Advanced Nonlinear Control for an Omnidirectional Spherical Robot Integrating Aerial and Ground Mobility

Davide Spitaleri[1][a], Gianluca Pepe[1][b], Maicol Laurenza[2][c], Silvia Milana[1][d],
Flavio Ceccarelli[1] and Antonio Carcaterra[1][e]

[1]*Department of Mechanical and Aerospace Engineering, University "La Sapienza", Eudossiana street, 18, Rome, Italy*
[2]*Department of Engineering, Niccolò Cusano University, Don Carlo Gnocchi street, 3, Rome, Italy*

Keywords: Spherical Drone, Rolling Robot, Nonlinear Control, Model-Based Control, Hybrid Dynamic System.

Abstract: This paper presents an innovative control strategy for an overactuated omnidirectional spherical drone, capable of both flying and rolling on the ground. The control system, based on Feedback Local Optimality Control (FLOP), utilizes a comprehensive dynamic model that facilitates smooth transitions between flight and rolling modes, optimizing energy efficiency and enhancing maneuverability. Key features include an advanced decision-making mechanism for contact detection and a constrained control allocation algorithm that respects physical limitations. Virtual simulations have demonstrated the control system's robustness and responsiveness. This spherical drone design not only extends the capability to navigate complex environments but also enables energy conservation during ground transport.

1 INTRODUCTION

In recent years, mobile robotics has increasingly focused on spherical robots due to their agility and stability in complex scenarios, making them ideal for industrial inspections, exploration of rugged terrains, urban search and rescue operations in difficult environment (Diouf et al., 2024). Integrating in a single system the flight capabilities of traditional multicopters with the ability to roll on surfaces offers unique versatility and energy-saving modes during ground transport. However, the realization of such systems remains challenging, necessitating innovations in mechanics, sensing, and control. Current solutions often present significant limitations that hinder the widespread adoption of such robots in real applications. Simpler designs mount traditional coaxial or quadrotor UAVs inside a protective cage using connecting rods, but these underactuated configurations typically suffer from low control capability, especially in rolling maneuvers (Hou et al., 2013; Zhou et al., 2010). Quadrotors with variable-pitch propellers, offer versatile operations on land, water and in the air, but are complex to manage and maintain (Yao et al., 2021). Improved maneuverability can be achieved with configurations allowing relative rotation between the quadrotor and the shell along a connecting rod equipped with ball bearings, but this solution favours rolling in one direction (Dudley et al., 2015; Kalantari & Spenko, 2014). Complete omnidirectional motion in flight and rolling can be attained with spherical flying robots equipped with a 3-axis gimbal mechanism. This design absorbs impacts by allowing the external cage to rotate passively, though it introduces high mechanical complexity (Atay et al., 2021; Briod et al., 2014). Finally, a hybrid aerial and ground mobility robot with a fully actuated multirotor system rigidly connected to the spherical shell offers promising omnidirectional mobility but requires further development to fully evaluate control and maneuverability in real-world scenarios (Sabet et al., 2019).

[a] https://orcid.org/0009-0009-4285-7567
[b] https://orcid.org/0000-0002-1444-7765
[c] https://orcid.org/0000-0002-3493-0965
[d] https://orcid.org/0000-0001-9484-0315
[e] https://orcid.org/0000-0002-0355-3925

Spitaleri, D., Pepe, G., Laurenza, M., Milana, S., Ceccarelli, F. and Carcaterra, A.
Advanced Nonlinear Control for an Omnidirectional Spherical Robot Integrating Aerial and Ground Mobility.
DOI: 10.5220/0013020300003822
In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024)* - *Volume 1*, pages 684-691
ISBN: 978-989-758-717-7; ISSN: 2184-2809

Among the various solutions, omnidirectional flying spherical robots with reversible propellers, over-actuated and gimbal-less, represent an emerging and promising area of research, offering high versatility with reduced mechanical complexity and improved structural robustness. However, this configuration requires more sophisticated control to manage impact moments and transitions between flight and ground rolling movement, such as in landing and take-off operations. To pursue this objective, in this study we implemented a nonlinear control algorithm that optimizes performance both in flight and rolling, and that can effectively manage the transitions during landing and take-off operations.

The Feedback Local Optimality Principle (FLOP) control, belonging to a special class of optimal controllers named Variational Feedback Controllers (Pensalfini et al., 2017; Pepe et al., 2018; Pepe et al., 2023; Pepe et al., 2019), allows for the integration of nonlinear behaviors, typical of spherical attitude control (Spitaleri et al., 2024), and for following assigned trajectories in feedback. The validation of the algorithm has been carried out through numerical simulations in a multibody environment, testing the robustness of the controller. The control hardware is emulated in the Simulink MathWorks virtual environment, where both the hardware and its integrated sensors are simulated, allowing for comprehensive testing and development without the need for physical components.

The paper is organized as follows: Section 2 describes the design of the newly assembled drone; Section 3 presents the equations of motion that characterize a rolling-flying sphere; Section 4 details the control system design; and Section 5 presents numerical results through simulations in flight, rolling and mixed modes.

