IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Momentum-Based Extended Kalman Filter for Thrust Estimation on Flying Multibody Robots

Hosameldin Awadalla Omer Mohamed[1, 2], Gabriele Nava[1], Giuseppe L'Erario[1, 3], Silvio Traversaro[1], Fabio Bergonti[1, 3], Luca Fiorio[1], Punith Reddy Vanteddu[1,], Francesco Braghin[2] and Daniele Pucci[1,3]

***Abstract*—Effective control design of flying vehicles requires a reliable estimation of the propellers' thrust forces to secure a successful flight. Direct measurements of thrust forces, however, are seldom available in practice and on-line thrust estimation usually follows from the application of fusion algorithms that process on-board sensor data. This paper proposes a framework for the estimation of the thrust intensities on flying multibody systems that are not equipped with sensors for direct thrust measurement. The key ingredient of the proposed framework is the so-called centroidal momentum of a multibody system, which combined with the propeller model. It enables the design of Extended Kalman Filters (EKF) for on-line thrust estimation. The presented approach tackles the additional complexity in thrust estimation due to the possibly *large* number of degrees of freedom of the system and uncertainties in the propeller model. For instance, a *covariance scheduling* approach based on the turbines RPM error is proposed to ensure a reliable estimation even in case of turbine failures. Simulations are presented to validate the proposed algorithm during robot flight. Moreover, an experimental setup is designed to evaluate the accuracy of the estimation algorithm using iRonCub, a jet-powered humanoid robot, while standing on the ground.**

***Index Terms*—Humanoid Robot Systems, Aerial Systems: Perception and Autonomy, Dynamics.**

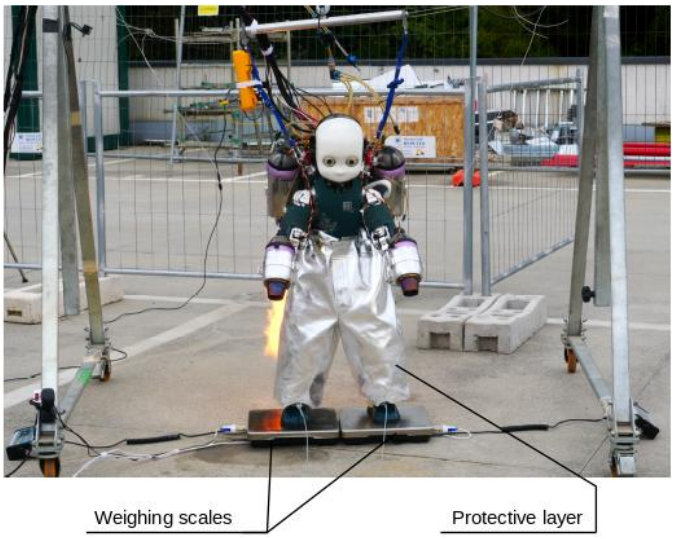

Figure 1. The humanoid robot iRonCub and the setup used for jet experimental campaigns.

I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) have been employed in several applications such as surveillance, remote monitoring and aerial photography, thus increasing the attention of the robotics community on these vehicles [1]. Many tasks involve interaction with the environment and thus exchanging forces with it, so estimating the forces and torques acting on the UAV can be essential for safe and useful interaction. In this sense, the estimation of thrust forces is essential for effective control design. The process of estimating the thrust forces acting on a flying vehicle becomes even harder when the underlying aircraft is composed of several rigid bodies, possibly interconnected by joints of various nature (e.g. rotational/prismatic). This paper contributes towards the on-line estimation of thrust forces that power flying multibody vehicles. The manuscript makes use of reduced models of the underlying multibody robot, that combined with the propellers' actuation models, enables the design of effective Extended Kalman Filters.

Thrust force estimation is typically achieved by exploiting the relationship between the thrust and the rotational speed of the propeller [2]. Thrust force estimation plays also a role for, and may depend upon, the process of estimating the external forces on the aircraft. The external forces acting on the UAV can be measured by means of Force/Torque (F/T) sensors, but they increase the overall cost of the flying system. Consequently, external forces are typically estimated by a suitable observer, which often exploits rigid-body dynamics and on-board proprioceptive sensors.

There is a wide literature on observers for external force estimation when the flying robot is modeled as a *single rigid body*. To cite a few, Lyapunov based nonlinear observers [3], *residual methods* [4], [5] – also used to estimate the generalized torques [6] – and Kalman and Unscented Kalman Filtering [7], [8] are some of the tools and methods that proved to be effective when estimating the external force on a single rigid body flying vehicle. All these methods do not apply directly, however, to the case where the flying robot is composed of several interconnected rigid bodies.

Manuscript received: July, 7, 2021; Revised September, 30, 2021; Accepted October, 29, 2021. This paper was recommended for publication by Editor-in-Chief T. Asfour and Editor A. Kheddar upon evaluation of the Editor and Reviewers' comments. This work was supported by Istituto Italiano di Tecnologia, 16163 Genova, Italy. (Corresponding author: H. A. O. Mohamed)

[1]H. A. O. Mohamed, G. Nava, G. L'Erario, S. Traversaro, F. Bergonti, L. Fiorio, P. R. Vanteddu and D. Pucci are with the Artificial and Mechanical Intelligence, Istituto Italiano di Tecnologia, 16163 Genova, Italy `firstname.surname@iit.it`

[2]H. A. O. Mohamed and F. Braghin are with the Department of Mechanical Engineering, Politecnico di Milano, 20156 Milan, Italy `hosameldinawadalla.mohamed@polimi.it`

[3]G. L'Erario, F. Bergonti and D. Pucci are also with the School of Computer Science, Univ. of Manchester, Manchester M13 9PL, U.K.

This letter has supplementary material available at https://ieeexplore.ieee.org Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Promising research directions, in fact, are trying to design UAVs with several degrees of freedom, which is a pivotal step for providing flying robots with manipulation or terrestrial locomotion capabilities [9], [10], [11]. These *flying multibody robots* are often exemplified by *aerial manipulators*, which are composed of a flying vehicle endowed with a robotic arm or gripper (see, e.g. [12], [9]). Aerial manipulators, however, are not the only family of multibody flying robots. To cite a few, one may mention: hexapod-quadrotor designs [13]; hybrid terrestrial and aerial quadrotors [14]; insect biobots that show aerial and terrestrial locomotion [15]; robotic structures that change their shapes to adapt to different modes of locomotion [16], [17]; and humanoid robots with thrusters to enhance bipedal locomotion [10], [11]. Additionally, an attempt to unify manipulation, aerial, and bipedal terrestrial locomotion on a single robotic platform is being carried out by the iRonCub project [18], [19], whose aim is to enable the humanoid robot iCub [20] to fly. As highlighted in Fig. 1, four jet engines located on the robot arms and shoulders, are used to lift the robot from the ground.

