Robust humanoid control using a QP solver with integral gains

Rafael Cisneros, Mehdi Benallegue, Abdelaziz Benallegue, Mitsuharu Morisawa, Hervé Audren, Pierre Gergondet, Adrien Escande, Abderrahmane Kheddar, Fumio Kanehiro

Robust humanoid control using a QP solver with integral gains

Rafael Cisneros[1,*], Mehdi Benallegue[1], Abdelaziz Benallegue[2,4], Mitsuharu Morisawa[1], Hervé Audren[5], Pierre Gergondet[2], Adrien Escande[2], Abderrahmane Kheddar[2,3] and Fumio Kanehiro[1]

***Abstract*— We propose a control framework for torque control of humanoid robots for efficiently minimizing the tracking error of multi-objective weighted tasks while satisfying dynamic constraints using a Quadratic Programming (QP) solver, such that this optimal dynamically-feasible reference (considering the state of the floating base) can be robustly tracked with exponential convergence in the presence of non modeled torque bias and low-frequency bounded disturbances. This is achieved by introducing integral gains in a Lyapunov-stable torque control which exploit the passivity properties of the dynamical model of the robot, and by the proper consideration of their effect on the dynamic constraints of the QP solver. The robustness of this framework is demonstrated in simulation by commanding our robot, the HRP-5P, to achieve simultaneously several objectives in configuration and Cartesian space, in the presence of non-modeled static and kinetic joint friction, as well as an uncertain torque scale.**

***Index Terms*— Robust control, Torque control, Passivity, Quadratic programming, Humanoid robots**

I. Introduction

A humanoid robot is designed to achieve simultaneously multiple tasks, including safety in uncertain environments, eventually subject to disturbances [1]. To achieve these tasks, a motion has to be generated through joint torques, produced by actuators, often through high-ratio gearboxes. The relation between the joint torques and the second order kinematics is well modeled using only inertial and geometry allowing for a perfect control of an ideal robot. However, these parameters are usually not exactly known [2] and the transmission of the torque often creates important reactive friction forces [3]. This tends to modify the robot dynamics substantially. These parameters could be partially identified [2], [3] but the accuracy and the precision of these methods are far from allowing seamless generation of kinematic acceleration.

To avoid facing these problems, a number of controllers do not use directly these dynamical equations, but resort to stiffly position-controlled joints [4]. This is for example the case of the HRP series. Another method is to equip the robot with joint-level torque sensors, which can measure the effect of friction and compensate for them [5]. However these sensors are usually costly, heavy and very fragile. Other techniques, grounded in the field of manipulators [6] rely rather on more robust control, using for example integral gains.

[1] Humanoid Research Group, AIST, Tsukuba, Japan.
[2] CNRS-AIST JRL, UMI3218/RL, Tsukuba, Japan.
[3] CNRS-University of Montpellier, LIRMM, France.
[4] Laboratoire d'Ingénierie des Systèmes de Versailles (LISV), Université de Versailles Saint-Quentin, Versailles, France.
[5] Ascent Robotics, Tokyo, Japan.
[*] Corresponding author E-mail: rafael.cisneros@aist.go.jp

These gains, referred also as joint velocity and position feedback gains, have been adapted to humanoid robots to improve robustness of torque control [7], [8], but this integration lacks theoretical grounding, especially at whole-body level.

Indeed, beyond the guarantee of zero static error, many other interesting properties can emerge from a careful choice of the integral gain. A nice example for manipulators is to exploit the natural passivity of the robot to design a controller which provides efficient motions with strong convergence proofs and effective parameter adaptation [9], [10]. We call this scheme *passivity-based control*.

The specificity that prevents humanoid robots from taking directly profit from these results is the under-actuation. The robot is not ground-fixed to the environment, and we model that by choosing a *floating base* among the body parts, generally the waist of the humanoid. The floating base can freely rotate and translate along 6 degrees of freedom, but has no dedicated actuator to control this motion, and relies on external contact forces instead. Since these forces are not directly controlled, we cannot simply add integral terms to the under-actuated modes of the motion and lose most of the properties of the integral gains.

In general, kinematic tasks require the tracking of reference joint and floating-base accelerations. The aforementioned under-actuation leads to feasibility constraints on these accelerations to be respected. This limitation, together with contact-related constraints and torque and joint limits are today already appropriately taken into account using efficient online controllers. One popular solution is based on using QP for the generation of constraint-compliant multi-objective second order kinematics [11].

In this paper we exploit this tool to introduce the integral gains on a torque-controlled humanoid robot without joint torque feedback. The QP allows to generate the optimal gradient-based acceleration with regard to the constraints and the integral gains, in closed-loop with a feedback composed by joint encoders and floating-base kinematics. This method allows also to decouple the kinematic task stiffness which describes the desired rate of convergence of the task, from the stiffness of the torque control which allows to control the compliance.

We show in high-fidelity simulations the humanoid performing multiple concurrent tasks, including stepping, subject to modeling errors and heavy non-modeled joint friction. The simulation allows to compare the performance of a robot equipped with passivity-based integral gains compared to a robot without integral gains and a simple high-gain QP.

II. Lyapunov-Stable Torque Control

A. Robot dynamics

We consider a rigid multi-body robot composed by m holonomic joints, such that the i-th one has $k_i \leq 6$ degrees of freedom (dof) and a configuration w.r.t. the parent body described by $\boldsymbol{Q_i} \in \mathcal{Q}_i \subseteq SE(3)$. We denote the m-tuple $\boldsymbol{Q} = (\boldsymbol{Q_1}, \ldots, \boldsymbol{Q_m})$ as the *configuration* of the robot [12], [13]. Each joint has a velocity represented by $\vartheta_i \in \mathbb{R}^{k_i}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ be the *configuration velocity*, resulting from stacking all the joint velocities ϑ_i, such that the robot has $n = \sum_{i=1}^{m} k_i$ dof. Then, $\dot{\boldsymbol{\alpha}} \in \mathbb{R}^n$ is regarded as the *configuration acceleration* [14].

It is possible to derive singularity-free dynamic equations using minimal and globally valid Euclidean and non-Euclidean configuration coordinates, see [15], [12], such that the dynamical model of the robot writes

$$\boldsymbol{M}(\boldsymbol{Q})\dot{\boldsymbol{\alpha}} + \boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{\alpha} + \boldsymbol{g}(\boldsymbol{Q}) = \boldsymbol{u} + \sum \boldsymbol{J_i^T F_i}, \quad (1)$$

where $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix representing the mass matrix of the robot, $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ is a specific matrix factorization accounting for the Coriolis and centripetal effects such that $\dot{\boldsymbol{M}}(\boldsymbol{Q}, \boldsymbol{\alpha}) - 2\boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha})$ is a skew-symmetric matrix[1], $\boldsymbol{g} \in \mathbb{R}^n$ is the vector of gravitational effects, $\boldsymbol{F_i} \in \mathbb{R}^6$ the i-th external wrench (force and torque), $\boldsymbol{J_i} \in \mathbb{R}^{6 \times n}$ the absolute Jacobian of its point of application, and $\boldsymbol{u} \in \mathbb{R}^n$ is a vector of input generalized forces which includes actuated and unactuated dof (zero entries).