2 SPHERICAL ROBOT DESIGN

The spherical robot developed in this study features a structure composed of three main elements, as depicted in Figure 1: an internal central unit, an external spherical frame, and eight cross-shaped rods. The internal central part houses the battery and electronic components, while the external spherical frame allows for rolling and provides protection. The aluminium rods connect the internal part to the external frame of radius R , ensuring structural rigidity and enabling the mounting of eight rotors in a star configuration, one for each rod, as proposed in (Brescianini & D’Andrea, 2018). The over actuation and the specific arrangement allow the motors to push uniformly in different directions, ensuring omnidirectional control. The other parts of the robot's chassis are 3D printed from polymer material (PLA), with the spherical shell featuring a thin and lattice-like geometry that provides both lightness and robustness, minimize aerodynamic interference and ensures smooth rolling. The central unit includes the Pixhawk 6X flight computer running the open-source PX4 control software. The onboard sensor system comprises three six-axis MEMS inertial units, two barometers, and a magnetometer, offering stabilization and noise reduction. The robot's rotors consist of brushless outrunner motors connected to fixed-pitch, three-blade symmetric propellers capable of generating variable thrust in both rotational directions. The electric motors are controlled by two ESCs, each with four independent outputs that modulate the voltage applied to the motors via pulse-width modulation (PWM) control and can reverse the direction of motor rotation. The six-cell 3300 mAh lithium polymer battery provides approximately ten minutes of hover time, delivering a peak power of 6.5 kW.

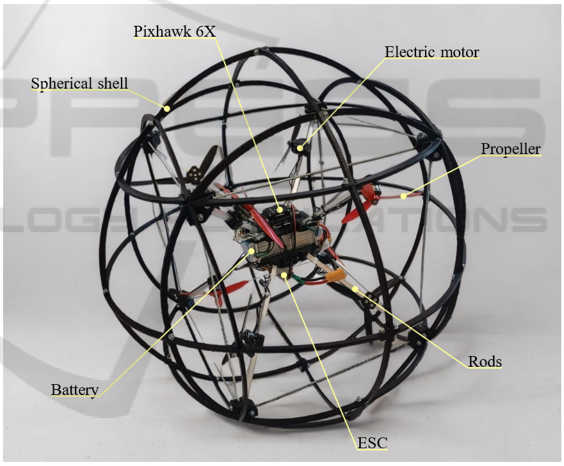

Figure 1: The newly assembled spherical robot prototype, featuring a 0.32 m radius spherical shell with a total weight of 2.3 kg. The robot is equipped with eight thrust motors and eight reversible three-bladed propellers. The onboard hardware is centrally positioned within the shell.

3 DYNAMIC MODEL

The dynamic model of a spherical drone capable of flying and rolling requires an adaptive modeling approach depending on the operational conditions: (i) during flight, the model must account for the mass, inertial and aerodynamic forces. This involves analyzing the flight dynamics, which includes the thrust of the rotors; (ii) during rolling, it is necessary to introduce a holonomic constraint that allows replicating the reaction force of the support plane and the tangential friction force. The entire dynamic model serves a dual purpose: on one hand, it enables testing and validation of the system in a simulated environment; on the other hand, it is crucial for implementing the model-based nonlinear control, described in the following sections. Given the omnidirectional spherical behavior of the robot, it is convenient to express the equations of motion using quaternions$\boldsymbol{q} = [q_0, q_1, q_2, q_3]^\top$, refer to (Nikravesh et al., 1985). Consequently, the rotation matrix $\boldsymbol{R}_b^n$ from the body frame $\{b\}$ to the inertial frame $\{n\}$ (East-North-Up, ENU) is defined as function of quaternions. Being $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ the body angular velocity expressed in the inertial reference frame, the body's rotational kinematics is described using quaternions as well through the transformation matrix $\boldsymbol{E}$, i.e. $\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{E}(\boldsymbol{q})\boldsymbol{\omega}$. The absolute angular velocity $\boldsymbol{\omega}_{r_i}$ of the r_i-th rotor comprises two components: the relative rotation of the electric motor $\dot{\theta}_{r_i}$ oriented along the unit vector $\vec{\mathrm{e}}_z^{r_i}$ of each rotating frame $\{r_i\}$ and the angular velocity component of the drone $\boldsymbol{\omega}$ (see Figure 2 for further details), leading to:

$$\boldsymbol{\omega}_{r_i} = \boldsymbol{R}_b^n(\boldsymbol{q})\boldsymbol{R}_{r_i}^b(\theta_{r_i})\dot{\theta}_{r_i}\vec{\mathrm{e}}_z^{r_i} + \boldsymbol{\omega} = \dot{\theta}_{r_i}\vec{\mathrm{e}}_{z_i} + 2\boldsymbol{E}^\top\dot{\boldsymbol{q}} \tag{1}$$

Where $\vec{\mathrm{e}}_{z_i} = \boldsymbol{R}_{r_i}^n\vec{\mathrm{e}}_z^{r_i}$ now indicates the direction of the action unit vector of the motor's relative rotation, expressed in reference frame $\{n\}$. The rotation matrix $\boldsymbol{R}_{r_i}^b$ allows transformation from the rotating rotor reference frame $\{r_i\}$ to the body reference frame $\{b\}$ and is defined by an XYZ rotation sequence $\boldsymbol{R}_{r_i}^b = \boldsymbol{R}_x(\alpha_i)\boldsymbol{R}_y(\beta_i)\boldsymbol{R}_z(\theta_{r_i})$, with the constant parameters α_i e β_i describing the specific mounting orientation of the individual stator r_i.

