Automatic Gain Tuning of a Momentum Based Balancing Controller for Humanoid Robots

Daniele Pucci, Gabriele Nava and Francesco Nori[1]

Abstract— This paper proposes a technique for automatic gain tuning of a momentum based balancing controller for humanoid robots. The controller ensures the stabilization of the centroidal dynamics and the associated zero dynamics. Then, the closed-loop, constrained joint space dynamics is linearized and the controller's gains are chosen so as to obtain desired properties of the linearized system. Symmetry and positive definiteness constraints of gain matrices are enforced by proposing a tracker for symmetric positive definite matrices. Simulation results are carried out on the humanoid robot iCub.

I. INTRODUCTION

Humanoid robotics is an undoubtedly flourishing field of research. Locomotion and manipulation have received a special attention from the control community, and the results shown at the DARPA robotics challenge are both stimulating and promising [1], [2]. Despite these advances, robust controllers for balancing and walking of humanoids still require a special focus of the robotics community. Furthermore, when these controllers are implemented on real platforms, the achievement of desired system performances usually requires time consuming tuning of the (often very numerous) gains characterizing the control laws. This paper proposes a technique to tune automatically the gains of a momentum based balancing strategy for humanoid robots.

A classical approach to the modelling and control of humanoid robots is based on considering the robot attached to ground, i.e. the robot is considered to be *fixed-base* [3]. In this case, in fact, well-known classical control strategies for manipulators can be directly applied for robot control. The limitations of this approach arise when attempting to tackle the general control problem of a humanoid interacting with its surrounding environment. For instance, running involves flight phases where the fixed-base approach clearly fails. At the modelling level, the Euler-Poincarè equations provide singularity free equations of motion for the humanoid robot [4, Chapter 13], and efficient algorithms can be applied for determining the components of these equations [5]. When considering these equations of motion, the mechanical system representing the humanoid robot is usually under actuated, and this forbids the full feedback linearization of the closed-loop system [6], [5]. The system underactuation is usually dealt with by means of constraints that arise from the contacts between the robot and the environment. This requires a close attention to the forces the robot exerts on the environment since uncontrolled forces may break the contacts, thus rendering the control of the robot critical. To ensure that the contact forces respect some physical constraints, different optimization procedures can be applied [7], [8].

Task-based control strategies have proven to be an efficient solution for balancing and walking of humanoid robots [9], [10], [11]. The aim of these strategies is the achievement of several control objectives, which are organized in a hierarchical structure. The possibility of defining different control objectives with different priorities is an efficient way to deal with manipulation tasks while balancing [12]. Furthermore, high priority tasks may be used to control directly the contact forces the robot exerts at the contact points, e.g. through the control of the center of mass dynamics [13].

When the above control algorithms are implemented in real applications, a long and tedious tuning of control gains is often required to achieve desired system performances. Despite the large number of gain optimization procedures for classical dynamical systems (see, e.g., [14], [15]) gain optimization techniques for floating base systems, and in particular in the field of humanoid robots, still needs more investigations. Preliminary results in this direction consist in applying classical LQR approaches to the linearized humanoid robot dynamics [16], [17]. In particular, LQR based optimization techniques can be applied to the so-called centroidal dynamics [16]. Another approach may consist of considering simplified robot models, and then apply classical gain scheduling procedures for balancing purposes [18].

In this paper, we propose a gain tuning method for the momentum-based control strategy we implemented on the iCub humanoid robot [19]. Contrary to [16], we optimize the gains by imposing desired properties of the linearized joint space dynamics. The approach handles any number of contacts between the robot and the environment. Symmetry and positive definiteness constraints of gain matrices are enforced via a nonlinear tracker for symmetric positive definite matrices. Simulation results verify the presented approach.

The paper is organized as follows. Section II introduces notation and system modelling. Section III recalls and complements the momentum-based control strategy [19]. Section IV presents the linearization and the gain optimization procedure. Section V presents simulations results using the iCub robot model. Conclusions and future works conclude the paper.

[1] The authors are with the iCub Facility department, Istituto Italiano di Tecnologia, Via Morego 30, Genoa, Italy `name.surname@iit.it`

II. BACKGROUND

A. Notation

- $\mathcal{I}$ defines an inertial frame of reference, with its z axis pointing against the gravity. The constant g denotes the norm of the gravitational acceleration.
- Given a matrix $A \in \mathbb{R}^{m \times n}$, we denote with $A^\dagger \in \mathbb{R}^{n \times m}$ its Moore Penrose pseudoinverse.
- $e_i \in \mathbb{R}^m$ is the canonical vector, consisting of all zeros but the i-th component that is equal to one.
- We denote with m the total mass of the robot.

B. Modelling

The robot is modelled as a multi-body system composed of $n + 1$ rigid bodies, called links, connected by n joints with one degree of freedom each. We also assume that none of the links has an *a priori* constant pose with respect to an inertial frame, i.e. the system is *free floating*.

The robot configuration space is the Lie group $\mathbb{Q} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n$ and it is characterized by the *pose* (position and orientation) of a base frame attached to a robot's link, and the joint positions. An element $q \in \mathbb{Q}$ can be defined as the following triplet: $q = (^{\mathcal{I}}p_{\mathcal{B}}, {}^{\mathcal{I}}R_{\mathcal{B}}, q_j)$ where $^{\mathcal{I}}p_{\mathcal{B}} \in \mathbb{R}^3$ denotes the position of the base frame with respect to the inertial frame, $^{\mathcal{I}}R_{\mathcal{B}} \in \mathbb{R}^{3 \times 3}$ is a rotation matrix representing the orientation of a *base frame*, and $q_j \in \mathbb{R}^n$ is the joint configuration. The velocity of the multi-body system can be characterized by the *algebra* of the group defined as $\mathbb{V} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^n$. An element of $\mathbb{V}$ is a triplet $\nu = (^{\mathcal{I}}\dot{p}_{\mathcal{B}}, {}^{\mathcal{I}}\omega_{\mathcal{B}}, \dot{q}_j) = (\mathrm{v}_{\mathcal{B}}, \dot{q}_j)$, where $^{\mathcal{I}}\omega_{\mathcal{B}}$ is the angular velocity of the base frame expressed w.r.t. the inertial frame, i.e. $^{\mathcal{I}}\dot{R}_{\mathcal{B}} = S(^{\mathcal{I}}\omega_{\mathcal{B}})^{\mathcal{I}}R_{\mathcal{B}}$. A more detailed description of the floating base model is provided in [19].