B. Fully actuated systems torque control

Let us first give a summary of some torque control schemes applied for fully actuated systems:

1) Inverse dynamics control: Inverse dynamics control [17] seeks a nonlinear feedback control law $\boldsymbol{u} = \boldsymbol{u_r}$ (reference torque) for (1) given by

$$\boldsymbol{u_r} = \boldsymbol{M}(\boldsymbol{Q})\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{\alpha} + \boldsymbol{g}(\boldsymbol{Q}) - \boldsymbol{u_e}, \quad (2)$$

where $\boldsymbol{u_e}$ is the vector of external torques as a result of ($\boldsymbol{u_e} = \sum \boldsymbol{J_i^T F_i}$) and $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ is the *reference configuration acceleration*. With this control on a perfect model, we achieve the ideal control $\dot{\boldsymbol{\alpha}} = \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$.

2) Integral gain torque control: Let us add an integral term to (2) to get the torque control law $\boldsymbol{u} = \boldsymbol{u_p}$ with:

$$\boldsymbol{u_p} = \boldsymbol{u_r} + \boldsymbol{L}\boldsymbol{s}, \quad (3)$$

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

Remark 1: The produced acceleration of this control is not necessarily $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ anymore but $\dot{\boldsymbol{\alpha}} = \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} + \boldsymbol{M}^{-1}(\boldsymbol{Q})\boldsymbol{L}\boldsymbol{s}$.

[1]The matrix $\boldsymbol{C}$ is not unique. The resulting generalized inertial force $\boldsymbol{C}\boldsymbol{\alpha}$ however, does not depend upon the particular choice of factorization. Furthermore, a matrix $\boldsymbol{C}$ may or not satisfy the skew-symmetric property, and if it does, the factorization is not unique. One analytical factorization that satisfies the skew-symmetric property was proposed in [16].

3) Passivity-based control: A great advantage of passivity-based controllers is that they provide certain robustness [18]. Let us consider a passivity-based controller by choosing an integral gain $\boldsymbol{L}$:

$$\boldsymbol{L} = \boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha}) + \boldsymbol{K}, \quad (4)$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is any strictly positive definite matrix; that is, $\boldsymbol{K} > 0$.

Using (4) in (3), we get the following torque control law:

$$\boldsymbol{u_p} = \boldsymbol{M}(\boldsymbol{Q})\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{\alpha_r} + \boldsymbol{g}(\boldsymbol{Q}) - \boldsymbol{u_e} + \boldsymbol{K}\boldsymbol{s}, \quad (5)$$

Proposition 1 (Passivity-based controller): The feedback control law given by $\boldsymbol{u} = \boldsymbol{u_p}$ from (5) achieves exponential stability for the velocity error $\boldsymbol{s}$.

Proof: By substituting (5) into (1), we get

$$\boldsymbol{M}(\boldsymbol{Q})\dot{\boldsymbol{s}} = -\boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{s} - \boldsymbol{K}\boldsymbol{s}. \quad (6)$$

Now, let us consider the following Lyapunov function:

$$V = \frac{1}{2}\boldsymbol{s^T M}(\boldsymbol{Q})\boldsymbol{s}. \quad (7)$$

Its time derivative is given by

$$\dot{V} = \frac{1}{2}\boldsymbol{s^T}\left(\dot{\boldsymbol{M}}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{s} + 2\boldsymbol{M}(\boldsymbol{Q})\dot{\boldsymbol{s}}\right), \quad (8)$$

which evaluated along the trajectories of (1) gives

$$\begin{aligned} \dot{V} &= \frac{1}{2}\boldsymbol{s^T}\left(\dot{\boldsymbol{M}}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{s} - 2\boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha})\boldsymbol{s} - 2\boldsymbol{K}\boldsymbol{s}\right) \\ &= -\boldsymbol{s^T K s} < 0, \end{aligned} \quad (9)$$

where we used the fact that $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is a skew-symmetric matrix, exploiting the passivity of the system.

From equation (9), we can write

$$\dot{V} \leq -2\frac{\sigma_{\min}(\boldsymbol{K})}{\sigma_{\max}(\boldsymbol{M})}V, \quad (10)$$

where $\sigma_{\min}(X)$ and $\sigma_{\max}(X)$ are the minimum and maximum eigenvalues of X, respectively. This represents a differential inequation whose solution is

$$V \leq V(0)\exp\left(-2\frac{\sigma_{\min}(\boldsymbol{K})}{\sigma_{\max}(\boldsymbol{M})}t\right), \quad (11)$$

giving the exponential convergence of the velocity error. ■

Note that given the expression of this Lyapunov function, it could be seen as the kinetic energy of the error, decaying exponentially with this controller. Furthermore, the controller is able to track the reference configuration acceleration:

Proposition 2: The control law expressed in (5) achieves exponential convergence of $\boldsymbol{s}$, and convergence of the acceleration error, $\dot{\boldsymbol{s}}$.

Proof: According to equation (6) and given that $\boldsymbol{s} \to 0$ as $t \to \infty$ exponentially, it can be concluded that $\dot{\boldsymbol{s}} \to 0$ as $t \to \infty$ since $\boldsymbol{M}(\boldsymbol{Q})$ is non-singular. ■

As seen from the proof of Proposition 1, $\boldsymbol{K}$ is associated with the exponential decay for convergence. The larger its elements are, the faster the joint trajectories will converge to the reference. This gain matrix $\boldsymbol{K}$ can be arbitrarily chosen as long as $\boldsymbol{K} > 0$. A simple solution is to choose it as an

identity matrix $\mathbf{1_n} \in \mathbb{R}^{n\times n}$ scaled by a factor λ, or in a more elegant way, according to the following remark:

Remark 2: If the gain matrix is chosen as time-varying $\boldsymbol{K} = \lambda \boldsymbol{M}$, where $\boldsymbol{M}$ is the mass matrix, then the time derivative of the Lyapunov function is given by

$$\dot{V} = -\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{K}\boldsymbol{s} = -\lambda \boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{M}\boldsymbol{s} = -2\lambda V < 0, \qquad (12)$$

which is a differential equation with exact analytical solution:

$$V = V(0)\exp(-2\lambda t). \qquad (13)$$

This choice for $\boldsymbol{K}$, proposed in this paper, not only appropriately gives a weighting factor that is related to the inertia driven by each of the generalized coordinate, but also to the coupling effect that lies between them, since the gain is non-diagonal.

C. Error dynamics in joint trajectory tracking

We show here an example on how these schemes are introduced in a closed-loop trajectory tracking and the properties of the produced error dynamics. Consider the case of a fully actuated robot in an Euclidean configuration space; that is, $\boldsymbol{Q}$ can be represented by a vector $\boldsymbol{q} \in \mathbb{R}^n$ such that $\boldsymbol{\alpha} = \dot{\boldsymbol{q}}$. Let us use the reference configuration acceleration $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ to track a desired configuration trajectory ($\boldsymbol{q_d}$, $\boldsymbol{\alpha_d}$, $\dot{\boldsymbol{\alpha}}_{\boldsymbol{d}}$) with a Proportional-Derivative (PD) compensation scheme:

$$\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} = k_p(\boldsymbol{q_d} - \boldsymbol{q}) + k_v(\boldsymbol{\alpha_d} - \boldsymbol{\alpha}) + \dot{\boldsymbol{\alpha}}_{\boldsymbol{d}} \qquad (14)$$

$$= k_p\boldsymbol{e} + k_v\dot{\boldsymbol{e}} + \dot{\boldsymbol{\alpha}}_{\boldsymbol{d}} \qquad (15)$$

where $\boldsymbol{e} = (\boldsymbol{q_d} - \boldsymbol{q})$ is the joint position error and k_p and k_v are positive scalars. Substituting (15) into (3) gives

$$\boldsymbol{u_p} = \boldsymbol{u_r} + \boldsymbol{K'_v}\dot{\boldsymbol{e}} + \boldsymbol{K'_p}\boldsymbol{e} + \boldsymbol{K'_i}\int_{t_0}^{t}\boldsymbol{e}(\iota)d\iota \qquad (16)$$

where $\boldsymbol{K'_i} = k_p\boldsymbol{L}$ and $\boldsymbol{K'_p} = (k_p\boldsymbol{M} + k_v\boldsymbol{L})$ whereas $\boldsymbol{K'_v} = (k_v\boldsymbol{M} + \boldsymbol{L})$; that is, it introduces an integral configuration term, while simultaneously increasing the effective gains of the proportional and derivative terms.

More specifically, in the case of passivity-based control, Proposition 1 provides exponential convergence to zero of $\boldsymbol{s} = \dot{\boldsymbol{e}} + k_v\boldsymbol{e} + k_p\int_{t_0}^{t}\boldsymbol{e}(\iota)d\iota$ which implies the exponential convergence of the error integral $\int_{t_0}^{t}\boldsymbol{e}(\iota)d\iota$. Similarly, Proposition 2 gives the convergence of the error $\boldsymbol{e}$ to zero.

The convergence of the error integral is known to provide robustness with regard to several kinds of perturbations and modeling errors: the constant biases are compensated with no static error and even for non constant bounded disturbances it produces only bounded errors [17].

D. The case of underactuated systems

Let us suppose that the articulated rigid multi-body robot described in Section II-A is underactuated in a number b of DoF. We can assume that $\boldsymbol{u}$ can be partitioned as follows:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u_B^T} & \boldsymbol{u_\theta^T} \end{bmatrix}^T, \qquad (17)$$

where $\boldsymbol{u_B} \in \mathbb{R}^b$ is a vector of "fictitious" generalized forces (that cannot be applied), $\boldsymbol{u_\theta} \in \mathbb{R}^{n-b}$ is the vector of torques (or forces) applied to each joint of the robot. This means that the acceleration $\dot{\boldsymbol{\alpha}}$ of (1) is feasible only if $\boldsymbol{u_B} = \boldsymbol{0}$ holds.

This feasibility constraint means that the reference acceleration $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ in the inverse dynamics control of (2) must ensure that the unactuated part of the reference torque $\boldsymbol{u_{r,B}}$ is null. However, when applying the passivity-based control $\boldsymbol{u_p}$, in order for the Proposition 1 and 2 to hold, the integral gains of (3) have to be added to the full torque vector $\boldsymbol{u_r}$ with a positive definite gain matrix $\boldsymbol{K}$ in (4). We show next that this control scheme is not possible.

Lemma 1: If $\boldsymbol{u_{r,B}} = 0$ there exists no gain matrix $\boldsymbol{K} > 0$ of (4) such that $\forall \boldsymbol{Q}, \boldsymbol{\alpha}, \boldsymbol{\alpha_r}$, the torque $\boldsymbol{u_p}$ of (3) is feasible (i.e. $\boldsymbol{u_{p,B}} = \boldsymbol{0}$, where $\boldsymbol{u_{p,B}}$ is the unactuated part of $\boldsymbol{u_p}$).

Proof: Assume that the conclusion of the lemma does not hold true, that is there exists $\boldsymbol{K} > 0$ such that $\boldsymbol{u_{p}}_B = 0$ if $\boldsymbol{u_r}_B = 0$. Based on (3) and considering the partition proposed in (17), $\boldsymbol{u_{p,B}} = 0$ with $\boldsymbol{u_{r,B}} = 0$ is equivalent to:

$$(\boldsymbol{C_B}(\boldsymbol{Q}, \boldsymbol{\alpha}) + \boldsymbol{K_B})(\boldsymbol{\alpha_r} - \boldsymbol{\alpha}) = \boldsymbol{0}, \qquad (18)$$

where $\boldsymbol{C_B}, \boldsymbol{K_B} \in \mathbb{R}^{b\times n}$ correspond to the first b rows of $\boldsymbol{C}, \boldsymbol{K}$ (the ones associated to the non-actuated dof). Let us take the case where $\boldsymbol{\alpha} = \boldsymbol{0}$, such that $\boldsymbol{C}(\boldsymbol{Q}, \boldsymbol{\alpha}) = \boldsymbol{0}$ too, and let us take $\boldsymbol{\alpha_r} \neq \boldsymbol{0}$. Then, we have $\boldsymbol{K_B}\boldsymbol{\alpha_r} = \boldsymbol{0}$. But because it is required that $\boldsymbol{K} > 0$, then $\boldsymbol{K_B}$ must be full rank, which gives that $\boldsymbol{K_B}\boldsymbol{\alpha_r} \neq \boldsymbol{0}$ contradicting (18), and this concludes the proof. ■

As a consequence of the above lemma, we see that if the reference acceleration $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ is feasible then the passivity-based control can generally not be applied. That means that $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ must not be feasible but must be rather constrained in order for the resulting passivity-based torque $\boldsymbol{u_{p,B}}$ to be feasible.

Within these conditions, Propositions 1 and 2 still hold for underactuated systems. However, the example of joint trajectory tracking in Section II-C cannot be totally applied since not all joint trajectories are feasible. Nevertheless, even if it cannot be proven formally, some of the resulting robustness shown in that example could be conserved with an appropriate control scheme. There are tools allowing to produce the best possible feasible passivity-based control, not only with regard to joint trajectory tracking but also for various kind of kinematic and dynamic tasks. In the next section we present the optimal multi-objective humanoid motion controller based on quadratic programming.

III. Multi-objective humanoid motion control

A. Humanoid robots

Humanoid robots consist of a highly redundant tree structure of several kinematic chains which, by means of external unilateral contact forces, provide a way to control the position and orientation of the underactuated *floating base*.

Let us consider the dynamic model of an articulated rigid multi-body robot described in Section II-A. The configuration of a humanoid robot with n dof can be described as $\boldsymbol{Q} = (\boldsymbol{p_B}, \boldsymbol{R_B}, \boldsymbol{q_\theta})$, where $\boldsymbol{p_B} \in \mathbb{R}^3$ is the position of the floating base, $\boldsymbol{R_B} \in SO(3)$ is a rotation matrix representing its orientation and $\boldsymbol{q_\theta} \in \mathbb{R}^{n-6}$ comprises all the joint angles of the robot (considering only rotational joints). That is, the number of non-controlled dof is $b = 6$, as we cannot exert directly a wrench on the base.

The linear and angular velocities of the floating base of the humanoid robot, $\boldsymbol{v_B}$ and $\boldsymbol{\omega_B}$, are computed as:

$$\boldsymbol{v_B} = \dot{\boldsymbol{p}}_{\boldsymbol{B}}, \qquad \boldsymbol{\omega_B} = \left(\dot{\boldsymbol{R}}_{\boldsymbol{B}}\boldsymbol{R}_{\boldsymbol{B}}^{\boldsymbol{T}}\right)^{\vee}, \tag{19}$$

where $(\cdot)^{\vee} : \mathbb{R}^{3\times3} \to \mathbb{R}^3$, such that if $\boldsymbol{S} = -\boldsymbol{S}^{\boldsymbol{T}} \in \mathbb{R}^{3\times3}$ we have $\boldsymbol{Sx} = (\boldsymbol{S})^{\vee} \times \boldsymbol{x}$, $\forall \boldsymbol{x} \in \mathbb{R}^3$. Then, the configuration velocity $\boldsymbol{\alpha}$ can be expressed by

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{v_B^T} & \boldsymbol{\omega_B^T} & \dot{\boldsymbol{q}}_{\boldsymbol{\theta}}^{\boldsymbol{T}} \end{bmatrix}^T. \tag{20}$$

A humanoid robot has a large number of dof, such that it can fulfill several objectives (or *tasks*) simultaneously, while satisfying kinematic and dynamic *constraints*, e.g. the ones imposed by external forces arising from unilateral contacts, whose force distribution is not unique either.

B. Unilateral contacts

A particular *surface* belonging to a link of the robot that is in contact with the *environment* can be described with a k-tuple of xy-points defined with respect to the surface, representing the vertices of a polygonal shape that approximates the contact region. See Fig. 1.

Let us suppose that the contact force distribution on that surface can be approximated by k lumped reaction force vectors placed at the vertices of the contact region. In order to hold a unilateral contact: (a) the surface frame must be constrained relative to the environment surface, (b) the normal component of each force vector must be positive and (c) each force vector must remain inside the corresponding friction cone, i.e. $\boldsymbol{f_c} \in \mathcal{C}$, to avoid tipping and slipping.

To fulfill both force requirements, we employ pyramidal approximations of the friction cones, $\mathcal{P} \subset \mathcal{C}$, described by 4 unitary bases ($\boldsymbol{\beta_j}$) arranged as columns of a matrix $\boldsymbol{\beta_c} \in \mathbb{R}^{3\times4}$, such that

$$\boldsymbol{f_c} = \boldsymbol{\beta_c}\boldsymbol{\rho_c} \tag{21}$$

where $\boldsymbol{\rho_c} \in \mathbb{R}^4$ is a vector of non-negative coefficients that constrains each force $\boldsymbol{f_c}$ to be inside of the pyramidal approximation of the friction cone, see Fig. 1 and [19].

C. Multi-objective motion solver

We use a QP solver to minimize the tracking error for several weighted tasks (to resolve conflicts), by computing an optimal reference configuration acceleration, $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$, and a feasible reference of external forces $\boldsymbol{u_{e,r}}$ parameterized by the vector $\boldsymbol{\rho_r}$ (made up by concatenating every $\boldsymbol{\rho_c}$), while satisfying linear equality, inequality and bounding constraints; that is, to solve

$$\begin{bmatrix} \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} \\ \boldsymbol{\rho_r} \end{bmatrix} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\left\|\boldsymbol{W}\left(\boldsymbol{A}_{\text{ob}}\boldsymbol{x} - \boldsymbol{b}_{\text{ob}}\right)\right\|^2 + \frac{1}{2}\gamma\left\|\boldsymbol{x}\right\|^2,$$
$$\text{s.t.} \quad \boldsymbol{A}_{\text{eq}}\boldsymbol{x} = \boldsymbol{b}_{\text{eq}}, \ \boldsymbol{Ax} \leq \boldsymbol{b}, \ \boldsymbol{l_b} \leq \boldsymbol{x} \leq \boldsymbol{u_b}, \tag{22}$$

where $\boldsymbol{W} = \text{blkdiag}\left(\boldsymbol{W_1}, \ldots, \boldsymbol{W_{g,k}}\right)$ is a block diagonal matrix made up of individual diagonal matrices that assign a weight to each component of the k tasks in order to solve conflicts among them [14] [19], and γ is a small weight (1E-4) introduced to minimize $\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}$ and $\boldsymbol{\rho_r}$ ensuring, at the same time, the positive definiteness of the Hessian matrix [20].

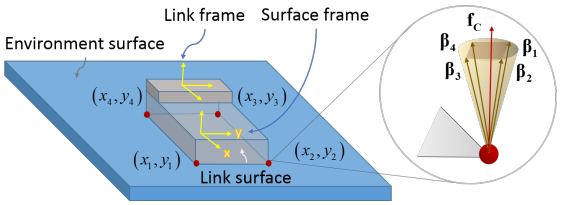

Fig. 1. Lumped reaction forces acting on contact surfaces, constrained in friction cones (or pyramidal approximations).

The matrices $\boldsymbol{A}_{\text{ob}}$, $\boldsymbol{A}_{\text{eq}}$, $\boldsymbol{A}$, and vectors $\boldsymbol{b}_{\text{ob}}$, $\boldsymbol{b}_{\text{eq}}$, $\boldsymbol{b}$, $\boldsymbol{l_b}$, $\boldsymbol{u_b}$ are made up by vertically concatenating the corresponding ones for each task or constraint.

D. Tasks

For the jth task, $\boldsymbol{A}_{\text{ob},\boldsymbol{j}}$ and $\boldsymbol{b}_{\text{ob},\boldsymbol{j}}$ are calculated as

$$\boldsymbol{A}_{\text{ob},\boldsymbol{j}} = \boldsymbol{J_g}, \qquad \boldsymbol{b}_{\text{ob},\boldsymbol{j}} = \ddot{\boldsymbol{g}}_{\text{ob}} - \dot{\boldsymbol{J}}_{\boldsymbol{g}}\boldsymbol{\alpha}, \tag{23}$$

where, $\ddot{\boldsymbol{g}}_{\text{ob}}$ is an acceleration objective and $\boldsymbol{J_g}$, $\dot{\boldsymbol{J}}_{\boldsymbol{g}}$ are the task Jacobian and its derivative, such that $\ddot{\boldsymbol{g}}_{\text{ob}} = \boldsymbol{J_g}\dot{\boldsymbol{\alpha}} + \dot{\boldsymbol{J}}_{\boldsymbol{g}}\boldsymbol{\alpha}$.

Here we consider only three different kind of tasks, tracked in a similar way as in (15):

1) Posture task: The acceleration objective is $\boldsymbol{q_\theta}$; that is, $\ddot{\boldsymbol{g}}_{\text{ob}} = \ddot{\boldsymbol{q}}_{\boldsymbol{\theta},\text{ob}}$, such that

$$\ddot{\boldsymbol{q}}_{\boldsymbol{\theta},\text{ob}} = k_p\left(\boldsymbol{q}_{\boldsymbol{\theta},\boldsymbol{d}} - \boldsymbol{q}_{\boldsymbol{\theta}}\right) + k_v\left(\dot{\boldsymbol{q}}_{\boldsymbol{\theta},\boldsymbol{d}} - \dot{\boldsymbol{q}}_{\boldsymbol{\theta}}\right) + \ddot{\boldsymbol{q}}_{\boldsymbol{\theta},\boldsymbol{d}}. \tag{24}$$

2) Position task: In this case, $\ddot{\boldsymbol{g}}_{\text{ob}} = \dot{\boldsymbol{v}}_{\text{ob}}$, the acceleration of a point described in the world frame, such that its desired trajectory ($\boldsymbol{p_d}$, $\boldsymbol{v_d}$, $\dot{\boldsymbol{v}}_{\boldsymbol{d}}$) is tracked as

$$\dot{\boldsymbol{v}}_{\text{ob}} = k_p\left(\boldsymbol{p_d} - \boldsymbol{p}\right) + k_v\left(\boldsymbol{v_d} - \boldsymbol{v}\right) + \dot{\boldsymbol{v}}_{\boldsymbol{d}}. \tag{25}$$

3) Orientation task: Here, $\ddot{\boldsymbol{g}}_{\text{ob}} = \dot{\boldsymbol{\omega}}_{\text{ob}}$, the angular acceleration of a particular frame with respect to the world frame, such that its desired trajectory ($\boldsymbol{R_d}$, $\boldsymbol{\omega_d}$, $\dot{\boldsymbol{\omega}}_{\boldsymbol{d}}$) is tracked as

$$\dot{\boldsymbol{\omega}}_{\text{ob}} = k_p\tilde{\boldsymbol{\Omega}} + k_v\left(\boldsymbol{\omega_d} - \boldsymbol{\omega}\right) + \dot{\boldsymbol{\omega}}_{\boldsymbol{d}}, \tag{26}$$

where $\tilde{\boldsymbol{\Omega}} = \left(\log\left\{\boldsymbol{R_d}\boldsymbol{R}^{\boldsymbol{T}}\right\}\right)^{\vee}$ represents the error vector in orientation.

E. Constraints

Here we consider only four types constraints:

1) Underactuation / torque constraint: The underactuation constraint ensures the generation of a feasible motion for the floating base, provided by adequate unilateral contact forces with the environment; it is an equality constraint. The torque constraint ensures that the required torques are within the limitations of the actuators (minimum and maximum torques: $\underline{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{\tau}}$); it is an inequality (box) constraint.

Let us define a matrix $\boldsymbol{D}$, such that $\boldsymbol{u_{e,r}} = \boldsymbol{D}\boldsymbol{\rho_r}$, as well as consider that $\boldsymbol{x} = \begin{bmatrix} \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}}^{\boldsymbol{T}} & \boldsymbol{\rho_r^T} \end{bmatrix}^T$. Then, the underactuation and torque constraints are, respectively, specified as

$$\begin{bmatrix} \boldsymbol{M_B} & -\boldsymbol{D_B} \end{bmatrix}\boldsymbol{x} = -\boldsymbol{C_B}\boldsymbol{\alpha} - \boldsymbol{g_B}, \tag{27}$$

$$\underline{\boldsymbol{\tau}} - \boldsymbol{C_j}\boldsymbol{\alpha} - \boldsymbol{g_j} \leq \begin{bmatrix} \boldsymbol{M_j} & -\boldsymbol{D_j} \end{bmatrix}\boldsymbol{x} \leq \bar{\boldsymbol{\tau}} - \boldsymbol{C_j}\boldsymbol{\alpha} - \boldsymbol{g_j}, \tag{28}$$

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

2) Joint limits constraints: Joint range and speed limits can be specified using inequality constraints, as done in [20].

3) Friction constraint: To meet the force constraints explained in Section III-B, it is necessary to ensure that $\mathbf{0} < \boldsymbol{\rho}$. This can be done by using bounding constraints.

4) Surface frame constraint: To constrain the surface frame (sf) relative to the environmental frame we track it down to a desired position and orientation, $\boldsymbol{p}_{\text{sf},\boldsymbol{d}}$ and $\boldsymbol{R}_{\text{sf},\boldsymbol{d}}$, with zero linear and angular velocity:

$$\boldsymbol{J_g}\dot{\boldsymbol{\alpha}}_{\text{ref}} + \dot{\boldsymbol{J}}_{\boldsymbol{g}}\boldsymbol{\alpha} = \boldsymbol{K_p}\begin{bmatrix} \boldsymbol{p}_{\text{sf},\boldsymbol{d}} - \boldsymbol{p}_{\text{sf}} \\ \tilde{\boldsymbol{\Omega}}_{\text{sf}} \end{bmatrix} - \boldsymbol{K_v}\begin{bmatrix} \boldsymbol{v}_{\text{sf}} \\ \boldsymbol{\omega}_{\text{sf}} \end{bmatrix}, \quad (29)$$

where $\tilde{\boldsymbol{\Omega}}_{\text{sf}}$ is defined as in (26), and $\boldsymbol{K_p}$, $\boldsymbol{K_v}$ are matrices of PD gains, such that it is possible to activate / deactivate the constraint on a specific dof of the contact.

F. Introduction of integral gains

We introduce the integral term $\boldsymbol{Ls}$ by adding it to the reference force input as in (3):

$$\boldsymbol{u_p} = \boldsymbol{M}(\boldsymbol{Q})\dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{Q},\boldsymbol{\alpha})\boldsymbol{\alpha} + \boldsymbol{g}(\boldsymbol{Q}) - \boldsymbol{u}_{\boldsymbol{e},\boldsymbol{r}} + \boldsymbol{Ls}, \quad (30)$$

This raises two main issues for the case of humanoids: (i) in order to take profit from the theoretical guarantees of the integral gains, the integral term has to be included into the whole vector of input generalized forces, including the unactuated components, and this term must then be compensated on these components; (ii) these additional terms alter the whole body dynamics. For instance when adding this term, the real acceleration of the robot is different from the reference one, that is:

$$\dot{\boldsymbol{\alpha}} = \dot{\boldsymbol{\alpha}}_{\boldsymbol{r}} + \boldsymbol{M}^{-1}\boldsymbol{Ls}, \quad (31)$$

and this difference needs to be accounted for when ensuring the feasibility of the resulting desired acceleration. These two aspects are a major part of our contribution in this paper that we describe in what follows.

1) Underactuation / torque constraint: Without the integral gains, this constraint, written as $\boldsymbol{u_B} = \mathbf{0}$, was respected thanks to (27). This constraint needs then to be modified to include the integral term:

$$\begin{bmatrix} \boldsymbol{M_B} & -\boldsymbol{D_B} \end{bmatrix}\boldsymbol{x} = -\boldsymbol{C_B}\boldsymbol{\alpha} - \boldsymbol{g_B} + \boldsymbol{L_B}\boldsymbol{s}, \quad (32)$$

where $\boldsymbol{L_B}$ are the unactuated rows of $\boldsymbol{L}$. This integral term can be regarded as an "artificial external generalized force", for which the QP solver needs to be aware to respect the underactuation constraint (as well as the others). The QP will then find the optimal acceleration with regard to this integral term, including, by doing so, the history of tracking error inside the task-based control.

Similarly, we also update the torque limit constraint by introducing the integral gain:

$$\underline{\boldsymbol{\tau}} - \boldsymbol{C_j}\boldsymbol{\alpha} - \boldsymbol{g_j} + \boldsymbol{L_j}\boldsymbol{s} \leq \begin{bmatrix} \boldsymbol{M_j} & -\boldsymbol{D_j} \end{bmatrix}\boldsymbol{x} \leq \bar{\boldsymbol{\tau}} - \boldsymbol{C_j}\boldsymbol{\alpha} - \boldsymbol{g_j} + \boldsymbol{L_j}\boldsymbol{s}, \quad (33)$$

where $\boldsymbol{L_j}$ are the actuated lines of $\boldsymbol{L}$.

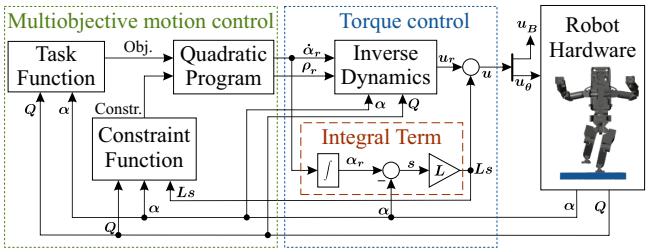

Fig. 2. Description of the framework architecture.

2) Other feasibility constraints: The most important additional feasibility condition is the surface frame constraint given (29). This can be updated using (31), giving:

$$\boldsymbol{J_g}\dot{\boldsymbol{\alpha}}_{\text{ref}} + \boldsymbol{J_g}\boldsymbol{M}^{-1}\boldsymbol{Ls} + \dot{\boldsymbol{J}}_{\boldsymbol{g}}\boldsymbol{\alpha} = \boldsymbol{K_p}\begin{bmatrix} \boldsymbol{p}_{\text{sf,des}} - \boldsymbol{p}_{\text{sf}} \\ \tilde{\boldsymbol{\Omega}}_{\text{sf}} \end{bmatrix} - \boldsymbol{K_v}\begin{bmatrix} \boldsymbol{v}_{\text{sf}} \\ \boldsymbol{\omega}_{\text{sf}} \end{bmatrix}. \quad (34)$$

Joint position and speed limit constraints can also be updated using the same process, but for the case of our simulations, we noticed that because they are already relatively conservative, the actual constraints remain respected. Fig. 2 illustrates the proposed control framework.

IV. Simulation Results

A. Simulation Environment

To test our control framework we use our humanoid robot, the HRP-5P. It is 1.819 m height when the legs are fully extended and weighs 104.835 kg. It has 35 structural dof: two legs of 6 dof each, two arms of 9 dof each, a waist with 3 dof and a head with a neck of 2 dof. Our humanoid is modeled in Matlab Simscape Multibody™, by using the mass distribution parameters given by the CAD model. The actuators on the joints are implemented by taking into account the mechanical effects of an ideal gear box, the rotor inertia and a friction model consisting of the sum of the Stribeck, Coulomb[2] and the viscous components.

The values of the gear ratio and the rotor inertia for each joint were taken from the real servo datasheet, whereas the friction parameters were chosen based on realistic values, inspired in the ones identified for the robot ABB IRB 6620 in [21] (see Table I), and applied equally to all the joints.

TABLE I

Friction parameters used for every joint.

Parameter	Value
Breakaway Friction Torque	0.05 Nm
Breakaway Friction Velocity	0.2 rad/s
Coulomb Friction Torque	0.04 Nm
Viscous Friction Coefficient	0.0005 Nm / (rad/s)

The input to these actuators is the joint torque, scaled by the gear ratio and by a non-modeled torque scale (0.9), which can either be considered to model a non-precisely known gain arising when considering actuators without electrical

[2]The Stribeck and Coulomb components correspond to a continuous model representing the effect of the static and kinetic friction [21].

current feedback – for which the motor torque reference is not respected. Electrical dynamics were neglected.

The contact model between the feet and the ground is implemented using the free Simscape Multibody Contact Forces Library [22]. This one considers that each vertex of the foot is attached to a tiny fictitious sphere, such that the wrench resulting from the contact of each sphere with the ground is calculated by using penalty for the normal component, and a Stick-Slip continuous friction model for the tangential component (see Table II). This allows to simulate collision and sliding motion.

TABLE II

CONTACT PARAMETERS BETWEEN FEET AND GROUND.

Parameter	Value
Contact Stiffness	1E7 N/m
Contact Damping	1E3 n/(m/s)
Static & Kinetic Friction Coefficients	0.4 & 0.3
Velocity Threshold	0.001 m/s

The controller is implemented in discrete time with a step size of $T = 0.005$ s by using common blocks of Simulink™ and the QP Solver (`quadprog`) provided by Matlab's Optimization Toolbox™.

The controller uses a model of the robot that does not consider the joint friction, neither the torque scale. Furthermore, it includes a modeling error of the inertial parameters induced by randomly incrementing or decrementing the mass of each link up to 10% of its value. The random seed is the same for every simulation, in order to consider the same mass distribution. The feedback received by the controller is a set of *clean* signals given as an output by the Simscape model; that is, the controller is also fully aware of the actual position and orientation of the body, which in reality can only be estimated but not measured. As for $\boldsymbol{\alpha}$, it is computed by using finite differences.

Let us compare the performance of the proposed passivity-based controller not only with the classical inverse dynamics controller, but also with a controller representing the approaches taken by different teams during the DARPA Robotics Challenge (DRC) [23], [19], which used the integral of the reference acceleration to close a velocity / position loop superimposed to the torque command, without theoretical proof nor the consideration of the integral term within the constraints of the QP and not based on passivity. This *decoupled* approach can be written using (3) with $\boldsymbol{L} = \lambda \text{diag}(\boldsymbol{M})$, providing a gain proportional to the mass driven by each joint. The comparison is done by testing two sequences of motions: *"Bowing"* and *"Lifting Feet"*, each one using (a) the Inverse Dynamics Control, (b) the Decoupled Integral Term Control and (c) the Passivity-Based Control, just to properly label the approaches.

The solver used to run the simulations in Matlab Simulink™ (R2017b) is `ode15s(stiff/NDF)`, with max-step size = 1E-2, rel. tolerance = 1E-4 and abs. tolerance = 1E-5.

The tasks considered are: (a) posture task (`q`), (b) position task of CoM (`com`), (c) pose task (position + orientation) of floating Base (`poseB`), (d) pose task of Chest (`poseCh`), (e) pose task of Right and Left Hands (`poseRH`, `poseLH`), and (f) pose task of Right and Left Feet (`poseRF`, `poseLF`).

The constraints considered are: (a) motion, (b) torque, (c) joint range and speed limits, and (d) surface frame constraints of Right and Left Feet Soles (`RFSole`, `LFSole`).

In both simulations the initial configuration corresponds to all joints angles at 0 deg. The default weight used by both simulations is shown in Table III[3], where $W = \text{diag}(\texttt{W\_mask}) \times \texttt{W\_val}$. The default PD gain matrices for the constraints used by both simulations are shown in Table IV. Notice that we have released the contact dof corresponding to the position and yaw orientation of the soles (not their velocities), as when the robot is standing it is not possible to correct those without sliding. However, the dof corresponding to the pitch and roll orientations are tracked back to 0 deg if the foot starts to tilt due to a disturbance. The default state of the task or constraint is also shown in both tables (`active`). Notice that if a sole constraint is active, the pose task for the corresponding foot is not, and *vice-versa*.

The value of the integral gain of the passivity-based control is $\lambda = 200$, whereas the one of the decoupled approach is $\lambda = 30$, the latter representing the largest value empirically found for which the closed-loop was stable.

TABLE III

DEFAULT TASK PARAMETER VALUES.

Task	Parameter	Value	Task	Parameter	Value
q	active	✓	poseCh	active	✓
	W_val	10		W_val	500
	W_mask	ones(1, n)		W_mask	[0 0 0 1 1 1]
com	active	✓	poseRH & poseLH	active	✓
	W_val	1000		W_val	100
	W_mask	[1 1 0]		W_mask	[1 1 1 1 1 1]
poseB	active	✓	poseRF & poseLF	active	✗
	W_val	200		W_val	500
	W_mask	[0 0 5 2 2 2]		W_mask	[1 1 1 1 1 1]

TABLE IV

DEFAULT CONSTRAINT PARAMETER VALUES.

Constraint	Parameter	Value
RFSole & LFSole	active	✓
	Kp	diag([0 0 0 1 1 0]) * 100
	Kv	diag([1 1 1 1 1 1]) * 200

B. Bowing simulation and results

Let the HRP-5P perform the following motion: (1) go to half-sitting, (2) incline the chest 60 deg and (3) return.

This bowing motion is performed by setting, as step command, the desired inclination of the chest of the corresponding task. Additionally, an arbitrary half-sitting configuration is used for the low-weighted posture task during

[3]Within this scheme, the weights (and masks) represent the most critical parameters, as switching off some dof or setting "low" values may degrade the quality of the task tracking, leading to a possible fall. In our case, these weights were heuristically tuned based on the "importance" of each task. On the other hand, the PD gains of the tasks can be seen as kinematic gains, used to build a trajectory reference (output of the QP) which tracks faster or slower the desired one, allowing some freedom in their specification.

the whole motion. Its purpose is to deal with the initial singularity to bend the knees correctly, and to help the solver to decide a final configuration for the 9 dof arms, or otherwise they would keep on moving within the null space of the cumulative task Jacobian.

Let us set the PD gains of almost all the tasks to the same values ($k_p = 50$, $k_v = 15$), except for `PoseCh`, and simulate this motion using two sets of PD gains for `PoseCh`: (a) ($k_p = 50$, $k_v = 15$) and (b) ($k_p = 100$, $k_v = 20$), as well as considering the three controllers (six simulations).

The plot showing the orientation of the Chest is shown in Fig. 3. As it can be seen, even though the desired motion is abrupt, the use of comparatively low PD gains lead to a reference trajectory that was slower than the desired one. However, the passivity-based control achieved a zero steady-state error. In the case of the inverse dynamics control, to diminish the steady-state error it is necessary to follow the desired trajectory faster (increasing the PD gains), leading to motions that may not be feasible. The comparatively faster motion of the passivity-based control with respect to the inverse dynamics one was explained in Section II-C. Fig. 4 shows one screenshot of the motion.

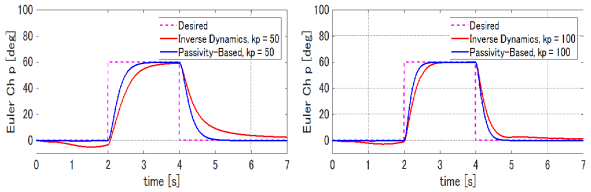

Fig. 3. Pitch Chest ("Bowing").

C. Lifting feet simulation and results

Let the HRP-5P perform the following motion: (1) go to *half-sitting*, (2) rise the hands (*hands*↑), (3) place the CoM over the right foot (*CoM*→ r), (4) lift the left ankle (*LF*↑), (5) put the left foot on the ground (*LF*↓), (6) place the CoM over the left foot (*CoM*→ l), (7) lift the right ankle (*RF*↑). (8) put the right foot on the ground (*RF*↓), and (9) *return*.

The PD gains of all tasks are set to the same values ($k_p = 100$, $k_v = 20$). Also, the desired trajectories are specified instead of set-point tasks. These ones are generated by using *Linear Segments Parabolic Blends* (LSPB) interpolation [17], using as final values the ones marked with $*$ in Table V.

This motion was simulated using each controller (three simulations), such that their performance can be compared through the plots of (i) the horizontal trajectory of the CoM together with the vertical position of the floating Base (Fig. 6), (ii) the trajectory of the Right Hand position (Fig. 7), (iii) the trajectory of the Left Foot position (Fig. 8), and (iv) the contact forces acting on the Left Foot, calculated vs. measured (Fig. 9).

As we can see, the performance of the inverse dynamics control is not good in the presence of modeling errors. Actually, when using this controller, the robot tilted before returning to half-sitting posture, but could recover (luckily)

Fig. 4. "Bowing", with passivity-based control (t = 4 s).

Fig. 5. "Lifting Feet", with passivity-based control (t = 6 s).

and didn't fall. The decoupled integral term control and the passivity-based control showed a very good performance until the moment when the CoM was transferred over the left foot. Then, the robot suddenly fell to the left when using the decoupled controller. This happens because the QP has no knowledge of the extra torque sent to the robot, assuming a wrong contact model and violating the conditions to hold the contact[4]. This can be seen in Fig. 9, where the difference between the calculated and measured normal force for this controller only is 500 N (the smaller discrepancy of the other two is because of the modeling errors). On the other hand, the passivity-based controller, whose QP is aware of the added torque, succeeds to perform the motion without falling, converging to the desired trajectory in almost all the cases: except for the z component of the final position of the right hand. This is because the tasks for the hand positions and for the CoM are in conflict, producing an averaged reference trajectory. Fig. 5 shows one screenshot of the motion.

V. Conclusions and Future Work

We have presented a framework for a QP-based multi-objective closed-loop torque-control with integral gains. The key idea is to use the QP to generate optimal reference accelerations which take into account integral gains in the feasibility constraint. The scheme provides not only robustness to modeling errors and joint friction, but also, in the case of passivity-based control, guarantees theoretically good performances regarding disturbances with a Lyapunov global exponential convergence. These developments were tested on a high-fidelity dynamical simulator, where we could observe the improvements compared to the classical control. The future developments include (i) a study on other sources of disturbances, such as flexible structures, or estimation errors, (ii) the integration of force feedback and stabilization control, (iii) a finer control of the control compliance, and (iv) conclude all these adding with real experiments.

Acknowledgments

This work was partially supported by NEDO's "Development of a highly dependable humanoid robot system that can work in unstructured environments" project, and in part by the COMANOID EU project.

[4]In the case of [23], [19], different controllers were used for the joints, i.e. the ankle joint had no integral term added. In our case, only one controller. Also, they are already considering stabilization control, which we have not.

TABLE V

OBJECTIVES FOR THE "LIFTING FEET" SIMULATION.

	Objective	Simulation phases								
Phase τ [s]		half-sitting [0, 1.5)	hands↑ [1.5, 2)	CoM→ r [2, 3)	LF↑ [3, 3.5)	LF↓ [3.5, 4)	CoM→ l [4, 5.5)	RF↑ [5.5, 6)	RF↓ [6, 6.5)	return [6.5, 7.5)
com	pos*[m]	[0 0 •]		[0 -0.1 •]			[0 0.1 •]			[0 0 •]
poseB	pos*[m]	[• • 0.7]								
poseRH	pos*[m]	[0 ∓0.4 0.8]	[0.25 ∓0.5 1.5]							[0 ∓0.4 0.8]
poseLH	PRY*[deg]	[0 ±45 0]	[-90 0 0]							[0 ±45 0]
poseRF	pos*[m]	✗						[0 -0.1 0.3]	[0 -0.1 0.15]	✗
poseLF	pos*[m]	✗			[0 0.1 0.3]	[0 0.1 0.15]	✗			

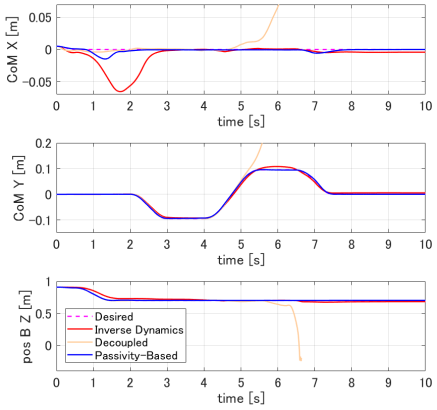

Fig. 6. CoM (xy) / Position (z) B.

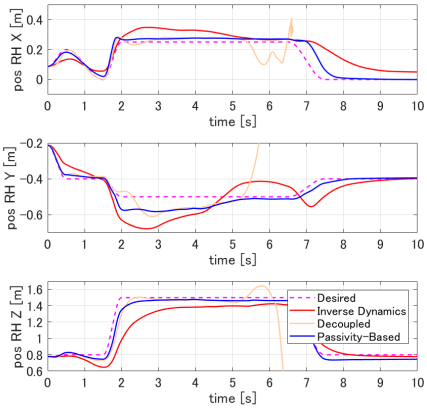

Fig. 7. Position RH.

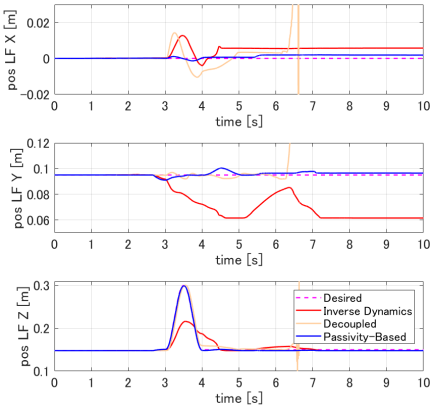

Fig. 8. Position LF.

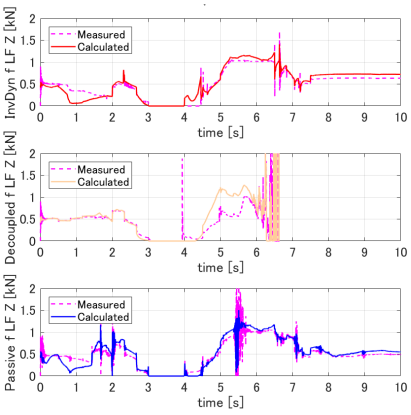

Fig. 9. Contact forces LF.

REFERENCES

[1] M. Morisawa et al. Combining suppression of the disturbance and reactive stepping for recovering balance. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[2] Y. Mikami et al. Identification of HRP-2 foot's dynamics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[3] S. Traversaro et al. Inertial parameter identification including friction and motor dynamics. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2013.

[4] K. Hirai et al. The development of Honda humanoid robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, 1998.

[5] J. Englsberger et al. Overview of the torque-controlled humanoid robot TORO. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.

[6] G. Morel et al. The precise control of manipulators with high joint-friction using base force/torque sensing. *Automatica*, 36(7), 2000.

[7] M. Mistry et al. Inverse Dynamics Control of Floating Base Systems Using Orthogonal Decomposition. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[8] S. Kuindersma et al. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3), 2016.

[9] J.-J.E. Slotine et al. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3), 1987.

[10] I.D. Landau et al. Applications of the Passive Systems: Approach to the Stability Analysis of Adaptive Controllers for Robot Manipulators. *Int. Journal of Adaptive Control and Signal Processing*, 3, 1989.

[11] K. Bouyarmane et al. On Weight-Prioritized Multi-Task Control of Humanoid Robots. *IEEE Transactions on Automatic Control*, 2017.

[12] V. Duindam et al. Lagrangian Dynamics of Open Multibody Systems with Generalized Holonomic and Nonholonomic Joints. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[13] R.M. Murray et al. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[14] K. Bouyarmane et al. Using a Multi-Objective Controller to Synthesize Simulated Humanoid Robot Motion with Changing Contact Configurations. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[15] P.J. From et al. Singularity-free dynamic equations of spacecraft-manipulator systems. *Acta Astronautica*, 69(11–12), 2011.

[16] M. Bjerkeng et al. A new Coriolis matrix factorization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[17] M.W. Spong et al. *Robot Modeling and Control*. John Wiley & Sons, 2006.

[18] A. Herzog et al. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[19] M.A. Hopkins et al. Compliant Locomotion Using Whole-Body Control and Divergent Component of Motion Tracking. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[20] J. Vaillant et al. Multi-contact vertical ladder climbing by an HRP-2 humanoid. *Autonomous Robots*, 40(3):561–580, March 2016.

[21] A. Bittencourt et al. Static Friction in a Robot Joint Modeling and Identification of Load and Temperature Effects. *Journal of Dynamic Systems, Measurement, and Control*, 134(5), 2012.

[22] S. Miller. Simscape Multibody Contact Force Library, 2017.

[23] M. Johnson et al. Team IHMC's lessons learned from the DARPA robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208, 2015.