From the outset, we choose to divide the vector of generalized coordinates into two parts, namely the body and the rotors, as we intend to use only the body dynamics, while still including the gyroscopic effects arising from the rotor dynamics. The differential equations of the dynamic model are derived using the Euler-Lagrange formulation with the vector of generalized coordinates $\boldsymbol{\eta} = [\boldsymbol{\eta}_b, \boldsymbol{\eta}_r]^\top$. Here, $\boldsymbol{\eta}_b = [x, y, z, q_0, q_1, q_2, q_3]^\top$ is associated to the rigid body dynamics, including the coordinates $\boldsymbol{r} = [x\ y\ z]^\top$ of the center of mass G and the orientation described by quaternions. The rotor rotation coordinates are collected in $\boldsymbol{\eta}_r = [\theta_{r_1}, \dots, \theta_{r_N}]^T$. The kinetic energy T and potential energy U of the body and the N-rotors r_i are:

$$T = \frac{1}{2}m\dot{\boldsymbol{r}}^\top\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{\omega}^\top\boldsymbol{R}_b^n\boldsymbol{I}_b\boldsymbol{R}_n^b\boldsymbol{\omega} + \\ + \sum_{i=1}^{N}\left[\frac{1}{2}m_r\dot{\boldsymbol{r}}_{r_i}^\top\dot{\boldsymbol{r}}_{r_i} + \frac{1}{2}\boldsymbol{\omega}_{r_i}^\top\boldsymbol{R}_{r_i}^n\boldsymbol{I}_r\boldsymbol{R}_n^{r_i}\boldsymbol{\omega}_{r_i}\right] \tag{2}$$

$$U = -m\boldsymbol{r}^\top\boldsymbol{g} - \sum_{i=1}^{N} m_r\boldsymbol{r}_{r_i}^\top\boldsymbol{g} \tag{3}$$

where m and $\boldsymbol{I}_b$ are respectively the mass and inertia matrix of the robot, including the spherical shell, frame, stator part of the electric motors, onboard electronics and batteries; m_r and $\boldsymbol{I}_r$ are the mass and inertia matrix of the rotor part of the electric motors, including the propeller; $\boldsymbol{g} = [0\ 0\ g]^\top$ is the gravitational acceleration vector; $\boldsymbol{r}_{r_i} = \boldsymbol{r} + \boldsymbol{R}_b^n\boldsymbol{d}_{r_i}^b$ represents the position of the center of mass of the rotating parts, with $\boldsymbol{d}_{r_i}^b$ the fixed geometric arrangement vector of the i-th rotor relative to the barycenter G, expressed in the body frame $\{b\}$.

The Lagrangian $\mathcal{L}$ is defined as the difference between the kinetic and potential energy of the system $\mathcal{L} = T - U$, and is associated with the problem's constraint conditions. For this reason, the extended Lagrangian $\mathcal{L}^*$, which includes holonomic constraints, is introduced:

$$\mathcal{L}^*(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \lambda, t) = \mathcal{L}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, t) + \sum_k \lambda_k\psi_k(\boldsymbol{\eta}, t) \tag{4}$$

where the Lagrange multipliers λ_k are associated with the constraint equations $\psi_k(\boldsymbol{\eta}, t) = 0$. It is important to note that, when representing the motion of a rigid body in 3D-space using quaternions, a holonomic constraint must be introduced to ensure the quaternion unit norm condition:

$$\psi_q(\boldsymbol{\eta}) = \|\boldsymbol{q}\| - 1 = 0 \tag{5}$$

In rolling mode, an additional holonomic constraint is required to express the contact with a support plane, assumed in this case to be horizontal:

$$\psi_{\mathrm{c}}(\boldsymbol{\eta}) = z - R = 0 \tag{6}$$

Proceeding with the Euler-Lagrange formulation, the following differential equations are obtained:

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{\eta}}}\right) - \dfrac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}} = \lambda_q\dfrac{\partial\psi_q}{\partial\boldsymbol{\eta}} + \lambda_c\dfrac{\partial\psi_c}{\partial\boldsymbol{\eta}} + \boldsymbol{\tau} \\ \dfrac{d^2\psi_q}{dt^2} = 0 \quad , \quad \dfrac{d^2\psi_c}{dt^2} = 0 \end{cases} \tag{7}$$

Where $\boldsymbol{\tau}$ is the vector of generalized forces acting on the robot, detailed in the following paragraph. The first system equation (7) can be rearranged in the classic form, comprising the mass matrix $\boldsymbol{M}(\boldsymbol{\eta})$, the Coriolis matrix $\boldsymbol{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$, the gravitational force vector $\boldsymbol{\zeta}(\boldsymbol{\eta})$, and the matrix $\boldsymbol{\Psi}(\boldsymbol{\eta})$ associated to the vector of

Lagrange multipliers $\boldsymbol{\lambda} = \left[\lambda_q, \lambda_c\right]^\top$ for the corresponding holonomic constraints:

$$\boldsymbol{M}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \boldsymbol{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{\zeta}(\boldsymbol{\eta}) = \boldsymbol{\Psi}(\boldsymbol{\eta})^\top \boldsymbol{\lambda} + \boldsymbol{\tau} \tag{8}$$