We assume that the robot interacts with the environment by exchanging n_c distinct wrenches. The equations of motion of the multi-body system can be described applying the Euler-Poincaré formalism [20, Ch. 13.5]:

$$M(q)\dot{\nu} + C(q,\nu)\nu + G(q) = B\tau + \sum_{k=1}^{n_c} J_{\mathcal{C}_k}^\top f_k \quad (1)$$

where $M \in \mathbb{R}^{n+6 \times n+6}$ is the mass matrix, $C \in \mathbb{R}^{n+6 \times n+6}$ is the Coriolis matrix, $G \in \mathbb{R}^{n+6}$ is the gravity term, $B = (0_{n \times 6}, 1_n)^\top$ is a selector matrix, $\tau \in \mathbb{R}^n$ is a vector representing the internal actuation torques, and $f_k \in \mathbb{R}^6$ denotes an external wrench applied by the environment to the link of the k-th contact. The Jacobian $J_{\mathcal{C}_k} = J_{\mathcal{C}_k}(q)$ is the map between the robot's velocity ν and the linear and angular velocity at the k-th contact link.

As described in [21, Sec. 5], it is possible to apply a coordinate transformation in the state space (q, ν) that transforms the system dynamics (1) into a new form where the mass matrix is block diagonal, thus decoupling joint and base frame accelerations. Also, in this new set of coordinates, the first six rows of Eq. (1) are the *centroidal dynamics*[1]. As

[1]In the specialized literature, the term *centroidal dynamics* is used to indicate the rate of change of the robot's momentum expressed at the center-of-mass, which then equals the summation of all external wrenches acting on the multi-body system [22].

an abuse of notation, we assume that system (1) has been transformed in this new set of coordinates, i.e.

$$M(q) = \begin{bmatrix} M_b(q) & 0_{6 \times n} \\ 0_{n \times 6} & M_j(q) \end{bmatrix}, \quad H = M_b \mathrm{v}_{\mathcal{B}}, \quad (2)$$

where $M_b \in \mathbb{R}^{6 \times 6}, M_j \in \mathbb{R}^{n \times n}$, $H := (H_L, H_\omega) \in \mathbb{R}^6$ is the robot momentum, and $H_L, H_\omega \in \mathbb{R}^3$ are the linear and angular momentum at the center of mass, respectively.

Lastly, it is assumed that a set of holonomic constraints acts on System (1). These holonomic constraints are of the form $c(q) = 0$, and may represent, for instance, a frame having a constant pose w.r.t. the inertial frame. In the case where this frame corresponds to the location at which a rigid contact occurs on a link, we represent the holonomic constraint as $J_{\mathcal{C}_k}(q)\nu = 0$. Hence, the kinematic constraint associated with all the rigid contacts can be represented as

$$J(q)\nu = \begin{bmatrix} J_{\mathcal{C}_1}(q) \\ \cdots \\ J_{\mathcal{C}_{n_c}}(q) \end{bmatrix} \nu = \begin{bmatrix} J_b & J_j \end{bmatrix} \nu = J_b \mathrm{v}_{\mathcal{B}} + J_j \dot{q}_j = 0, \quad (3)$$

with $J_b \in \mathbb{R}^{6n_c \times 6}, J_j \in \mathbb{R}^{6n_c \times n}$, and $\dot{q}_j \in \mathbb{R}^n$ the joint space velocity. The base frame velocity is denoted by $\mathrm{v}_{\mathcal{B}} \in \mathbb{R}^6$, which in the new coordinates yielding a block-diagonal mass matrix is given by $\mathrm{v}_{\mathcal{B}} = (\dot{p}_c, \omega_o)$, where $\dot{p}_c \in \mathbb{R}^3$ is the velocity of the system's center of mass $p_c \in \mathbb{R}^3$, and $\omega_o \in \mathbb{R}^3$ is the so-called system's *average angular velocity*. By differentiating the kinematic constraint Eq. (3), one obtains

$$J\dot{\nu} + \dot{J}\nu = J_b \dot{\mathrm{v}}_{\mathcal{B}} + J_j \ddot{q}_j + \dot{J}_b \mathrm{v}_{\mathcal{B}} + \dot{J}_j \dot{q}_j = 0. \quad (4)$$

III. RECALLS AND COMPLEMENTS ON THE MOMENTUM-BASED CONTROL STRATEGY

We recall and complement the *momentum-based* control strategy implemented on our iCub humanoid robot [19]. The control objective is the stabilization of a desired robot momentum and the stability of the associated zero dynamics.

A. Momentum Control

Recall that the rate-of-change of the robot momentum equals the net external wrench acting on the robot, which in the present case reduces to the contact wrench $f := (f_1, \cdots, f_{n_c}) \in \mathbb{R}^{6n_c}$ plus the gravity wrenches. Then, in view of Eq. (2), the rate-of-change of the robot momentum can be expressed as:

$$\frac{\mathrm{d}}{\mathrm{d}t}(M_b \mathrm{v}_{\mathcal{B}}) = \dot{H}(f) = J_b^\top f - mge_3, \quad (5)$$

where $e_3 \in \mathbb{R}^6$.

