BALANCE CONTROL OF A HUMANOID ROBOT BASED ON THE COOPERATIVE DUAL TASK-SPACE FRAMEWORK

Ana Christine de Oliveira*, Bruno Vilhena Adorno*

*Graduate Program in Electrical Engineering - Federal University of Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil

Emails: `anacol@ufmg.br`, `adorno@ufmg.br`

Abstract— This paper presents a technique for the modeling and balance control of humanoid robots based on dual quaternion algebra and the Cooperative Dual Task-Space Framework. A strategy for controlling the Center of Mass (CoM) position that takes into account the system's stability constraints is presented and validated in a realistic simulation. The results show that the presented control strategy is able to track a desired 3D trajectory for the CoM while ensuring the robot's balance, and may potentially be extended to perform bipedal locomotion.

Keywords— Humanoid robots, Center of Mass (CoM), CoM control, Dual quaternion, Cooperative Dual Task-Space (CDTS).

Resumo— Este artigo apresenta uma técnica de modelagem e controle de equilíbrio de robôs humanóides baseada na álgebra de quatérnios duais e no Espaço de Cooperação Dual. Uma estratégia para controlar a posição do Centro de Massa (CM), que leva em consideração as restrições de estabilidade do sistema, é apresentada e validada em um ambiente de simulação realístico. Os resultados obtidos mostram que esta estratégia de controle é capaz de realizar o rastreamento de uma trajetória 3D desejada para o CM, enquanto garante o equilíbrio do robô, e pode, potencialmente, ser estendida para a locomoção de robôs bípedes.

Palavras-chave— Robôs humanóides, Centro de Massa, Controle do Centro de Massa, Quatérnios duais, Espaço de Cooperação Dual.

1 Introduction

The autonomous locomotion of legged robots, such as humanoids, is a very challenging subject. During the locomotion, some dynamic constraints must be satisfied to guarantee a balanced motion. However, this class of robots has a large number of Degrees of Freedom (DOF), and obtaining a dynamic model in this case requires extensive calculations. To avoid this, a simplified model can be used to represent the system and generate balanced trajectories (Kajita et al., 2003; Kim, 2007). This approach is often used to generate walking patterns, however the motion still must be controlled to avoid disturbances and to guarantee stability.

The robot's Center of Mass (CoM) is directly related to the robot's balance, and its control is of great interest to design a balanced walking motion. As a consequence, the CoM control is a problem widely treated in the literature. Some works focus on the robot stabilization during the walking motion, in which the walking pattern parameters are manipulated in order to adjust the CoM to its reference trajectory. Hof (2008) proposed a new concept named "extrapolated center of mass" (XcoM), which combines the CoM projection onto the ground and its velocity. In addition to disturbance rejection, Graf (2010) eliminated the double support phase—i.e., the one where both feet are on the ground—to obtain a faster walking motion, planning the next swing-foot trajectory regarding the current CoM. Missura and Behnke (2011) proposed a control method capable of rejecting lateral disturbances in the CoM trajectory during the walking motion.

Other approaches addressed the problem of balance control of standing robots through the CoM manipulation. The goal of those approaches is to lead the CoM projection onto the ground into the robot's support polygon, which are stable regions characterized by the convex hull of the feet contact-points with the ground. Hofmann et al. (2004) adopted a strategy based on a PD controller and the robot's dynamic model that allows the stabilization of a robot in unstable initial conditions, through the control of both CoM, robot's orientation, and swing-foot pose. The controller maneuver the robot's limbs that are not in contact with the ground, excluding the upper-body limbs. Stephens (2007) combined two strategies: the ankle strategy and the hip strategy. The first one designs a controller that keeps all joints torques neglected, except the ankle joint. As this behavior can lead the CoM to unstable regions, the hip strategy is applied, which consists of maneuvering the hip angles in order to incline the torso and lead the CoM to the desired target. This strategy also does not takes the upper-body limbs into consideration. Cotton et al. (2009), on the other hand, used a concept defined as "Statically Equivalent Serial Chain" to obtain the CoM Jacobian matrix relating the joint velocities and the CoM linear velocity. The aim of that work was to build the CoM Jacobian without the robot's dynamic parameters, in order to control its CoM. To do that, the joint angles for some stable configurations of the robot are recorded and used in a system of linear equations to obtain the parameters of the CoM Jacobian. Boulic et al. (1995) and

Phillips and Badler (1991) also solved the balance control of standing configurations, but instead of controlling specifically biped robots, they focused on posture optimization of articulated bodies in general. More specifically, Boulic et al. (1995) introduced the concept of augmented bodies and used the pseudoinverse of the CoM Jacobian in the control strategy, whereas Phillips and Badler (1991) considered the robot's CoM attached to the torso's lower part and used inverse kinematics to control the approximated CoM.

All these strategies are very useful for the purpose they were designed for, but none of them is flexible enough in order to control the CoM *both* in walking motion and in standing configurations. The solution of this problem using the whole-body was addressed by Choi et al. (2007), which used the pseudoinverse of the CoM Jacobian matrix, obtained by a weighted sum of the CoM Jacobian matrix of each link, in the CoM control strategy. Sentis et al. (2010) solved a slightly different problem—the control of multicontact compliant tasks— while the balance constraints are satisfied.

The aforementioned balance control techniques rely on a suitable model, which may be intrinsically complex for humanoid robots, mainly due to their high number of DOF and also because of their hybrid kinematic structure. More specifically, the two legs and the two arms may be regarded as two parallel kinematic chains, respectively, whereas they are serially coupled through the torso. Regarding the kinematic modeling, humanoid robots are, generally, modeled as a free-floating point with attached limbs in contact with the ground. The kinematic model of each one of these limbs is then obtained separately, in general using the Denavit-Hartenberg convention. Toscano et al. (2014) proposed a method using the Screw Theory to obtain the kinematic model of the robot's limbs, including virtual joints within the model in order to apply the Davie's Method to solve the inverse kinematics of the robot. Park and Lee (2013) decoupled the robot's body in upper and lower body, and used the approach proposed by Adorno et al. (2010), named Cooperative Dual Task-Space (CDTS), to model the cooperation between the robot's legs.

In this paper we present a modeling method for humanoid robots by using dual quaternion (DQ) algebra, and we also implement a balance control strategy based on the work of Park and Lee (2013), which uses the CDTS together with the CoM control.

The remainder of this paper is organized as follows. Section 2 presents the mathematical background, whereas Section 4 presents the modeling of a humanoid robot using DQs and the CDTS. In Section 5, the balance control strategy is described. Finally, Section 6 presents the results, and Section 7 concludes the paper and presents a suggestion of future works.

2 Mathematical Background

According to Adorno (2011), dual quaternion (DQ) algebra has gained popularity in the last two decades within the context of kinematics, robotics, and control systems. A specific class of DQs, the ones with unit norm, is particularly interesting because it represents rigid motions in a very compact way without suffering from representational singularities. For instance, DQs are more compact than homogeneous transformation matrices (HTM), since the former has only eight parameters whereas the latter has twelve. In addition, DQ multiplications are less expensive than HTM multiplications (Adorno, 2011). Also, the DQ coefficients can be directly used in a control law, differently from HTMs.

This section reviews the basic concepts and definitions about dual quaternions and establishes the basic notation that will be used throughout the rest of the paper. The concepts and notations presented herein are based on the work of Adorno (2011).

Quaternions, which can be regarded as an extension of complex numbers, are defined as $\boldsymbol{h} \triangleq h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$, where $h_1, h_2, h_3, h_4 \in \mathbb{R}$, and $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ are imaginary units that obey the properties

$$\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1. \tag{1}$$

The conjugate of $\boldsymbol{h}$ is $\boldsymbol{h}^* \triangleq h_1 - \left(h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}\right)$. Complex numbers are a particular case of quaternions by letting $h_3 = h_4 = 0$.

While quaternions are primarily known to represent rotations (Kuipers, 1999), they may also be used to represent translations (Adorno, 2011). For instance, a translation $\begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$ is represented by the quaternion $\boldsymbol{t} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k}$. On the other hand, a rotation of an angle ϕ around the unit norm rotation axis $\begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$ is represented by the unit-norm quaternion $\boldsymbol{r} = \cos(\phi/2) + \boldsymbol{n}\sin(\phi/2)$, where $\boldsymbol{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$.

DQs are composed of two quaternions in addition to the dual unit ε, which is nilpotent (Selig, 2005); that is,

$$\varepsilon \neq 0, \text{ but } \varepsilon^2 = 0. \tag{2}$$

More specifically, a DQ is defined as $\underline{\boldsymbol{h}} \triangleq h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k} + \varepsilon\left(h_5 + h_6\hat{\imath} + h_7\hat{\jmath} + h_8\hat{k}\right)$, where $h_1, \ldots, h_8 \in \mathbb{R}$. We also define $\mathcal{P}(\underline{\boldsymbol{h}}) \triangleq h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$ and $\mathcal{D}(\underline{\boldsymbol{h}}) \triangleq h_5 + h_6\hat{\imath} + h_7\hat{\jmath} + h_8\hat{k}$, which are the primary and the dual part of $\underline{\boldsymbol{h}}$, respectively. The conjugate of $\underline{\boldsymbol{h}}$ is $\underline{\boldsymbol{h}}^* = \mathcal{P}(\underline{\boldsymbol{h}})^* + \varepsilon\mathcal{D}(\underline{\boldsymbol{h}})^*$. The multiplication between DQs behave the same way

as the multiplication between complex numbers; indeed, we must only respect the axioms (1) and (2) of the algebra. (From (1), it is easy to see that the imaginary units do not commute, hence DQ multiplication is not commutative.)

We also define an operator to map a DQ into an eight-dimensional vector:[1]

$$\text{vec}_8 \, \underline{\boldsymbol{h}} \triangleq \begin{bmatrix} h_1 & \cdots & h_8 \end{bmatrix}^T. \tag{3}$$

Furthermore, given the DQ multiplication $\underline{\boldsymbol{c}} = \underline{\boldsymbol{ab}}$, the Hamilton operators $\overset{+}{\boldsymbol{H}}(\cdot)$ and $\overset{-}{\boldsymbol{H}}(\cdot)$ are the matrices that satisfy the following relation (Adorno, 2011):

$$\text{vec}_8 \, \underline{\boldsymbol{c}} = \overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{a}}) \, \text{vec}_8 \, \underline{\boldsymbol{b}} = \overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{b}}) \, \text{vec}_8 \, \underline{\boldsymbol{a}}. \tag{4}$$

Rigid motions may be represented by DQs, which describe simultaneously position and orientation. More specifically, given the translation quaternion $\boldsymbol{t}$ and the unit-norm rotation quaternion $\boldsymbol{r}$, the unit DQ that represents the rigid motion is given by $\underline{\boldsymbol{x}} = \boldsymbol{r} + \varepsilon(1/2)\boldsymbol{tr}$ (Selig, 2005).

Since a rigid motion represents a transformation between different frames, in this paper we adopt the following convention: a rigid motion from frame $\mathcal{F}_a$ to frame $\mathcal{F}_b$ is represented by $\underline{\boldsymbol{x}}_b^a$. A sequence of rigid motions is represented by a multiplication of DQs. For instance, a rigid motion from $\mathcal{F}_a$ to $\mathcal{F}_b$, followed by a rigid motion from frame $\mathcal{F}_b$ to $\mathcal{F}_c$, is represented by $\underline{\boldsymbol{x}}_c^a = \underline{\boldsymbol{x}}_b^a \underline{\boldsymbol{x}}_c^b$.

3 Cooperative Dual Task-Space

Consider a serial kinematic chain with n joints (e.g., humanoid arms or humanoid legs), where one extremity is the base and the other one is the end-effector (e.g., hands in case of arms and feet in case of legs). The forward kinematics model (FKM) provides the mapping between the vector $\theta = \begin{bmatrix} \theta_1 & \cdots & \theta_n \end{bmatrix}^T$ of joint angles and the DQ $\underline{\boldsymbol{x}}$ representing the pose of the end effector; that is, $\underline{\boldsymbol{x}} = \underline{\boldsymbol{f}}(\theta)$. The differential FKM (DFKM) is given by $\text{vec}_8 \, \underline{\dot{\boldsymbol{x}}} = J\dot{\theta}$, where $J = \partial \, \text{vec}_8 \, \underline{\boldsymbol{x}}/\partial \theta$ is the system's Jacobian matrix. Both FKM and DFKM can be found for any serial kinematic chain by using dual quaternion algebra (Adorno, 2011).

The CDTS was proposed by Adorno et al. (2010) to represent the geometrical relationships between two kinematic chains using dual quaternion algebra. In this framework, the state of the system is completely determined by two variables: the relative pose $\underline{\boldsymbol{x}}_r$ and the absolute pose $\underline{\boldsymbol{x}}_a$. The relative pose $\underline{\boldsymbol{x}}_r$ provides the relationship between the poses of two chain extremities—denoted by $\underline{\boldsymbol{x}}_1$ and $\underline{\boldsymbol{x}}_2$—, and $\underline{\boldsymbol{x}}_a$ is an intermediate pose between them. These variables, shown in Fig. 1, are called "cooperative variables", as they are often used in

---

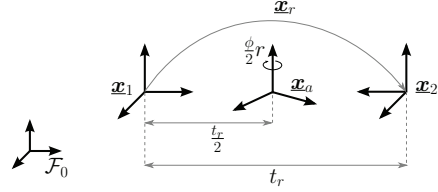[1]We also define the vec_4 operator in an analogous way for quaternions.

Figure 1: Cooperative variables of two chains.

cooperative systems, and are given by (Adorno et al., 2010)

$$\underline{\boldsymbol{x}}_r \triangleq \underline{\boldsymbol{x}}_1^* \underline{\boldsymbol{x}}_2, \tag{5}$$

$$\underline{\boldsymbol{x}}_a \triangleq \underline{\boldsymbol{x}}_1 \underline{\boldsymbol{x}}_r^{\{1/2\}}, \tag{6}$$

where $\underline{\boldsymbol{x}}_r^{\{1/2\}}$ represents a rigid motion given by half of the translation and half of the rotation of $\underline{\boldsymbol{x}}_r$. More specifically, for $\underline{\boldsymbol{x}}_r = \boldsymbol{r}_r + \varepsilon(1/2)\boldsymbol{t}_r\boldsymbol{r}_r$, with $\boldsymbol{r}_r = \cos(\phi_r/2) + \boldsymbol{n}_r \sin(\phi_r/2)$, the relative pose is given by (Adorno et al., 2010)

$$\underline{\boldsymbol{x}}_r^{\{1/2\}} \triangleq \boldsymbol{r}_r^{\{1/2\}} + \frac{1}{4}\varepsilon \boldsymbol{t}_r \boldsymbol{r}_r^{\{1/2\}}, \tag{7}$$

where $\boldsymbol{r}_r^{\{1/2\}} \triangleq \cos(\phi_r/4) + \boldsymbol{n}_r \sin(\phi_r/4)$.

The main advantages of using the CDTS is that the coordination between two kinematic chains working cooperatively can be completely described using only two variables. Moreover, the cooperative variables explicitly take into account the kinematic constraints of the system, which are imposed by the task.

Deriving (5) and using the vec_8 operator, defined in (3), we obtain

$$\text{vec}_8 \, \underline{\dot{\boldsymbol{x}}}_r = J_r \dot{q}, \tag{8}$$

where $q = \begin{bmatrix} q_1^T & q_2^T \end{bmatrix}^T$ corresponds to the joints values of the first and second kinematics chains, and $J_r = \begin{bmatrix} \overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_2) \, C_8 J_1 & \overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_1^*) \, J_2 \end{bmatrix}$ is the Jacobian matrix for the relative variable, with J_1 and J_2 being the Jacobian matrix of the first and second kinematic chains, respectively, and C_8 is the conjugating matrix (Adorno et al., 2010).

Likewise, deriving (6) and using the vec_8 operator, we obtain

$$\text{vec}_8 \, \underline{\dot{\boldsymbol{x}}}_a = J_a \dot{q},$$

with $J_a = \begin{bmatrix} \overset{-}{\boldsymbol{H}}\left(\underline{\boldsymbol{x}}_r^{\{1/2\}}\right) J_1 & 0_{8 \times n_2} \end{bmatrix} + \overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_1) \, J_{r/2}$ being the Jacobian matrix for the absolute variable, n_2 is the dimension of vector q_2, and $J_{r/2}$ satisfies the relation $d\left(\underline{\boldsymbol{x}}_r^{\{1/2\}}\right)/dt = J_{r/2}\dot{q}$ (Adorno et al., 2010).

In the next section, the CDTS will be used to model a biped robot.

4 Modeling

This section describes the kinematic modeling of a humanoid robot, using the dual quaternion theory and the CDTS. The modeling is decoupled in
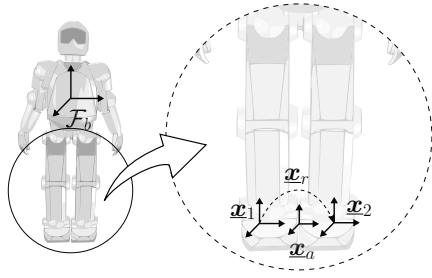
Figure 2: Cooperative variables of the robot's legs.

two parts: a) geometrical relationship between the robot's feet and b) influence of the joints velocities in the robot's CoM linear velocity.

4.1 Feet coordination

Consider the coordinate systems related to each foot, represented by the dual quaternions $\underline{\boldsymbol{x}}_1$ and $\underline{\boldsymbol{x}}_2$, as shown in Fig. 2.

Because the forces and torques actuating on the system are not considered by the kinematic model, the choice of the reference frame may have influence on the control output, since a poor choice may yield movements that violate the kinematic constraints of balance, i.e. which consists of keeping the supporting-foot (SF) fixed on the ground and the CoM projection onto the ground into the robot's support polygon. Therefore, to ensure that this constraints will be respected, the reference frame must be attached to the SF, in the single-support cases, and in any of the feet, in the double-support cases.

The robot's body orientation may be captured by the absolute variable of the robot's legs referenced in a frame attached to its upper-body, represented in Fig. 2 as $\mathcal{F}_b$. Since the poses of the robot's feet are known in $\mathcal{F}_b$ (i.e. $\underline{\boldsymbol{x}}_1^b$ and $\underline{\boldsymbol{x}}_2^b$), and that the reference frame is attached to the right leg, $\underline{\boldsymbol{x}}_a^b$ is given by

$$\underline{\boldsymbol{x}}_a^b = \underline{\boldsymbol{x}}_1^b \underline{\boldsymbol{x}}_a^1, \tag{9}$$

where $\underline{\boldsymbol{x}}_a^1$ is given by (6).

The primary part of $\underline{\boldsymbol{x}}_a^b$ captures the body orientation, and $\underline{\boldsymbol{x}}_r$—which is given by (5)—captures the position and orientation of the swing foot regarding the SF. It will be used later in the control strategy.

4.2 Center of Mass

Consider a humanoid robot with n limbs, each one with $k(i)$ links. The method adopted to model the influence of the joints velocities in the CoM linear velocity is similar to the proposed by Choi et al. (2007). It consists of obtaining equivalent kinematic models for the CoM of each link. Assuming that the links' CoM are known in the reference frame, there is a FKM for each link CoM such

that $\boldsymbol{c}_\lambda = f_c(q_\lambda)$, where q_λ represents the angles of the joints that have influence on link λ.

The global robot's CoM $\boldsymbol{c}$ is computed according to

$$\boldsymbol{c} = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{k(i)} m_{ij} \boldsymbol{c}_{ij}, \tag{10}$$

where M represents the robot's total mass, m_{ij} is the mass of link j in limb i, and $\boldsymbol{c}_{ij}$ is the CoM of this same link. The DFKM of the global CoM is given by

$$\text{vec}_4 \, \dot{\boldsymbol{c}} = J_{com} \dot{q}_{wb}, \tag{11}$$

with $q_{wb} = \begin{bmatrix} q_1^T & q_2^T & q_3^T & q_4^T \end{bmatrix}^T$, where q_1, q_2, q_3 and q_4 represent the joints angles of the right and left arm and the right and left leg, respectively, and

$$J_{com} = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{k(i)} m_{ij} J_{ij},$$

where J_{ij} is the Jacobian matrix relating the whole-body joints velocities and the CoM linear velocity of link j in the limb i. It is possible to decouple the matrix J_{com} in two parts, i.e. $J_{com} = \begin{bmatrix} J_{com_{UB}} & J_{com_{LB}} \end{bmatrix}$, where $J_{com_{UB}}$ is related to the arms' joints and $J_{com_{LB}}$ to the legs' joints.