The first seven rows of equation (8) describe the rigid body dynamics, while the remaining rows pertain to the rotor dynamics. The vector $\boldsymbol{\tau}$ expresses the generalized forces acting on the robot, coming from: the thrust forces $\boldsymbol{f}_{r_i} = f_{r_i}\vec{e}_{z_i}$ generated by propellers along their respective axes of rotation; the contact forces $\boldsymbol{f}_c$ generated by the rolling surface and the spherical shell through the contact point C (see Figure 2); the aerodynamic resistance torques from the propellers $\boldsymbol{t}_{r_i} = t_{r_i}\vec{e}_{z_i}$ and the torques from the electric motors $\boldsymbol{t}_{m_i} = t_{m_i}\vec{e}_{z_i}$. In our case, the contribution produced by aerodynamic drag force acting on the sphere is neglected because of the low operating translational speed. Due to the typically high values of the rotors, around tens of thousands of RPM, the body is significantly influenced by the Coriolis terms associated to $\dot{\boldsymbol{\eta}}_r$. In contrast, the rotors' dynamics is not directly affected by Coriolis terms and is typically faster than the body's dynamics because of a shorter transient time. To develop a model suitable for the controller, we streamline the system by substituting $\ddot{\boldsymbol{\eta}}_r$, obtained from the last eight rows of equation (8), into the first rows related to body dynamics. In the end, the new compact system model, along with the corresponding constraint equations, takes the form:

$$\begin{cases} \boldsymbol{M}_b\ddot{\boldsymbol{\eta}}_b + \boldsymbol{C}_{11}\dot{\boldsymbol{\eta}}_b + \boldsymbol{C}_{12}\dot{\boldsymbol{\eta}}_r + \boldsymbol{\zeta}_b = \dfrac{\partial\psi_{\mathrm{q}}}{\partial\boldsymbol{\eta}_b}\lambda_{\mathrm{q}} + \dfrac{\partial\psi_c}{\partial\boldsymbol{\eta}_b}\lambda_c + \boldsymbol{\tau}_c + \boldsymbol{\tau}_m \\ \dfrac{d^2\psi_{\mathrm{q}}}{dt^2} = 0 \quad , \quad \dfrac{d^2\psi_c}{dt^2} = 0 \end{cases} \tag{9}$$

with $\boldsymbol{M}_b = \boldsymbol{M}_{11} - \boldsymbol{M}_{12}\boldsymbol{M}_{22}^{-1}\boldsymbol{M}_{21}$. The remaining forces and torques include those generated by contact interactions and by the electromotive torques $\boldsymbol{t}_{m_i}$:

$$\boldsymbol{\tau}_c = \left[\boldsymbol{f}_c^\top, \left(2\boldsymbol{E}(\boldsymbol{d}_c \times \boldsymbol{f}_c)\right)^\top\right]^\top \tag{10}$$

$$\boldsymbol{\tau}_m = \begin{bmatrix} \sum_{i=1}^{N} k\boldsymbol{t}_{m_i} \\ 2\boldsymbol{E}\sum_{i=1}^{N}(\boldsymbol{d}_{r_i} \times k\boldsymbol{t}_{m_i} - \boldsymbol{t}_{m_i}) \end{bmatrix} \tag{11}$$

In fact, experimental bench tests show a linear relationship $f_{r_i} = kt_{m_i}$ between the thrust forces of the rotors and their electromotive torque through a constant parameter k (as further explained in Section 4.2). The contact force vector $\boldsymbol{f}_c$ is simply the tangential friction force, directed along the velocity of the contact point $\dot{\boldsymbol{r}}_c$ and defined as:

$$\boldsymbol{f}_c = \frac{\dot{\boldsymbol{r}}_c}{\|\dot{\boldsymbol{r}}_c\|}\mu(\sigma)\,\lambda_c \tag{12}$$

Where the Lagrange multiplier λ_c represents the normal reaction force of the plane, while the friction coefficient μ is selected as a nonlinear function of the relative slip $\sigma = \|\dot{\boldsymbol{r}}_c\|$ given by $\mu(\sigma) = \mu_d \tanh(\epsilon\sigma)$, with μ_d the dynamic friction coefficient and ϵ a smoothing coefficient.

When the drone is in flight, the governing equation (9) is simplified by removing the contact constraint ψ_c with the corresponding Lagrange multiplier λ_c, and the vector of contact interaction $\boldsymbol{\tau}_c$. Finally, the set of second-order differential equations describing the robot's motion in rolling and flight modes can be reduced to affine first-order systems to optimally utilize the FLOP control algorithm described in the next section. Defining the state vector as $\boldsymbol{x} = \left[\boldsymbol{\eta}_b^\top, \dot{\boldsymbol{\eta}}_b^\top, \int\lambda_q dt\,, \int\lambda_c dt\right]^\top \in \mathbb{R}^{16}$ for the rolling mode and $\boldsymbol{x} = \left[\boldsymbol{\eta}_b^\top, \dot{\boldsymbol{\eta}}_b^\top, \int\lambda_q dt\right]^\top \in \mathbb{R}^{15}$ for the flying mode, we have:

$$\dot{\boldsymbol{x}} = \boldsymbol{\phi}(\boldsymbol{x}, \dot{\boldsymbol{\eta}}_r) + \boldsymbol{D}(\boldsymbol{x})\boldsymbol{t}_m \tag{13}$$

Where $\boldsymbol{t}_m = \left[t_{m_1}, \ldots t_{m_i}, \ldots, t_{m_N}\right]^\top$ collects the rotors' electromotive torques and is considered the control action of our system. Here, $\boldsymbol{\phi}(\boldsymbol{x}, \dot{\boldsymbol{\eta}}_r)$ highlights the parametric dependence on the motor RPMs, appropriately estimated from the commands sent to the electric motor controllers.

