

# Position and Attitude Control of an Underactuated Flying Humanoid Robot

Gabriele Nava, Luca Fiorio, Silvio Traversaro, Daniele Pucci<sup>1</sup>

**Abstract**—This paper proposes a control strategy for the stabilization of a jet-powered flying humanoid robot. In particular, the contribution of the paper concerns the design of a control framework capable of tracking a desired robot position and orientation trajectory while flying. Asymptotic stability of the closed loop system is shown by means of a Lyapunov analysis. Simulations are carried out on a model of the humanoid robot iCub to verify the soundness of the proposed approach.

## I. INTRODUCTION

The recent development of complex aerial systems such as *aerial manipulators* [1] calls for the design of *whole-body* control laws capable of stabilizing the overall system dynamics while guaranteeing robustness with respect to external disturbances and modeling errors [2],[3],[4]. This paper takes a step in this direction by developing a whole-body control framework for stabilizing position and *attitude*, i.e. the overall orientation in space, of a jet-powered flying humanoid robot.

From the modeling and control point of view, the dynamics of a complex aerial system is, at least theoretically, not far from the dynamics of other multi-body systems, for example a humanoid robot, subject to a set of external actuation forces. Differently from the other aerial systems, a jet-powered humanoid robot would also be capable to perform *legged locomotion*, that may be more energetically efficient than flying especially when operating in confined spaces. On the other hand, a humanoid robot that is endowed with the capability of flying may have an advantage with respect to other walking robots in terms of speed of travel and energy consumption on rough terrain or when the operating area is vast. For this reason we believe there may be a technological benefit in developing a single platform capable to both fly and walk.

There is a wide and rich literature concerning the control of aerial systems [5],[6]. Researchers focused in particular on ensuring stability properties and robustness while performing manipulation tasks or in case of a propeller failure [7],[8]. One possibility is to apply linear control techniques based on linear approximation of the system's dynamics, but on the other hand nonlinear controllers can guarantee more robustness to external disturbances and unmodeled phenomena, especially in presence of highly nonlinear dynamical systems [9],[10]. Another widely used control strategy is the so-called *vectored-thrust* paradigm, that exploits the coupling between the vehicle's position and attitude dynamics. In particular, the robot's attitude together with the thrust force magnitude

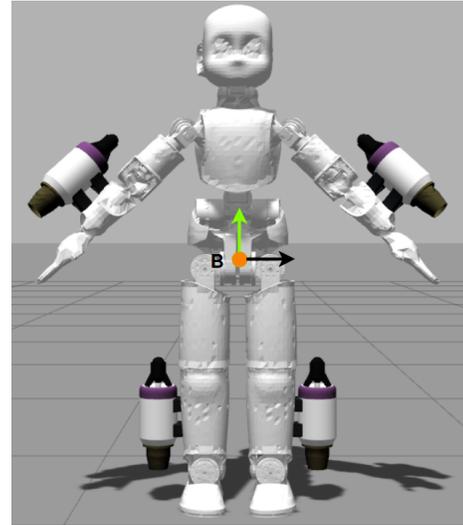


Fig. 1. The jet-powered humanoid robot. The base frame B is attached to the robot's base link.

are used as control inputs to first stabilize the robot position along the desired trajectory. Then, the attitude is stabilized towards the desired values by means of *backstepping* or *high gains control* approach. Eventual actuation redundancy is finally exploited, for example, for tracking a desired robot's orientation [9],[11].

Concerning instead humanoid robots, to assume that the robot is always attached to the ground may be a limitation for achieving tasks such as walking. An alternative solution is to make use of the *floating base* formalism, that does not assume any of the robot's links to have an *a priori* constant pose with respect to an inertial reference frame [12]. An effective technique for controlling floating base robots with rigid joints is the operational space control, where the control objective is often the stabilization of the robot *centroidal momentum* [13]. The controllers designed for this objective are usually referred to as *momentum-based* controllers [14]. To get rid of the (eventual) actuation redundancy associated with momentum control, a lower priority task is usually added during the stabilization of the robot momentum. This secondary task aims at imposing a desired joint robot configuration, and plays a pivotal role for the stabilization of the system's *zero dynamics* [15].

In a previous work, we proposed a momentum-based strategy for controlling a flying humanoid robot [16]. However, one of the main drawbacks of controlling the momentum is that in general it cannot be associated with a quantity that

<sup>1</sup> All authors belong to the Italian Institute of Technology, Via Morego 30, Genoa, Italy name.surname@iit.it

somehow represents the *overall* orientation of the system in space, and therefore it may not be easy to achieve a good attitude tracking, which is fundamental for performing complex maneuvers while flying. The contribution of this paper is the development of a task-based control algorithm for position and attitude stabilization of complex multi-body flying systems, and its implementation on a novel platform, that is, a jet-powered humanoid robot. The proposed algorithm ensures global tracking of the robot's center of mass position, and local asymptotic tracking of the attitude trajectory. Other important control objectives such as performing take off and landing or achieving manipulation tasks while flying may require further modifications and improvements of the proposed control algorithm and will be addressed in future works.

The remainder of the paper is organized as follows: Sec II recalls notation, system modeling, and the momentum-based control strategy for flying proposed in [16]. In Section III, the task-based control for position and attitude tracking is presented. Simulation results on the robot iCub [17] are presented in Section IV. Conclusions and perspectives conclude the paper.

## II. BACKGROUND

### A. Notation

- 1)  $\mathcal{I}$  denotes an inertial frame of reference, with its  $z$  axis pointing against the gravity.  $\mathcal{B}$  denotes the *base* frame, i.e. a frame attached to the robot *base link*.  $\mathcal{G}[\mathcal{I}]$  denotes a frame with the origin at the robot center of mass, and the same orientation of the inertial frame.  $\mathcal{G}[\mathcal{B}]$  is a frame with the origin at the robot center of mass, and the orientation of the base frame.
- 2)  $1_n \in \mathbb{R}^{n \times n}$  is the identity matrix of dimension  $n$ .  $0_{n \times m} \in \mathbb{R}^{n \times m}$  is the zero matrix of size  $n \times m$ .
- 3)  $S(x) \in \mathbb{R}^{3 \times 3}$  is the skew symmetric matrix such that  $S(x)y = x \times y$ , where  $\times$  denotes the cross product operator in  $\mathbb{R}^3$ .
- 4)  $skew(A) \in \mathbb{R}^{3 \times 3}$  is the skew symmetric matrix such that  $skew(A) = \frac{A - A^\top}{2}$ , with  $A \in \mathbb{R}^{3 \times 3}$ .
- 5)  $(\cdot)^\vee$  is the operator defined by  $x = S(x)^\vee$ .

