W_f \rho_r} - \boldsymbol{0_{3L}}\right\|^2$ by setting

$$\boldsymbol{A}_{\text{ob},\boldsymbol{t}} = \left(\boldsymbol{I_{3L}} - \boldsymbol{G_\sigma J_t}\right)\boldsymbol{W_f}, \qquad \boldsymbol{b}_{\text{ob},\boldsymbol{t}} = \boldsymbol{0_{3L}}. \tag{40}$$

4.2.2 Local CoP Task

Let us consider $\boldsymbol{W_{l,n}}$ as defined in (4) and assume that the anchor point $\boldsymbol{p_l}$ corresponds to the desired local CoP at the contacting surface of link l. Then, we just build a task in the space of the wrenches to minimize the moment around that anchor point; that is, to minimize $\frac{1}{2}\left\|\boldsymbol{W_{l,n}\rho_{l,r}} - \boldsymbol{0_3}\right\|^2$ or $\frac{1}{2}\left\|\left[\boldsymbol{0}_{\text{ini}}\ \boldsymbol{W_{l,n}}\ \boldsymbol{0}_{\text{fin}}\right]\boldsymbol{\rho_r} - \boldsymbol{0_3}\right\|^2$ by setting

$$\boldsymbol{A}_{\text{ob},\boldsymbol{t}} = \left[\boldsymbol{0}_{\text{ini}}\ \boldsymbol{W_{l,n}}\ \boldsymbol{0}_{\text{fin}}\right], \qquad \boldsymbol{b}_{\text{ob},\boldsymbol{t}} = \boldsymbol{0_3}. \tag{41}$$

4.3 *Admittance Control*

The QP-based approach calculates a set of feasible and optimal contact forces and moments (wrenches), parametrized by $\boldsymbol{\rho_r}$. These forces can be realized if the reference acceleration $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ is adequately tracked provided a perfect model of the environment.

In practice, the pose of the floating-base is estimated. Therefore, there will always be a non-negligible discrepancy related to the assumed pose of the contact surfaces. Consequently, the actual forces and moments will differ from the calculated ones. To ensure the application of a calculated contact wrench we apply task-space force control.

With the QP it is straightforward to specify tasks in Cartesian space. Because of that, it is straightforward to implement a position-based force control within the QP framework. This control aims to indirectly regulate the wrench on a link by modifying its position reference.

In [16] we proposed the following force control:

$$\dot{\boldsymbol{V}}_{l,\text{ob}} = \begin{bmatrix} \dot{\boldsymbol{v}}_{l,\text{ob}} \\ \dot{\boldsymbol{\omega}}_{l,\text{ob}} \end{bmatrix} = -\boldsymbol{K_p}\left(\boldsymbol{F_{l,r}} - \boldsymbol{F_l}\right) - \boldsymbol{K_v}\left(\dot{\boldsymbol{F}}_{\boldsymbol{l,r}} - \dot{\boldsymbol{F}}_{\boldsymbol{l}}\right). \tag{42}$$

Here, $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$ are diagonal matrices of positive PD gains. $\dot{\boldsymbol{v}}_{l,\text{ob}}$ and $\dot{\boldsymbol{\omega}}_{l,\text{ob}}$ are linear and angular acceleration objectives of the end-effector. $\boldsymbol{F_{l,r}}$ and $\boldsymbol{F_l}$ are the *reference* and *actual* wrenches. $\dot{\boldsymbol{F}}_{\boldsymbol{l,r}}$ and $\dot{\boldsymbol{F}}_{\boldsymbol{l}}$ stand for their time derivatives (calculated by finite differences and filtered). We calculate $\boldsymbol{F_{l,r}}$ from the output of the QP during the previous iteration. This introduces one time-step delay (z^{-1}). In [16] we show the implementation details as well as the convergence properties of this control to a constant reference in the case of visco-elastic contact. However, this formulation cannot to distinguish the desired force distribution from the internal forces.

In this paper, we improve the force control law by exploiting these internal forces. We project them into the null space of the contact Jacobian related to the force distribution. Let us first introduce the idea of the null-space projection by referring to the bipedal locomotion. It is known that controlling the actual vertical force while walking leads to a drift of the CoM due to measurement and modeling errors [17]. For this reason, in [27] the difference of vertical forces is used instead to mitigate this effect. The idea behind of this control is to assume that the difference is caused by internal forces, which can be projected into the null space of the force distribution, such that the latter one remains unaffected. To extend this idea to multi-contact locomotion we use the projector $\mathbf{\Phi}$ shown in (11). Although there are several ways to define it, we use the following closed-form expression:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi_1} \\ \vdots \\ \mathbf{\Phi_L} \end{bmatrix} = \boldsymbol{I_{3L}} - \boldsymbol{G_\sigma J_t} = \boldsymbol{I_{3L}} - \begin{bmatrix} \boldsymbol{\sigma_1} & \cdots & \boldsymbol{\sigma_1} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\sigma_L} & \cdots & \boldsymbol{\sigma_L} \end{bmatrix}. \tag{43}$$

First, let us define

$$\boldsymbol{f_{l,\text{proj}}} = \mathbf{\Phi_l} \left(\boldsymbol{f_{c,r}} - \boldsymbol{f_c} \right), \tag{44}$$

where $\boldsymbol{f_{c,r}}$ and $\boldsymbol{f_c}$ are sets of reference and measured contact forces, respectively, arranged as shown in (8). Once $\boldsymbol{f_{l,\text{proj}}}$ is defined, we modify (42) as

$$\dot{\boldsymbol{V}}^{\text{null}}_{\boldsymbol{l},\text{ob}} = -\boldsymbol{K_p} \begin{bmatrix} \boldsymbol{f_{l,\text{proj}}} \\ \boldsymbol{n_{l,r}} - \boldsymbol{n_l} \end{bmatrix} - \boldsymbol{K_v} \begin{pmatrix} \dot{\boldsymbol{f}}_{\boldsymbol{l,r}} - \dot{\boldsymbol{f}}_{\boldsymbol{l}} \\ \dot{\boldsymbol{n}}_{\boldsymbol{l,r}} - \dot{\boldsymbol{n}}_{\boldsymbol{l}} \end{pmatrix}. \tag{45}$$

Then, we define the *Null-Space Admittance Task* in the space of the accelerations as written in (19), by setting

$$\boldsymbol{A}_{\text{ob},\boldsymbol{t}} = \boldsymbol{J_G}(\boldsymbol{q}), \qquad \boldsymbol{b}_{\text{ob},\boldsymbol{t}} = \dot{\boldsymbol{V}}^{\text{null}}_{\boldsymbol{G},\text{ob}} - \dot{\boldsymbol{J}}_{\boldsymbol{G}}(\boldsymbol{q},\boldsymbol{\alpha})\boldsymbol{\alpha}, \tag{46}$$

4.4 *Centroidal Angular Momentum*

As mentioned in Section 2.3, we assume that the centroidal angular momentum rate is zero ($\dot{\boldsymbol{k}}_{\boldsymbol{G}} = \mathbf{0}$) for $\boldsymbol{p_{cmp}} = \boldsymbol{p_{zmp}}$ to hold and generate its desired trajectory. So, it is necessary to minimize the centroidal angular momentum, within the QP framework, by defining a task.

The relationship between the centroidal momenta rate of change (with respect to the CoM), $\dot{\boldsymbol{h}}_{\boldsymbol{c}}$, and the configuration acceleration is given by

$$\dot{\boldsymbol{h}}_{\boldsymbol{c}} = \begin{bmatrix} \dot{\boldsymbol{l}} \\ \dot{\boldsymbol{k}}_{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A_{c,v}} \\ \boldsymbol{A_{c,\omega}} \end{bmatrix} \dot{\boldsymbol{\alpha}} + \begin{bmatrix} \dot{\boldsymbol{A}}_{\boldsymbol{c,v}} \\ \dot{\boldsymbol{A}}_{\boldsymbol{c,\omega}} \end{bmatrix} \boldsymbol{\alpha}, \tag{47}$$

where $\boldsymbol{A_c}(\boldsymbol{q}) \in \mathrm{R}^{6\times(n+6)}$ is the centroidal momentum matrix [28] [29] and $\dot{\boldsymbol{A}}_{\boldsymbol{c}}(\boldsymbol{q},\boldsymbol{\alpha}) \in \mathrm{R}^{6\times(n+6)}$, its time derivative. The subscripts v and ω indicate the linear and angular components of the matrices, respectively.

A Centroidal Angular Momentum Task is specified in the space of the accelerations using a centroidal angular momentum objective, $\dot{\boldsymbol{k}}_{\boldsymbol{c,ob}} = \boldsymbol{K}\left(\boldsymbol{k_c^d} - \boldsymbol{k_c}\right)$, such that

$$\boldsymbol{A_{c,\omega}}\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} = \boldsymbol{K}\left(\boldsymbol{k_c^d} - \boldsymbol{k_c}\right) - \dot{\boldsymbol{A}}_{\boldsymbol{c,\omega}}\boldsymbol{\alpha}. \tag{48}$$

Table 1. Friction parameters used for every joint.