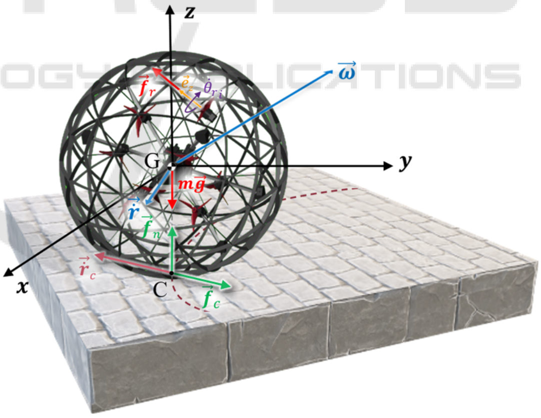

Figure 2: Model of the flying-rolling sphere, showing the inertial coordinate system $\{x, y, z\}$ translated in G; the sphere velocity vectors $\dot{\boldsymbol{r}}$, $\boldsymbol{\omega}$; the i-th rotor axis $\vec{e}_z^{\,r_i}$ along with its relative velocity $\dot{\theta}_{r_i}$; the contact point velocity $\dot{\boldsymbol{r}}_c$; and the contact, thrust and gravity forces.

4 CONTROL SYSTEM DESIGN

The control architecture designed for the autonomous navigation of the drone is illustrated in Figure 3. The flight operator can plan a trajectory $\boldsymbol{x}_T(t)$, that

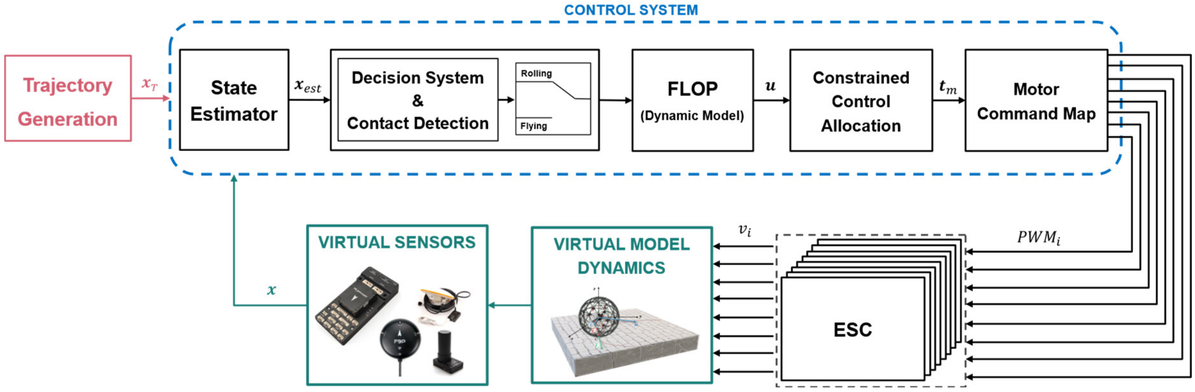

Figure 3: Architecture of the Control System designed for the spherical drone, which receives as input the target trajectory to be followed and the virtualized onboard sensor data and gives as output the PWM signal to the virtual model.

includes both flight segments and rolling segments. The control system receives this trajectory and onboard sensor data as inputs and outputs PWM commands to the electric motors controllers. The control system consists of five sequential blocks: (i) the State Estimator assesses the drone's position, orientation, and translational and rotational velocities; (ii) the Decision System and Contact Detection determines whether the drone is in flight, rolling, landing, or take-off phases; (iii) the FLOP Control Algorithm calculates the desired "fictitious" force and torque actions needed to follow the trajectory; (iv) the Constrained Control Allocation Algorithm converts the desired actions into physical actions, specifically the rotors' electromotive torques t_{m_i}, while respecting actuators limits; (v) the Motor Commands maps the electromotive torques to commands for the ESCs. To test and tune the control system, we developed a virtual reference model that simulates the dynamics in a multibody environment.

In the State Estimator block, the drone's state estimation is achieved by properly processing data from the sensors. Specifically, the system's orientation is estimated using accelerometer, gyroscope, and magnetometer data from the IMU platform, implementing filtering and data fusion techniques typical of the Attitude and Heading Reference System (AHRS) To enhance the position and velocity estimates, a Kalman filter was implemented, integrating accelerometer data and GPS data from an RTK (Real-Time Kinematic) dual-antenna system, which provides centimeter-level accuracy with an update frequency of 10 Hz.

The Decision System, based on onboard sensor measurements and the output from the Contact Detection System, determines the operating state of the drone, such as being in free flight or in rolling contact. The Contact Detection System analyzes the vertical acceleration data provided by the IMU and compares it with the acceleration predicted by both the rolling contact model (9) and the associated free-flight model. The system then identifies the state based on which predicted acceleration value is closest to the measured data and select the corresponding model to employ in the control algorithm. A properly defined moving average filter helps the evaluation system to reduce false positives caused by noisy measurements and potential signal oscillations.