The control objective is defined as the stabilization of a desired robot momentum $H^d \in \mathbb{R}^6$. Let $\tilde{H} = H - H^d$ define the momentum error. Assuming that the contact wrenches f can be chosen at will, then we choose f such that [19]:

$$\dot{H}(f) = \dot{H}^* := \dot{H}^d - K_p \tilde{H} - K_i I_{\tilde{H}} \quad (6a)$$

$$\dot{I}_{\tilde{H}} = \begin{bmatrix} J_G^L(q_j) \\ J_G^\omega(q_j^d) \end{bmatrix} \dot{q}_j \quad (6b)$$

$K_p, K_i \in \mathbb{R}^{6\times6}$ two symmetric, positive definite matrices and

$$\bar{J}_G(q_j) := -M_b J_b^\dagger J_j = \begin{bmatrix} J_G^L(q_j) \\ J_G^\omega(q_j) \end{bmatrix} \in \mathbb{R}^{6\times n}, J_G^L, J_G^\omega \in \mathbb{R}^{3\times n}$$

If $n_c > 1$, there are infinite contact wrenches f that satisfy Eq. (6a). We parametrize the set of solutions f to (6a) as:

$$f = f_1 + N_b f_0 \tag{7}$$

with $f_1 = J_b^{\top\dagger}\left(\dot{H}^* + mge_3\right)$, $N_b \in \mathbb{R}^{6n_c \times 6n_c}$ the projector into the null space of $J_b^\top$, and $f_0 \in \mathbb{R}^{6n_c}$ the wrench redundancy that does not influence $\dot{H}(f) = \dot{H}^*$. To determine the control torques that instantaneously realize the contact wrenches given by (7), we use the dynamic equations (1) along with the constraints (4), which yields:

$$\tau = \Lambda^\dagger(JM^{-1}(h - J^\top f) - \dot{J}\nu) + N_\Lambda\tau_0 \tag{8}$$

with $\Lambda = J_j M_j^{-1} \in \mathbb{R}^{6n_c \times n}$, $N_\Lambda \in \mathbb{R}^{n\times n}$ the projector onto the nullspace of Λ, the vector $h := C(q,\nu)\nu + G(q) \in \mathbb{R}^{n+6}$, and $\tau_0 \in \mathbb{R}^n$ a free variable.

B. Stability of the Zero Dynamics

The stability of the zero dynamics is attempted by means of a so called *postural task*, which exploits the free variable τ_0. Partition the vector h as follows: $h = (h_b, h_j)$, $h_b \in \mathbb{R}^6, h_j \in \mathbb{R}^n$. A choice of the postural task that ensures the stability of the zero dynamics on one foot is [19]:

$$\tau_0 = h_j - J_j^\top f + u_0 \tag{9}$$

where $u_0 := -K_p^j N_\Lambda M_j(q_j - q_j^d) - K_d^j N_\Lambda M_j \dot{q}_j$, and $K_p^j \in \mathbb{R}^{n\times n}$ and $K_d^j \in \mathbb{R}^{n\times n}$ two symmetric, positive definite matrices. An interesting property of the closed loop system (1)–(8)–(9)–(7) is stated in the following Lemma.

Lemma 1. *The closed loop joint space dynamics $\ddot{q}_j$ does not depend upon the wrench redundancy f_0.*

The proof of Lemma 1 is in the appendix. This result implies also that the linearization of the closed-loop joint dynamics does not depend on f_0, thus rendering the gain tuning procedure presented in this paper independent from the choice of the contact wrenches redundancy. This redundancy is exploited to minimize the joint torques τ in Eq. (8). In the language of *Optimization Theory*, we can rewrite the control strategy as follows:

$$f_0^* = \underset{f_0}{\operatorname{argmin}} |\tau^*(f_0)|^2 \tag{10a}$$

$$s.t.$$

$$Cf_0 < b \tag{10b}$$

$$\tau^*(f_0) = \underset{\tau}{\operatorname{argmin}} |\tau(f_0) - \tau_0(f_0)|^2 \tag{10c}$$

$$s.t.$$

$$\dot{J}(q,\nu)\nu + J(q)\dot{\nu} = 0 \tag{10d}$$

$$\dot{\nu} = M^{-1}(B\tau + J^\top(f_1 + N_b f_0) - h) \tag{10e}$$

$$\tau_0 = h_j - J_j^\top(f_1 + N_b f_0) + u_0. \tag{10f}$$

The constraints (10b) ensure the satisfaction of friction cones, normal contact surface forces, and center-of-pressure constraints. The control torques are then given by $\tau = \tau^*(f_0^*)$.

IV. GAIN TUNING PROCEDURE

A. Problem Statement

The goal is to impose desired local properties of the joint dynamics. The choice of focusing on the joint dynamics over other output functions reflects the aim of choosing stiffness and damping at the joint level, without perturbing the task hierarchy of momentum control and stability of the associated zero dynamics via postural control.