For multibody flying vehicles such as iRonCub, the process of estimating thrust forces is further complexified by the multibody nature of the underlying robot. Two common approaches for thrust estimation on flying multibody robots are: i) often applied to aerial manipulators, one may consider only the dynamics of the flying platform and account for the wrenches applied by the manipulator arm as external disturbances; then the application of single-body estimation methods may lead to an estimate of the thrust forces [21]; ii) one may consider the full dynamics of the system as well as the interaction forces between the flying robot and the manipulator arm [22]. It is not straightforward, however, to generalize the above approaches to more complex flying multibody robots, such as a jet-powered humanoid robot. For instance, the first approach is not applicable because there is no distinction between a "flying part" and a "manipulating part" in jet-powered humanoid robots. The propellers, in fact, could be mounted at any link of the multibody system. The second approach seems more favorable, but its complexity increases considerably since humanoid robots usually possess much more degrees of freedom (DoF)s than classical aerial manipulators. It is also worth mentioning the thrust estimation framework for the multi-link robot HYDRUS [23]. In that work, the external force, torque and the position of contact points are estimated by mounting an IMU for each link to estimate the linear and angular acceleration of the link's center of mass (CoM) [24]. However, the approach may not scale well if the multibody system is as complex as a humanoid robot and the propellers are jet engines. If jet engines are used as propellers, in fact, the complexity of their dynamics – due to also the dependence on the environmental conditions and the type of fuel being used – limits the application of the above mentioned estimation strategies. [25].

Another route one may follow for thrust estimation on flying multibody robots is to extend the methods developed for external force estimation on humanoid robots. More precisely, the thrust forces may be treated as external forces with known directions acting on the robot, considered as a *floating base* system [26]. One approach to estimate external forces for humanoid robots is to use *residual methods* tailored for humanoid robots [4], [27]. However, the approach requires a measurement or estimation of the robot joint torques, which may not always be available in practice. Furthermore, it requires to use full robot dynamics in the estimation process, and this can increase the complexity of the estimator design for systems with high number of DoFs [28].

To remove the need of torque sensors, the external wrench acting on a robot can be estimated using an F/T sensor attached to a robot link combined with Newton-Euler algorithms [29], [30], possibly framed as Maximum a Posteriori estimation problems that use Guassian noise models [31]. However, these methods provide instantaneous estimates of external forces and therefore they are prone to noise and outliers.

Other state-of-the-art techniques for external wrench estimation make use of the so-called robot's centroidal momentum [32]: a reduced robot model that relates the external wrenches acting on the robot with its kinematics. One can use the robot centroidal momentum to estimate a time-varying external wrench [33], or even the CoM kinematics [34], [35]. In addition, a simultaneous estimation for floating-based kinematics and contact force measurements was developed following an Extended Kalman Filtering (EKF) formulation [28]. The aforementioned methods can give inspiration to estimate thrust intensities of complex flying mutlibody robots.

This paper presents a framework to estimate on-line thrust forces for flying multibody systems with actuator dynamics. The estimator combines the robot centroidal momentum and propellers models in a single EKF formulation in order to increase the robustness and reliability of the estimated thrust. We consider the robot iRonCub [18], [19] as a case study and hence we use the jet engines models identified in our previous work [25]. Moreover, we further improve the estimator design by implementing a *covariance scheduling* strategy, which changes the filter behavior to account for a jet engine's failure in maintaining its own rotational speed. The proposed estimation framework is validated on the real iRonCub standing on weighing scales in order to measure the change of the robot's weight due to applied thrusts. The error between the estimated weight and the ground truth did not exceed 7% of the robot's total weight.

The reminder of the paper is organized as follows. Section II reviews some preliminary concepts that will be used in in the estimator design, section III details the procedure of designing the estimator. Section IV presents simulation results where the proposed estimator is used in flight simulations with the robot iRonCub. In addition, section V illustrates experiments performed with the robot iRonCub and the validation procedure. Further, it shows some test results of applying *covariance scheduling* to a case where failures in the jets have occurred. Finally, Section VI presents final remarks and future directions.

II. BACKGROUND

A. Notation and preliminaries

- $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is identity matrix of dimension n.
- $\boldsymbol{0}_{n\times m} \in \mathbb{R}^{n\times m}$ is the zero matrix of size $n \times m$.
- $\boldsymbol{a} \in \mathbb{R}^j$ is a vector of dimension j;
- $\boldsymbol{A} \in \mathbb{R}^{j\times k}$ is a matrix of size j times k;

- $\mathcal{S}(x) \in \mathbb{R}^{3\times3}$ is the skew symmetric matrix such that $\mathcal{S}(x)y = x \times y = -y \times x$, x and $y \in \mathbb{R}^3$, where $\times$ denotes the cross product operator in $\mathbb{R}^3$.
- $\mathcal{I}$ denotes the inertial frame of reference;
- $\mathcal{G}[\mathcal{I}]$ is a reference frame with the origin at the robot center of mass, and the orientation of the inertial frame.
- $\boldsymbol{e}_3 = [0, 0, 1, 0, 0, 0]^T$; $i \in [1, \dots, t]$;
- t is the number of thrusters attached to the robot;
- ${}^{\mathcal{I}}o_{\mathcal{G}} \in \mathbb{R}^3$ is the center of mass position represented in the inertial frame;
- ${}^{\mathcal{I}}o_i \in \mathbb{R}^3$ is the position of thruster i represented in the inertial frame;
- $\boldsymbol{r}_i = {}^{\mathcal{I}}\boldsymbol{o}_i - {}^{\mathcal{I}}\boldsymbol{o}_{\mathcal{G}}$; $\bar{\mathcal{S}}(\boldsymbol{r}_i) = \begin{bmatrix} \boldsymbol{I}_3 \\ \mathcal{S}(\boldsymbol{r}_i) \end{bmatrix}$.