Parameter	Value
Static Friction Coefficient (τ_S)	5 Nm
Kinetic Friction Coefficient (τ_K)	0.3 Nm
Viscous Friction Coefficient (υ)	4 Nm / (rad/s)
Breakaway Friction Velocity (ω_S)	0.05 rad/s

We use $\boldsymbol{k_c^d} = \mathbf{0_3}$. Then,

$$\boldsymbol{A}_{\text{ob},\boldsymbol{t}} = \boldsymbol{A_{c,\omega}}, \qquad \boldsymbol{b}_{\text{ob},\boldsymbol{t}} = -\boldsymbol{K k_c} - \dot{\boldsymbol{A}}_{\boldsymbol{c,\omega}} \boldsymbol{\alpha}. \tag{49}$$

5. Simulation Results

To assess the performance and robustness of our multi-contact stabilization framework, we performed some simulations in Choreonoid dynamics simulator[1] [30]. We used HRP-5P, which is a 37-dof humanoid robot with a height of 1.83 m and a weight of 101 kg [31]. The links of this robot are rigid but we consider in the simulation a rubber bush at the feet. This introduces flexibility that modifies the configuration of the robot, leading to a geometric error.

The motor drivers of HRP-5P accept current commands which are proportional to motor torque but do not compensate for frictions. We simulate the joint friction by using a discontinuous friction model, which includes kinetic, static, and viscous friction. The friction model considered for each joint is

$$\tau_f = (\tau_S - \tau_K) \exp\left(-\left|\frac{\omega}{\omega_S}\right|\right) \operatorname{sgn}(\omega) + \tau_K \operatorname{sgn}(\omega) + \upsilon\omega, \tag{50}$$

where τ_f is the friction torque applied to the joint, ω is the joint velocity, τ_S is the static friction, τ_K is the kinetic friction, υ is the viscous friction coefficient, ω_S is the breakaway friction velocity and $\operatorname{sgn}(x)$ is the sign function.

The friction parameters were chosen based on realistic values estimated from the torque plots of the real robot's right wrist joint[2], (see Table 1) and applied equally to all the joints. The simulation of this friction was performed using the method described in [33] which is free of chattering and unbounded drift.

The PD gains, weights, and masks, if any, that were used to configure the tasks of the QP considered for the simulations are shown in Table 2. Notice that for the Null-Space Admittance Tasks the PD gains of the force and moment components (`f-` and `n-`) are different. The constraints being considered were (a) under-actuation, (b) torque limits, (c) contact unilaterality / friction, (d) joint range and speed limits, and (e) self-collision. Furthermore, we applied the anti-windup solution described in Section 3.4.3 with parameters $p = 0.1$, $\dot{\boldsymbol{v}}_{\boldsymbol{B},\max} = (0.5, 0.5, 10)$ m/s^2 and $\dot{\boldsymbol{\omega}}_{\boldsymbol{B},\max} = (5, 5, 5)$ deg/s^2.

Two kinds of simulations were performed, each one under the effect of different types of external perturbations: (a) bipedal locomotion and (b) multi-contact locomotion.

5.1 *Bipedal locomotion*

We designed a bipedal locomotion pattern of 11 steps assuming a flat floor and no obstacles. The step length was set to 0.25 m and a toe-lift heel-strike rotation profile was used for the swinging foot. The step cycle was 1.05 s (0.7 s in single support and 0.35 s in double support).

[1]Choreonoid, available at `http://www.choreonoid.org`.
[2]These values can be found of the same order as the ones reported in joint friction identification-related papers like [32].

Table 2. Task description, code and parameters used for the biped and multicontact locomotion.

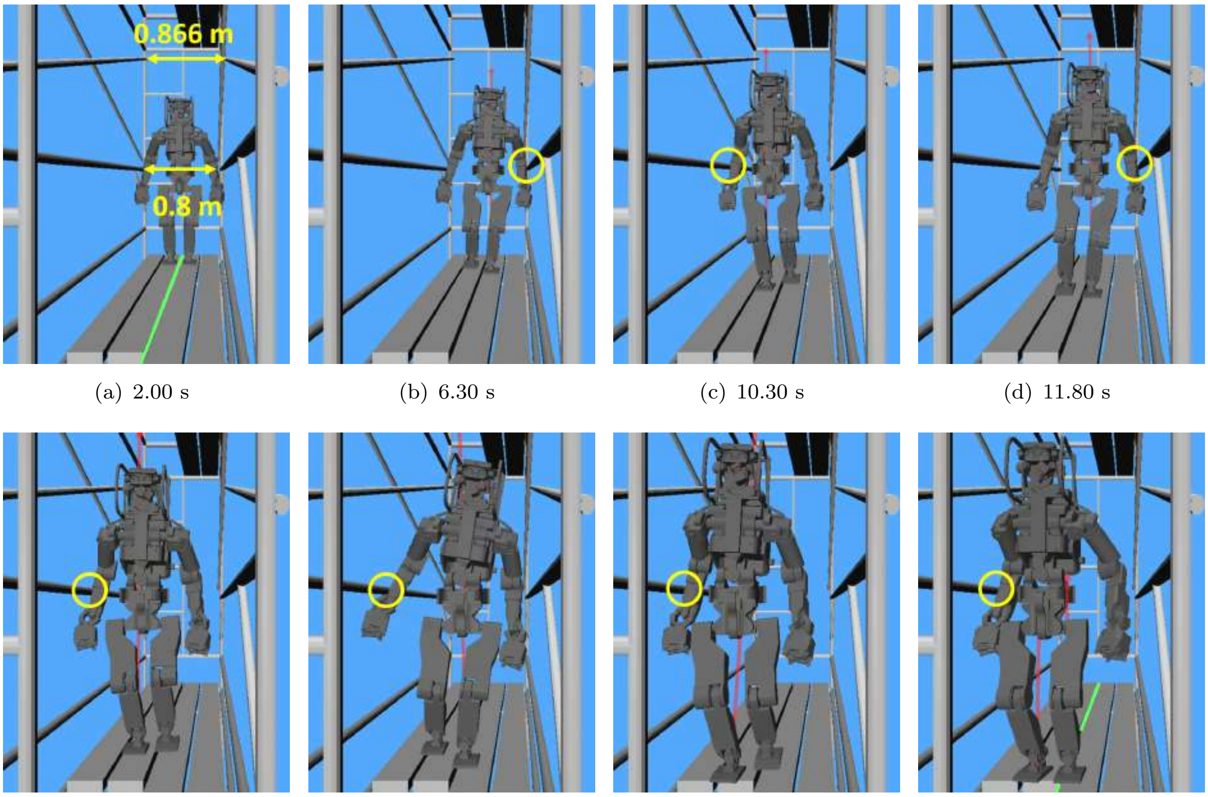
Task description	Code	Parameter	Bipedal	Multi-contact
posture (joint configuration)	`q`	PD weight	800, 200 10	800, 200 10
pose of floating Base	`poseB`	PD weight mask	500, 100 4000 0 0 1 1 1 1	500, 100 10000 0 0 1 1 1 1
orientation of Chest	`rotCh`	PD weight mask	100, 100 1000 1 1 1	500, 100 1000 1 0.1 1
pose of Right and Left Feet	`poseRF` `poseLF`	PD weight	500, 50 3000	1000,100 10000
pose of Right and Left Hands	`poseRH` `poseLH`	PD weight	40, 10 1000	1000, 100 10000
zmp-based com task	`com`	weight mask	1e10 1 1 0	1e10 1 1 0
zmp task	`zmp`	weight mask	50 1 1 0.1	50 1 1 0.5
force distribution	`fdist`	weight	1	10
centroidal angular Momentum	`angMom`	gain weight mask	1 50 1 1 5	1 50 1 1 1
local cop of Right and Left Feet	`lcopRF` `lcopLF`	cop weight	0 0 0 10	0 0 0 10
local cop of Right and Left Hands	`lcopRH` `lcopLH`	cop weight	0 0 0 10	0 0 0 10
null-space admittance of Right and Left Feet	`admiRF` `admiLF`	f-PD n-PD weight	1e-3, 2e-4 5e-2, 2e-4 3000	5e-3, 1e-4 5e-2, 1e-4 10000
null-space admittance of Right and Left Hands	`admiRH` `admiLH`	f-PD n-PD weight	5e-3, 1e-7 1e-2, 1e-6 2000	5e-2, 1e-4 5e-1, 1e-4 10000

The force distribution ratio for each foot at the beginning and the end of the motion the robot is (0.5, 0.5, 0.5). During the single support phases, the force distribution ratio of each foot alternates between (1.0, 1.0, 1.0) and (0.0, 0.0, 0.0), and during the double support phases, it is interpolated. The desired ZMP and CoM trajectories are automatically generated from this specification by using the online motion generation algorithm of [18]. This motion generator is capable to shrink or enlarge the support phases within a range of ± 0.2 s based on contact detection. The QP task parameters are described in the column *bipedal* of Table 2. Concerning the passivity-based torque control, we used a gain $\boldsymbol{K} = 2\,\mathrm{diag}\,(\boldsymbol{M})$.

Let us consider that the same biped motion is executed on three different scenarios:

(a) A flat floor without obstacles (for reference).
(b) Inside of a narrow scaffold.
(c) On rough uneven terrain, passing through the narrow entrance of a scaffold.