### B. Robot Modelling

The humanoid robot can be modeled as a multi-body system, composed of  $n+1$  rigid bodies, usually referred as *links*, connected by  $n$  one degree of freedom joints. Following up the floating base formalism, the system's *configuration space* is defined as  $\mathbb{Q} \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n$ . An element of  $\mathbb{Q}$  is given by the following triplet:  $q = (\mathcal{I}o_{\mathcal{B}}, \mathcal{I}R_{\mathcal{B}}, s)$ , where  $(\mathcal{I}o_{\mathcal{B}}, \mathcal{I}R_{\mathcal{B}})$  represents the position and orientation of the base frame  $\mathcal{B}$  expressed in the inertial frame, while  $s \in \mathbb{R}^n$  denotes the internal joint angles. The velocity of the system is characterized by the set  $\mathbb{V} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^n$ . An element of  $\mathbb{V}$  is given by  $\nu = (v_{\mathcal{B}}, \dot{s})$  where  $v_{\mathcal{B}} = (\mathcal{I}\dot{o}_{\mathcal{B}}, \mathcal{I}\omega_{\mathcal{B}})$  is the linear and angular velocity of the base frame w.r.t. the inertial frame, while  $\dot{s}$  are the joint velocities. We recall that the base angular velocity satisfies  $\mathcal{I}\dot{R}_{\mathcal{B}} = S(\mathcal{I}\omega_{\mathcal{B}})\mathcal{I}R_{\mathcal{B}}$ .

We now apply the Euler-Poincaré formalism [18, Ch. 13.5] to write the system's equations of motion:

$$M(q)\dot{\nu} + C(q, \nu)\nu + G(q) = \begin{bmatrix} 0_6 \\ \tau \end{bmatrix} + \sum_{k=1}^m J_k^\top F_k \quad (1)$$

where  $M, C \in \mathbb{R}^{n+6 \times n+6}$  are the mass and Coriolis matrix,  $G \in \mathbb{R}^{n+6}$  is the gravity vector,  $\tau \in \mathbb{R}^n$  are the internal actuation torques.

We assume that the robot is powered by  $m = 4$  thrust forces, located on the robot's end effectors as in Fig. 1. Each  $F_k$  in Eq. (1) represents the thrust force applied on the robot by the  $k$ -th jet. In particular,  $F_k = \mathcal{I}l_k(q)T_k$ , where  $\mathcal{I}l_k \in \mathbb{R}^3$  is the thrust direction, and  $T_k \in \mathbb{R}$  the thrust magnitude. The thrust directions move accordingly with the robot's joints, while each thrust intensity can be regulated by controlling its rate of change  $\dot{T}_k$ . The jacobian  $J_k(q)$  is the map between the system's velocity  $\nu$  and the linear velocity  $\mathcal{I}\dot{o}_k$  of the  $k$ -th thrust application point. Lastly, by defining  $T := (T_1, T_2, T_3, T_4)^\top$ , we can compactly rewrite  $f(q, T) = \sum_{k=1}^m J_k^\top F_k$ .

### C. Recalls on the Momentum-Based Flying Controller

We summarize now the momentum-based control strategy for flying presented in our previous work. The key point is to recall that the rate of change of the robot *centroidal momentum* (i.e., the robot's total momentum expressed w.r.t. the aforementioned frame  $\mathcal{G}[\mathcal{I}]$ ) equals the summation of all the external forces and moments acting on the system, that in the case study are the thrust forces and the gravity force. The centroidal momentum dynamics is then given by:

$$\mathcal{G}[\mathcal{I}]\dot{h} = A(q)T - mge_3 \quad (2)$$

where we define:

$$\begin{aligned} e_3 &= (0, 0, 1, 0, 0, 0)^\top \\ A(q) &= \left( \bar{S}(r_1)\mathcal{I}l_1, \dots, \bar{S}(r_m)\mathcal{I}l_m \right) \\ r_i &= \mathcal{I}o_i - \mathcal{I}o_{\mathcal{G}}, \quad \forall i \in (1, \dots, m) \\ \bar{S}(r_i) &= \begin{pmatrix} 1_3 \\ S(r_i) \end{pmatrix} \end{aligned}$$

$\mathcal{I}o_{\mathcal{G}} \in \mathbb{R}^3$  is the center of mass position. In this Section, we omit the superscript  $\mathcal{G}[\mathcal{I}]$  unless it is necessary to explicitly write it for the sake of clarity, and we write  $\mathcal{G}[\mathcal{I}]\dot{h} = \dot{h}$ . The goal is to design a control algorithm for stabilizing the momentum dynamics along a reference trajectory  $h_d(t)$ . The advantage on controlling Eq. (2) w.r.t., for example, the acceleration of a generic frame  $\dot{v}_t = \dot{v}_t(q, \nu, \tau, T)$  associated with one of the robot links, resides in its independence from the joint velocities  $\dot{s}$  and torques  $\tau$ . This clearly identifies the key role of the thrust forces in the stabilization of the underactuated part of system (1), i.e. the floating base dynamics. More precisely, define the momentum error  $\tilde{h} = h - h_d$ . Then, at the equilibrium configuration  $(\tilde{h}, \dot{\tilde{h}}) = (0, 0)$  the effect of the thrust forces must oppose the gravity plus the term  $\dot{h}_d(t)$ , i.e.:

$$0 = A(q)T - mge_3 - \dot{h}_d(t). \quad (3)$$

Reminiscent of the *vectored-thrust* paradigm [9], we may think of using the joint velocities  $\dot{s}$  and the thrusts rate of change  $\dot{T}$  to align the total thrust force  $A(q)T$  to the gravity and the desired momentum rate of change effect. Therefore we make the following assumption:

**Assumption 1:** the joint velocities  $\dot{s} := u_2$  and the thrust rate of change  $\dot{T} := u_1$  can be chosen at will and then considered as control inputs. In particular, the joint velocities may be stabilized towards the desired values by means of a *low-level* joint velocity control loop. Otherwise, the joint velocity dynamics can be controlled by properly choosing the joint torques  $\tau$ .

More generally, we define:

$$\begin{aligned}\tilde{\xi} &= A(q)T + F, \\ F &:= -mge_3 - \dot{h}_d + K_D \tilde{h} + K_P I(t)\end{aligned}$$

with  $K_P, K_D \in \mathbb{R}^{6 \times 6}$  two symmetric and positive definite matrices. The variable  $I(t)$  represents the integral of  $\tilde{h}$ . Then, we rewrite Eq. (2) as follows:

$$\dot{I} = \tilde{h} \quad (4a)$$

$$\dot{\tilde{h}} = \tilde{\xi} - K_D \tilde{h} - K_P I \quad (4b)$$

$$\dot{\tilde{\xi}} = Au_1 + \Lambda_s u_2 + \Lambda_b \nu_B - \ddot{h}_d + K_D \dot{\tilde{h}} + K_P \tilde{h} \quad (4c)$$

where matrix  $A$  is given by Eq. (2), while:

$$\begin{aligned}\Lambda_b &:= \Lambda \begin{pmatrix} 1_6 \\ 0_{n \times 6} \end{pmatrix}, \Lambda_s := \Lambda \begin{pmatrix} 0_{6 \times n} \\ 1_n \end{pmatrix} \\ \Lambda &:= - \left( \tilde{S}_1, \dots, \tilde{S}_m \right) J_r \\ \tilde{S}_i &:= T_i \begin{pmatrix} 0_3 & S(\mathcal{I}l_i) \\ S(\mathcal{I}l_i) & S(r_i)S(\mathcal{I}l_i) \end{pmatrix}\end{aligned}$$

and  $J_r \in \mathbb{R}^{6m \times n+6}$  the jacobian mapping the system velocity  $\nu$  into the velocities  $\Omega := (\dot{r}_1, \omega_1, \dots, \dot{r}_m, \omega_m) \in \mathbb{R}^{6m}$ , where  $\omega_i \in \mathbb{R}^3$  is the angular velocity associated with a frame attached to the  $i$ -th thrust application point.

In our previous work, we managed to define the conditions for the existence of a smooth control input  $(u_1, u_2)$  such that the closed loop equilibrium point  $(I, \tilde{h}, \tilde{\xi}) = (0, 0, 0)$  of the system (4) is globally asymptotically stable. In particular, recall that the centroidal momentum is linear versus the robot velocity  $\nu$ , i.e.  $h = (J_h^B, J_h^s)\nu = J_h^B \nu_B + J_h^s \dot{s}$ . The matrix  $J_h(q) = (J_h^B, J_h^s)$  is usually referred as the *centroidal momentum matrix* [13]. Then, define:

$$\begin{aligned}\sigma &:= (\Lambda_b + \tilde{K} J_h^B) \nu_B + (K_D + 1_3) \dot{\tilde{h}} + K_P I - \ddot{h}_d - \tilde{K} h_d \\ B &:= \Lambda_s + \tilde{K} J_h^s \\ \tilde{K} &:= K_P + K_D + K_O^{-1}\end{aligned}$$

with  $K_O \in \mathbb{R}^{6 \times 6}$  a symmetric and positive definite matrix. If there exist  $(u_1, u_2)$  such that:

$$\sigma + Au_1 + Bu_2 = 0_6, \quad (5)$$

then the closed loop equilibrium point  $(I, \tilde{h}, \tilde{\xi}) = (0, 0, 0)$  is globally asymptotically stable. The proof has been presented

in [16]. In particular, as long as  $\text{rank}(\begin{bmatrix} A & B \end{bmatrix}) = 6$  a solution to Eq. (5) always exists. Also, being the control input of dimension  $(u_1, u_2) \in \mathbb{R}^{n+m}$ , one is left with a redundancy of dimension  $n+m-6$ . We exploit this redundancy to attempt the stabilization of the system's *zero dynamics* by defining a *postural* task of the form:

$$p := -K_P^p (s - s_r) \quad (6)$$

where  $K_P^p \in \mathbb{R}^{n \times n}$  a symmetric and positive definite matrix and  $s_r \in \mathbb{R}^n$  a reference position for the joints configuration. The tasks (5)-(6) can be combined in a weighted optimization problem as follows:

$$(u_1^*, u_2^*) = \underset{(u_1, u_2)}{\text{argmin}} (\lambda_p |u_2 - p|^2 + \lambda_m |\sigma + Au_1 + Bu_2|^2 + \lambda_s |u_2|^2 + u_1^T W u_1) \quad (7a)$$

s.t.

$$lb_1 < u_1 < ub_1, \quad lb_2 < u_2 < ub_2 \quad (7b)$$

with  $\lambda_p, \lambda_m, \lambda_s$  positive weighting constants and  $W \in \mathbb{R}^{m \times m}$  a positive weighting matrix. To better understand if the control objectives could be also achieved on a real platform, we relaxed Assumption 1 by limiting the inputs  $u_1$  and  $u_2$ .  $lb_1, ub_1$  and  $lb_2, ub_2$  are the lower and upper bounds of  $u_1$  and  $u_2$ , respectively. The cost function (7a) also contains regularization terms depending on  $u_1$  and  $u_2$ .

The solution to the problem (7) is the pair  $(u_1^*, u_2^*)$ , i.e. the instantaneous rate of change of the thrust intensities  $\dot{T}^*$  and the joint velocities  $\dot{s}^*$ . In case the actual control input of system (1) are the joint torques  $\tau$ , we interpret  $\dot{s}^*$  as a desired value for the joint velocities to be stabilized by a torque-control law. More precisely, we partition Eq. (1) as follows:

$$\begin{aligned}M &= \begin{pmatrix} M_B & M_{B_s} \\ M_{B_s}^T & M_s \end{pmatrix} \\ b &:= \begin{pmatrix} b_B \\ b_s \end{pmatrix} = C\nu + G \\ f(q, T) &= \begin{pmatrix} f_B \\ f_s \end{pmatrix}\end{aligned}$$

with  $M_B \in \mathbb{R}^{6 \times 6}$ ,  $M_{B_s} \in \mathbb{R}^{6 \times n}$ ,  $M_s \in \mathbb{R}^{n \times n}$ ,  $b_B, f_B \in \mathbb{R}^6$  and  $b_s, f_s \in \mathbb{R}^n$ . It is now possible to isolate the joints dynamics from Eq. (1), which yields:

$$\bar{M}_s \ddot{s} + \bar{b} = \tau \quad (8)$$

where we defined  $\bar{M}_s := M_s - M_{B_s}^T M_B^{-1} M_{B_s}$  and  $\bar{b} := b_s - f_s + M_{B_s}^T M_B^{-1} (f_B - b_B)$ . In view of Eq. (8), the stabilization of the desired joint velocities may be attempted by applying the following high-gain control law:

$$\tau = \bar{b} - \bar{M}_s \left( K_P^s (\dot{s} - \dot{s}^*) + K_I^s \int_0^t (\dot{s} - \dot{s}^*) dt \right). \quad (9)$$

#### D. Complements on Attitude Stabilization

One of the main limitations of the control law (5)-(9) is that it does not guarantee good performances for tracking a desired robot attitude. More precisely, the angular momentum of a rigid multi-body system is given by  ${}^{\mathcal{G}[X]}h_\omega =$

$\mathcal{G}^{[Z]} \mathbb{I} \mathcal{I} \omega_o$  where  $\mathcal{G}^{[Z]} \mathbb{I}(q) \in \mathbb{R}^{3 \times 3}$  is the total robot's inertia expressed in the frame  $\mathcal{G}^{[Z]}$  and  $\mathcal{I} \omega_o \in \mathbb{R}^3$  is the so-called *locked* (or average) angular velocity [13]. When all the joint velocities are locked ( $\dot{s} = 0$ ),  $\mathcal{I} \omega_o$  represents the angular velocity of the robot, that now behaves as a single rigid body. However, in general  $\mathcal{I} \omega_o$  is not associated with the derivative of a rotation matrix  ${}^{\mathcal{I}}R_o$  somehow representing the overall robot orientation in space. There are precise conditions for the existence of such matrix, that are not guaranteed to be always satisfied for the case under study [19].