B. Modeling of flying multibody robots

A flying multibody robot is usually composed of $n+1$ rigid bodies, connected by n one-DoF joints. Following the *floating base* formalism, we define the system *configuration space* as $\mathbb{Q} \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n$, and an element of $\mathbb{Q}$ is given by the triplet: $\boldsymbol{q} = ({}^{\mathcal{I}}\boldsymbol{o}_{\mathcal{B}}, {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}, \boldsymbol{s})$, where $({}^{\mathcal{I}}\boldsymbol{o}_{\mathcal{B}}, {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}})$ represents the position and orientation of the base frame $\mathcal{B}$ expressed in the inertial frame $\mathcal{I}$, while $\boldsymbol{s} \in \mathbb{R}^n$ are the 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 $\boldsymbol{\nu} = (\mathrm{v}_{\mathcal{B}}, \dot{\boldsymbol{s}})$ where $\mathrm{v}_{\mathcal{B}} = ({}^{\mathcal{I}}\dot{\boldsymbol{o}}_{\mathcal{B}}, {}^{\mathcal{I}}\boldsymbol{\omega}_{\mathcal{B}})$ is the linear and angular velocity of the base frame w.r.t. the inertial frame, while $\dot{\boldsymbol{s}}$ are the joint velocities. Recall that the base angular velocity satisfies ${}^{\mathcal{I}}\dot{\boldsymbol{R}}_{\mathcal{B}} = \mathcal{S}({}^{\mathcal{I}}\boldsymbol{\omega}_{\mathcal{B}})\,{}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}$.

III. ESTIMATOR DESIGN

The section details the thrust estimator with a focus on the process and measurement models for the EKF. In our design, we make the following assumptions:

1) the thrusters are rigidly attached to the robot links;
2) the direction of the thrust forces with respect to the associated link is known a-priori.

Assumption 1) implies we do not account for tilting propellers. However it is straightforward to extend the theoretical framework to this use case when assumption 2) is satisfied.

A. Centroidal Momentum Dynamics

In the estimation framework proposed in this paper, the relation between the robot and the jet thrust forces is obtained by exploiting the properties of the *centroidal momentum* dynamics, that is, the robot total momentum expressed w.r.t. the frame $\mathcal{G}[\mathcal{I}]$. More specifically, the rate of change of the centroidal momentum equals the summation of all the external forces and moments acting on the system [32]. In the case under analysis they are the thrust forces, the gravity force and the feet contact forces (while balancing on the ground). So, one has:

$$ {}_{\mathcal{G}[\mathcal{I}]}\dot{\boldsymbol{L}} = {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{feet}} - mg\boldsymbol{e}_3 + \sum_{k=1}^{t} \bar{\mathcal{S}}(\boldsymbol{r}_i)\,{}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{f}_i, \qquad (1) $$

where ${}_{\mathcal{G}[\mathcal{I}]}\dot{\boldsymbol{L}}$ indicates the rate of change of the robot centroidal momentum, m is the robot total mass, g is the norm of the gravitational acceleration and ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{f}_i \in \mathbb{R}^3$ is the force applied by thruster i, from which we can further highlight the thrust intensity T_i:

$$ {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{f}_i = {}^{\mathcal{I}}\boldsymbol{l}_i(\boldsymbol{q})T_i, \qquad (2) $$

${}^{\mathcal{I}}l_i(\boldsymbol{q}) \in \mathbb{R}^3$ is the direction of thruster i and $T_i \in \mathbb{R}_{\geq 0}$ is its thrust intensity. In addition, ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{feet}} \in \mathbb{R}^6$ represents the effect of left and right foot wrenches in the momentum dynamics:

$$ {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{feet}} = {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{X}^{\mathcal{LF}}\,{}_{\mathcal{LF}}\boldsymbol{W} + {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{X}^{\mathcal{RF}}\,{}_{\mathcal{RF}}\boldsymbol{W}, $$

where ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{X}^{\mathcal{LF}}(\boldsymbol{q})$ and ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{X}^{\mathcal{RF}}(\boldsymbol{q})$ are the so called Adjoint transformation matrices [36]. For the sake of clarity we rewrite the sum of thrust forces in Eq. (1) as:

$$ {}_{\mathcal{G}[\mathcal{I}]}\dot{\boldsymbol{L}} = {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{feet}}(\boldsymbol{q}) - mg\boldsymbol{e}_3 + \boldsymbol{A}(\boldsymbol{q})\boldsymbol{T}, \qquad (3) $$

where $\boldsymbol{T} = \begin{bmatrix} T_1 & T_2 & \dots & T_t \end{bmatrix}^T$ is the vector of thrust intensities and $\boldsymbol{A}(\boldsymbol{q}) = \begin{bmatrix} \bar{\mathcal{S}}(\boldsymbol{r}_1)^{\mathcal{I}}\boldsymbol{l}_1 & \bar{\mathcal{S}}(\boldsymbol{r}_2)^{\mathcal{I}}\boldsymbol{l}_2 & \dots & \bar{\mathcal{S}}(\boldsymbol{r}_t)^{\mathcal{I}}\boldsymbol{l}_t \end{bmatrix}$.

B. Thrusters dynamics

Depending on the type of thrusters used to power the flying robot, it may be necessary to include also the motors dynamics in the estimator design. More specifically, this is required when the dynamic response of the thrusters cannot be considered *instantaneous* with respect to the thrusters' inputs that regulate the generated thrust (e.g. rotor velocity, throttle, etc.). For the case study considered in this paper, we consider a second order dynamics for the thurst generation. In particular, we employ a jet-engines model that was designed to describe the relationship between the input throttle command (u_i for jet engine i) and the output thrust intensity (T_i) [25]. It was verified that a second order model can effectively describe the dynamics of jet engines, and this model encompasses a large class of existing propellers. Following this consideration, *Sparse Identification method for Nonlinear Dynamics* (SINDy) [37] was used to identify the governing terms in this model. After, a model structure was selected such that the nonlinear dynamics are affine with respect to the input u_i. For each jet engine i, the 2nd derivative of the thrust intensity is given by:

$$ \ddot{T}_i = f_{jet}(T_i, \dot{T}_i, u_i) = d_J(T_i, \dot{T}_i) + g_J(T_i, \dot{T}_i)v(u_i), $$

where:

$$ \begin{aligned} d_J(T_i, \dot{T}_i) &= K_T T_i + K_{TT} T_i^2 + K_D \dot{T}_i + K_{DD} \dot{T}_i^{\,2} + K_{TD} T_i \dot{T}_i + c \\ g_J(T_i, \dot{T}_i) &= B_U + B_T T_i + B_D \dot{T}_i \\ v(u_i) &= u_i + B_{UU} u_i^2, \end{aligned} $$

The values of the parameters:
$\boldsymbol{\theta} = \begin{bmatrix} K_T, K_{TT}, K_D, K_{DD}, K_{TD}, B_U, B_{TT}, B_D, c \end{bmatrix}^T$ can be computed by means of an EKF-based identification strategy, using data series that contain the input command u_i and the output measurement T_i samples collected during the jet engines experimental activities [25].