5 Control Strategy

There are two stability conditions that must be satisfied to guarantee the robot's balance: 1) the CoM projection must be kept within the support polygon—i.e., the convex hull of the feet's contact-points with the ground—, and 2) the SF must remain stationary on the ground. Furthermore, to avoid fluctuations of the angular momentum about the CoM, which may drive the CoM to unstable regions, the upper-body must be kept vertically oriented during the motion.

The CoM and feet reference trajectories are defined a priori to satisfy the first condition. In order to fulfill the second condition, the reference frame is attached to the SF, as mentioned in Section 4. Thus, the control strategy must satisfy, simultaneously, three objectives: a) CoM tracking, b) swing-foot pose tracking and c) upper-body orientation regulation.

Because the reference frame is attached to the SF, the pose tracking of the swing-foot is equivalent to the pose tracking of the relative variable $\underline{\boldsymbol{x}}_r$. Furthermore, the upper-body orientation regulation can be regarded as the regulation of the absolute variable orientation $\mathcal{P}\left(\underline{\boldsymbol{x}}_a^b\right)$ from (9).

The control strategy that considers only actuation on the legs joints is given by

$$\dot{q}_l = J^+ K e, \tag{12}$$

where $q_l = \begin{bmatrix} q_3^T & q_4^T \end{bmatrix}^T$, K is a positive definite gain matrix, e is the error between the reference and

current values, i.e.,

$$e = \begin{bmatrix} \text{vec}_8 \left(\underline{\boldsymbol{x}}_{r_{REF}} - \underline{\boldsymbol{x}}_{r_{CURR}} \right) \\ \text{vec}_4 \left(\mathcal{P} \left(\underline{\boldsymbol{x}}_{a_{REF}}^b \right) - \mathcal{P} \left(\underline{\boldsymbol{x}}_{a_{CURR}}^b \right) \right) \\ \text{vec}_4 \left(\boldsymbol{c}_{ref} - \boldsymbol{c}_{curr} \right) \end{bmatrix}, \quad (13)$$

and J^+ is the right pseudoinverse of $J = \begin{bmatrix} J_r^T & J_{\mathcal{P}a}^T & J_{com_{LB}}^T \end{bmatrix}^T$, in which $J_{\mathcal{P}a}$ represents the four upper rows of J_a.

6 Results

The system was simulated using the softwares Matlab and V-Rep (Rohmer et al. (2013)) with the humanoid robot ASTI, which has 18 joints. The simulations were executed under the following specifications: processor Intel Core I3 CPU M380 @ 2.53GHz and 4,00GB Memory RAM, operating system Windows 7 64-bit, Matlab version R2012a 64-bit and V-Rep version PRO EDU 3.1.3 32-bit.

To verify the influence of the cooperative variables in the robot balance control, both $\underline{\boldsymbol{x}}_r$ and $\mathcal{P} \left(\underline{\boldsymbol{x}}_a^b \right)$ were removed from the control strategy. This simulation is labeled as "Pure CoM Control", while the simulation using the complete control strategy (12)–(13) is labeled as "CoM and Feet Control."

In all simulations, the feet are kept stationary, and some trajectories for the CoM were defined within the support polygon—which is known a priori, because the feet poses do not change—to validate the robot's model and to evaluate the control strategy. Eight different trajectories were tested to observe the robot's stability close to the support-polygon edges, as well as to check the occurrence of self collisions during the motion. Fig. 3 shows the results obtained for two trajectories. The evolution of the coefficients of $\underline{\boldsymbol{x}}_r$ and $\mathcal{P} \left(\underline{\boldsymbol{x}}_a^b \right)$, for a circular trajectory, are shown in Fig. 4.

The robot's actual CoM trajectory does not follow the given reference in the "Pure CoM Control." This behavior is expected since the stability constraints were not taken into consideration in this control strategy. On the other hand, in the "CoM and Feet Control," the CoM tracking is successfully performed, as well as the regulation of the body orientation and feet poses.

A tree-dimensional CoM tracking was also tested. The system's performance is depicted in Fig. 5.

Fig. 5b shows that the actual CoM trajectory follows the reference with an error in the order of millimeters, and a phase delay between the reference and the actual trajectory can be noticed. The delay is an expected behavior, because the control strategy designed does not have a feedforward component to nullify it. In spite of these deviations in the controlled motion, the system keeps its balance and the control objectives are accomplished.

(a) Sinusoidal trajectory. (b) Circular trajectory.
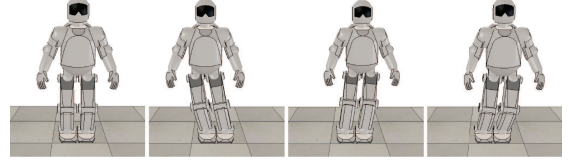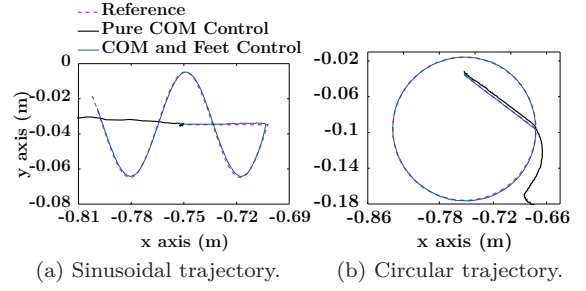
(c) Screen captures of the robot during the circular trajectory.

Figure 3: The charts at the top show a comparative result, where the *dashed* line represents the reference CoM trajectory, the *thick black* line represents the actual CoM trajectory for the "Pure CoM Control" and the *thin blue* line represents the actual CoM trajectory for the "CoM and Feet Control." The figure at the bottom shows four screen captures of the robot's motion during the execution of the circular trajectory in the "CoM and Feet Control."

Fig. 4 shows that the constraints are respected during the entire motion for the "CoM and Feet Control," which is also confirmed by the screen captures in Fig 3c and 5c , where the upper-body is vertically oriented and the feet are stationary.

7 Conclusion

In this paper, a modeling method for humanoid robots using dual quaternion algebra was presented, and a balance control strategy based on the CDTS together with the CoM position control was shown and validated in simulation us-

(a) Coefficients of $\mathcal{P} \left(\underline{\boldsymbol{x}}_a^b \right)$. (b) Coefficients of $\underline{\boldsymbol{x}}_r$.
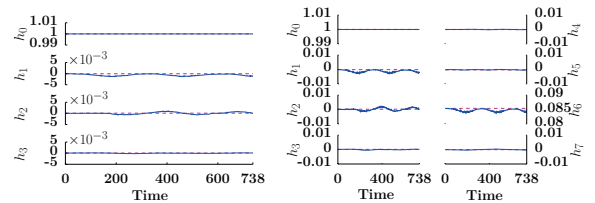
Figure 4: The figure on the *left* shows the evolution of the primary part coefficients of the absolute variable, and the figure on the *right* shows the evolution of the relative variable coefficients. The *dashed* line represents the reference, and the *thin blue* line represents the actual CoM trajectory for the "CoM and Feet Control."
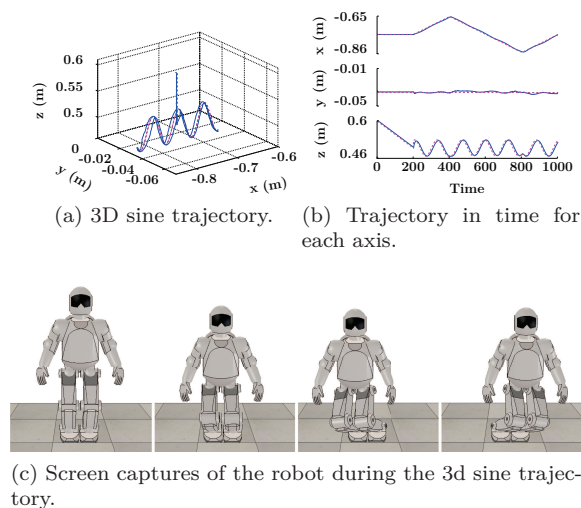
(a) 3D sine trajectory.

(b) Trajectory in time for each axis.

(c) Screen captures of the robot during the 3d sine trajectory.

Figure 5: CoM trajectory for a sinusoidal reference in xz-plane (*top left*) using the complete control strategy, where the *dashed* curve represents the reference and the *solid* curve represents the actual CoM; and the decomposed trajectory in each axis (*top right*). The figure at the *bottom* shows four screen captures of the robot's motion in the "CoM and Feet Control."

ing V-Rep and Matlab. The full control strategy (i.e., which takes into account both CoM control and balance constraints) was tested and compared with the case when only the CoM position control is active, without the balance control. The results show that the full controller keeps the robot balanced in a standing configuration, while the CoM position is successfully tracked. Without the balance control, the CoM position tracking is not achieved. Future works will focus on the application of the modeling method and the control strategy presented in this paper to control the locomotion of a humanoid robot under disturbances.

Acknowledgments

References

Adorno, B. V. (2011). *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration.* PhD thesis.

Adorno, B. V., Fraisse, P., and Druon, S. S. (2010). Dual position control strategies using the cooperative dual task-space framework. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 3955–3960, Taipei. Ieee.

Boulic, R., Mas, R., and Thalmann, D. (1995). Inverse Kinetics for Center of Mass Position Control and Posture Optimization. In *Image Processing for Broadcast and Video Production.*, number 1, pages 234–249. Springer, London.

Choi, Y., Kim, D., Oh, Y., and You, B.-j. J. (2007). Posture/walking control for humanoid robot based on kinematic resolution of CoM Jacobian with embedded motion. *IEEE Transactions on Robotics*, 23(6):1285–1293.

Cotton, S., Murray, A. A., and Fraisse, P. (2009). Estimation of the center of mass using statically equivalent serial chain modeling. In *ASME 2009 IDETC/CIE*, pages 1–7.

Graf, C. (2010). A Closed-loop 3D-LIPM Gait for the RoboCup Standard Platform League Humanoid. pages 18–22.

Hof, A. L. (2008). The 'extrapolated center of mass' concept suggests a simple control of balance in walking. *Human Movement Science*, 27(1):112–25.

Hofmann, a., Massaquoi, S., Popovic, M., and Herr, H. (2004). A sliding controller for bipedal balancing using integrated movement of contact and noncontact limbs. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, volume 2, pages 1952–1959.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Intl. Conference on Robotics and Automation*, pages 1620–1626. Ieee.

Kim, J.-h. (2007). Walking pattern generation of a biped walking robot using convolution sum. In *IEEE-RAS Intl. Conference on Humanoid Robots*, pages 539–544. IEEE.

Kuipers, J. (1999). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality.* Princeton University Press.

Missura, M. and Behnke, S. (2011). Lateral capture steps for bipedal walking. In *IEEE-RAS Intl. Conference on Humanoid Robots*, number October.

Park, H. A. and Lee, C. S. G. (2013). Cooperative-Dual-Task-Space-based whole-body motion balancing for humanoid robots. In *Proceedings - IEEE Intl. Conference on Robotics and Automation*.

Phillips, C. B. and Badler, N. I. (1991). Interactive behaviors for bipedal articulated figures.

Rohmer, E., Singh, S. P. N., and Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *IEEE Intl. Conference on Intelligent Robots and Systems*, pages 1321–1326.

Selig, J. M. (2005). *Geometric fundamentals of robotics.* Springer-Verlag New York Inc., 2nd edition.

Sentis, L., Park, J., and Khatib, O. (2010). Compliant Control of Multicontact and Center-of-Mass Behaviors in Humanoid Robots. *IEEE Transactions on Robotics*, 26(3):483–501.

Stephens, B. (2007). Integral control of humanoid balance. In *IEEE Intl. Conference on Intelligent Robots and Systems*, pages 4020–4027. Ieee.

Toscano, G. S., Simas, H., and Castelan, E. B. (2014). Screw-based modeling of a humanoid biped robot. In *Anais do XX Congresso Brasileiro de Automática*, pages 944–951, Belo Horizonte.