To overcome this limitation, the solution proposed in our previous work was to rewrite the integral of the angular momentum error  $I_\omega \in \mathbb{R}^3 := I = (I_l, I_\omega)^\top$  in order to take into account an orientation correction term of the form:

$$I_\omega = \mathcal{G}^{[Z]} \mathbb{I}(q) (\text{skew}({}^{\mathcal{I}}R_B R_d^\top)^\vee). \quad (10)$$

where  ${}^{\mathcal{I}}R_B \in SO(3)$  represents the base frame orientation. The particular form of Eq. (10) is derived as follows. Assume that the angular velocity  $\omega$  can be used as a control input of the dynamics  $\dot{R} = S(\omega)R$ . Then, a choice of  $\omega$  that guarantees singularity free control laws and the quasi-global stability of the equilibrium point  $R = R_d$  is [20, Sec 5.11.6]:

$$\omega = -k(\text{skew}(R R_d^\top)^\vee), \quad k > 0. \quad (11)$$

**Errata Corrige:** the equations (19)-(20) presented in [16, Sec 3E] both contain the term  $\text{skew}(R_d^\top R)^\vee$ . However, if the orientation dynamics is defined as  $\dot{R} = S(\omega)R$  the correct formulation is  $\text{skew}(R R_d^\top)^\vee$ .

We recall that  $\mathcal{I} \omega_o$  is not related to the derivative of  ${}^{\mathcal{I}}R_B$  and therefore the asymptotic stability of (5)-(9) with  $I_\omega$  as in (10) is not guaranteed anymore. Furthermore, this approach assumes  $R_d$  to be constant and it didn't show good tracking performances when  $R_d = R_d(t)$ .

### III. TASK-BASED CONTROL

This section proposes a task-based control approach derived from the momentum-based flying control presented in Sec. II, that allows the stabilization robot base frame rotation  ${}^{\mathcal{I}}R_B \in SO(3)$  towards desired values  $R_d(t)$ .

The key point is to observe that the locked angular velocity can be expressed as a linear function of the joint velocities and the base velocity [21]. In particular:

$$\mathcal{I} \omega_o = \mathcal{I} \omega_B + J_\omega^s \dot{s}, \quad (12)$$

where  $\mathcal{I} \omega_B$  is the base angular velocity in the inertial frame and  $J_\omega^s \in \mathbb{R}^{3 \times n}$  maps the joint velocities into the locked angular velocity. Therefore, the angular momentum in centroidal coordinates is given by:

$$\mathcal{G}^{[Z]} h_\omega = \mathcal{G}^{[Z]} \mathbb{I}(q) \mathcal{I} \omega_B + \mathcal{G}^{[Z]} \mathbb{I}(q) J_\omega^s(q) \dot{s}, \quad (13)$$

Assume that the joint velocities can be considered as control input, i.e.  $\dot{s} = u_2$ . Then, one may think of choosing  $u_2$  to instantly influence the angular momentum equation Eq. (13), in order to impose  $\mathcal{G}^{[Z]} h_\omega = \mathcal{G}^{[Z]} h_\omega^*$ .

The effectiveness of this approach strongly depends on the feasibility of the desired angular momentum  $\mathcal{G}^{[Z]} h_\omega^*$ . In

particular, it must be verified that for the given choice of  $\mathcal{G}^{[Z]} h_\omega^*$  the matrix  $J_\omega^s(q)$  never loses rank while executing the task. Without loss of generality, we can express the angular momentum w.r.t. the frame  $\mathcal{G}^{[B]}$ , by multiplying Eq. (13) times the rotation matrix  ${}^{\mathcal{B}}R_Z$ :

$$\mathcal{G}^{[B]} h_\omega = \mathcal{G}^{[B]} \mathbb{I} \omega_B + \bar{J}_\omega^s \dot{s}, \quad (14)$$

where  $\mathcal{G}^{[B]} \mathbb{I} = {}^{\mathcal{B}}R_Z \mathbb{I} {}^{\mathcal{I}}R_B$  is the total inertia matrix in the new coordinates,  ${}^{\mathcal{B}}\omega_B$  the angular velocity in base coordinates and  $\bar{J}_\omega^s = {}^{\mathcal{B}}R_Z \mathcal{G}^{[Z]} \mathbb{I}(q) J_\omega^s(q)$ .

We design the desired angular momentum expressed in  $\mathcal{G}^{[B]}$  frame to be:

$$\mathcal{G}^{[B]} h_\omega^* = \mathcal{G}^{[B]} \mathbb{I}_0 \omega_B, \quad (15)$$

with  $\mathcal{G}^{[B]} \mathbb{I}_0 = \mathcal{G}^{[B]} \mathbb{I}(t=0)$ . The main advantage of performing this change of coordinates and of choosing  $\mathcal{G}^{[B]} h_\omega^*$  as in (15) is that, being the base frame  $\mathcal{B}$  attached to the robot, the inertia matrix  $\mathcal{G}^{[B]} \mathbb{I}$  only depends on the joints configuration, i.e.  $\mathcal{G}^{[B]} \mathbb{I} = \mathcal{G}^{[B]} \mathbb{I}(s)$ . While the robot is flying, the position of most of the joints only has *small* and *smooth* changes with respect to the initial robot configuration. This is true in particular for the joints associated to the links that mostly contributes to the robot's total inertia, such as the robot chest and upper legs. It is possible to verify numerically that for the case study the inertia rate of change is  $\mathcal{G}^{[B]} \dot{\mathbb{I}} \approx 0_3$ , and therefore the constraint applied with Eq. (15) is not far from the system's original behavior.

The joint velocities satisfying (15) are given by:

$$u_2 = (\bar{J}_\omega^s)^\dagger (\mathcal{G}^{[B]} \mathbb{I}_0 - \mathcal{G}^{[B]} \mathbb{I}(s)) \omega_B + N_\omega^s u_3, \quad (16)$$

where  $(\bar{J}_\omega^s)^\dagger$  is the Moore-Penrose pseudoinverse of matrix  $\bar{J}_\omega^s$  and  $N_\omega^s \in \mathbb{R}^{n \times n}$  is the projector into its null space.  $u_3 \in \mathbb{R}^n$  is a free variable. Now that we achieved  $\mathcal{G}^{[B]} h_\omega = \mathcal{G}^{[B]} h_\omega^*$ , we make use of the angular momentum acceleration  $\mathcal{G}^{[B]} \dot{h}_\omega = \mathcal{G}^{[B]} \mathbb{I}_0 \omega_B$  to control the base orientation dynamics. In particular:

**Lemma 1:** Consider the following system:

$$\mathcal{I} \dot{R}_B = \mathcal{I} R_B S({}^{\mathcal{B}}\omega_B) \quad (17a)$$

$$\mathcal{G}^{[B]} \dot{h}_\omega = \mathcal{G}^{[B]} \mathbb{I}_0 \omega_B \quad (17b)$$

$$\mathcal{G}^{[B]} \ddot{h}_\omega = \mathcal{G}^{[B]} \ddot{h}_\omega^*, \quad (17c)$$

and assume that the angular momentum acceleration  $\mathcal{G}^{[B]} \ddot{h}_\omega^* \in \mathbb{R}^3$  can be chosen at will. Apply the control law

$$\mathcal{G}^{[B]} \ddot{h}_\omega = \mathcal{G}^{[B]} \ddot{h}_\omega^d - (1_3 + c_0 \mathcal{G}^{[B]} \mathbb{I}_0^{-1}) \mathcal{G}^{[B]} \dot{h}_\omega + \mathcal{G}^{[B]} \dot{h}_\omega^d - c_1 \mathcal{G}^{[B]} \mathbb{I}_0^{-1} \mathcal{G}^{[B]} \dot{h}_\omega - \dot{s} k v - c_o \mathcal{G}^{[B]} \mathbb{I}_0^{-1} s k v, \quad (18a)$$

with:

$$s k v = \text{skew}(R_d^\top \mathcal{I} R_B)^\vee, \quad (18b)$$

$$\mathcal{G}^{[B]} \dot{h}_\omega = \mathcal{G}^{[B]} h_\omega - \mathcal{G}^{[B]} h_\omega^d, \quad c_0, c_1 \in \mathbb{R}^+ \quad (18c)$$

to system (17). Then, the closed loop equilibrium  $(\mathcal{G}^{[B]} \dot{h}_\omega, \mathcal{G}^{[B]} h_\omega, \mathcal{I} R_B) = (\mathcal{G}^{[B]} \dot{h}_\omega^d(t), \mathcal{G}^{[B]} h_\omega^d(t), R_d(t))$  is locally asymptotically stable.

The proof is given in the Appendix. Now, the angular momentum acceleration  ${}^{\mathcal{G}[\mathcal{B}]}\ddot{h}_\omega^*$  that stabilizes the base orientation and angular momentum dynamics can be achieved by resorting to the remaining free joint velocities  $u_3$  and the thrust intensities rate of change  $\dot{T} = u_1$ . Similarly to Eq. (4) in Sec. II C, we define the following dynamical system:

$$\dot{I}_l = \tilde{h}_l \quad (19a)$$

$$\dot{\tilde{h}}_l = \tilde{\xi}_l - K_{Dl}\tilde{h}_l - K_{Pl}I_l \quad (19b)$$

$$\begin{aligned} \dot{\tilde{\xi}}_l &= [1_3, 0_3](\Lambda u_1 + \Lambda_s N_\omega^s u_3 + \lambda_{bias} + \Lambda_b v_B) + \\ &\quad - \dot{h}_l^d + K_{Dl}\dot{\tilde{h}}_l + K_{Pl}\tilde{h}_l \end{aligned} \quad (19c)$$

$${}^{\mathcal{G}[\mathcal{Z}]}\ddot{h}_\omega = [0_3, 1_3](\Lambda u_1 + \Lambda_s N_\omega^s u_3 + \lambda_{bias} + \Lambda_b v_B) \quad (19d)$$

where the equations (19a)-(19b)-(19c) correspond to the first three rows of (4a)-(4b)-(4c), respectively, and represent the linear momentum dynamics. Eq. (19d) is the angular momentum dynamics and can be obtained by selecting the last three rows of the first three elements on the right-hand side of Eq. (4c). In both Eq. (19c) and (19d) we also substituted  $u_2$  with its expression from Eq. (16). This implies  $\Lambda_s u_2 = \Lambda_s [(\bar{J}_\omega^s)^\dagger ({}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0 - {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}(s))^{\mathcal{B}} \omega_B + N_\omega^s u_3] = \Lambda_s N_\omega^s u_3 + \lambda_{bias}$ .

A control input  $(u_1, u_3)$  that stabilizes system (19) towards the closed loop equilibrium point  $(I_l, \tilde{h}_l, \tilde{\xi}_l) = (0, 0, 0)$  while also ensuring  ${}^{\mathcal{G}[\mathcal{Z}]}\ddot{h}_\omega = {}^{\mathcal{G}[\mathcal{Z}]}\ddot{h}_\omega^*$  must satisfy the following equivalence:

$$\bar{\sigma} + A u_1 + \bar{B} u_3 = 0_6 \quad (20)$$

with  $A(q)$  as in Eq. (5), while  $\bar{B}$ ,  $\bar{\sigma}$  must be recomputed as:

$$\begin{aligned} \bar{\sigma} &= \begin{bmatrix} [1_3, 0_3](\sigma + \lambda_{bias} + \gamma_{bias}) \\ [0_3, 1_3](\lambda_{bias} + \Lambda_b v_B) - {}^{\mathcal{G}[\mathcal{Z}]}\ddot{h}_\omega^* \end{bmatrix} \\ \bar{B} &= \begin{bmatrix} [1_3, 0_3](\Lambda_s N_\omega^s + \tilde{K} J_h^s N_\omega^s) \\ [0_3, 1_3](\Lambda_s N_\omega^s) \end{bmatrix} \\ \gamma_{bias} &= \tilde{K} J_h^s (\bar{J}_\omega^s)^\dagger ({}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0 - {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}(s))^{\mathcal{B}} \omega_B \end{aligned}$$

Eq. (20) is equivalent to Eq. (5) presented in Sec. II C. As before, as long as  $\text{rank}(\begin{bmatrix} A & \bar{B} \end{bmatrix}) = 6$ , a solution to Eq. (20) always exists. Furthermore, it is still possible to apply Eq. (6) to design the robot's postural task, and the desired joint velocities  $u_2$  (and consequently  $u_3$ ) can be obtained by applying the high-gain control technique presented in Eq. (9). The tasks (6)-(20) can still be combined in the weighted optimization problem (7).

**Remark:** the desired angular momentum in centroidal coordinates is obtained using the following relationship:  ${}^{\mathcal{G}[\mathcal{Z}]}\ddot{h}_\omega^* = {}^{\mathcal{I}}R_B {}^{\mathcal{G}[\mathcal{B}]}\ddot{h}_\omega^*$ . Furthermore, it can be verified that as long as Eq. (16) holds,  ${}^{\mathcal{G}[\mathcal{Z}]}\ddot{h}_\omega^*$  does not depend on the control inputs  $\dot{s}$ ,  $\dot{T}$  and  $\tau$  (hence avoiding algebraic loops).

## IV. SIMULATION RESULTS

### A. Simulation Environment

The algorithm presented in Sec III is implemented in a Simulink controller interfacing with Gazebo simulator [22].

The controller frequency is 100 [Hz]. Gazebo offers different physic engines to integrate the system's dynamics. Among all the possibilities, we chose the Open Dynamics Engine (ODE), that uses a fixed step semi-implicit Euler integration scheme, with a simulation time step of 1 [ms]. An advantage of using Gazebo consists in the ability to test directly on the real robot the same control software used in simulation.

### B. Performances Comparison

At first, we performed a comparison between the control law (5)-(9) with the modification presented in Sec. II D and the task-based control law (18)-(20). Concerning the constraints on the inputs  $u_1$  and  $u_2$  in the optimization problem (7a), we consider  $|u_1| < 100 \frac{N}{s}$  and  $|u_2| < 45 \frac{deg}{s}$ . At the moment, no bounds on the angular momentum acceleration are considered. We kept equal for both controllers all the control gains and parameters but the ones that mostly affect the orientation tracking performances, that are  $K_{P\omega} := K_P = \begin{bmatrix} K_{Pl} & 0_3 \\ 0_3 & K_{P\omega} \end{bmatrix}$  for the controller described in Sec. II and  $c_0, c_1$  for the task based controller. In particular, we choose  $K_{P\omega} = c_2 1_3$ ,  $c_2 > 0$ . Both controllers are required to track a desired base frame orientation  $R_d(t)$ . In particular, the base frame roll, pitch and yaw desired angles are designed as a chirp signal of the form  $\alpha = \alpha_0 + A_m \sin(2\pi(f_0 t + \frac{k}{2} t^2))$  with  $k = 0.0175$ ,  $A_m = 7.5$  [deg] and  $f_0 = 0.2$  [Hz]. First, we evaluate the range of  $c_0, c_1, c_2$  that always guarantees a stable behavior while executing the given task. In particular:

$$20 < c_2 < 60,$$

$$30 < c_0 < 75,$$

$$30 < c_1 < 75.$$

Then, we performed 30 experiments by choosing randomly the values of  $c_0, c_1, c_2$  in between the stable range. The results are compared in Fig. 2, that represents the average norm of the rotation error  $\text{rot} = [\text{roll}, \text{pitch}, \text{yaw}]^\top$ . The semi-transparent colored regions represent the variance of the error norm over 30 experiments. The task-based controller shows better tracking performances for all the frequencies considered in the chirp signal, despite the variance over the experiments is wider than for the momentum-based controller. This can be explained by considering that the closed loop orientation dynamics is affected by the ratio between  $c_0$  and  $c_1$ , but we chose the two gains independently. This is anyways a worst-case scenario w.r.t. choosing, for example, something similar to  $c_1 = 2\sqrt{c_0}$  as for the classical critically damped systems.

### C. Task 1: Orientation Tracking

This task consists in the robot hovering at 5 [m] from the ground. While hovering, the robot is spinning along the yaw angle with a spinning speed of 1  $\frac{\text{rad}}{s}$  and it is tracking a sinusoidal trajectory along both pitch and roll angle with an amplitude of 15 [deg] and a frequency of 0.25 [Hz]. Figure 3 shows the reference and actual values of the roll, pitch and yaw angles. Despite the control law (18) is coupling the roll, pitch and yaw dynamics, it is possible to achieve

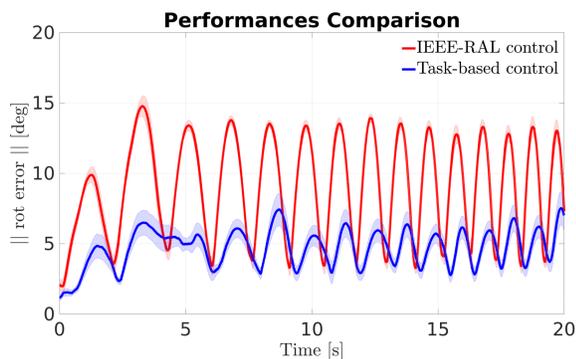


Fig. 2. Norm of the rotation tracking error. The solid lines represent the average norm over 30 experiments, while the transparent region is the associated variance.

reasonably good tracking performances along all the three angles. The coupling of the roll, pitch and yaw dynamics in the closed loop system may be the responsible of the residual steady-state errors along the pitch angle. A video showing the experiment is attached to the paper.

#### D. Task 2-3: Aggressive Maneuvers

We tested the effectiveness of the task-based control while performing aggressive maneuvers. In particular, Task 2 consists in the humanoid robot flying at  $30 \left[ \frac{km}{h} \right]$ , with a constant pitch orientation of  $-60$  [deg]. The robot has to suddenly stop and bring its velocity to zero in 5 [s]. Furthermore, it also has to bring the pitch angle to zero, and, after two seconds, rotate along the yaw angle by  $90$  [deg]. During Task 3, the robot is flying at  $30 \left[ \frac{km}{h} \right]$  along the  $x$  CoM direction, with a constant pitch orientation of  $-60$  [deg]. Then, it has to turn by  $90$  [deg] and fly along the  $y$  CoM direction, always at  $30 \left[ \frac{km}{h} \right]$ . Meanwhile turning, the yaw angle must be updated in order to keep the robot head always parallel to the flying direction. The pitch angle must remain constant at  $-60$  [deg]. The turning maneuver happens in 5 [s]. Figures 4-6 represent the pitch and yaw references and actual values during Tasks 2-3. The actual values remain close to the references, despite the robot is performing aggressive tasks. Figure 5 shows the thrust forces magnitude required for achieving Task 2, that resulted to be the most challenging of the three tasks under consideration. The robot weight, including the turbines, is around  $40$  [kg]. The turbines weight is that of the commercial products JetCat P130-RX (max thrust  $130$  [N]) and JetCat P220-RXi (max thrust  $220$  [N]), that can also achieve the required thrust magnitude. A video showing the experiments is attached to the paper.

Finally, in Fig. 7 we compared the center of mass error along the directions  $[x, y, z]$  during the three tasks. It is possible to observe that during Task 1-3 the CoM error remain small along all directions, with a magnitude smaller than  $0.06$  [m]. During Task 2 there is a peak of  $0.3$  and  $-0.15$  [m] of error along the  $x$  and  $z$  directions, corresponding to the instant the robot performs the stop from  $30$  to  $0 \left[ \frac{km}{h} \right]$ . However, after few seconds stability is recovered and the

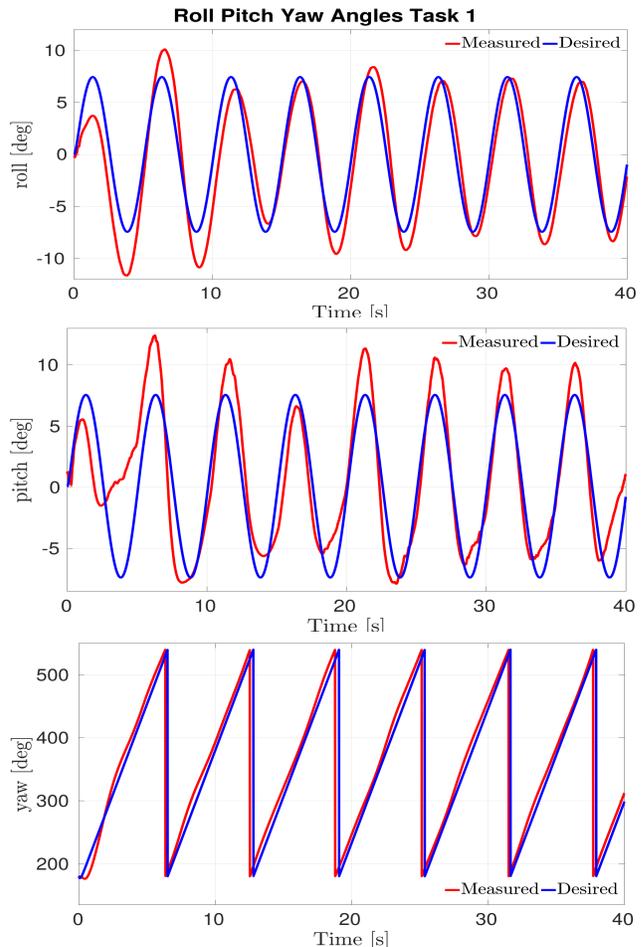


Fig. 3. Reference and actual base orientation while performing Task 1. The closed loop system shows good tracking performances for all the three angles.

error rapidly decreases.

#### E. Feasibility of the Angular Momentum Constraint

We verified that the matrix  $\bar{J}_\omega^s$  remains full rank during all the tasks, that is the condition that guarantees that the angular momentum can be instantly regulated by (16). In particular, we performed a Singular Value Decomposition analysis on matrix  $\bar{J}_\omega^s$ . Both the average and the variance of the singular values  $[\sigma_1, \sigma_2, \sigma_3]$  remained the same during all three tasks. In particular,  $[\sigma_1^{avg}, \sigma_2^{avg}, \sigma_3^{avg}] = [1.2, 1.15, 0.45]$  while the variance is:

$$\begin{aligned} 1.3 < \sigma_1 < 1.15, \\ 1.2 < \sigma_2 < 1.1, \\ 0.35 < \sigma_3 < 0.55. \end{aligned}$$

Numerical results therefore show that achieving Eq. (15) is always possible during the three tasks.

## V. CONCLUSIONS

In this paper, a task-based control for the stabilization of a jet-powered flying humanoid robot is proposed. The control algorithm can guarantee global tracking of a desired CoM

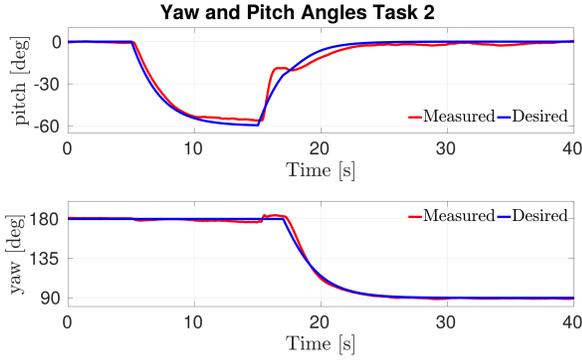


Fig. 4. Reference and actual pitch and yaw angles while performing Task 2. Despite the aggressive maneuver, the closed loop system still shows good tracking performances.

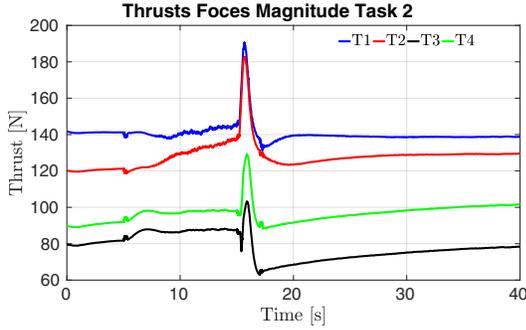


Fig. 5. Thrust forces magnitude while performing Task2 (the most challenging task).

position, and local asymptotic stabilization of a reference attitude trajectory. Tests in a simulation environment with the humanoid robot iCub and performances comparisons with a different control approach verify the soundness of the proposed algorithm.

Nevertheless, throughout the paper we made several assumptions, such as neglecting the aerodynamics effects and the jets' own dynamics. Also the take-off and landing phases, which are not considered in this work, have shown to be critical, and may require a dedicated control design. An analysis of the maximum load acting on the robot joints while flying is also required. Finally, further modifications of the proposed control algorithm are necessary in order to perform manipulation tasks while flying. Future work may address these issues in order to lead to a practical implementation of the control algorithm on the real robotic platform iCub.

## VI. APPENDIX

### A. Proof of Lemma 1

For the sake of clarity we omit the subscripts and superscripts and we define  $R = {}^I R_B$ ,  $\tilde{h}_\omega = {}^G[B]\tilde{h}_\omega$ ,  $\omega = {}^B\omega_B$ ,  $\mathbb{I}_0^{-1} = {}^G[B]\mathbb{I}_0^{-1}$ . Now, consider the following Lyapunov

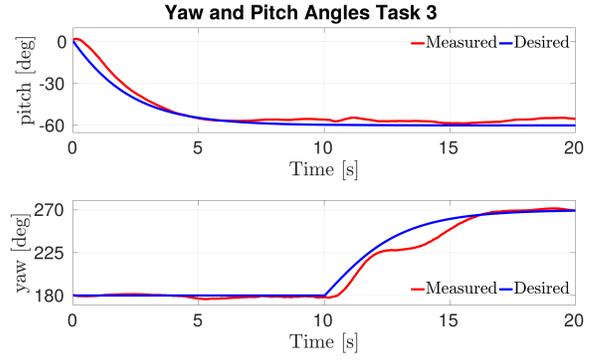


Fig. 6. Reference and actual pitch and yaw angles while performing Task 3. Despite the aggressive maneuver, the closed loop system still shows good tracking performances.

function candidate:

$$\begin{aligned} V &= V_1 + V_2 + V_3, \\ V_1 &= \frac{c_0 + c_1}{2} \text{tr}(1_3 - R_d^\top R), \\ V_2 &= \frac{c_0 + c_1}{2} \tilde{h}_\omega^\top \mathbb{I}_0^{-1} \tilde{h}_\omega, \\ V_3 &= \frac{1}{2} |\dot{\tilde{h}}_\omega + \tilde{h}_\omega + skv|^2. \end{aligned} \quad (21)$$

It is possible to verify that (21) is a valid Lyapunov function candidate. The term  $V_1$  is always positive, and it is zero iff  $R = R_d$  (see also [20, Sec 5.11.6]). Recall  $\mathbb{I}_0^{-1}$  is symmetric and positive definite. Then,  $V_2$  is always positive and  $V_2 = 0$  iff  $\tilde{h}_\omega = 0$ . The last term  $V_3$  is always positive and it has several solutions such that  $V_3 = 0$ , but the only one that guarantees also  $V_1$  and  $V_2$  to be zero is  $R = R_d$ ,  $\tilde{h}_\omega = 0$ ,  $\dot{\tilde{h}}_\omega = 0$  (in particular, recall that  $skv(R = R_d) = 0$ ). Therefore one has  $V \geq 0$ ,  $V = 0$  iff  $R = R_d$ ,  $\tilde{h}_\omega = 0$ ,  $\dot{\tilde{h}}_\omega = 0$ . The time derivative of  $V$  is given by:

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3, \\ \dot{V}_1 &= (c_0 + c_1) \tilde{h}_\omega^\top \mathbb{I}_0^{-1} skv, \\ \dot{V}_2 &= (c_0 + c_1) \tilde{h}_\omega^\top \mathbb{I}_0^{-1} \dot{\tilde{h}}_\omega, \\ \dot{V}_3 &= (\dot{\tilde{h}}_\omega + \tilde{h}_\omega + skv)^\top (\ddot{\tilde{h}}_\omega + \dot{\tilde{h}}_\omega + \dot{skv}). \end{aligned} \quad (22)$$

The derivatives  $\dot{V}_2$  and  $\dot{V}_3$  are straightforward to compute, while the derivative  $\dot{V}_1$  can be obtained by recalling that  $\frac{d(\text{tr}(1_3 - R_d^\top R))}{dt} = 2\tilde{\omega}^\top skv(R, R_d)$  with  $\tilde{\omega} = \omega - \omega_d$  (see also [20, Sec 5.11.6]). Also, if Eq. (15) holds, then  $\tilde{\omega} = \mathbb{I}_0^{-1} \tilde{h}_\omega$ , which in turns leads to (22).

Direct substitution of  $\dot{\tilde{h}}_\omega$  from Eq. (18) into  $\dot{V}$  gives, after long but straightforward calculations:

$$\begin{aligned} \dot{V} &= -c_1 \tilde{h}_\omega^\top \mathbb{I}_0^{-1} \tilde{h}_\omega + \\ &\quad -c_0 (\dot{\tilde{h}}_\omega^\top \mathbb{I}_0^{-1} \dot{\tilde{h}}_\omega + skv^\top \mathbb{I}_0^{-1} skv + 2\dot{\tilde{h}}_\omega^\top \mathbb{I}_0^{-1} skv) \\ &= -c_1 \tilde{h}_\omega^\top \mathbb{I}_0^{-1} \tilde{h}_\omega - c_0 (\dot{\tilde{h}}_\omega + skv)^\top \mathbb{I}_0^{-1} (\dot{\tilde{h}}_\omega + skv) \end{aligned} \quad (23)$$

Being  $\mathbb{I}_0^{-1}$  symmetric and positive definite, one has  $\dot{V} \leq 0$ . In particular it holds that:

- 1)  $\dot{V} \leq 0$  implies that  $\tilde{h}$  and  $\dot{\tilde{h}}$  are bounded, being  $V$  a non-increasing function ( $R$  is bounded by definition);

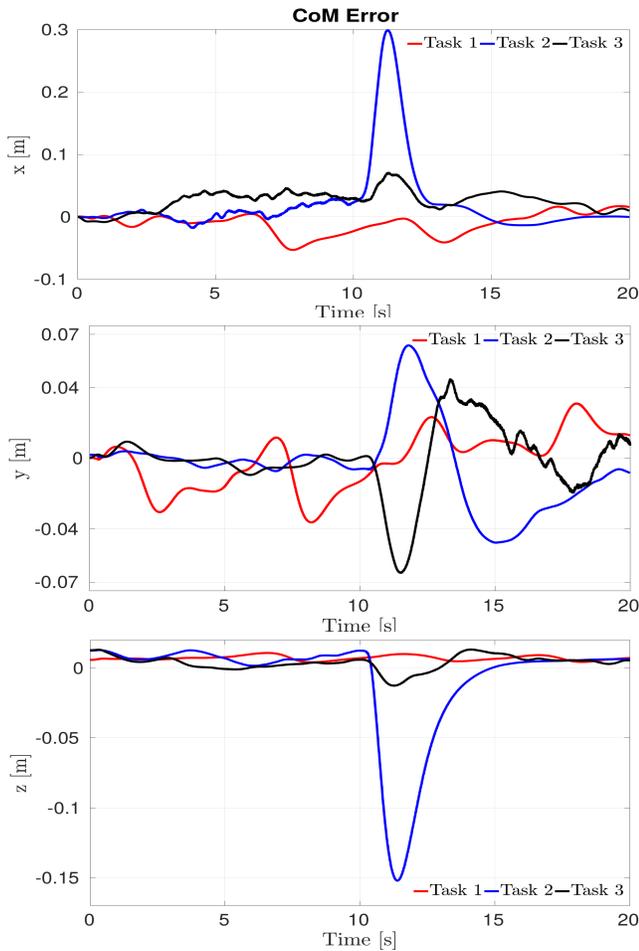


Fig. 7. CoM error during Tasks 1-2-3. The error remains bounded also during the most aggressive task (task 2).

- 2)  $\ddot{V}$  is bounded because of 1) and because of the choice of  $\ddot{h}$  as in (18);
- 3) following the Barbalat's Lemma,  $\ddot{V}$  bounded implies that  $\dot{V} \rightarrow 0$ ;
- 4)  $\dot{V} \rightarrow 0$  implies  $\dot{h} \rightarrow 0$  and  $(\dot{h} + skv) \rightarrow 0$ ;
- 5) being  $\ddot{h}$  bounded, then  $\dot{h} \rightarrow 0$ ;
- 6) finally, because of 4) and 5) one has that also  $skv \rightarrow 0$  and this implies the local convergence of  $R \rightarrow R_d$  as detailed in [20, Sec 5.11.6].

## REFERENCES

- [1] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, July 2018.
- [2] H. Yang and D. Lee, "Dynamics and control of quadrotor with robotic manipulator," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5544–5549.
- [3] G. Heredia, A. E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. A. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 3417–3422.
- [4] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, "6d physical interaction with a fully actuated aerial robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5190–5195.

- [5] X. DING, P. GUO, K. XU, and Y. YU, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chinese Journal of Aeronautics*, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1000936118301894>
- [6] H. B. Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation literature survey," *Robotics and Autonomous Systems*, vol. 107, pp. 221 – 235, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889017305535>
- [7] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2491–2498.
- [8] V. Lippiello, F. Ruggiero, and D. Serra, "Emergency landing for a quadrotor in case of a propeller failure: A backstepping approach," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4782–4788.
- [9] M. D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated vtolvehicles: A review of basic control design ideas and principles," *IEEE Control Systems*, vol. 33, no. 1, pp. 61–75, Feb 2013.
- [10] S. Rajappa, M. Ryll, H. H. Blthoff, and A. Franchi, "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4006–4013.
- [11] R. Naldi, M. Furci, R. G. Sanfelice, and L. Marconi, "Robust global trajectory tracking for underactuated vtol aerial vehicles using inner-outer loop control paradigms," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 97–112, Jan 2017.
- [12] R. Featherstone, *Rigid Body Dynamics Algorithms*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.
- [13] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, 2013.
- [14] 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>
- [15] 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.
- [16] D. Pucci, S. Traversaro, and F. Nori, "Momentum control of an underactuated flying humanoid robot," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 195–202, Jan 2018.
- [17] L. Natale, C. Bartolozzi, D. Pucci, A. Wykowska, and G. Metta, "icub: The not-yet-finished story of building a robot child," *Science Robotics*, vol. 2, no. 13, 2017. [Online]. Available: <http://robotics.sciencemag.org/content/2/13/eaq1026>
- [18] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*. Springer Publishing Company, Incorporated, 2010.
- [19] A. Saccon, S. Traversaro, F. Nori, and H. Nijmeijer, "On centroidal dynamics and integrability of average angular velocity," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 943–950, April 2017.
- [20] R. Olfati-Saber, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles," Ph.D. dissertation, Cambridge, MA, USA, 2001, aAI0803036.
- [21] S. Traversaro, D. Pucci, and F. Nori, "A unified view of the equations of motion used for control design of humanoid robots," Jan 2017. [Online]. Available: [https://www.researchgate.net/publication/312200239\\_A\\_Unified\\_View\\_of\\_the\\_Equations\\_of\\_Motion\\_used\\_for\\_Control\\_Design\\_of\\_Humanoid\\_Robots](https://www.researchgate.net/publication/312200239_A_Unified_View_of_the_Equations_of_Motion_used_for_Control_Design_of_Humanoid_Robots)
- [22] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, pp. 2149 – 2154, 2004.