It is important to underline that some jet engines have their rotor speed measurements available, and this data can be used for the estimation of the thrust intensities. However, we decided to use a model that does not take into account the rotor speed, with the aim of designing a general estimation framework applicable to each type of flying multi-body robots.

C. Thrust Estimator Design

The general formulation of the thrust estimator considers the following non-linear system:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t), \quad \boldsymbol{w}(t) \sim \mathcal{N}(0, \boldsymbol{Q}(t)), \quad (4a)$$

$$\boldsymbol{z}(t) = h(\boldsymbol{x}(t)) + \boldsymbol{v}(t) \quad \boldsymbol{v}(t), \sim \mathcal{N}(0, \boldsymbol{R}(t)), \quad (4b)$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the system state, $\boldsymbol{z} \in \mathbb{R}^m$ is the measurement vector, $\boldsymbol{w}$ is the process noise and $\boldsymbol{v}$ is the measurement noise. The noises are uncorrelated, normally distributed with zero mean, and the known co-variance matrices: $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ describe the process and the measurement noise, respectively. In our design, the state $\boldsymbol{x}$ consists of the thrust intensities $\boldsymbol{T}$, their derivatives $\dot{\boldsymbol{T}}$, the centroidal momentum ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{L}$ and the feet contact wrenches ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$, i.e.:

$$\boldsymbol{x} = \left[\boldsymbol{T}^T, \dot{\boldsymbol{T}}^T, {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{L}^T, {}_{\mathcal{LF}}\boldsymbol{W}^T, {}_{\mathcal{RF}}\boldsymbol{W}^T\right]^T \in \mathbb{R}^{26}. \quad (5)$$

The input $\boldsymbol{u}(t)$, consists of the jets throttle inputs generated by a dedicated controller, namely $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 & \dots & u_t \end{bmatrix}^T$.

1) Measurement model: The measurements are obtained combining the on-board sensors outputs with robot kinematics and dynamics model. They consist of the centroidal momentum, and the feet contact wrenches expressed in their corresponding local frames ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$. Therefore the measurement vector $\boldsymbol{z}(t)$ can be written as follows:

$$\boldsymbol{z} = \left[{}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{L}^T, {}_{\mathcal{LF}}\boldsymbol{W}^T, {}_{\mathcal{RF}}\boldsymbol{W}^T\right]^T \in \mathbb{R}^{18}. \quad (6)$$

The centroidal momentum ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{L}$ is computed from the system kinematics as per [32], using the relationship below:

$${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{L} = \boldsymbol{J}_h(\boldsymbol{q})\boldsymbol{\nu}, \quad (7)$$

where $\boldsymbol{J_h(q)}$ is the *Centroidal Momentum Matrix* [32] that maps the system velocity to the centroidal momentum expressed in $\mathcal{G}[\mathcal{I}]$. In contrast, the feet contacts wrenches ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$ are obtained utilizing two 6-axis Force/Torque (F/T) sensors mounted on the robot.

2) Process model: The process dynamics combine the jets dynamics and the centroidal dynamics presented in section III-A. The reason to adding the two dynamical models is to overcome the limitations of each individual model. On one hand, the jet dynamics change with environmental conditions as stated previously. Also, they do not take into account the failure of a jet engine to deliver the expected thrust [25]. On the other hand, the centroidal dynamics assume the only contact forces applied to the robot are those of the thrusters, in addition to the contact wrenches with the ground. Further, they are affected by the uncertainties of the robot model, as well as the measurement bias in the left and right contact wrenches ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$ information from the F/T sensors. The process model is written as:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{T}} \\ \ddot{\boldsymbol{T}} \\ {}_{\mathcal{G}[\mathcal{I}]}\dot{\boldsymbol{L}} \\ {}_{\mathcal{LF}}\dot{\boldsymbol{W}} \\ {}_{\mathcal{RF}}\dot{\boldsymbol{W}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{T}} \\ f_{JET}(\boldsymbol{T}, \dot{\boldsymbol{T}}, \boldsymbol{u}) \\ \boldsymbol{A}(\boldsymbol{q})\boldsymbol{T} - mg\boldsymbol{e}_3 + {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{feet}}(\boldsymbol{q}) \\ \boldsymbol{0}_{6\times 1} \\ \boldsymbol{0}_{6\times 1} \end{bmatrix}.$$

The dynamics $f_{JET}(\boldsymbol{T}, \dot{\boldsymbol{T}}, \boldsymbol{u})$ defines the dynamics of all the four engines attached to the robot and is written as:

$$f_{JET}(\boldsymbol{T}, \dot{\boldsymbol{T}}, \boldsymbol{u}) = \begin{bmatrix} f_{jet}(T_1, \dot{T}_1, u_1) \\ f_{jet}(T_2, \dot{T}_2, u_2) \\ \vdots \\ f_{jet}(T_t, \dot{T}_t, u_t) \end{bmatrix}, \quad (8)$$

where each element $f_{jet}(T_i, \dot{T}_i, u_i)$ is the model of jet engine i described in III-B. The feet contact wrenches ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$ are added as states in order to provide filtered values to the centroidal dynamics instead of using direct measurements. Their dynamics are assumed to be unknown, and we assume that their change is purely driven by noise. Thus, their time derivatives are set to zero. The filtering of contact wrenches can be tuned using the corresponding elements of the process co-variance matrix $\boldsymbol{Q}$. Furthermore, the evolution of ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$ is assumed to be slow such that the filtering does not significantly modify their dynamics.

Remark: both the process and the measurement model depend, nonlinearly, on the robot configuration $\boldsymbol{q}$, while the measurement model depends also on the system velocity $\boldsymbol{\nu}$. To keep the filter design simpler, we decided not to add the robot configuration and velocity as part of the system state $\boldsymbol{x}$. We therefore assume that the robot configuration $\boldsymbol{q}$ and velocity $\boldsymbol{\nu}$ are *ideal*, so we treat them as perfect measurements to be used in the process dynamics and measurements models.

IV. SIMULATION RESULTS

Validation of the estimation strategy proposed in Sec. III is performed first in Gazebo simulation environment [38], with a simulated iRonCub robot. The flight controller is implemented in MATLAB-Simulink and provides to Gazebo reference throttle and robot joint torques at a frequency of 100 [Hz] [19]. In the testing scenario, the robot starts from a standing position, balancing on the ground with jet thrusts, then it takes-off, moves forward, rotates counter-clockwise around its vertical axis, and finally, it moves forward again. To emulate the jet dynamics, the simulator also includes the nonlinear jet-engine model described in Section III. To verify the estimator robustness w.r.t. model uncertainties, we modify the parameters of the jets model used by the estimator and jet control by adding to each parameter a uniformly-distributed random variable on the range of 5% of the actual parameter value. For better understanding of the simulated scenario, please refer to the associated video.

Fig. 2 shows the estimated thrusts of all four engines, compared to their actual values from Gazebo simulator. We also compared the performances of our estimation framework with a modified version of the estimator in III-C, which assumes that the jet dynamics are unknown (substituting the state $\dot{\boldsymbol{T}} = \boldsymbol{0}_{4\times 1}$ and assigning it with high uncertainty in the corresponding part of the co-variance matrix $\boldsymbol{Q}$). The rest of the gains in the co-variance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the same for both estimators, and they are tuned to achieve the best fit possible with the validation data. In both cases, the estimated thrusts converge to values close to the thrusts from Gazebo. Also, Fig. 3. shows that the change in the robot's centroidal

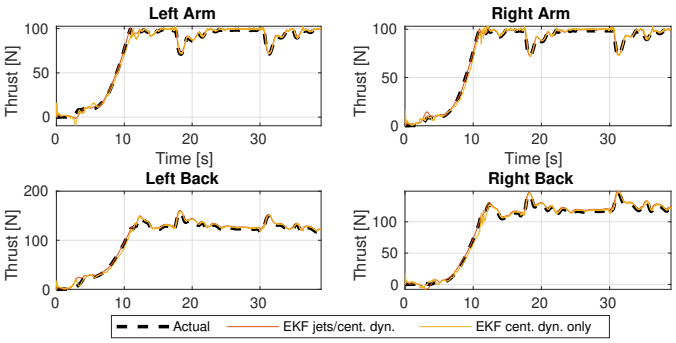

Figure 2. Thrust intensities values in simulation. The estimated thrust are in agreement with the values retrieved from Gazebo.

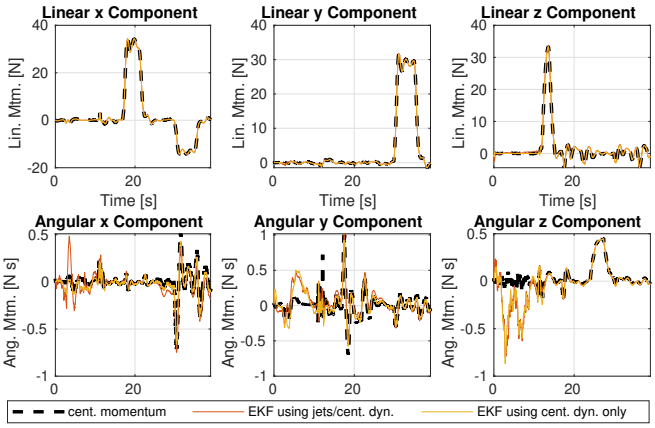

Figure 3. Centroidal momentum with its estimates using different EKF estimators during iRonCub simulated flight.

momentum are also being tracked by both estimators. However, limitations in our current control framework [19] do not allow to achieve high variations of the angular momentum.

V. EXPERIMENTAL VALIDATION

A. Overview

In this section we describe the experimental setup used to validate the thrust estimation algorithm presented in Section III. The experimental setup consists of the robot iCub [39] equipped with model jet engines: each forearm is equipped with a jet engine of the model number P100-RX [40], while the chest link is equipped with a jet-pack that consists of two jet engines of the model number P220-RXi [41]. The jet fuel is supplied from a tank placed off-board, and the selected fuel is kerosene Jet-A1. To protect the legs of the robot from the heat generated from the exhaust of the jet engines, several layers of a heat and fire resistant material (aerogel) are used to cover the lower part of the robot as shown in Fig. 1. Concerning the feet contact wrenches measurements ${}_{\mathcal{L}\mathcal{F}}\boldsymbol{W}$ and ${}_{\mathcal{R}\mathcal{F}}\boldsymbol{W}$, two F/T sensors internally mounted on the hip joints were used to compute the external wrenches with the algorithm proposed in [30]. The robot has also F/T sensors mounted at the feet, but they were not used because they suffered from considerable drift due to the high heat generated from the engines exhaust. The drift in the hips sensors was instead negligible for the duration of the experiment.

B. Ground truth and its temperature compensation

To obtain a source of ground truth, the robot is placed on weighting scales [42] to measure and record the weight of the robot throughout the experiment. Later, the recorded weight is compared with the vertical component of the wrench resulting from all the estimated thrust intensities, defined as:

$$ {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{jets}}(t) = mg\boldsymbol{e}_3 - \boldsymbol{A}(\boldsymbol{q})\hat{\boldsymbol{T}}(t), \tag{9}$$

where $\hat{\boldsymbol{T}}(t)$ is the estimated thrust intensities. The comparison is performed by computing the error $E_{\text{T}}(t)$ as:

$$E_{\text{T}}(t) = \big(W_{\text{scale1}}(t)+W_{\text{scale2}}(t)\big) \cdot g - {}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{jets}}(t) \cdot \boldsymbol{e}_3^T,$$

where $W_{\text{scale1}}(t)$ and $W_{\text{scale2}}(t)$ are the weights of the scales.

1) Temperature correction of scales data: The recorded weight from the scales suffered from a drift, that was caused by the change of their operating temperature due to the heat produced by the jet engines. It is known that weighting sensors usually have dependency on the temperature [43].

The F/T sensors placed on the robot feet are also equipped with temperature sensors. The temperature data have been recorded during the experiment, and they provide a reasonable indicator of the temperature of the scales due to their proximity. Fig. 4 shows the collected weight data along with the temperature data from the left and right foot F/T sensors.

The weight data is corrected exploiting the fact that, when the jets are turned OFF (in the initial and final part of the collected dataset), the measured robot weight should be the same. In addition, the temperature data shows a (approximately) linear increase. Therefore, a linear correction is applied to the weight data as shown below:

$$W_{\text{scale}}(T_{\text{scale}}) = W + \alpha_{\text{scale}} \cdot (T_{\text{scale}}(t) - T_{\text{scale0}}),$$

where $W_{\text{scale}}(T_{\text{scale}}(t))$ is the measured weight, which varies with the operating temperature, W is the true weight of the robot, $T_{\text{scale}}(t)$ is any given temperature value, and T_{scale0} is a initial temperature taken at the beginning of the test.