Some snapshots of the biped locomotion simulation inside of the scaffold can be seen in Figure 4. We can see that the robot can complete the motion without losing balance. This is in great part due to our torque control and the low gains of `poseRH` and `poseLH`, set on purpose to allow them to behave more compliantly. It is also due to the stabilizing control, which modifies the desired ZMP (and the CoM accordingly) to keep balance, even in the case of unexpected collisions. See the comparison between the ZMP and CoM trajectories for the case of the flat floor (Figure 7) and the scenario of the narrow scaffold (Figure 8). It is possible to see the effect of the unexpected collisions with the guardrail of the scaffold as spikes in the estimated ZMP



(a) 2.00 s (b) 6.30 s (c) 10.30 s (d) 11.80 s

(e) 15.80 s (f) 16.80 s (g) 18.30 s (h) 20.30 s

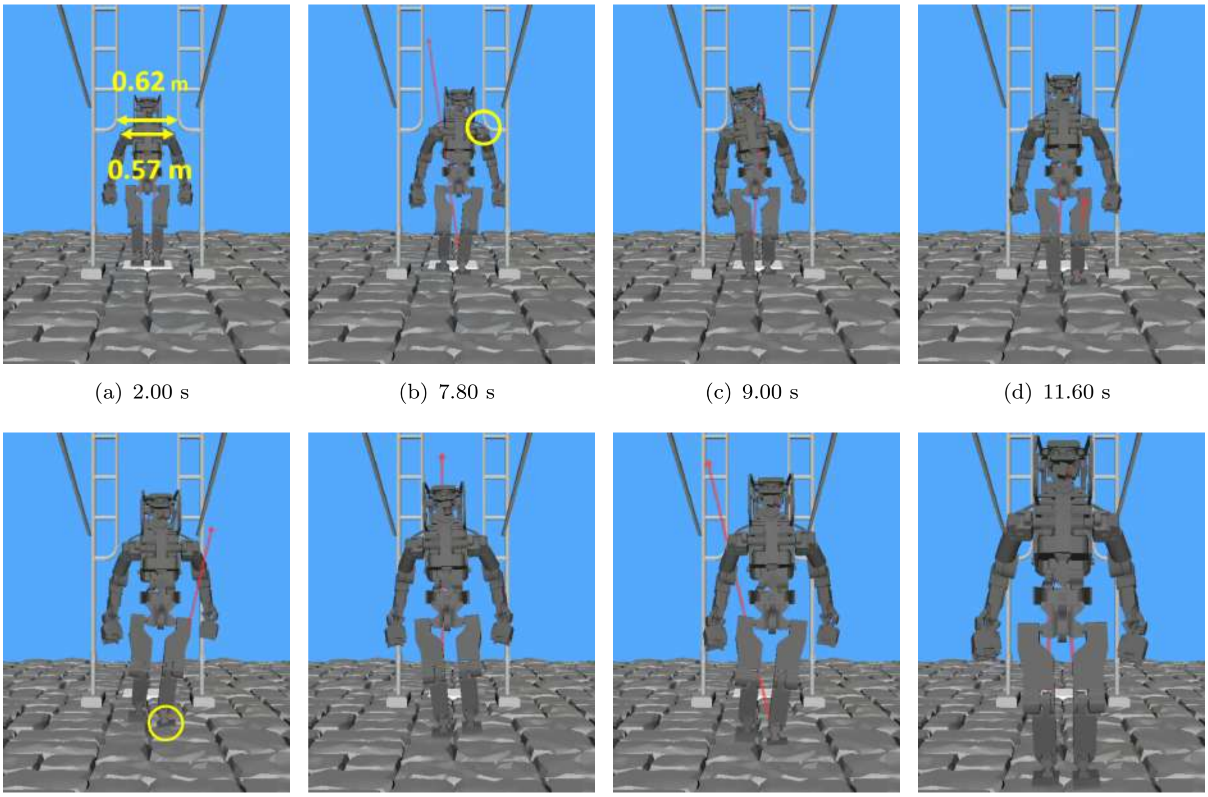
Figure 4. Bipedal locomotion inside of a narrow scaffold that is 0.866 m wide. The arms of the robot open up in such a way that the width of the robot at the height of the guardrail is 0.8 m. Unavoidably the arms hit the frame of the scaffold due to the sway of the walking motion, causing yaw deviation. The contacts are highlighted with yellow circles. The green line at the center is used to emphasize such yaw deviation.

and from the shape of the estimated CoM in the y-direction. We can also notice that the robot experiences yaw deviation due to unexpected contacts. As a result of this deviation, the robot finishes its motion with its arm in contact with the guardrail, also unexpected. This situation would result in instability if it were not for the anti-windup solution discussed in Section 3.4.3.

As for the third scenario, the corresponding snapshots can be seen in Figure 5. The corresponding ZMP and CoM trajectories are depicted in Figure 9. We can see that the robot can maintain balance despite the collision against its shoulder with the entrance of the scaffold. As the chest (and shoulders) cannot be given higher compliance (to be able to walk), the stabilizing control is the main responsible for coping with the unexpected collision. This is done by modifying the desired ZMP. The difference between the desired ZMP and the reference one due to the modification, as well as the disturbance caused by the collision to the estimated CoM can be seen in Figure 9 before $t = 8$ s.

5.2 *Multi-contact locomotion*

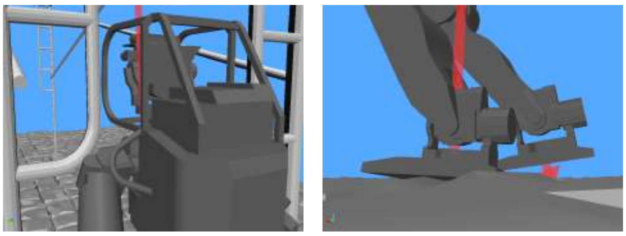
Let us consider that the robot needs to continue its locomotion inside of the scaffold. Given that the scaffold is very narrow and there is a gap in the floor, the robot needs to walk sideways and use its hands to continue its locomotion by using the guardrail as support (see Figure 1(b)). We placed a slanted plate over the guardrail at 45° to allow the hands to make full palm contact. Then, we designed a multi-contact locomotion pattern for the robot to move by following the sequence: left foot → left hand → right hand → right foot. This one is performed three times. The step cycle was 2.5 s (1.5 s in triple support and 1.0 s in quadruple support). When initially placing the hands on the slanted plate, the force distribution ratio of each foot and each hand is set to (0.5, 0.5, 0.4) and (0.0, 0.0, 0.1), respectively. After that, the force distribution ratios are



(a) 2.00 s (b) 7.80 s (c) 9.00 s (d) 11.60 s

(e) 13.30 s (f) 14.80 s (g) 16.07 s (h) 22.30 s

Figure 5. Bipedal locomotion on an uneven floor passing through the entrance of a scaffold that is 0.62 m wide. The shoulders of the robot, which are prone to hit the frame, are 0.57 m wide. Unavoidably the shoulder hit the frame of the entrance of the scaffold due to the sway of the walking motion (7.8 s). The contact is highlighted with a yellow circle, as well as one moment (13.3 s) when the foot experienced vibration due to edge contact. See Figure 6 for another point of view of these highlighted events.



(a) Collision with the frame (b) Edge contact

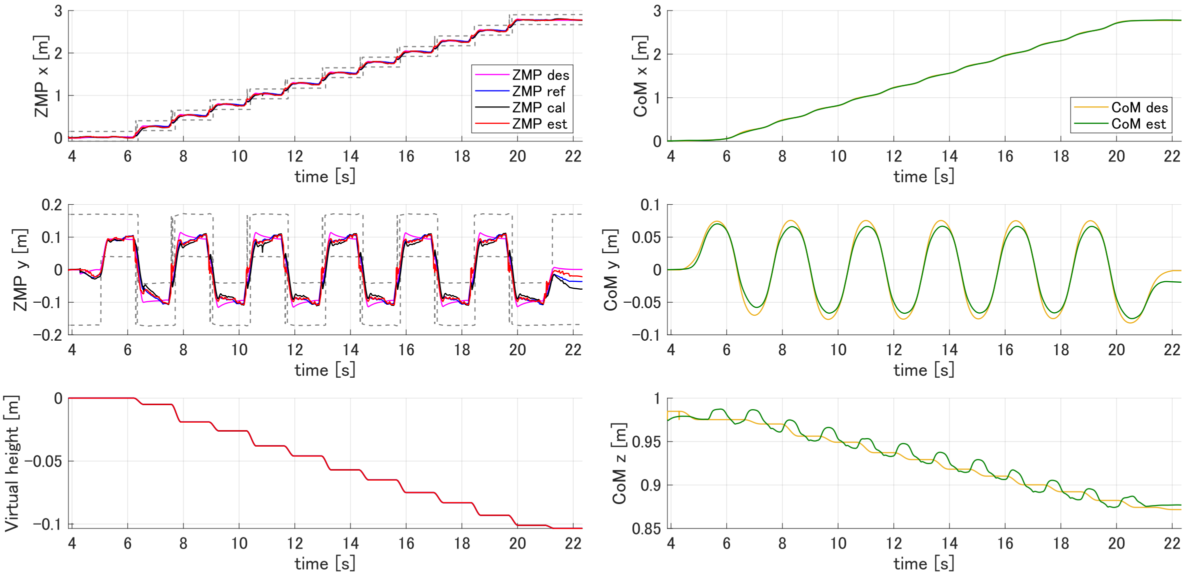
Figure 6. Two events arising during the simulation on the uneven floor: (a) a collision of the shoulder against the frame of the scaffold, and (b) the state of the foot that leads to its vibration (edge contact).

set as shown in Table 3 during the triple support phases and interpolated during the quadruple support phases. Notice that we ask the tangential force to be generated mainly by the feet, but the reference tangential force to be tracked is eventually decided by the QP.