4.1 Nonlinear FLOP Control

The control system designed for guiding and controlling the drone is a nonlinear control type known as FLOP (Pepe et al., 2018). Instead of directly providing the N electromotive torques t_{m_i} as control variables, this system uses seven fictitious actions $\boldsymbol{u} = [u_1, \dots, u_7]^\top$. These actions include three translational forces along the drone's body axes xyz and four pseudo-torque actions affecting the orientation described by quaternions. This approach allows for the subsequent resolution of a control allocation problem for the electromotive torque variables, taking into account the actual constraints that exist. This process is explained in more detail in the following section. As illustrated in (Pepe et al., 2018), the FLOP technique utilizes an affine reference dynamic model of the type (13), linear in $\boldsymbol{u}$, to provide an approximate solution to the optimal local control problem of a constrained minimization problem of the form:

$$\min_{\boldsymbol{u}} J = \int_0^T \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_T)^\top \boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{x}_T) + \frac{1}{2}\boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}\, dt \tag{14}$$

$$\textit{Subject to:} \qquad \dot{\boldsymbol{x}} = \boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{B}\boldsymbol{u}$$

The control vector $\boldsymbol{u}$ is premultiplied by the input matrix $\boldsymbol{B}$, properly derived by introducing the fictitious actions in our model. The solution provided by the FLOP technique yields an explicit feedback control formulated as:

$$\boldsymbol{u} = \boldsymbol{R}^{-\top}\boldsymbol{B}^{\top}\left[\frac{\partial\boldsymbol{\phi}^{\top}}{\partial\boldsymbol{x}} - \frac{1}{\Delta t}\boldsymbol{I}\right]^{-1}\boldsymbol{Q}^{\top}(\boldsymbol{x}-\boldsymbol{x}_{T}) + \boldsymbol{B}^{+}[\dot{\boldsymbol{x}}_{T} - \boldsymbol{\phi}(\boldsymbol{x}_{T})] \quad (15)$$

Here, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are matrices containing the control tuning gains, Δt is a positive control parameter, and the symbol $+$ denotes the pseudo-inverse of matrix $\boldsymbol{B}$.

4.2 Constrained Control Allocation and Motor Command

To determine the necessary electromotive torques for controlling the drone, a constrained allocation problem is solved, which involves a linear programming problem outlined as follows:

$$\begin{aligned}&\min_{\boldsymbol{t}_m} \quad \chi_1\|\boldsymbol{Bu}-\boldsymbol{Dt}_m\|^2 + \chi_2\|\boldsymbol{t}_m - \boldsymbol{t}_m^{*}\|^2 \\ &\textit{Subject to:} \qquad \boldsymbol{t}_{m_-} \le |\boldsymbol{t}_m| \le \boldsymbol{t}_{m_+}\end{aligned} \quad (16)$$

The goal is to minimize the sum of two terms, each weighted by a scaling factor, χ_i. The first term, the most critical, requires that the electromotive torques $\boldsymbol{t}_m$ multiplied by the associated Jacobian matrix $\boldsymbol{D}$ closely match the "fictitious" components $\boldsymbol{u}$ suggested by the FLOP through $\boldsymbol{B}$. The second term ensures that the new $\boldsymbol{t}_m$ is as close as possible to the $\boldsymbol{t}_m^{*}$ determined in the previous time step to obtain a smoother control action. Moreover, the minimization problem (16) is constrained by the hard limits that the electromotive torques $\boldsymbol{t}_m$ must be within the actual maximum and minimum values, denoted by $\boldsymbol{t}_{m-}$ and $\boldsymbol{t}_{m+}$, representing the lower and upper boundaries, respectively.

Finally, the Motor Command block is responsible for mapping the electromotive torques from the Constrained Control Allocation to the corresponding PWM commands to be sent to the ESCs, defined here in percentage terms. Based on the hypothesis described in (Brescianini & D'Andrea, 2018), the PWM signal can be expressed as a polynomial function of the electromotive torque, defining the conversion map:

$$PWM_i = c_1 t_{m_i} + c_2\sqrt{|t_{m_i}|}\cdot sign(t_{m_i}) + c_3 \quad (17)$$

Using a load cell, a series of experimental measurements were conducted directly on the motor-propeller assembly to determine the operating points related to thrust force and transmitted torque. The conversion map was then analytically calculated by determining the values of $c_1 = 218.98\ \mathrm{Nm}, c_2 = 41.96\sqrt{\mathrm{Nm}}$ and c_3=50 through a least-squares problem. Additionally, the coefficient $k = 131.14$ introduced in (11) was evaluated.

5 NUMERICAL SIMULATION

The system was tested using numerical simulations in the Simscape multibody environment within MATLAB & Simulink, serving as a tool for design support and verification of the control system. The virtual model comprises a multibody setup with 14-degree-of-freedom, six for the sphere and eight for the rotors, and with specific modifications that introduce misalignments compared to the model used in the control algorithm. This testing and validation approach allowed us to assess the robustness of the algorithm under the following conditions: uncertainty in the system's mass and inertia of about 5%; actual electromotive torques with a random variation of around 10%; thrust forces and aerodynamic torques on the blades misaligned by up to 10%; contact dynamics calculated using the penalty method, where the normal force $\boldsymbol{f}_n$ and tangential friction force $\boldsymbol{f}_t$ are modelled differently:

$$\boldsymbol{f}_n = [0,0,f_n]^{\top}, \qquad \boldsymbol{f}_t = \frac{\dot{\boldsymbol{r}}_c}{\|\dot{\boldsymbol{r}}_c\|}\mu f_n$$

$$f_n = \begin{cases} k_s\delta + c_s\dot{\delta} & \delta \ge 0,\ \dot{\delta} \ge 0 \\ k_s\delta & \delta \ge 0, \dot{\delta} < 0 \\ 0 & \delta < 0 \end{cases} \quad (18)$$

where $\delta = R - z$ is the virtual penetration of the sphere into the plane, k_s and c_s are the contact stiffness and damping, respectively, and the friction coefficient μ is defined as a piecewise linear function of the slip. The virtualization of the sensors was implemented by introducing noise signals into the state measured from the virtual system to emulate the real onboard sensors of the Pixhawk 6X.