Assuming that (10b) is always satisfied, the control torques obtained by solving the optimization problem (10) depend only on the system state, i.e. $\tau = \tau(q,\nu)$. Since it is assumed that the robot stands on (at least) one foot, one can express the system state in terms of the joint position and velocity, i.e. $(q,\nu) = (q(q_j), \nu(q_j, \dot{q}_j))$. Then, the joint space dynamics depends only on joint position and velocity, i.e. $\ddot{q}_j = f(q_j, \dot{q}_j)$ and we can linearize this dynamics about an equilibrium point $(q_j^d, 0)$. The process of finding the linearized joint dynamics is similar to that presented in [19], which yields

$$\ddot{q}_j = -Q_1(q_j - q_j^d) - Q_2\dot{q}_j$$

where $Q_1, Q_2 \in \mathbb{R}^{n\times n}$ are given by

$$Q_1 = C_1(q_j^d)K_iC_2(q_j^d) + C_3(q_j^d)K_p^jC_4(q_j^d) \tag{11a}$$

$$Q_2 = C_1(q_j^d)K_pC_2(q_j^d) + C_3(q_j^d)K_d^jC_4(q_j^d) \tag{11b}$$

and $C_1 = M_j^{-1}\Lambda^\dagger J_b M_b^{-1}$, $C_2 = M_b J_b^\dagger J_j$, $C_3 = M_j^{-1}N_\Lambda$ and $C_4 = N_\Lambda M_j$. Now, let $x \in \mathbb{R}^{2n}$ be defined as follows $x := \begin{bmatrix} x_1^\top & x_2^\top \end{bmatrix}^\top = \begin{bmatrix} q_j^\top - q_j^{d\top} & \dot{q}_j^\top \end{bmatrix}^\top$. The linearized joint space dynamics around an equilibrium point $(q_j^d, 0)$ is given by

$$\dot{x} = \begin{bmatrix} \partial_{q_j}\dot{x}_1 & \partial_{\dot{q}_j}\dot{x}_1 \\ \partial_{q_j}\dot{x}_2 & \partial_{\dot{q}_j}\dot{x}_2 \end{bmatrix} x = \begin{bmatrix} 0_n & 1_n \\ -Q_1 & -Q_2 \end{bmatrix} x = Ax. \tag{12}$$

Then, the optimization problem we attempt at solving is stated next.

$$K_p^{j*}, K_d^{j*}, K_i^*, K_p^* = \underset{K_p^j, K_d^j, K_i, K_p}{\operatorname{argmin}} |A(K_p^j, K_d^j, K_i, K_p) - A^d|^2 \tag{13}$$

$$s.t.$$

$$K_i = K_i^\top > 0 \tag{14a}$$

$$K_p = K_p^\top > 0 \tag{14b}$$

$$K_p^j = K_p^{j\top} > 0 \tag{14c}$$

$$K_d^j = K_d^{j\top} > 0 \tag{14d}$$

where A^d is the desired state matrix of the following form: $A^d = \begin{bmatrix} 0_n & 1_n \\ -Q_1^d & -Q_2^d \end{bmatrix}$ with $Q_1^d, Q_2^d \in \mathbb{R}^{n\times n}$ the desired stiffness and damping matrices. The optimization problem (13)-(14) may be solved with any nonlinear available optimizer. Yet, finding numerical solutions to the optimization problem may be time consuming, which may forbid the on-line use of the optimizer when the desired stiffness and damping are time-varying. In this case, the solutions to (13)-(14) may also become discontinuous at some time instants. We propose below a method for solving on-line the problem (13) that provide continuous solutions for the control gains.

B. Solution to the unconstrained problem

Assume that the constraints (14) do not hold. When the robot stands on one foot, intuition would suggest that the joint space dynamics can be imposed at will, i.e. there always exist control gains such that the matrices Q_1^d, Q_2^d can be chosen arbitrarily. This section shows, however, that this is not possible because of the two strict stack-of-task control strategy defined in section III.

To show this, we prove that there exist some matrices Q_1^d such that no choice of the control gains renders $Q_1^d = Q_1(K_i, K_p^j)$ satisfied.

Now, if the constraints (14) do not hold, then the optimization problem (13) can be rewritten as

$$y_1 = \Gamma \begin{bmatrix} k_i \\ k_p^j \end{bmatrix} \tag{15a}$$

$$\Gamma = \left[\begin{bmatrix} C_2^\top \otimes C_1 \end{bmatrix} \quad \begin{bmatrix} C_4^\top \otimes C_3 \end{bmatrix} \right] \in \mathbb{R}^{n^2 \times (36+n^2)} \tag{15b}$$

where $y_1, k_p^j \in \mathbb{R}^{n^2}$, $k_i \in \mathbb{R}^{36}$ are the vectorization of matrices Q_1, K_p^j and K_i obtained by reordering their columns into a single column vector, and $\otimes$ the Kronecker product. Then, the following result holds.

Lemma 2. *Assume that* $rank(J) = 6n_c$, *and that* $n > 6n_c$. *Then, the matrix* Γ *is not full row rank, i.e.* $rank(\Gamma) < n^2$.

The proof of the above Lemma is in the Appendix. As a consequence of the above Lemma, there exist some desired matrices Q_1^d, i.e. y_1, such that no control gain implies the exact solution to $Q_1^d = Q_1(K_i, K_p^j)$. On the other hand, the least square solution to the problem (15a) is given by

$$\begin{bmatrix} k_i^* \\ k_p^{j*} \end{bmatrix} = \Gamma^\dagger y_1 \tag{16}$$

Clearly, the gains K_i^*, K_p^{j*} obtained by the above solution do not in general satisfy the symmetry and positive definiteness constraints (14). A similar procedure can be applied to find the gains K_d^{j*}, K_p^* that do not in general satisfy (14) but solve in the least square sense $Q_2^d = Q_2(K_d^{j*}, K_p^*)$.