Thus, the values of $W_{\text{scale}}(T_{\text{scale}}(t))$ are recorded at time instants $t = t_0$ and $t = t_{\text{f}}$, and a correction $\alpha_{\text{scale}} \cdot (T_{\text{scale}}(t) - T_{\text{scale0}})$ is computed such that the aforementioned values are the same, in particular using the relation shown below:

$$\alpha_{\text{scale}} = \frac{W_{\text{scale}}(T_{\text{scale}}(t_{\text{f}})) - W_{\text{scale}}(T_{\text{scale}}(t_0))}{T_{\text{scale}}(t_{\text{f}}) - T_{\text{scale}}(t_0)}$$

We decided to use external sensors in the validation procedure rather than the feet F/T sensors, because temperature compensation of the F/T sensor is usually more complex than calibrating the weighing scales. In particular, it was hard to apply the assumption above to the feet F/T sensors because of nonlinear, temperature-related effects in the F/T measurements.

C. Experimental layout

The thrust intensities estimator is implemented in MATLAB/Simulink environment. YARP [44] is used to obtain

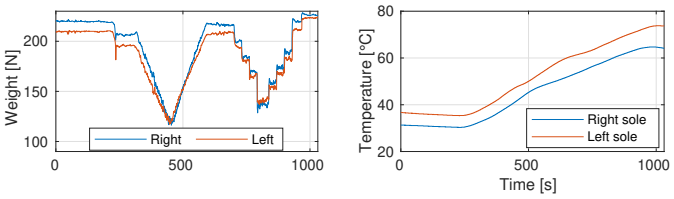

Figure 4. The left plot shows the drift in weight measurement of the scales. The right shows the temperature variation of the robot's soles.

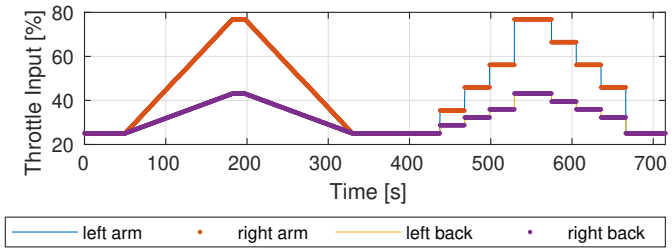

Figure 5. Plots shows the input throttle commands applied to the jet engines.

the information from the robot sensors as well as the jet engines' and to handle the communication between various software modules. Whole Body Toolbox [45] is used for integrating YARP into Simulink environment. Moreover, custom-made YARP-based drivers are used to obtain the measurements and to send control commands to the actuators.

The validation experiment consists of the robot iRonCub standing on the scales while the joints are set to a fixed state, this allowed the robot to be standing on two feet in a take-off configuration. The jet engines are commanded with input throttle values shown in Fig. 5. The values of the robot joint positions commands and the ratios of the throttle commands are taken from simulations of a take-off maneuver run in Gazebo simulation environment. The throttle commands, along with the sensors measurements, are logged and then applied to the estimator offline. Since the robot is standing still throughout all the experiment, the floating base pose and velocity are estimated using legged odometry and assuming that the feet do not move. A different and more appropriate method for flight estimation of the robot base kinematics is a subject of study in our future works.

The identified parameters of the jets model have the same values reported in [25]. In order to demonstrate the contribution of the jets dynamics (III-B), the centroidal dynamics (III-A) and the combined dynamics (III-C2), an estimator is implemented for each case. For instance, simulated jets dynamics are implemented to illustrate their response to the applied throttle input $\boldsymbol{u}$. The centroidal dynamics only estimator is implemented with modified co-variances as detailed in Sec. IV. The estimator with only the centroidal dynamics uses the sensors from the robot besides its centroidal dynamics, similar to some state-of-the-art external force estimators for humanoid robots [28] and [33]. We have stated the limitations of using each dynamics independently in III-C2. However, the goal of experiment is to show the validity of all sources of information (jets dynamics and centroidal dynamics).

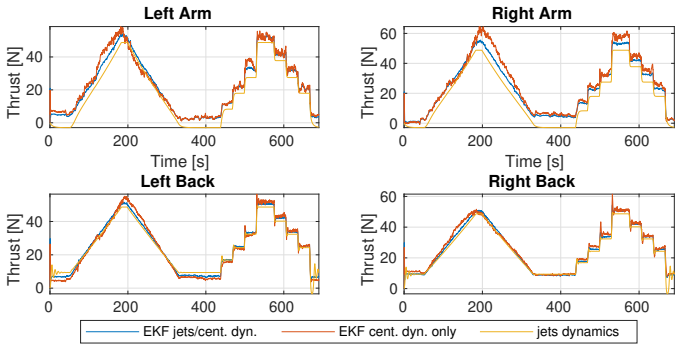

Figure 6. Plot of the thrust intensities estimation using EKF estimators, in addition to simulated jets dynamics.

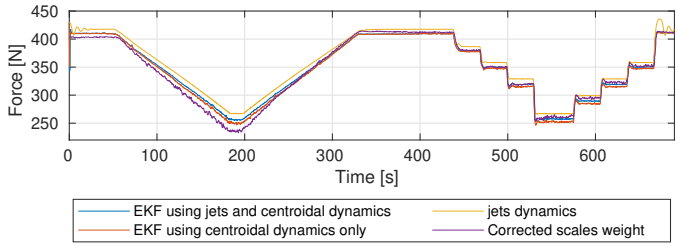

Figure 7. Plot of ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{jets}}(t) \cdot \boldsymbol{e}_3^T$ corresponding to different estimators along with the robot weight measured by the scales.

D. Experimental Results

Fig. 6 shows individual thrust intensities estimates for each estimation version to highlight the effect of using the centroidal dynamics or the jets dynamics alone, and combining both in a single estimator (the proposed estimator in III-C). The latter, despite a similar tuning, seems to provide smoother results w.r.t. the estimator version that implements only the centroidal dynamics. This is beneficial for feeding the estimates to a flight controller.

Fig. 7 compares the estimated quantity ${}_{\mathcal{G}[\mathcal{I}]}\boldsymbol{W}_{\text{jets}}(t) \cdot \boldsymbol{e}_3^T$ of each estimator versions mentioned in V-C compared to the corrected scales measurement. From Fig. 7, we can see that for both estimators the estimates resemble the corrected scales measurement. The absolute error does not exceed $35N$ for all estimators. During the ramp in the time period $t = 100s - 300s$, the estimation error is higher than in other time periods. This is due to many factors. For instance, the temperature correction model assumed in V-B1 may not be always adequate to describe the scales measurements dependency on the temperature, and there may be un-modeled non-linearities that we are, at the moment, unable to identify. Another possibility is that the temperature measurements have some inaccuracies because the temperature sensors are not directly attached to the load cells inside the scales. As a result, the error in the period $t = 100s - 300s$ seems larger than the differences in errors between each estimator and the other. However, the relative error with respect to the full weight of the robot always remains less than 7%. This can be considered as a general indication of the goodness of the estimation qualities for the considered use case.

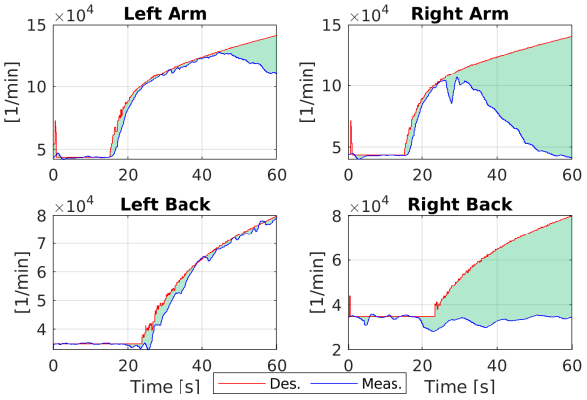

Figure 8. Rotational speeds of the jet engines. The red plot shows the desired speed, the blue plot shows the measured speed. The green highlighted area indicates the tracking error, which is very large in presence of a failure.

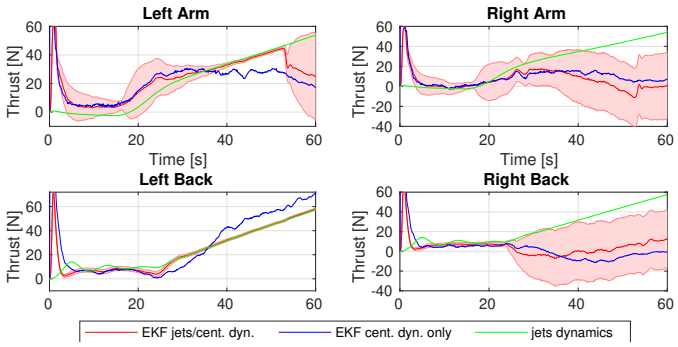

Figure 9. Estimated thrust intensities along with the corresponding uncertainty. The time periods when the uncertainty increases are associated with the failure of the engine.

E. Covariance scheduling experiment

The combination of the robot centroidal dynamics and the jet dynamics proposed in the thrust estimator design allowed us to implement a *covariance scheduling* algorithm, which changes the covariance of the estimated thrusts according to the failure of one or more jet engines. In order to detect the turbine's failure we computed the rotor speed error, which is a straightforward information to obtain from the jet engines.

The error indicates how far the jet engine is from the *nominal* behaviour, for which the model [25] was identified. When the error is larger than a chosen threshold, then the corresponding covariance of the estimated thrust in the matrix $\boldsymbol{Q}(t)$ is set to a value high enough for the estimator to disregard the jet dynamics during the failure period. It is worth mentioning that any other error indicator can be used to update the covariance matrices, if a suitable model is available to describe it.

During our experimental campaign, there were few tests where some of the jet engines failed to deliver the desired thrust as commanded by some chosen throttle input values. The failure was due to a defect in the fuel line, and it appeared in the rotational speed measurements of each jet engine, as shown in Fig. 8. The thrust estimator then changed its behaviour based on the error of the rotational speeds as shown in Fig. 9, where the estimator is tuned to follow the jet model when the jet engine's rotational speed error was smaller than a chosen threshold. In case of a rotational speed error higher than the threshold value, the estimator follows the robot centroidal dynamics by assuming high uncertainty in the corresponding jet engine model. In Fig. 9, even though there is not real ground truth to be compared with, we can see that the estimator tries to follow the most reliable source of information available. In this experiment, the centroidal-based estimation was affected by disturbances due to the bias in the feet contact wrenches ${}_{\mathcal{LF}}\boldsymbol{W}$ and ${}_{\mathcal{RF}}\boldsymbol{W}$ measurements, but still it was more reliable than jet dynamics-based during engines failures.

VI. CONCLUSIONS

This paper presented a thrust intensities estimation strategy for flying multibody robots, with a focus on the use case of a jet-powered humanoid robot. In order to overcome the difficulty of achieving direct thrusts measurements on the humanoid robot, the proposed algorithm implements a variation of centroidal momentum estimation algorithms used in humanoid robotics, and combines it with the jet engines dynamical models. Experimental validation is also presented using simulation and the real robot iRonCub. The experimental validation procedure consists of applying thrusts commands while the robot is standing on scales and logging the measured weight, then comparing this measurement with the equivalent weight computed from the estimated thrust intensities. Further, a *covariance scheduling* strategy was implemented to account for the failures in jet engines, Exploiting the combined dynamics, the estimator shifts its belief towards robot model in case one or more thrusters exhibits unmodeled phenomena.

The experimental validation procedure did not include the case of the robot detaching from the ground (flying), this was due to many reasons. Because of the complexity of the setup and the higher safety requirements of experiments with jet engines, limiting the experiment to the robot standing was necessary at this stage of developing and implementing control and estimation algorithms. Furthermore, an experimental setup different from the scales to validate the estimation was not available at the moment. As demonstrated, we resorted to a non-invasive way to measure the weight of the robot, which even if it does not validate the individual estimated thrust intensities, it can still provide a good indication of the quality of estimation. Another limitation concerns the floating base estimation to be used for the proposed thrust estimation: since the robot was kept fixed on the ground, forward kinematics was used to estimate the floating base pose. When the robot detaches from the ground, it will then be essential to incorporate additional information in the estimator, such as the IMU or Visual/Lidar Odometry. Addressing all these requirements and therefore test the estimator while performing the first flight is certainly our main focus for future activities.

Moreover, would be of particular importance to perform robustness analysis on the estimator against model and measurements uncertainties: more specifically, the effect of the measurements bias and other non-linearities in the F/T sensors information, as well as the inevitable uncertainties in the robot's inertial parameters (that might change during flight) and in the jet dynamics (that are affected by environmental conditions).

REFERENCES

[1] C. F. Liew, D. DeLatte, N. Takeishi, and T. Yairi, "Recent developments in aerial robotics: A survey and prototypes overview," *arXiv preprint arXiv:1711.10085*, 2017.

[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[3] B. Yüksel, C. Secchi, H. H. Bülthoff, and A. Franchi, "A nonlinear force observer for quadrotors and application to physical interactive tasks," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2014, pp. 433–440.

[4] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1623–1630.

[5] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance control of vtol uavs with a momentum-based external generalized forces estimator," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2093–2099.

[6] T. Tomić, C. Ott, and S. Haddadin, "External wrench estimation, collision detection, and reflex reaction for flying robots," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1467–1482, 2017.

[7] J. Cayero, B. Morcego, and J. Cugeró, "Estimating external disturbances in uavs using linear and spin momenta," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4317–4323.

[8] C. D. McKinnon and A. P. Schoellig, "Unscented external force and torque estimation for quadrotors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5651–5657.

[9] 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.

[10] K. Kim, P. Spieler, E. Lupu, A. Ramezani, and S. Chung, "A bipedal walking robot that can fly, slackline, and skateboard," *Science Robotics*, vol. 6, no. 59, p. eabf8136, 2021.

[11] Z. Huang, B. Liu, J. Wei, Q. Lin, J. Ota, and Y. Zhang, "Jet-hr1: Two-dimensional bipedal robot step over large obstacle based on a ducted-fan propulsion system," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017.

[12] 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*, Sep. 2014, pp. 3417–3422.

[13] M. Pitonyak and F. Sahin, "A novel hexapod robot design with flight capability," in *2017 12th System of Systems Engineering Conference (SoSE)*, June 2017, pp. 1–6.

[14] A. Kalantari and M. Spenko, "Design and experimental validation of hytaq, a hybrid terrestrial and aerial quadrotor," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4445–4450.

[15] A. Bozkurt, A. Lal, and R. Gilmour, "Aerial and terrestrial locomotion control of lift assisted insect biobots," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sept 2009, pp. 2058–2061.

[16] L. Daler, S. Mintchev, C. Stefanini, and D. Floreano, "A bioinspired multi-modal flying and walking robot," *Bioinspiration & Biomimetics*, vol. 10, no. 1, p. 016005, 2015. [Online]. Available: http://stacks.iop.org/1748-3190/10/i=1/a=016005

[17] L. Daler, J. Lecoeur, P. B. Hählen, and D. Floreano, "A flying robot with adaptive morphology for multi-modal locomotion," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nov 2013, pp. 1361–1366.

[18] 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, 2017.

[19] G. Nava, L. Fiorio, S. Traversaro, and D. Pucci, "Position and attitude control of an underactuated flying humanoid robot," in *2018 IEEE-RAS 18th Int. Conf. on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.

[20] 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: https://robotics.sciencemag.org/content/2/13/eaaq1026

[21] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor uavs equipped with a servo robot arm," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4014–4020.

[22] M. Wilmsen, C. Yao, M. Schuster, S. Li, and K. Janschek, "Nonlinear wrench observer design for an aerial manipulator," *IFAC-PapersOnLine*, vol. 52, no. 22, pp. 1–6, 2019.

[23] M. Zhao, K. Kawasaki, K. Okada, and M. Inaba, "Transformable multirotor with two-dimensional multilinks: modeling, control, and motion planning for aerial transformation," *Advanced Robotics*, vol. 30, no. 13, pp. 825–845, 2016.

[24] F. Shi, M. Zhao, T. Anzai, X. Chen, K. Okada, and M. Inaba, "External wrench estimation for multilink aerial robot by center of mass estimator based on distributed imu system," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1891–1897.

[25] G. L'Erario, L. Fiorio, G. Nava, F. Bergonti, H. A. O. Mohamed, E. Benenati, S. Traversaro, and D. Pucci, "Modeling, identification and control of model jet engines for jet powered robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2070–2077, 2020.

[26] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[27] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 409–415.

[28] M. Benallegue, P. Gergondet, H. Audrerr, A. Mifsud, M. Morisawa, F. Lamiraux, A. Kheddar, and F. Kanehiro, "Model-based external force/moment estimation for humanoid robots with no torque measurement," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3122–3129.

[29] A. Del Prete, L. Natale, F. Nori, and G. Metta, "Contact force estimations using tactile sensors and force/torque sensors," in *Human Robot Interaction*, vol. 2012, 2012, pp. 0–2.

[30] S. Traversaro, "Modelling, estimation and identification of humanoid robots dynamics," Ph.D. dissertation, Italian Institute of Technology, 2017. [Online]. Available: https://github.com/traversaro/traversaro-phd-thesis

[31] C. Latella, "Human whole-body dynamics estimation for enhancing physical human-robot interaction," *arXiv preprint arXiv:1912.01136*, 2019.

[32] D. E. Orin and A. Goswami, "Centroidal momentum matrix of a humanoid robot: Structure and properties," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 653–659.

[33] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," in *2015 IEEE-RAS 15th Int. Conf. on Humanoid Robots (Humanoids)*, 2015, pp. 556–563.

[34] L. Hawley, R. Rahem, and W. Suleiman, "Kalman filter based observer for an external force applied to medium-sized humanoid robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1204–1211.

[35] S. Piperakis, M. Koskinopoulou, and P. Trahanias, "Nonlinear State Estimation for Humanoid Robot Walking," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3347–3354, Oct 2018.

[36] S. Traversaro and A. Saccon, *Multibody dynamics notation (version 2)*. Technische Universiteit Eindhoven, Nov. 2019, dept. of Mechanical Engineering. Report locator DC 2019.100.

[37] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016. [Online]. Available: https://www.pnas.org/content/113/15/3932

[38] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2004. (IROS 2004)*, pp. 2149 – 2154, 2004.

[39] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. S. Victor, F. Becchi, G. Sandini, and G. Metta, "The design of the icub humanoid robot," *International journal of humanoid robotics*, vol. 9, no. 04, p. 1250027, 2012.

[40] "Jetcat p100-rx," 2019. [Online]. Available: https://www.jetcat.de/en/productdetails/produkte/jetcat/produkte/hobby/Engines/p100_rx

[41] "Jetcat p220-rxi," 2019. [Online]. Available: https://www.jetcat.de/en/productdetails/produkte/jetcat/produkte/hobby/Engines/p220-rxi

[42] *CPWplus Weighing Scales - Datasheet*, Adam Equipment, 2020, this datasheet is subject to modifications. [Online]. Available: https://www.adamequipment.com/cpwplus-200

[43] D. Helmick, A. Okon, and M. DiCicco, "A comparison of force sensing techniques for planetary manipulation," in *2006 IEEE Aerospace Conference*, 2006, pp. 14 pp.–.

[44] F. Paul, C. Elena, D. Daniele, P. Ali, M. Giorgio, and N. Lorenzo, "A middle way for robotics middleware," *Journal of Software Engineering for Robotics*, 2014.

[45] D. Ferigo, S. Traversaro, F. Romano, and D. Pucci, "A generic synchronous dataflow architecture to rapidly prototype and deploy robot controllers," *International Journal of Advanced Robotic Systems*, 2020.