The numerical simulations include three types of tests: flight mode control, ground rolling mode control, and a mixed mode with landing and take-off.

5.1 Flying Mode Test

The first test was conducted to verify the drone's performance in flight mode, independently controlling its position and orientation along a predetermined trajectory. The drone was tasked with following a helical trajectory with a oscillation period of 10 s, while maintaining a constant arbitrary angular velocity $\boldsymbol{\omega}_T$. The simulation results, shown in Figure

4 demonstrate that the robot can follow the assigned trajectory well, both in position and orientation, with a settling position error less than 0.3 m despite starting from an initial condition different from the trajectory's start point.

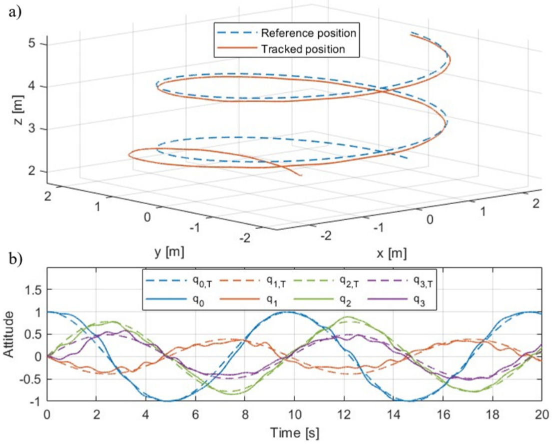

Figure 4: Flying mode test results, showing the position trajectory in 3D view (a) and the attitude via quaternion (b).

5.2 Rolling Mode Test

A second simulation was conducted to assess the performance of ground rolling mobility. In this case, the robot was subjected to a hard maneuvering test under low-traction conditions, simulating an icy surface with a dynamic friction coefficient of 0.1.

Additionally, the control system was assumed to be unaware of the actual traction conditions, considering a dynamic friction coefficient of 0.5 into (12). This allows to test the system's robustness under unexpected slippery conditions. The target trajectory involved a sharp curve of radius 0.5 m, to be navigated at a speed of 2 m/s. The results shown in Figure 5 indicate that the maximum deviation from the target was about 20 cm, considered satisfactory given the challenging conditions of the test.

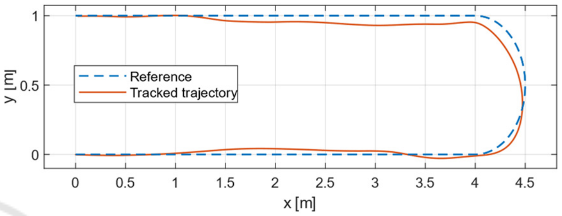

Figure 5: Tracked trajectory in rolling mode.

5.3 Mixed Mode

The final test evaluated the drone's ability to transition between flight and rolling modes. The robot

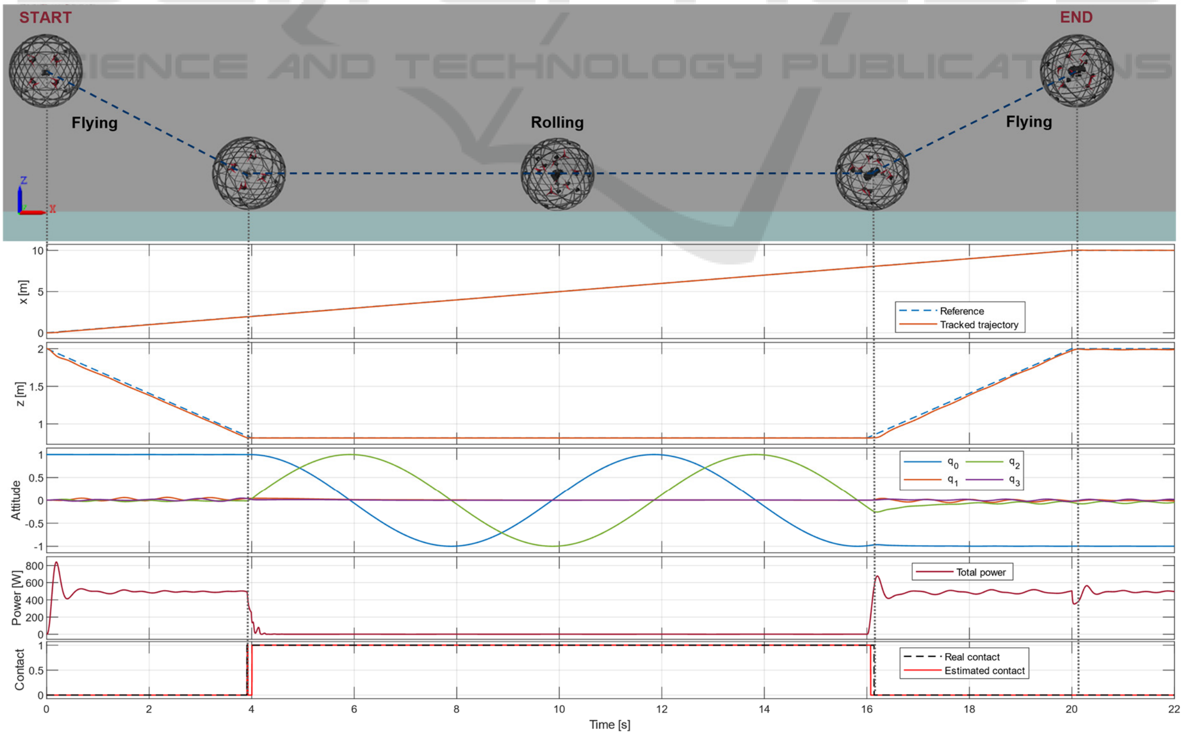

Figure 6: Results of the simulation in mixed mode, showing the behavior of the spherical drone during the three phases of landing, rolling on ground and take-off.