The trajectories of the desired ZMP and the desired CoM were generated based on the force distribution ratios. Figure 10 illustrates how the stable region of the CoM changes during the

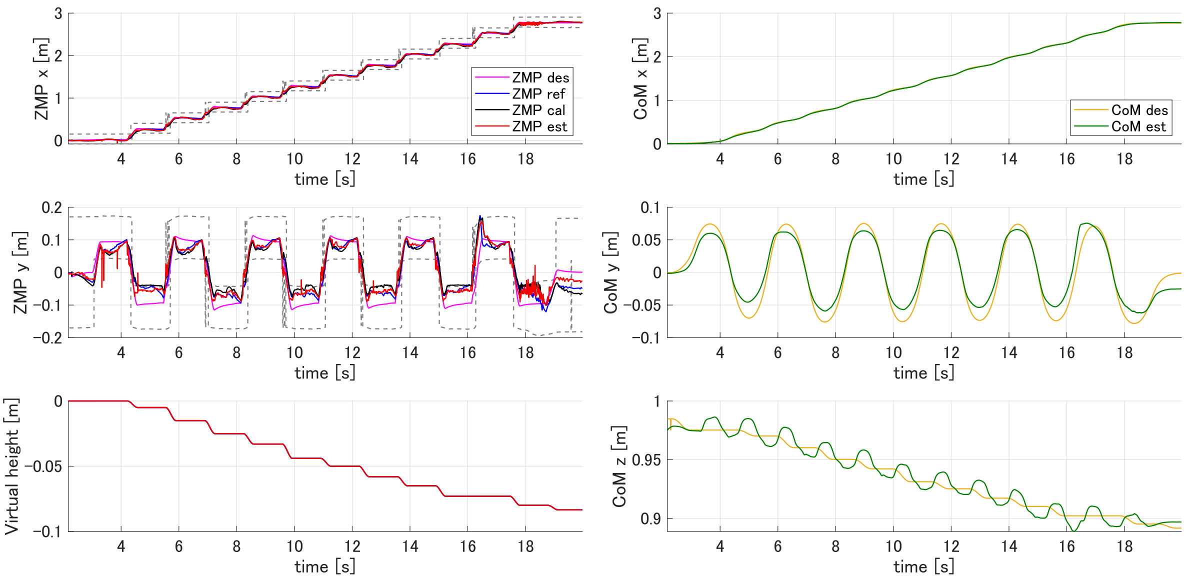
Table 3. Force distribution ratios used for the triple support phases.

Support	$\boldsymbol{\sigma_{RF}}$	$\boldsymbol{\sigma_{LF}}$	$\boldsymbol{\sigma_{RH}}$	$\boldsymbol{\sigma_{LH}}$
RF-RH-LH	(1.0, 1.0, 0.7)	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.15)	(0.0, 0.0, 0.15)
RF-LF-RH	(0.5, 0.5, 0.35)	(0.5, 0.5, 0.35)	(0.0, 0.0, 0.3)	(0.0, 0.0, 0.0)
RF-LF-LH	(0.5, 0.5, 0.35)	(0.5, 0.5, 0.35)	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.3)
LF-RH-LH	(0.0, 0.0, 0.0)	(1.0, 1.0, 0.7)	(0.0, 0.0, 0.15)	(0.0, 0.0, 0.15)



(a) ZMP (b) CoM

Figure 7. The behavior of the ZMP and the CoM on a flat floor without obstacles. The gray dotted-lines represent the polygon of support. *ZMP des* and *CoM des* are the desired values given by the motion generator. The "virtual height", as defined after (12), can be seen to vary due to the contact detection-based modification of the swing phase. *ZMP ref* is the reference one resulting from the modification of the desired value in order to keep balance. *ZMP cal* is the one calculated from the optimal reference wrenches given as an output by the QP. *ZMP est* and *CoM est* are the estimated current values, calculated from the force/torque sensors, the estimation of the floating-base, and the current joint angles.



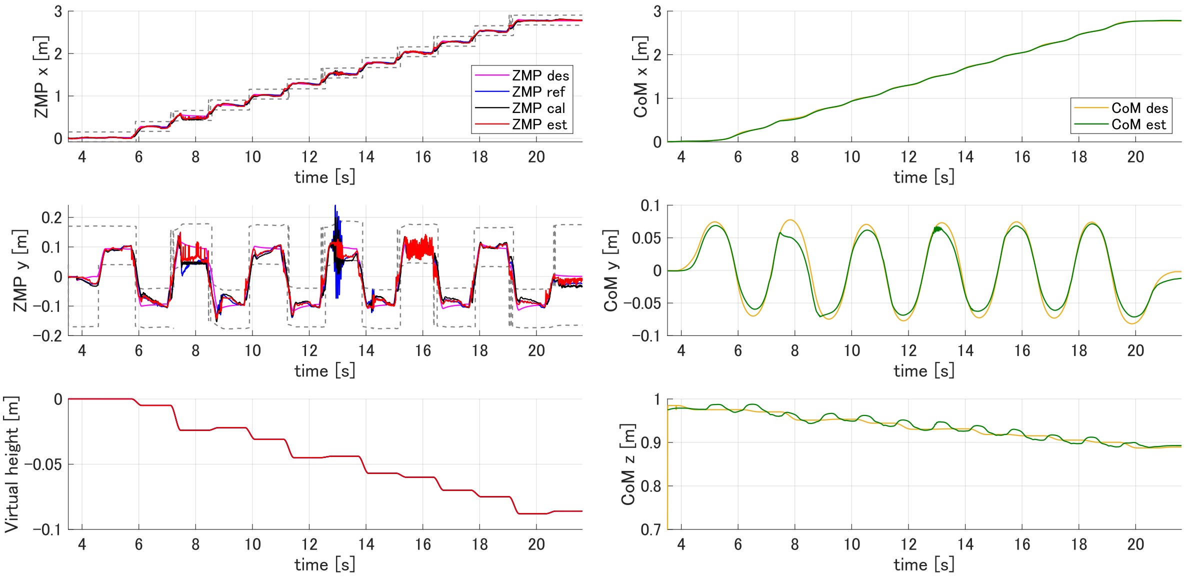
(a) ZMP (b) CoM

Figure 8. The behavior of the ZMP and the CoM within the narrow scaffold. The explanation of the signals is as in Figure 7.

initial steps. Each region was projected to a horizontal plane from the CoM, the contact position, and its friction cones, as in [34] [35]. As these are the stable regions of the static CoM we can see that the CoM was generated within a fully sufficient stable region.

The QP task parameters for the multi-contact locomotion are described in the column *Multi-contact* of Table 2. Concerning the passivity-based torque control, we used a gain $\boldsymbol{K} = 20\mathrm{diag}\,(\boldsymbol{M})$.

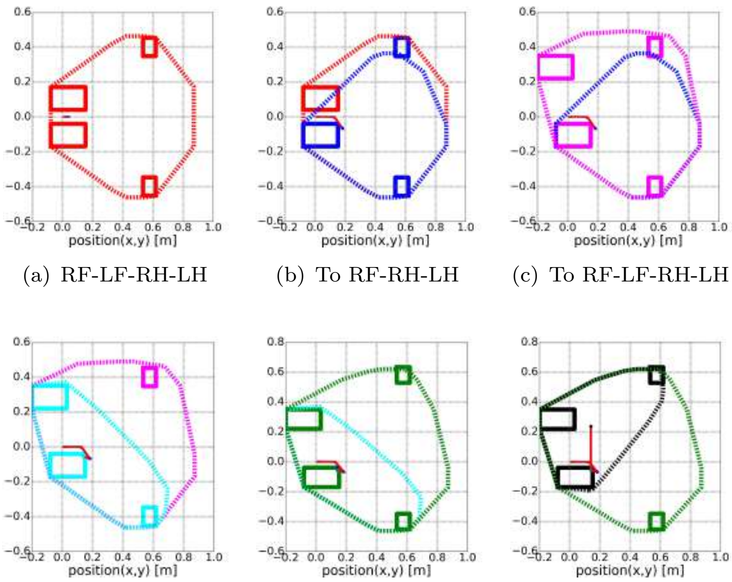
Some snapshots of the multi-contact locomotion simulation inside of the scaffold can be seen from two different perspectives in Figure 11 (rear view) and Figure 12 (right view). From the rear view we can see the sway of the body required to shift the CoM and realize the desired



(a) ZMP (b) CoM

Figure 9. The behavior of the ZMP and the CoM on the rough terrain. The explanation of the signals is as in Figure 7. It is possible to see the effect of the unexpected collisions with the frame from the y component of the estimated CoM, as well as the vibration of the foot from the plot of the ZMP caused by edge contact (see Figure 6).

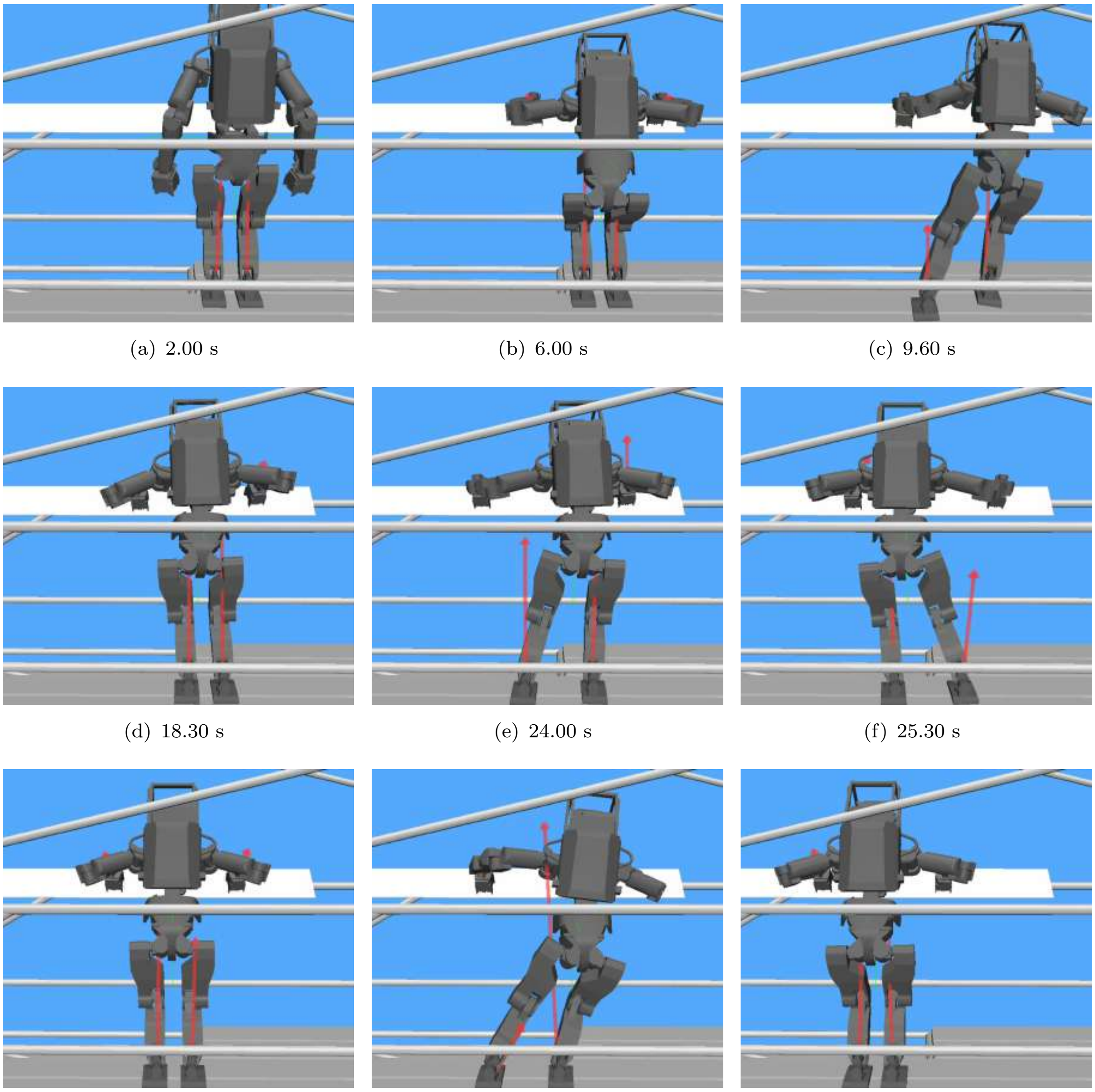
force distribution ratios. From the right view we can see that the projection of the CoM of the robot is clearly in front of the feet (from $t = 15$ s). This is possible only if the hands are used as supporting links. The same can be verified from the corresponding ZMP and CoM trajectories, which are depicted in Figure 13 together with the polygon of support of feet and hands. There, we can see that the ZMP trajectory is outside of the polygon of support of the feet (lower pair of dotted lines); that is, in front of them. Furthermore, it is also possible to observe a good tracking of the CoM in y, which is the motion direction.



(a) RF-LF-RH-LH (b) To RF-RH-LH (c) To RF-LF-RH-LH

(d) To RF-LF-RH (e) To RF-LF-RH-LH (f) To RF-LF-LH

Figure 10. Sequence of CoM Feasible Regions (CFR) [35]. The solid red and blue lines are the ZMP and the CoM respectively, which are almost overlapped. Each marker denotes the position of the ZMP and CoM at the beginning of the next support phase. (a) shows the moment in which the robot initially contacts the slanted plate (RF-LF-RH-LH). The multi-contact locomotion is realized by repeating the following transitions: switching the ZMP from RF-LF-RH-LH to RF-RH-LH (b), from RF-RH-LH to RF-LF-RH-LH (c), from RF-LF-RH-LH to RF-LF-RH (d), from RF-LF-RH to RF-LF-RH-LH (e), and from RF-LF-RH-LH to RF-LF-LH (f).



(a) 2.00 s (b) 6.00 s (c) 9.60 s

(d) 18.30 s (e) 24.00 s (f) 25.30 s

(g) 30.80 s (h) 34.80 s (i) 43.30 s

Figure 11. Multi-contact locomotion inside of a scaffold (rear view).

5.3 *Validations*

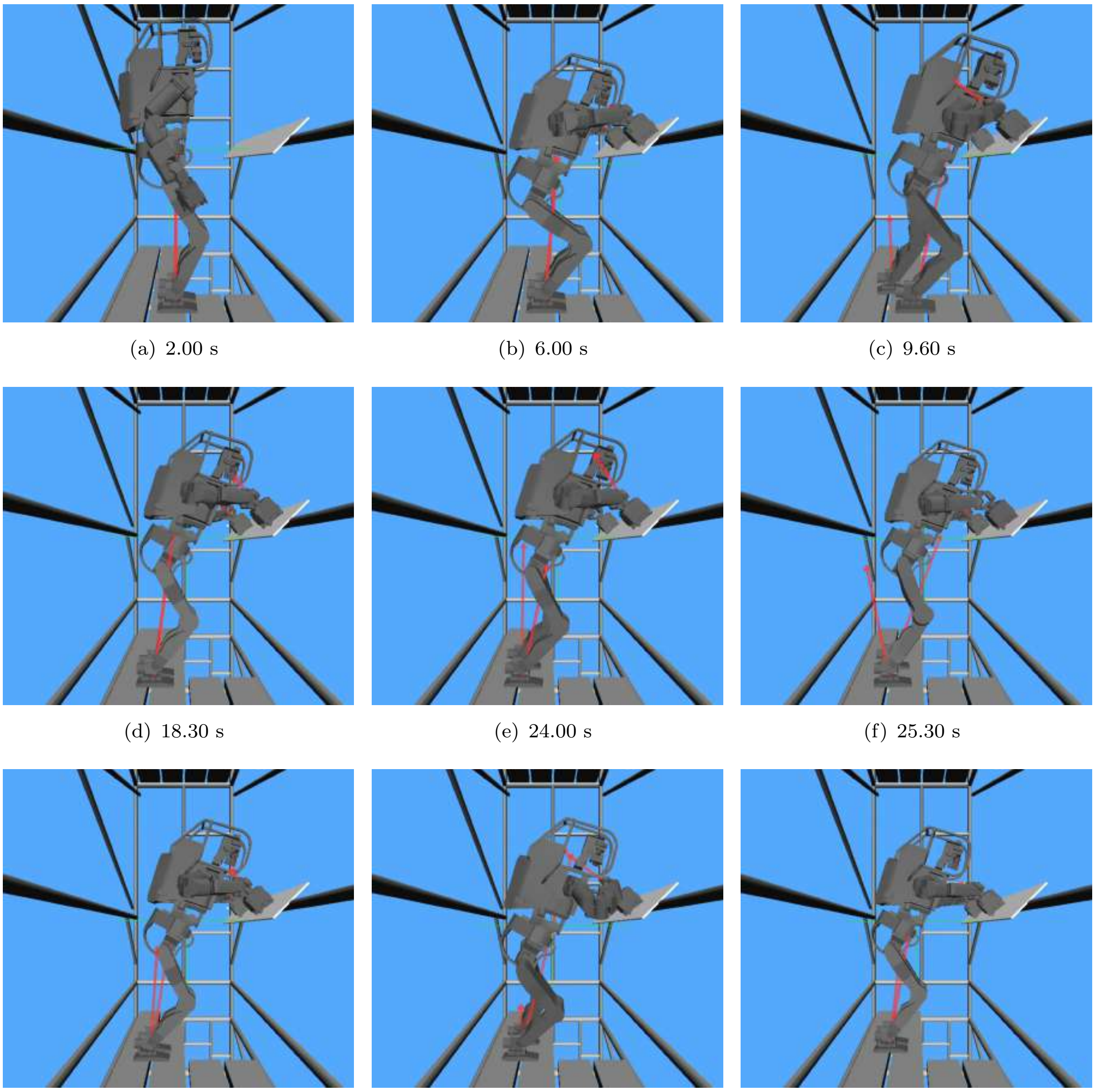
To assess the importance of some components of our framework we ran additional simulations having them deactivated. Particularly, we assessed (a) the anti-windup solution (Section 3.4.3) and (b) the null-space admittance task (Section 4.3).

5.3.1 Anti-Windup Solution

To show why the anti-windup solution is necessary we ran the four simulation cases mentioned above under three different conditions:

(a) Using the anti-windup solution; that is, as in Section 5.1 and Section 5.2.
(b) Using $\boldsymbol{K} = 0.2\,\mathrm{diag}\,(\boldsymbol{M})$ for the integral term without an anti-windup solution.
(c) Using $\boldsymbol{K} = 2.0\,\mathrm{diag}\,(\boldsymbol{M})$ for the integral term without an anti-windup solution.

In summary, without an anti-windup solution, the only case that succeeded was over a flat floor and using $\lambda = 0.2$. Even without unexpected contacts, there are errors due to floating-base estimation and joint friction which windup. For the rest of the cases, the robot became unstable.



(a) 2.00 s (b) 6.00 s (c) 9.60 s

(d) 18.30 s (e) 24.00 s (f) 25.30 s

(g) 30.80 s (h) 34.80 s (i) 43.30 s

Figure 12. Multi-contact locomotion inside of a scaffold (right view). The shown frames correspond to the ones in Figure 11.

Here, we chose only one case in which the robot did not fail early to compare plots: walking on an uneven floor. Figure 14 shows the evolution of the integral term added to the torque of all the joints of the robot. Figure 15 shows the evolution of the integral term (the virtual extra force) added to the waist of the robot. As we can see, without an anti-windup solution the integral terms are prone to grow until the torque sent reaches the joint torque limits. Due to the joint torque limits constraint of the QP, no higher torque will be sent, but the motion will be modified by the QP until it finds no stable solution, leading to instability.

This problem is mainly due to unexpected contacts (unexpected collisions or contacts happening earlier than planned). With longer unexpected contact, more windup is generated and the motion fails earlier. Without using an anti-windup solution we can only set lower gains for the integral term, at the expense of the tracking performance. Figure 16 shows a comparison of the tracking performance for the right foot during the locomotion on an uneven floor. The poor tracking in the y-direction causes a motion failure even with low gains for the integral term.

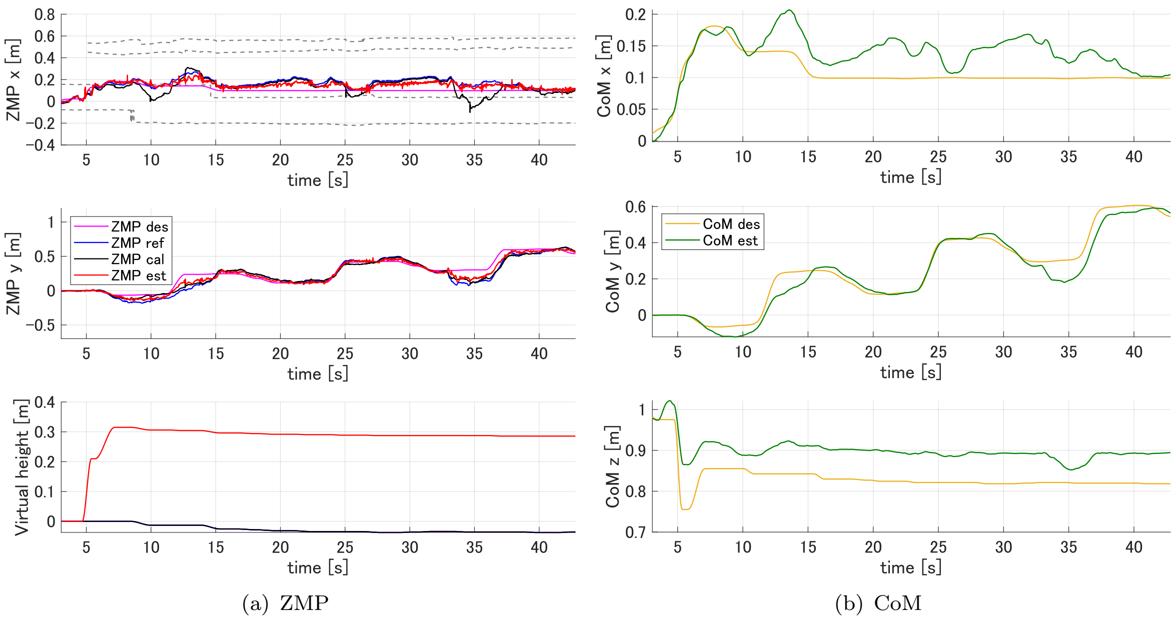


Figure 13. The behavior of the ZMP and the CoM during the multi-contact locomotion. Here, the motion is performed in the y-direction (with respect to the robot). The explanation of the signals is as in Figure 7 with the exception of the gray dotted-lines, which here represent the polygon of support of feet and hands, separately; that is why we see 4 lines in the sub-plot for the x-direction of the ZMP.

5.3.2 Null-Space Admittance Task

To show the effect of the null-space projection within the admittance task we also ran the previous four simulation cases under two different conditions:

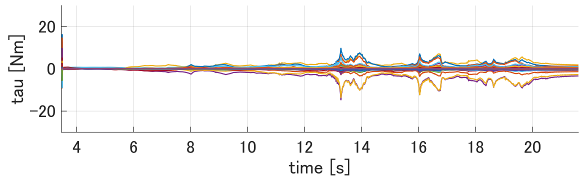
(a) Using the null-space admittance task described in (45).
(b) Using the original admittance task described in (42).

In summary, there was no noticeable difference in all the cases of bipedal locomotion. This is because the projection acts only during the double support phase, which is noticeably short. However, the outcome was different for the multi-contact locomotion where we have at least three contacts most of the time. These, in practice, tend to generate internal forces that disrupt the force distribution. The null-space approach deals with these internal forces. Figure 19 shows the evolution of the z-component of the force distribution for each end effector when using the null-space admittance task. Figure 20 shows the evolution of the z-component of the force distribution for each end effector when using the original admittance task. These plots confirm that the null-space projection helps to exert a force distribution closer to the desired one. Furthermore, this approach also improves the force tracking. Figure 17 shows the force tracking performance on the right foot and the right hand when using the null-space admittance task. Figure 18 shows the force tracking performance on the right foot and the right hand when using the original admittance task.

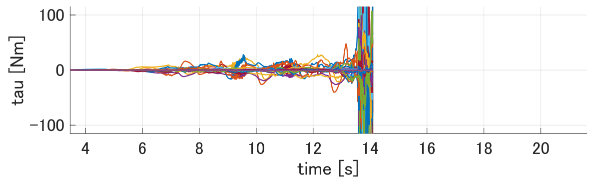
6. Conclusions and Future Work

We developed a QP-based stabilizing framework based on a force distribution formulation that inherently defines the desired ZMP trajectory. This trajectory is modified by a DCM-based balance controller and realized by manipulating the optimal force distribution. The trajectory of the CoM is then adjusted from this compensated ZMP by using the LIP model.

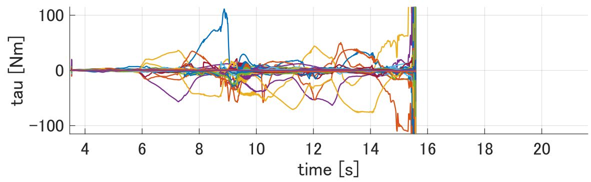
By using this controller, it was possible to achieve stable bipedal or multi-contact locomotion, at least in simulation, despite the presence of unexpected collisions, allowing the robot to evolve in narrow spaces. The robustness against unexpected contacts is partially due to the inherent



(a) Using $\lambda = 2.0$ and anti-windup solution

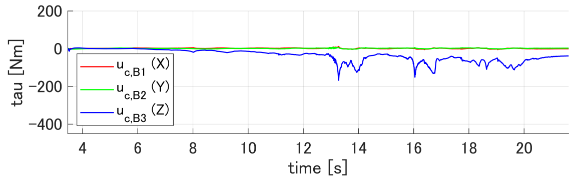


(b) Using $\lambda = 0.2$ without an anti-windup solution

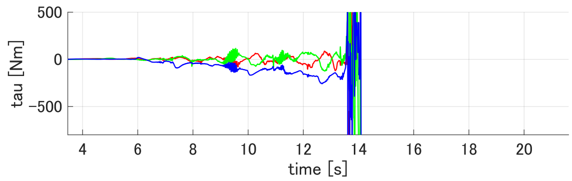


(c) Using $\lambda = 2.0$ without an anti-windup solution

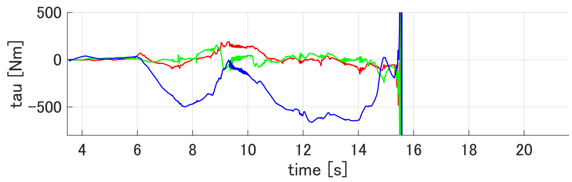
Figure 14. The integral term added to the torque of all the joints of the robot for the third scenario of the bipedal locomotion; that is, for the locomotion on the uneven floor. The legends for the signals are omitted on purpose due to the high number of joints (53). The parameter λ is used to compute the gain matrix $\boldsymbol{K} = \lambda \text{diag}\,(\boldsymbol{M})$ (see (24)). The running time of this simulation was shorter for the cases not using the anti-windup solution due to a motion failure. Notice also that the vertical scale of the plot is different for these cases.



(a) Using $\lambda = 2.0$ and anti-windup solution



(b) Using $\lambda = 0.2$ without an anti-windup solution



(c) Using $\lambda = 2.0$ without an anti-windup solution

Figure 15. The integral term added to the waist of the robot (a virtual extra force) for the third scenario of the bipedal locomotion; that is, for the locomotion on the uneven floor. The signals $u_{c,B1}$ (X), $u_{c,B2}$ (X) and $u_{c,B3}$ (Z) are the Cartesian components of the virtual extra force, corresponding to the floating-base-related elements of $\boldsymbol{u_c}$ (see (28)). The parameter λ is used as in Figure 14. Notice also that the vertical scale of the plot is different for these cases.

compliance achieved by the inverse dynamics-based control with integral gain. This control allows additional contact forces to last only for a short time and the disturbance to be eventually absorbed. Also, it can track, with exponential convergence, the dynamically feasible trajectories that were generated by the set of QP tasks implementing the balance control.

As future work, we will implement this framework on the real HRP-5P and test it on real narrow environments.

Funding

This work was partially supported by NEDO's "Development of a highly dependable humanoid robot system that can work in unstructured environments" project (15102348-0).

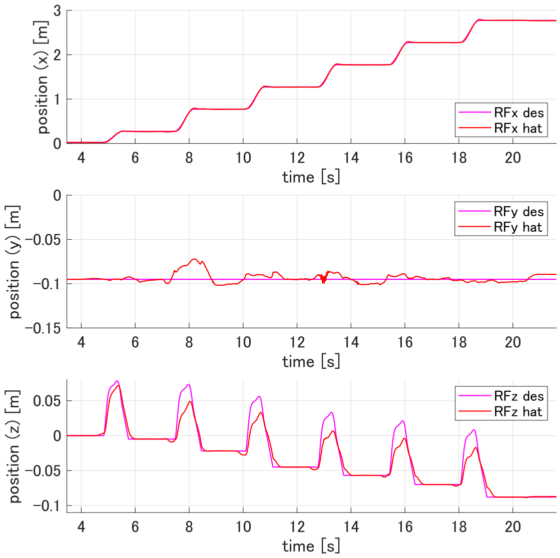
References

[1] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *The International Journal of Humanoid Robotics*, 2(4):505–518, 2005.

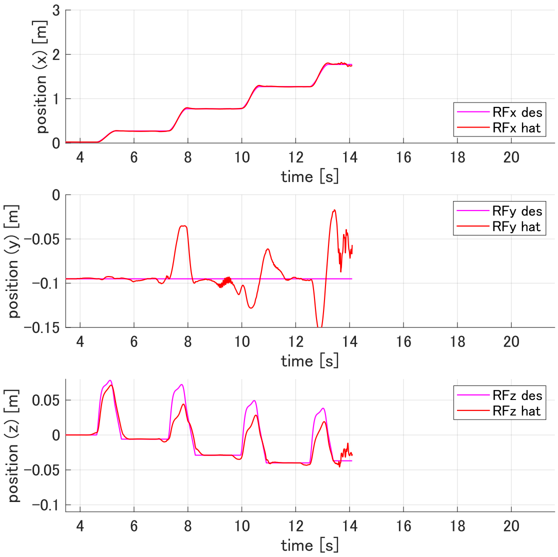
[2] L. Sentis, J. Park, and O. Khatib. Compliant control of mul-ticontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics*, 26(3):483–501, 2010.

[3] S-H. Hyon, J. Hale, and G. Cheng. Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces. *IEEE Transactions on Robotics*, 25(1):171–178, 2009.

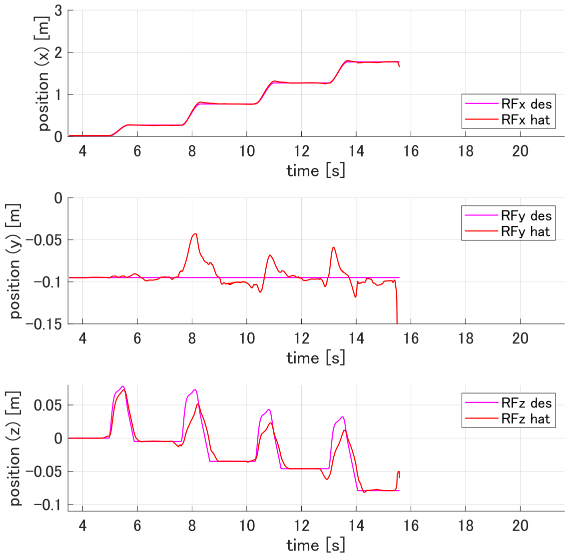
[4] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal. Optimal distribution of contact forces with inverse dynamics control. *The International Journal of Robotics Research*, 32(3):280–298, 2013.



(a) Using $\lambda = 2.0$ and anti-windup solution



(b) Using $\lambda = 0.2$ without an anti-windup solution



(c) Using $\lambda = 2.0$ without an anti-windup solution

Figure 16. Position of the right foot during the locomotion on the uneven floor. *RF des* is the desired position of the foot given by the motion generator. *RF hat* is the estimated current position of the foot, calculated from the estimation of the floating-base and the current joint angles.

[5] S. Benjamin and C. Atkeson. Dynamic balance force control for compliant humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

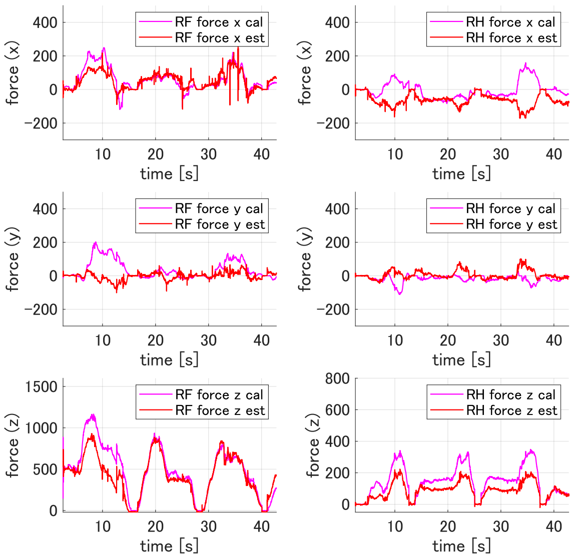
[6] B. Henze, Maximo. A. Roa, and C. Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *International Journal of Robotics Research*, 35(12):1522–1543, 2016.

[7] D. Orin, A. Goswami, and S. Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35:161–176, 2013.

[8] S. Feng, E. Whitman, X. Xinjilefu, and C. Atkeson. Optimization-based Full Body Control for the DARPA Robotics Challenge. *Journal of Field Robotics*, 32(2):293–312, 2015.

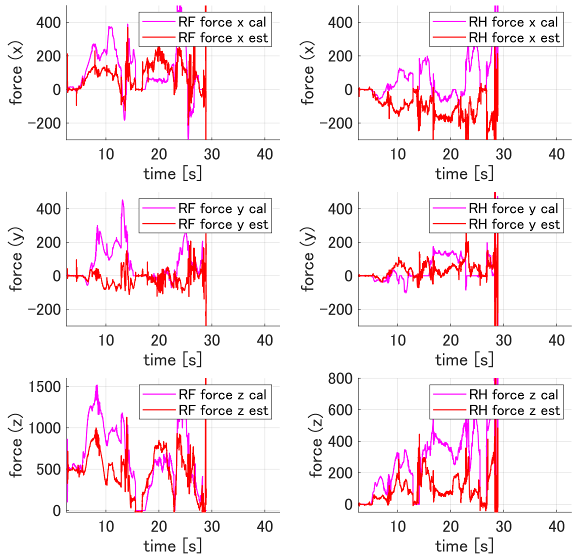
[9] M.A. Hopkins, D.W. Hong, and A. Leonessa. Compliant Locomotion Using Whole-Body Control and Divergent Component of Motion Tracking. In *IEEE Internacional Conference on Robotics and Automation (ICRA)*, 2015.

[10] T. Koolen, T. Bertrand, T. de Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt. Design of a momentum-based control framework and application to the humanoid robot Atlas. *The International Journal of Humanoid Robotics*, 13(1):1650007, 2016.



(a) Right Foot (b) Right Hand

Figure 17. Forces exerted by one foot and one hand during the multi-contact locomotion when using the null-space admittance task described in (45). *Force cal* is the reference force given as an output by the QP. *Force est* is the actual force measured by the force/torque sensor of the end effector.



(a) Right Foot (b) Right Hand

Figure 18. Forces exerted by one foot and one hand during the multi-contact locomotion when using the original admittance task described in (42). The explanation of the signals is as in Figure 17.

[11] G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer. Dynamic Walking on Compliant and Uneven Terrain using DCM and Passivity-based Whole-body Control. In *IEEE-RAS International Conference on Humanoid Robots*, 2019.

[12] T. Ko, K. Yamamoto, K. Murotani, and Y. Nakamura. Compliant Biped Locomotion of Hydra, an Electro-Hydrostatically Driven Humanoid. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2018.

[13] K. Bouyarmane et al. Quadratic Programming for Multirobot and Task-Space Force Control. *IEEE Transactions on Robotics*, 35(1), 2019.

[14] S. Caron, A. Kheddar, and O. Tempier. Stair Climbing Stabilization of the HRP-4 HumanoidRobot using Whole-body Admittance Control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[15] R. Cisneros, M. Benallegue, A. Benallegue, M. Morisawa, H. Audren, P. Gergondet, A. Escande, A. Kheddar, and F. Kanehiro. Robust humanoid control using a QP solver with integral gains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[16] R. Cisneros, M. Benallegue, M. Morisawa, and F. Kanehiro. QP-based task-space hybrid / parallel control for multi-contact motion in a torque-controlled humanoid robot. In *IEEE-RAS International Conference on Humanoid Robots*, 2019.

[17] M. Morisawa, M. Benallegue, R. Cisneros, I. Kumagai, A. Escande, K. Kaneko, and F. Kanehiro. Multi-Contact Stabilization of a Humanoid Robot for Realizing Dynamic Contact Transitions on Non-coplanar Surfaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[18] M. Morisawa, R. Cisneros, M. Benallegue, I. Kumagai, A. Escande, and F. Kanehiro. Online 3D CoM Trajectory Generation for Multi-Contact Locomotion Synchronizing Contact. In *IEEE-RAS International Conference on Humanoid Robots*, 2018.

[19] M.B. Popovic and H. Herr. Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications. *The International Journal of Robotics Research*, 24(12):1013–1032, 2005.

[20] M. Benallegue, A. Benallegue, and Y. Chitour. Tilt estimator for 3D non-rigid pendulum based on a tri-axial accelerometer and gyrometer. In *IEEE-RAS International Conference on Humanoid Robots*, 2017.

[21] K. Bouyarmane and A. Kheddar. Using a Multi-Objective Controller to Synthesize Simulated Humanoid Robot Motion with Changing Contact Configurations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[22] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro. Multi-contact vertical ladder climbing by an HRP-2 humanoid. *Autonomous Robots*, 40(3):561–580, March 2016.

[23] F. Kanehiro, M. Morisawa, W. Suleiman, K. Kaneko, and E. Yoshida. Integrating Geometric Constraints into Reactive Leg Motion Generation. In *IEEE/RSJ Internacional Conference on Intelligent Robots and Systems (IROS)*, 2010.

[24] K.J. Åström and L. Rundqwist. Integrator Windup and How to Avoid It. In *American Control Conference*, 1989.

[25] J. Englsberger, C. Ott, and A. Albu-Schäffer. Three-dimensional bipedal walking control using Divergent Component of Motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[26] M. Morisawa, K. Nobuyuki, S. Nakaoka, K. Kaneko, S. Kajita, and F. Kanehiro. Biped Locomotion Control for Uneven Terrain with Narrow Support Region. In *IEEE/SICE International Symposium on System Integration (SII)*, 2014.

[27] 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 *IEEE/RSJ Internacional Conference on Intelligent Robots and Systems (IROS)*, 2010.

[28] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved Momentum Control: Humanoid Motion Planning based on the Linear and Angular Momentum. In *IEEE/RSJ Internacional Conference on Intelligent Robots and Systems (IROS)*, 2003.

[29] D. Orin and A. Goswami. Centroidal Momentum Matrix of a Humanoid Robot Structure and Properties. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.

[30] S. Nakaoka. Choreonoid: Extensible Virtual Robot Environment Built on an Integrated GUI Framework. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 79–85, 2012.

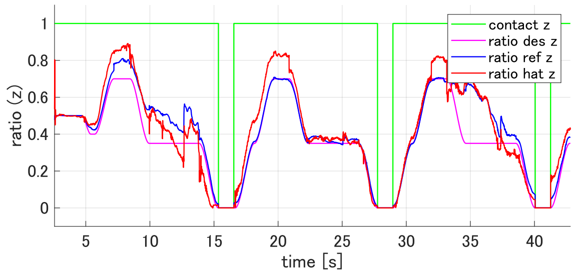
[31] K. Kaneko, H. Kaminaga, T. Sakaguchi, S. Kajita, M. Morisawa, I. Kumagai, and F. Kanehiro. Humanoid Robot HRP-5P: an Electrically Actuated Humanoid Robot with High Power and Wide Range Joints. *IEEE Robotics and Automation Letters (RA-L)*, 4(2), 2019.

[32] A. Albu-Schäffer and G. Hirzinger. Parameter identification and passivity based joint control for a 7DOF torque controlled light weight robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

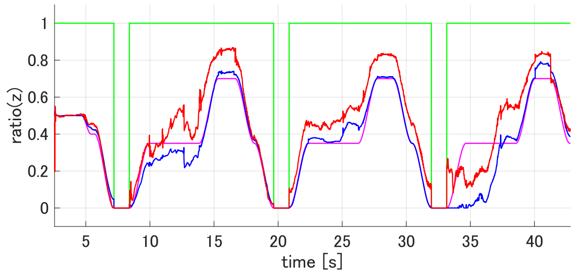
[33] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto. Admittance and Impedance Representations of Friction Based on Implicit Euler Integration. *IEEE Transactions on Robotics*, 22(6):1176–1188, 2006.

[34] S. Bretl and S. Lall. Testing static equilibrium for legged robots. *IEEE Transactions on Robotics*, 24(4), 2008.

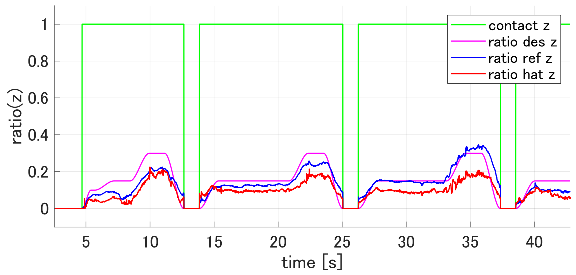
[35] S. Nozawa, M. Kanazawa, Y. Kakiuchi, K. Okada, Y. Yoshiike, and M. Inaba. Three-dimensional humanoid motion planning using COM feasible region and its application to ladder climbing tasks. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 49–56, 2016.



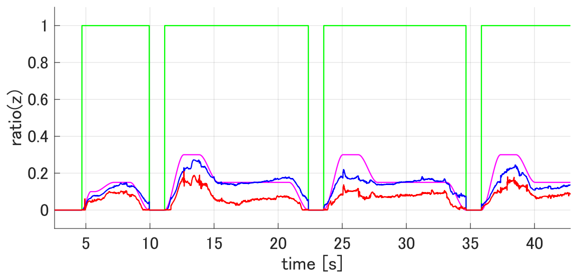
(a) Right Foot



(b) Left Foot

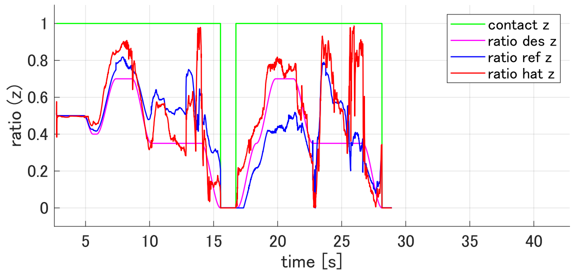


(c) Right Hand

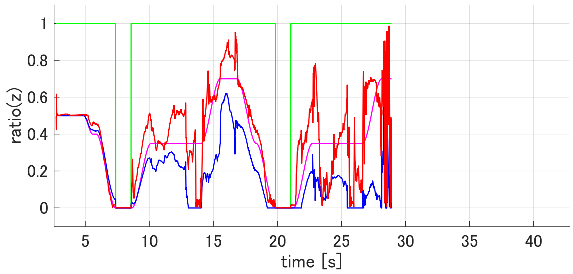


(d) Left Hand

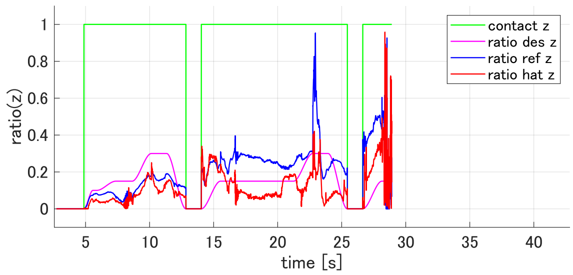
Figure 19. Force distribution ratios for each of the end effectors during the multi-contact locomotion when using the null-space admittance task described in (45). Only the z-component is shown. *Contact z* indicates if the contact for the corresponding link has been registered (1) or not (0) to the QP solver. *Ratio des z* is the desired ratio given by the motion generator. *Ratio ref z* is the ratio calculated from the reference forces given as an output by the QP. *Ratio hat z* is the ratio calculated from the actual forces measured by the force/torque sensors of the end effectors. The measurement provided by each end effector is considered for the force distribution ratio computation only if the contact is registered; otherwise it is assumed that the reading corresponds to the inertial force due to the acceleration of the link.



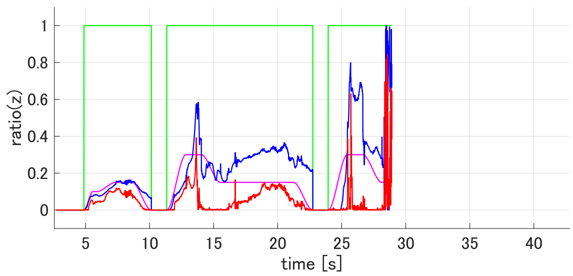
(a) Right Foot



(b) Left Foot



(c) Right Hand



(d) Left Hand

Figure 20. Force distribution ratios for each of the end effectors during the multi-contact locomotion when using the original admittance task described in (42). Only the z component is shown. The explanation of the signals is as in Figure 19. The running time of this simulation was shorter due to a motion failure in which the robot became unstable.