C. Enforcing symmetry and positive definiteness constraints

In the previous section, we solved the problem (13) by assuming that the constraints (14) do not hold. Hence, we are now given with a set of gains $K_p^{j*}, K_d^{j*}, K_i^*, K_p^*$ that may not be symmetric and positive definite. Define $K^* := \{K_p^{j*}, K_d^{j*}, K_i^*, K_p^*\}$ a matrix of proper dimension. Then, to enforce symmetry and positive definiteness constraints, we solve (on-line) a second optimization problem for each of the unconstrained optimal gain. More precisely, the problem we solve follows:

$$O^*, L^* = \underset{(O,L)}{\operatorname{argmin}} |K^* - X(O,L)|^2 \tag{17}$$

$$\text{s.t.}$$

$$OO^\top = 1$$

$$L \quad \text{diagonal matrix.}$$

with $X := O^\top \exp(L)O$, O an orthogonal matrix, and L a diagonal matrix. The solution to the problem (17) are the matrices O^*, L^*. Therefore, the constrained optimized gain matrix is given by:

$$X^* = O^{*\top} \exp(L^*)O^* \tag{18}$$

Clearly, at this point we just moved the problem from solving the optimization problem (13) with the constraints (14) to solving the problems (17) with the constraints of the kind $OO^\top = 1$. Now, being O an orthogonal matrix, then $\dot{O} = OS(v)$, with v a vector of proper dimension depending on the dimension of O, and $S(\cdot)$ a skew symmetric matrix. Assuming v and $\dot{L}$ as exogenous control inputs, one can find Lyapunov-like solutions to the problem (17). More precisely, the following result holds.

Lemma 3. *Let* $O, L \in \mathbb{R}^{m \times m}$ *denote an orthogonal and a diagonal matrix, respectively. Consider the following system:*

$$\dot{L} = U \tag{19a}$$

$$\dot{O} = OS(v) \tag{19b}$$

where the vector $v \in \mathbb{R}^{m(m-1)/2}$ *and the diagonal matrix* U *are considered as exogenous control inputs. Define* $\tilde{K} = K^* - X(O, L)$, *and the operator* $S^{-1}(v)$ *such that* $v = S^{-1}(S(v))$. *Apply the control inputs*

$$U = -K_U \exp(-L)diag(B_1) \tag{20a}$$

$$v = K_v S^{-1} \left(\frac{B_2 - B_2^\top}{2} \right) \tag{20b}$$

to system (19), where K_U *is a positive definite diagonal matrix,* K_v *is a symmetric positive definite matrix,* $B_2 = O^\top \exp(L)OK^{*\top} - K^{*\top}O^\top \exp(L)O$, $B_1 = \exp(L) - OK^*O^\top$, *and diag$(B_1)$ defined as follows*

$$diag(B_1)_{(i,j)} = B_{1(i,j)} \quad \text{if } i = j$$
$$diag(B_1)_{(i,j)} = 0 \quad \text{if } i \neq j.$$

Then, the following results hold:

- *If* K^* *is symmetric and positive definite, the equilibrium point of the closed-loop dynamics* $\tilde{K} = 0$ *is stable;*
- *The system trajectories* $\tilde{K}(t)$ *are globally bounded for any* $K^* \in \mathbb{R}^{m \times m}$;
- $|\tilde{K}(t)| \leq |\tilde{K}(0)|$ *for any* $K^* \in \mathbb{R}^{m \times m}$.

The proof of Lemma 3 is in the Appendix. The above Lemma points out that the *distance* between the optimal, unconstrained solution K^* (obtained in Section IV-B) and the constrained (symmetric, positive definite) gain $X(O(t), L(t))$ is non increasing, i.e. $|\tilde{K}(t)| \leq |\tilde{K}(0)|$. Then, the control laws (20) can be viewed as a tracker for symmetric positive definite matrices even when the matrix has to track a non symmetric positive definite matrix (i.e. it does not belong to the same manifold). Let us remark that convergence of the tracking error $\tilde{K}$ to zero is not ensured *a priori*. Simulations we have performed, however, tend to show that the cases when $\tilde{K}$ does not converge to zero are limited, and the analysis on this convergence is currently being developed.

Note also that if the optimal, unconstrained solution K^* varies in time slowly, the tracker preserves its properties by continuity. Then, one may think of applying the solution (16) (20) on-line for time-varying desired stiffness and damping $Q_1^d(t), Q_2^d(t)$. Let us finally observe that we could have avoided to find the intermediate solution (16), and define the optimization problem (13) in terms of the parametrization $X(O, L)$, and then apply the procedure explained above to find time evolutions for the constrained gains. Simulations we have performed tend to show that this approach performs worse than the route we propose, and further investigations in this direction are being conducted.

D. Desired matrix correction when two feet balancing

If the constraint (3) acting on the system represents more than one robot frame fixed with respect to the inertial frame (e.g. two feet balancing), the matrices Q_1 and Q_2 in (11) may not be full rank. As a matter of fact, the minimal coordinates describing the constrained mechanical system are fewer than n in this case. Then, the ranks of the desired matrices Q_1^d and Q_2^d must be equal to those of the matrices Q_1 and Q_2. In general, the desired matrices Q_1^d and Q_2^d must be corrected according to the constraints acting on the system.

To do this, observe that the *feasible* joint accelerations according to the constraints (3) are given by:

$$\ddot{q}_j = -\overline{J}_j^\dagger \dot{\overline{J}}_j \dot{q}_j + N_J \ddot{q}_{j0}, \qquad (21)$$

where $\overline{J}_j = (1_{6n_c} - J_b J_b^\dagger)J_j$, $\dot{\overline{J}}_j$ is the time derivative of $\overline{J}_j$ and N_J is the projector onto the null space of $\overline{J}_j$. In the above equation, we have used $v_{\mathcal{B}} = -J_b^\dagger J_j \dot{q}_j$ and its derivative. Then, given two desired matrices Q_1^d, Q_2^d, we project them as follows

$$\begin{aligned} \overline{Q}_1^d &= N_J Q_1^d \\ \overline{Q}_2^d &= N_J Q_2^d. \end{aligned} \qquad (22)$$

to correct the ranks and structure of the desired stiffness and damping according to the constraints (3) when $n_c > 1$.

V. SIMULATIONS RESULTS

A. Simulation Environment

Simulations are performed on a 23 degree of freedom robot model representing the humanoid iCub. The simulation software is developed in MATLAB. The time evolution of the dynamical system is obtained through the integration of the system dynamics (1) subject to the constraints (4). We parametrize the orientation of the base frame using a quaternion representation $\mathcal{Q} \in \mathbb{R}^4$. The system state is then defined as: $\chi := (p_{\mathcal{B}}, \mathcal{Q}, q_j, \dot{p}_{\mathcal{B}}, \omega_{\mathcal{B}}, \dot{q}_j)$, and the time derivative is given by $\dot{\chi} = (\dot{p}_{\mathcal{B}}, \dot{\mathcal{Q}}, \dot{q}_j, \dot{\nu})$. The state is integrated through time by means of the numerical integrator MATLAB *ode15s*. For the purpose of this paper, both problems (16) (20) are solved offline, before starting the state integration. Online implementations of this tuning algorithm will be the subject of a forthcoming publication. To integrate the variables $\dot{L}, \dot{O}$, we use a fixed step integrator. The constraints (4), as well as $|\mathcal{Q}| = 1$ and $OO^\top = 1$, need to be enforced during the integration phase, and we added correction terms to $\dot{\mathcal{Q}}, \dot{O}$.
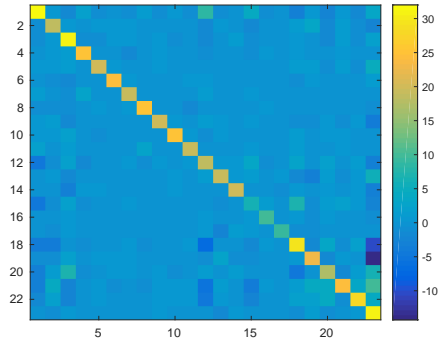

Fig. 1. Matrix Q_1 after the gain tuning procedure in case of one foot balancing. Simulations run in MATLAB environment.
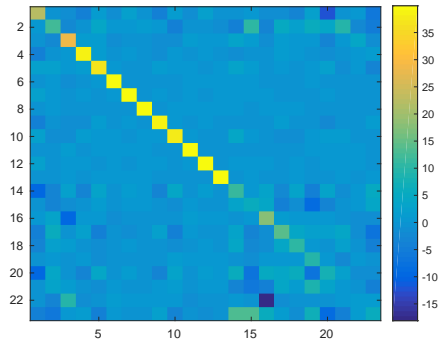

Fig. 2. Matrix Q_1 after the gain tuning procedure in case of two feet balancing. Simulations run in MATLAB environment.

B. Results

Simulations are performed for both the robot balancing on one foot and two feet. We choose Q_1^d to be positive definite and diagonal, and $Q_2^d = 2\sqrt{Q_1^d}$. In this case, the desired joint space dynamics aims at the following properties:

- The joint space dynamics be locally decoupled, i.e. each joint can be tuned separately;
- The linearized system is critically damped. This will avoid excessive overshoots in the joint space dynamics.

When the robot is balancing on two feet, the matrices Q_1^d, Q_2^d are corrected as in (22).

Figures 1-2 show the shape of matrix Q_1 after gain tuning. Observe that in the case of one foot balancing, the matrix Q_1 is close to a diagonal matrix, and this implies that the joint space dynamics is almost locally decoupled. In case of two feet balancing, it is interesting to observe the effectiveness of the gain tuning procedure by looking at the difference between the first 11 rows of Q_1, and the last 12 rows, which correspond to the closed chained formed by the legs.

To verify that the joint space dynamics is close to the desired dynamics, we evaluate the dynamical response of each joint to a step input. In particular, we focused on the settling time t_s. It is possible to approximate t_s as $t_s \approx \frac{-3}{\mathrm{Re}(\lambda)}$, where $\mathrm{Re}(\lambda)$ is the real part of system's eigenvalues. Figures 3-4 show the dynamics of torso pitch for both one foot and two feet balancing. In both cases, $t_s \approx t_s^d$, and the system dynamics is then closed to the desired dynamics.
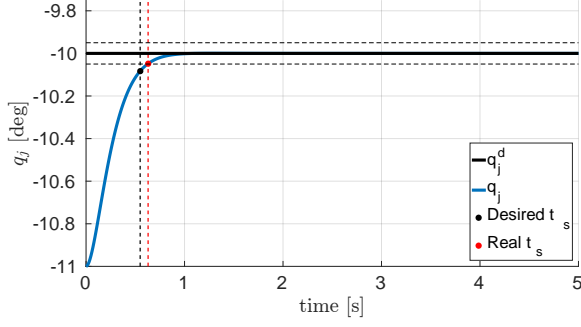
Fig. 3. Response of torso pitch to a step input in case of 1 foot balancing. The black dot indicates the desired settling time, while the red dot is the real settling time. The dashed horizontal lines indicate $\pm 5\%$ of the reference position. Simulations run in MATLAB environment.
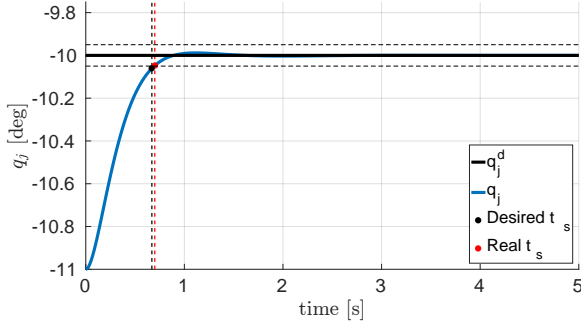

Fig. 4. Response of torso pitch to a step input in case of 2 feet balancing. In this case, the real and desired settling time are almost coincident. Simulations run in MATLAB environment.

VI. CONCLUSIONS

This paper has presented a gain tuning procedure for constrained floating base systems controlled through momentum-based control. The objective is the achievement of a desired local dynamics for the system's joint space. The optimization is performed on the linearization around an equilibrium point of the closed-loop system's joint space. The constraints on symmetry and positive definiteness of gain matrices are enforced thanks to a tracker for symmetric positive definite matrices. This allows fast resolutions of the constrained optimization problem, which allows one for on-line implementations of the presented algorithms. Simulation results on the humanoid robot iCub show the effectiveness of the gain tuning procedure for both the robot balancing on one foot and two feet.

Further improvements on the gain tuning procedure may be developed in future works. The gains optimization presented in this paper can be applied in different equilibrium points along a joint reference trajectory, and may be an efficient tuning strategy in case of humanoid walking. On line implementations of the presented algorithm on the real humanoid robot is being investigated.

APPENDIX

A. Proof of Lemma 1

Recall that M is block diagonal. The joint space dynamics is given by the last n rows of Eq. (1):

$$M_j \ddot{q}_j = J_j^\top f - h_j + \tau. \tag{23}$$

Moreover, we can rewrite the term $JM^{-1}(h - J^\top f)$ in the control torques equations Eq. (8) as follows:

$$JM^{-1}(h - J^\top f) = J_b M_b^{-1}(h_b - J_b^\top f) + \Lambda(h_j - J_j^\top f)$$

In view of $N_\Lambda = 1_n - \Lambda^\dagger \Lambda$ and (9), the control torques (8) become:

$$\tau = h_j - J_j^\top f + \Lambda^\dagger (J_b M_b^{-1}(h_b - J_b^\top f) - \dot{J}\nu) + N_\Lambda u_0 \tag{24}$$

Substituting Eq. (24) into Eq. (23) gives:

$$M_j \ddot{q}_j = \Lambda^\dagger (J_b M_b^{-1}(h_b - J_b^\top f) - \dot{J}\nu) + N_\Lambda u_0. \tag{25}$$

The only term which contains the wrenches f in Eq. (25) is multiplied by $J_b^\top$. Since $f = f_1 + N_b f_0$, and by definition $J_b^\top N_b = 0_{6n_c}$, we have that $J_b^\top f = J_b^\top f_1$. Hence, vector f_0 does not influence the joint space dynamics Eq. (25).

B. Proof of Lemma 2

Given two rectangular matrices A, B, recall the properties $\mathrm{rk}\,(AB) \leq \min(\mathrm{rk}\,(A), \mathrm{rk}\,(B))$ and $\mathrm{rk}\,(B \otimes A) = \mathrm{rk}\,(A)\,\mathrm{rk}\,(B)$, where $\mathrm{rk}\,(\cdot)$ denotes the rank of a matrix. We apply the above properties to evaluate the rank of the matrices C_1, C_2, C_3, C_4 in Eq. (11a). It is straightforward to verify that: $\mathrm{rk}(C_1) \leq 6$, $\mathrm{rk}(C_2) \leq 6$, $\mathrm{rk}(C_3) \leq n - 6n_c$, $\mathrm{rk}(C_4) \leq n - 6n_c$. It is now possible to evaluate the rank of matrix

$$\Gamma = \left[\begin{bmatrix} C_2^\top \otimes C_1 \end{bmatrix} \quad \begin{bmatrix} C_4^\top \otimes C_3 \end{bmatrix} \right] \in \mathbb{R}^{n^2 \times (n^2 + 36)},$$

i.e.

$$\mathrm{rk}(\Gamma) \leq \mathrm{rk}(C_2^\top \otimes C_1) + \mathrm{rk}(C_4^\top \otimes C_3) \leq 36 + (n - 6n_c)^2.$$

The condition for Γ to be full row rank is $\mathrm{rk}(\Gamma) = n^2$. This condition may be verified if $36 + (n - 6n_c)^2 = n^2$. Recall that n, n_c must be positive integers, and that $n > 6n_c$. Assume that $n = 6n_c + k$, with $k \in \mathbb{N}$. Then, one can verify that $36 + (n - 6n_c)^2 < n^2$ yields $36(n_c^2 - 1) + 12kn_c > 0$, which is always satisfied for any $n_c, k \in \mathbb{N}$. As a consequence, $\mathrm{rk}(\Gamma) < n^2$.

C. Proof of Lemma 3

Using Frobenius matrix norms, one has that $|\tilde{K}|^2 = |K^* - X|^2 = \mathrm{tr}\,((K^* - X)^\top (K^* - X))$, where $\mathrm{tr}(\cdot)$ denote the trace operator. Consider now the candidate Lyapunov function

$$V = |\tilde{K}|^2 = \mathrm{tr}\,((K^* - X(O, L))^\top (K^* - X(O, L))) \tag{26}$$

Observe that V is always positive, and $V = 0$ iif $\tilde{K} = 0$. Then, to prove the three statements in Lemma 3, it suffices to show that $\dot{V} \leq 0$. To do this, recall that O is an orthogonal

matrix, i.e. $OO^\top = 1$. Also, observe that Eq. (26) can be rewritten as:

$$V = \mathrm{tr}\left(K^* K^{*\top} - 2K^{*\top} O^\top \exp(L)O + \exp(2L)\right) \quad (27)$$

where we used the properties $\mathrm{tr}(K^{*\top} O^\top \exp(L)O) = \mathrm{tr}(O^\top \exp(L)OK^*)$ and $\mathrm{tr}(O^\top \exp(2L)O) = \mathrm{tr}(OO^\top \exp(2L))$. To compute the time derivative of V, recall that $\dot{O} = OS(v)$, with $S(v)$ a skew-symmetric matrix, and $\dot{L} = U$. Then, $\dot{V}$ becomes:

$$\dot{V} = 2\,\mathrm{tr}(B_1 \exp(L)U) + 2\,\mathrm{tr}(B_2 S(v)),$$

with $B_2 = O^\top \exp(L)OK^{*\top} - K^{*\top} O^\top \exp(L)O$, and $B_1 = \exp(L) - OK^* O^\top$. Observe that $\dot{V} \leq 0$ if both $\mathrm{tr}(B_1 \exp(L)U) \leq 0$ and $\mathrm{tr}(B_2 S(v)) \leq 0$. Now, note that $\mathrm{tr}(B_1 \exp(L)U)$ can be rewritten as $\sum_{i=1}^n e_i^\top B_1 \exp(L)Ue_i$. Since the product $\exp(L)U$ is diagonal, then $\sum_{i=1}^n e_i^\top B_1 \exp(L)Ue_i = \sum_{i=1}^n e_i^\top B_1 e_i \exp(l_i)u_i$, where we indicate with $\exp(l_i)u_i$ the i-th element along the diagonal of $\exp(L)U$. The trace can then be rewritten as $\sum_{i=1}^n e_i^\top B_1 e_i \exp(l_i)u_i = \sum_{i=1}^n b_{1i} \exp(l_i)u_i$ where b_{1i} is the i-th element along the diagonal of B_1. A possible choice of u_i that ensures $\mathrm{tr}(B_1 \exp(L)U) \leq 0$ is:

$$u_i = -k_{Ui} \exp(-l_i)b_{1i} \quad k_{Ui} > 0. \quad (28)$$

Since u_i is the i-th element along the diagonal of U, (28) implies $U = -K_U \exp(-L)\mathrm{diag}(B_1)$ with K_U a diagonal matrix of positive constants.

Now, recall that the matrix B_2 can be decomposed as follows:

$$B_2 = \frac{(B_2+B_2^\top)}{2} + S(\omega) \quad (29)$$

where $\omega = S^{-1}\left(\frac{(B_2-B_2^\top)}{2}\right)$. Recall also that the trace of a product between a symmetric and a skew-symmetric matrix is zero. Then, $\mathrm{tr}\left(\frac{(B_2+B_2^\top)}{2}S(v)\right) = 0$. We are now left to evaluate $\mathrm{tr}(S(\omega)S(v))$. The trace of a matrix product can also be written as $\mathrm{tr}(X^\top Y) = \mathrm{vec}(X)^\top \mathrm{vec}(Y)$, where $\mathrm{vec}(\cdot)$ is the vectorization operator. Then $\mathrm{tr}(S(\omega)S(v)) = -\mathrm{tr}(S(\omega)^\top S(v)) = -\mathrm{vec}((S(\omega))^\top \mathrm{vec}(S(v))$. Note that $\mathrm{vec}(S(x)) = Tx \ \forall x$, where the matrix T satisfies $\frac{T^\top T}{2} = 1$ due to the skew-symmetry of $S(\cdot)$. Hence,

$$\mathrm{tr}(S(\omega)^\top S(v)) = -\omega^\top T^\top Tv = -2\omega^\top v$$

and this suggests that a possible choice of v is $v = K_v S^{-1}(S(\omega)) = K_v S^{-1}\left(\frac{(B_2-B_2^\top)}{2}\right)$.

References

[1] T. Koolen, S. Bertrand, G. Thomas, 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," *International Journal of Humanoid Robotics*, vol. 13, p. 34, March 2016.

[2] M. Hopkins, R. Griffin, A. Leonessa, B. Lattimer, and T. Furukawa, "Design of a compliant bipedal walking controller for the darpa robotics challenge," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, Nov 2015, pp. 831–837.

[3] Q. Huang, K. Kaneko, K. Yokoi, S. Kajita, T. Kotoku, N. Koyachi, H. Arai, N. Imamura, K. Komoriya, and K. Tanie, "Balance control of a biped robot combining off-line pattern with real-time modification," *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000.

[4] J. E. Marsden and T. Ratiu, *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Springer Science & Business Media, 2013, vol. 17.

[5] R. Featherstone, *Rigid Body Dynamics Algorithms*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[6] J. A. Acosta and M. Lopez-Martinez, "Constructive feedback linearization of underactuated mechanical systems with 2-DOF," *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, 2005.

[7] P. Wensing and D. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3103–3109.

[8] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 2589–2594.

[9] B. Stephens and C. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 1248–1255.

[10] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 981–988.

[11] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "iCub whole-body control through force regulation on rigid noncoplanar contacts," *Frontiers in Robotics and AI*, vol. 2, no. 6, 2015.

[12] E. Farnioli, M. Gabiccini, and A. Bicchi, "Optimal contact force distribution for compliant humanoid robots in whole-body loco-manipulation tasks," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5675–5681, May 2015.

[13] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, vol. 33, no. 4, pp. 399–414, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9294-z

[14] M. Teshnehlab and K. Watanabe, "Self tuning of computed torque gains by using neural networks with flexible structures," *IEE Proceedings - Control Theory and Applications (Volume:141 , Issue: 4)*, Feb 2002.

[15] N. Aphiratsakun and M. Parnichkun, "Fuzzy based Gains Tuning of PD controller for joint position control of AIT Leg Exoskeleton-I (ALEX-I)," *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, Feb 2009.

[16] S. Mason, L. Righetti, and S. Schaal, "Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting," *2014 IEEE-RAS International Conference on Humanoid Robots*, Nov 2014.

[17] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic LQR tuning based on Gaussian process global optimization," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[18] D. Xing, C. G. Atkeson, J. Su, and B. J. Stephens, "Gain scheduled control of perturbed standing balance," *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[19] G. Nava, F. Romano, F. Nori, and D. Pucci, "Stability analysis and design of momentum-based controllers for humanoid robots," *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, Oct 2016.

[20] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*. Springer Publishing Company, Incorporated, 2010.

[21] S. Traversaro, D. Pucci, and F. Nori, "On the base frame choice in free-floating mechanical systems and its connection to centroidal dynamics," 2016. [Online]. Available: https://traversaro.github.io/preprints/changebase.pdf

[22] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, 2013.