was tasked with three phases: landing, rolling, and take-off, while maintain a constant target speed $\dot{x}_T$ of 0.5 m/s (Figure 6Figure 6). During landing and take-off, a controlled vertical speed $\dot{z}_T$ of 0.3 m/s was required at specific key times. Before landing, the control system did not have information about the contact plane's location and imposed a constant descent speed, until the Decision and Contact Detection system identified contact around the 4th second mark, and promptly switched to rolling control. Due to the low speed of impact, there was no noticeable bounce, and the motors were managed by the controller to achieve the desired rolling motion, resulting in a significant drop in the required power. At the 16th second mark, the take-off phase was initiated and the Contact Detection system, unaware of the take-off command, automatically recognized the new flight state and subsequently enabled the free flight control, where the orientation became stable again and the required power increased accordingly.

6 CONCLUSIONS

This study presents a comprehensive approach to the design and control of a flying-rolling spherical drone. The key contributions include the development of a nonlinear control system, referred to as FLOP, which effectively manages the drone’s complex dynamics in both aerial and ground-based operations. The control architecture integrates state estimation, decision-making, and force allocation to achieve precise control in varying operational modes, including flight, rolling, landing, and take-off. Numerical simulations validated the proposed control methods, demonstrating the drone's ability to follow planned trajectories and maintain stability under different conditions. Future work involves applying the FLOP algorithm to the newly constructed prototype and conducting experimental tests to validate its performance and robustness in real-world scenarios.

REFERENCES

Atay, S., Bryant, M., & Buckner, G. (2021). The Spherical Rolling-Flying Vehicle: Dynamic Modeling and Control System Design. *Journal of Mechanisms and Robotics*, *13*(5).

Brescianini, D., & D’Andrea, R. (2018). An omni-directional multirotor vehicle. *Mechatronics*, *55*, 76-93.

Briod, A., Kornatowski, P., Zufferey, J.-C., & Floreano, D. (2014). A Collision-resilient Flying Robot. *Journal of Field Robotics*, *31*(4), 496-509.

Diouf, A., Belzile, B., Saad, M., & St-Onge, D. (2024). Spherical rolling robots—Design, modeling, and control: A systematic literature review. *Robotics and Autonomous Systems*, *175*, 104657.

Dudley, C. J., Woods, A. C., & Leang, K. K. (2015). A micro spherical rolling and flying robot. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),

Hou, K., Sun, H. X., Jia, Q. X., Zhang, Y. H., Wei, N. Z., & Meng, L. (2013). Analysis and Design of Spherical Aerial Vehicle's Motion Modes. *Applied Mechanics and Materials*, *411-414*, 1836-1839.

Kalantari, A., & Spenko, M. (2014). Modeling and Performance Assessment of the HyTAQ, a Hybrid Terrestrial/Aerial Quadrotor. *IEEE Transactions on Robotics*, *30*(5), 1278-1285.

Nikravesh, P. E., Wehage, R. A., & Kwon, O. K. (1985). Euler Parameters in Computational Kinematics and Dynamics. Part 1. *Journal of Mechanisms, Transmissions, and Automation in Design*, *107*(3), 358-365.

Pensalfini, S., Coppo, F., Mezzani, F., Pepe, G., & Carcaterra, A. (2017). Optimal control theory based design of elasto-magnetic metamaterial. *Procedia Engineering*, *199*, 1761-1766.

Pepe, G., Antonelli, D., Nesi, L., & Carcaterra, A. (2018). Flop: Feedback local optimality control of the inverse pendulum oscillations. Proceedings of ISMA 2018 - International Conference on Noise and Vibration Engineering and USD 2018 - International Conference on Uncertainty in Structural Dynamics, Leuven.

Pepe, G., Doria, A., Roveri, N., & Carcaterra, A. (2023). Vibration energy harvesting for cars: semi-active piezo controllers. *Archive of Applied Mechanics*, *93*(2), 663-685.

Pepe, G., Laurenza, M., Antonelli, D., & Carcaterra, A. (2019). A new optimal control of obstacle avoidance for safer autonomous driving. 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2019,

Sabet, S., Agha-Mohammadi, A. A., Tagliabue, A., Elliott, D. S., & Nikravesh, P. E. (2019). Rollocopter: An Energy-Aware Hybrid Aerial-Ground Mobility for Extreme Terrains. 2019 IEEE Aerospace Conference,

Spitaleri, D., Pepe, G., Laurenza, M., Milana, S., & Carcaterra, A. (2024). Enhancing Spherical Rolling Robot Control for Slippery Terrain. 2024 13th International Workshop on Robot Motion and Control (RoMoCo),

Yao, Y., Deng, Z., Zhang, X., & Lv, C. (2021). Design and Implementation of a Quadrotor-Based Spherical Robot. 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC),

Zhou, Q. L., Zhang, Y., Qu, Y. H., & Rabbath, C. A. (2010). Dead reckoning and Kalman filter design for trajectory tracking of a quadrotor UAV. Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications,