machines

MDPI

Article

Computer-Aided Design, Multibody Dynamic Modeling, and Motion Control Analysis of a Quadcopter System for Delivery Applications

Carmine Maria Pappalardo *, Marco Del Giudice, Emanuele Baldassarre Oliva, Littorino Stieven and Alessandro Naddeo

Department of Industrial Engineering, University of Salerno, Via Giovanni Paolo II 132, 84084 Fisciano, Italy
* Correspondence: cpappalardo@unisa.it; Tel.: +39-089-963472

Abstract: This paper elaborates on the modeling and control of an Unmanned Aerial Vehicle (UAV) for delivery purposes, thereby integrating computer-aided design, multibody dynamic modeling, and motion control analysis in a unified framework. The UAV system designed in this study and utilized for item delivery has a quadcopter structure composed of four arms connected to a central trunk. In this investigation, the proposed design of the delivery drone is systematically modeled employing the multibody approach, while SIMSCAPE MULTIBODY is the software used for performing the dynamic analysis and for devising the final design of the control system. To this end, starting from the CAD model designed using SOLIDWORKS, the control system of the quadcopter is developed by performing dynamic simulations in the MATLAB/SIMULINK environment. Additionally, another fundamental contribution of this paper is the analytical derivation of the nonlinear set of algebraic constraint equations peculiar to the present multibody system, which characterizes the kinematics of the delivery drone and describes the relative angular velocity imposed between two rigid bodies as nonholonomic constraints. Furthermore, as discussed in detail in this paper, the choice of the propulsion system and the design of the individual components heavily depends on the structural and functional needs of the UAV under study. On the other hand, the control system devised in this work is based on cascaded Proportional-Integral-Derivative (PID) controllers, which are suitable for achieving different maneuvers that are fundamental for the motion control of the delivery drone. Therefore, the final performance of the UAV system is a consequence of the regulation of the feedback parameters that characterize the PID controllers. In this respect, the paper presents the refining of the parameters characterizing the PID controllers by using both an internal MATLAB tool, which automatically tunes the controller gains of single-input single-output systems, and by observing the resulting transient behavior of the UAV system, which is obtained through extensive dynamical simulations. The set of numerical results found in this investigation demonstrates the high performance of the dynamical behavior of the UAV system designed in this work.

Keywords: UAV system; delivery drone; multibody dynamics; holonomicandnonholonomic constraints; PID control; SOLIDWORKS; MATLAB/SIMULINK; SIMSCAPE MULTIBODY

check for updates

Citation: Pappalardo, C.M.; Del Giudice, M.; Oliva, E.B.; Stieven, L.; Naddeo, A. Computer-Aided Design, Multibody Dynamic Modeling, and Motion Control Analysis of a Quadcopter System for Delivery Applications. *Machines* **2023**, *11*, 464. https://doi.org/10.3390/machines11040464

Academic Editor: Xuping Zhang

Received: 27 February 2023
Revised: 29 March 2023
Accepted: 6 April 2023
Published: 8 April 2023

1. Introduction

1.1. Formulation of the Problem of Interest for This Investigation

The acronym UAV stands for Unmanned Aerial Vehicle. This family of aerial mechanical systems represents a class of aircraft that can be remotely piloted by following a given flight plan or by using a ground-based command station [1]. The principal differences between conventional aircraft and UAVs can be found in avionics, communication systems, data management, and flight autonomy [2,3]. Originally, UAVs were typically used for military applications, such as reconnaissance missions and aerial attacks [4]. Subsequently, their use also shifted to civil applications due to their potential for exploration and monitoring. The main advantage of UAV systems compared to conventional aircraft is the capacity to carry out complex and dangerous missions at lower economic costs [5,6]. Another peculiarity of UAV systems is the possibility of serving as a support for robotic mechanical systems for performing multiple repetitive tasks in the case of large-scale industrial applications [7–9]. In fact, since UAVs can work autonomously, their modern use is becoming interconnected with several robotic systems [10].

In mechanical and industrial engineering applications, the use of Computer-Aided Design (CAD) programs and Computer-Aided Engineering (CAE) software plays a central role [11,12]. In this context, the need for the Integration of Computer-Aided Design and Analysis (I-CAD-A) is day by day becoming of paramount importance in several engineering areas [13–15]. To this end, Multibody System Dynamics (MBD) represents a fundamental tool for performing reliable and advanced computer simulations of the kinematic and dynamic behavior of rigid–flexible articulated mechanical systems. General-purpose multibody software allows for automatically generating and numerically solving the differential-algebraic form of the equations of motion describing the dynamic behaviors of complex mechanical systems. This process is based on the preliminary geometric analysis carried out employing computer-aided design software, which is currently used exclusively for the creation of complex geometric shapes [16,17]. It is therefore apparent that the integration of computer-aided design and analysis tools will be prominent in modern engineering applications.

From a general perspective, the process of analysis can be seen as the verification of the performance of an already existing mechanical system. The synthesis process, on the other hand, is represented by the design of a mechanical system that is aimed at achieving the required specifications of the project [18,19]. In the mathematical modeling of a multibody system, the system degrees of freedom are the minimum number of independent parameters necessary and sufficient to uniquely define the configuration of the mechanical system [20,21]. For example, for a rigid body moving in space, six degrees of freedom are defined: three rotational and three translational coordinates. In the mathematical modeling of multibody mechanical systems made up of rigid bodies, two general formulation approaches can be followed. These are the Minimal Coordinate Formulation (MCF) and the Redundant Coordinate Formulation (RCF). In the former approach, the analysis is based on a number of generalized coordinates equal to the number of degrees of freedom of the dynamical system to be modeled. In the latter approach, on the other hand, the number of generalized coordinates employed in the description of the motion of the mechanical system of interest is greater than the system degrees of freedom. While the first approach is adopted only in the case of machines and mechanisms having a simple structure, the second approach is typically used in the dynamic analysis of complex multibody systems since it allows for the systematic development of the equations of motion [22,23]. Therefore, the RCF is the general multibody formulation utilized in this work for modeling the case study of this investigation.

1.2. Literature Review

The scientific literature over the last twenty years has mainly been focused on the construction of UAV systems and on the development of various control methods. Kant et al. statically analyzed individual components of pre-existing drone models [24]. Ballous et al. designed a drone to transport medical items [25]. Zhafri et al. focused on optimizing the materials for the mechanical design of drones [26]. Meenakshipriya et al. designed and fabricated a 3D-printed UAV model, and then tested its dimensions and roughness [27]. Benito et al. suggested a series of criteria to design a UAV with a payload [28]. Tnunay et al. designed an auto-tuner to make the proportional-derivative controller more robust [29]. De Simone et al. designed a proper set of control inputs for a UAV system, considering the wind disturbance action [30]. Hai-long et al. devised a proportional-integral nonlinear controller for the attitude control and the trajectory tracking of drone systems [31]. Amir and Abbass proposed a simplified mathematical model of a UAV system [32]. Razinkova et al. studied an adaptive compensation controller for the position tracking error of quadrotor UAVs [33].

Fernando et al. focused on a simulator and a laboratory platform to facilitate the future development of control algorithms, automatic navigation systems, and trajectory planning for quadrotors [34]. Pessen et al. proposed a method for tuning the parameters of the PID controller of a drone [35].

The body of knowledge apparent from the literature concerning UAV systems is quite extended, thereby demonstrating the growing importance of such topics for engineering applications. In recent years, the popularity of UAVs has grown because they can meet a wide range of needs. In particular, quadrotors play a central role in aerial development due to their versatility in both the civil and military sectors. Fixed-wing drones such as airplanes require a proper track for takeoff and landing, while quadcopters can take off vertically from a given airstrip [36]. Furthermore, drones can be used for the delivery of goods, as proposed in this research. In this vein, the challenges to address concern modeling, such as dimensioning the components and choosing the best frame configuration, as well as in controlling the various movements, such as hovering, rolling, and pitching [37]. Additionally, another important topic concerning the mathematical modeling of UAVs is the identification of their structural model, and for example, the state estimation of the system dynamics for attaining the attitude control [38–40].

A rigid multibody system is a dynamical system that consists of solid bodies connected by kinematic joints that limit their relative motion [41–43]. These are mechanical devices that restrict the motion of the members of a machine or mechanism, and they are formed by an appropriate coupling of surfaces between two contiguous bodies. These contacting surfaces are called conjugate surfaces and they identify a kinematic element, while the set of two kinematic elements forms a kinematic pair [44]. From a mathematical point of view, the equations describing the motion of multibody mechanical systems are nonlinear ordinary differential equations, while the equations modeling the motion restrictions of the kinematic pairs are nonlinear algebraic equations, thereby leading to a highly nonlinear set of differential-algebraic equations [45]. The study of multibody dynamics, also called direct dynamics, is the analysis of how these systems move under the influence of specific applied forces, torques, and moments. The study of the opposite problem, i.e., which set of generalized forces is required to make the mechanical system move in a specific manner, is called inverse dynamics, and this is an important topic that is closely related to the field of nonlinear control and state estimation [46,47].

1.3. Scope and Contributions of This Study

The goal of this research paper is to thoroughly describe the modeling of a UAV system serving as a delivery drone, an autonomous flying machine that was specifically conceived for the transportation and delivery of light goods. This derives from complete CAD model creation to multibody model development and the subsequent design of an ad hoc control system. The first challenge to face in this research endeavor is the choice of the components of the propulsion system, which depends on the object to carry, representing the payload of the UAV system. In the case of the study considered in this work, the delivery drone must carry six beers, forming a pack of about two kilograms. Although the present delivery drone is designed for beer delivery, it can easily carry any object of the same weight and therefore can be used for different purposes. The virtual prototype of the delivery drone designed in this study is medium-sized, needing a space of about two square meters to land. However, it is still suitable for delivery in modern apartment buildings. The design concept devised in this investigation envisages a transport that is fully automated, with no intervention of the customer on the drone required in order to recover the goods.

In this paper, a CAD model of the virtual prototype of the UAV system serving as a delivery drone was developed first. Then, by using the geometric information provided by the CAD model, a multibody model was created. By doing so, the resulting simplified CAD model was subsequently imported into the MATLAB simulation environment using the software called SIMSCAPE MULTIBODY. The delivery drone is provided with the input of the trajectory to follow, which was designed using MATLAB and SIMULINK. This is necessary to determine the position error and the force quota needed for achieving the system guidance. For this purpose, the motion control is realized using separate PID controllers, which were designed with one for each movement. More specifically, while the PID controller regulating the drone altitude was manually calibrated by observing the system feedback behavior, the cascade PID controllers for the two-dimensional cruise motion were adjusted with an internal MATLAB/SIMULINK tool called PID Tuner. In addition to the flight scenarios concerning the take-off, cruise, and landing maneuvers, a dynamical simulation of the unloading of the beers was also performed employing SIMSCAPE MULTIBODY. The dynamical simulations carried out in this study using SIMSCAPE MULTIBODY are robust and efficient. The numerical results obtained in this investigation using MATLAB are fully satisfying for the computer-aided design and analysis of the virtual prototype of the proposed delivery system.

One of the main contributions of the present research work is the development of a multibody model of the UAV system serving as a delivery drone that was promptly designed in this investigation. The analytical development of an accurate multibody model, as well as its subsequent computer implementation and dynamical simulations, is aimed at the design and testing of the laws of motion to be provided as the input functions to control the motion of the mechanical system under study. As shown in detail in the manuscript, the delivery drone is modeled as a multibody system composed only of rigid bodies endowed with a three-dimensional motion. In the multibody model devised in this paper, the proper formulation of a nonlinear set of differential equations that describes the dynamic behavior of the system under study is of primary importance. Furthermore, the introduction of an additional set of nonlinear algebraic equations is equally essential for the correct modeling of the effect of the mechanical joints, as well as the presence of driving constraints imposed on the dynamical system to be controlled. Additionally, in the mathematical description of all the mechanical actions applied to the delivery drone, an appropriate set of externally applied forces, torques, and moments is included in the multibody model. In the multibody model of the UAV system analyzed in this investigation, for the mathematical description of the driving constraints associated with the imposed rotational motion of the propellers, a set of nonholonomic algebraic constraints is specifically devised and analytically formulated, thereby representing one of the most important novelties and contributions presented in this research work.

To conceptually summarize the approach proposed and employed in this investigation, one can refer to the schematic flowchart represented in Figure 1.

In the flowchart reported in Figure 1, all the fundamental steps followed in this paper are schematically illustrated to clarify the scope and the contributions of the present research work.

1.4. Organization of the Manuscript

The remaining parts of this manuscript are organized as follows. Section 2 is focused on the description and the mathematical modeling of the propulsion system of the virtual prototype developed in this work. In Section 3, the main steps followed in the computer-aided design of the UAV system of interest for this investigation are illustrated, the individual mechanical components designed using SOLIDWORKS are described, and the reasons behind the design choices made are specified. Section 4 is devoted to the multibody modeling of the virtual prototype of the UAV system designed in this research, including the formulation of a new algebraic set of nonholonomic constraints suitable for mathematically modeling the relative angular velocity imposed between the propellers and the chassis of the delivery drone. In Section 5, the key features investigated in the development of the control system devised for the proposed delivery drone are explained. Section 6 describes the computer simulations carried out for analyzing the dynamical behavior of the delivery drone, while Section 7 presents and discusses the numerical results found in this investigation. In Section 8, a summary of the completed work is presented, the conclusions drawn in this research are reported, and some directions for future works are commented on.

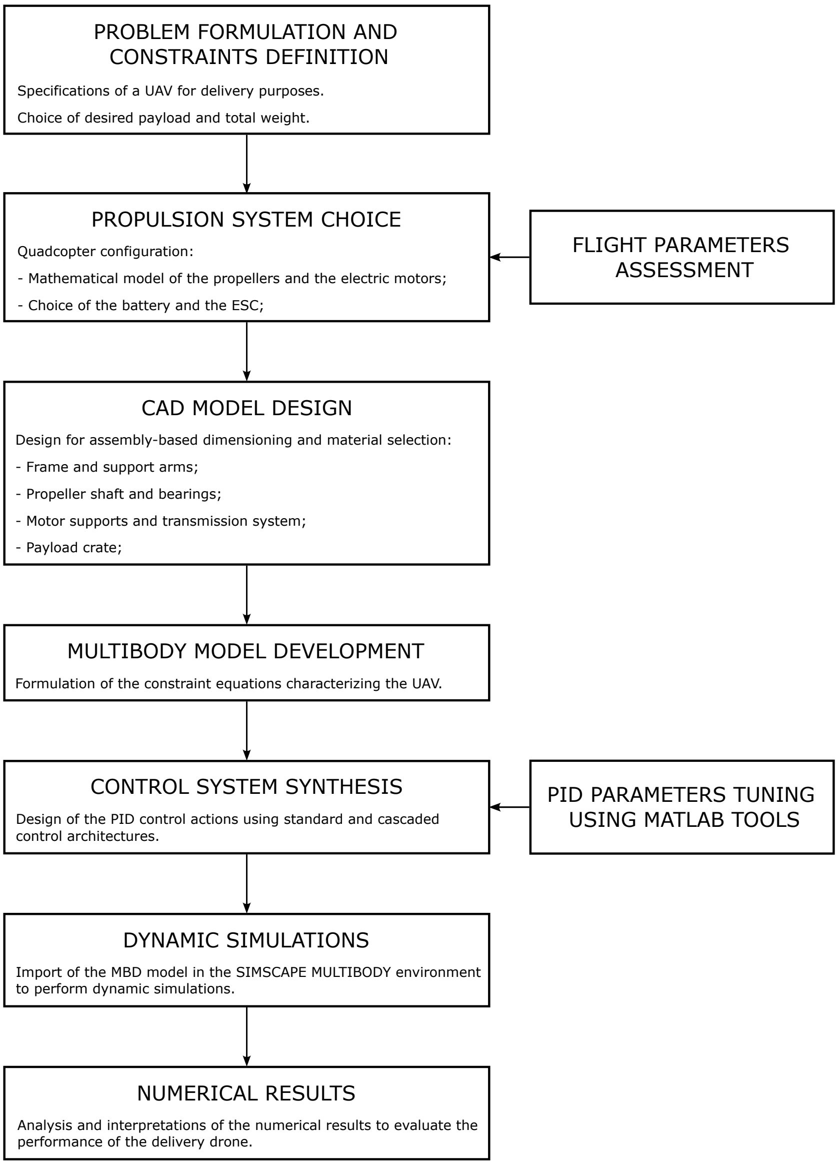

PROBLEM FORMULATION AND CONSTRAINTS DEFINITION
Specifications of a UAV for delivery purposes.
Choice of desired payload and total weight.
PROPULSION SYSTEM CHOICE
Quadcopter configuration:
- Mathematical model of the propellers and the electric motors;
- Choice of the battery and the ESC;
FLIGHT PARAMETERS ASSESSMENT
CAD MODEL DESIGN
Design for assembly-based dimensioning and material selection:
- Frame and support arms;
- Propeller shaft and bearings;
- Motor supports and transmission system;
- Payload crate;
MULTIBODY MODEL DEVELOPMENT
Formulation of the constraint equations characterizing the UAV.
CONTROL SYSTEM SYNTHESIS
Design of the PID control actions using standard and cascaded control architectures.
PID PARAMETERS TUNING USING MATLAB TOOLS
DYNAMIC SIMULATIONS
Import of the MBD model in the SIMSCAPE MULTIBODY environment to perform dynamic simulations.
NUMERICAL RESULTS
Analysis and interpretations of the numerical results to evaluate the performance of the delivery drone.

Figure 1. Flowchart of the approach followed in the paper.

2. Propulsion System Description and Modeling

2.1. Main Concept

The virtual prototype of the UAV system serving as a delivery drone that is designed in this study is a quadcopter. Consequently, the first problem to be addressed in the design phase is the ideation of the propulsion system. The propulsion system is fundamental in the design of quadcopters because it considerably influences their performance. In general, the propulsion system consists of four fundamental parts, that is, motors, propellers, batteries, and electronic speed controllers. The specifics of these components must be properly matched to work efficiently and for avoiding accidents. For instance, the selection of motors and propellers is carried out simultaneously to obtain the best combination in terms of thrust, efficiency, and dimensions of the frame. Therefore, this section is devoted to the propulsion system description and modeling, thereby addressing one by one all the requirements for the design of the desired quadcopter working as a delivery drone.

2.2. Propellers

Propellers provide the required thrust to the prototype of the quadcopter designed in this work. Propeller function is of paramount importance and is analogous to the that of the fixed-wing design of airplanes. The geometric parameters that mainly affect propeller performance are the diameter and the pitch. Therefore, in engineering practice, propellers are described by an identification number of four digits. The first two numbers specify the diameter measured in inches, while the second two numbers define the propeller pitch measured in inches. The pitch is a measure of the theoretical forward travel of a propeller during one revolution, assuming no loss of energy to slippage. Propeller pitch is used to select a propeller for a specific application based on the desired thrust and RPM range. Another fundamental parameter that characterizes each propeller refers to the number of blades. For instance, standard propellers available for sale in the market have two or three blades. Propellers having two blades are more efficient, but they generate less thrust than those having three blades. The materials used for manufacturing the propellers also affect their dynamic response. Typical materials employed for the manufacturing of propellers are plastic, wood, and carbon fiber. Plastic propellers are very common but they suffer from severe self-induced vibrations and are quite fragile. Propellers made of wood are heavy and stiff, but less responsive to speed changes. Carbon fiber propellers are the best option because they are stiff and light, and they generate less mechanical vibrations.

2.3. Electric Motors

Electric motors convert the electrical energy of batteries into the mechanical energy of the propellers. The selection of these components strongly depends on the payload of the UAV system of interest. Heavier UAVs need larger motors due to their higher power requirements. The electric motors mainly used for quadcopters are DC motors since they are more efficient and cheaper than AC motors. The geometric dimensions of an electric motor are described by a number having four digits. The first two numbers indicate the width of the motor, measured in millimeters, while the last two numbers indicate the height, measured in millimeters. Another important parameter used for identifying DC motors is referred to as the KV value. This parameter approximately specifies the number of RPMs that the motor revolves under a single volt of electric tension. For example, electric motors having high values of the KV parameter are faster and drive smaller propellers. For these reasons, these kinds of DC motors are employed in racing drones. Electric motors having low values of the KV parameter, on the other hand, are slower and more suitable for supporting the propulsion systems of delivery drones.

2.4. Batteries

The batteries provide power to the electric motors. Currently, there are three different types of batteries, namely lithium polymer (LiPo), nickel metal hydride (NiMH), and lithium-ion. Because of their higher energy density and their ability to bear a higher stress without burning out, the most commonly used family of batteries are LiPo batteries, which are suitable for UAV systems in general. A fundamental parameter that characterizes the performances of batteries to be considered is the discharge rate, which defines the maximum current tolerable by the propulsion system without accidents. For the UAV system designed in this investigation as a delivery drone, the battery chosen is a $12S$, 10,000 (mAh) battery with a discharge rate of 30 (C). The symbol S stands for series, indicating that the cells are connected in series to increase the voltage of the battery pack. Each cell in a lithium polymer (LiPo) battery pack has a nominal voltage of 3.7 (V), so a $12S$ battery pack would have a nominal voltage of 12 × 3.7 (V) = 44.4 (V). The number of cells in a battery pack affects both the voltage and the capacity of the battery.

2.5. Electronic Speed Controllers

Electronic Speed Controllers (ESCs) are electronic components that guide the speed of the motors by processing the autopilot signal. Another function of the ESC is to transform the DC current coming from the batteries into a three-phase AC current to supply power to the electric motor. There are also other auxiliary functions such as battery protection. The ESC chosen for the virtual prototype of the proposed UAV system has an amperage of 80 (A) supplied to the electric motors.

2.6. Propeller Modeling

By knowing the diameter D_p expressed in (m), the pitch H_p expressed in (m), and the number of blades B_p of a propeller, it is possible to estimate the thrust force F_T expressed in (N) and the torque moment M_p expressed in $(N \times m)$. The thrust and the torque can be obtained from the following formulas:

$$F_T = \rho C_T \left(\frac{N}{60}\right)^2 {D_P}^4 \tag{1}$$

$$M_p = \rho C_M \left(\frac{N}{60}\right)^2 {D_P}^5 \tag{2}$$

where C_T and C_M are the dimensionless thrust and torque coefficients, respectively, while N denotes the revolutions per minute of the propeller. The parameter ρ represents the air density expressed in (kg/m^3), which varies with the altitude h expressed in meters according to the following simple law:

$$\rho = \frac{273 P_a}{101325(273 + T_t)} \rho_0 \tag{3}$$

where P_a represents the atmospheric pressure, T_t is the atmospheric temperature, and the standard air density ρ_0 is equal to 1.293 (kg/m^3) when the temperature is 273 (K). The atmospheric pressure P_a measured in (Pa) can be estimated as follows:

$$P_a = 101325 \left(1 - 0.0065 \frac{h}{273 + T_t}\right)^{5.2561} \tag{4}$$

The coefficients C_T and C_M are functions of propeller data. These two dimensionless coefficients can be determined by using the following formulas:

$$C_T = 0.25 \pi^3 \lambda \zeta^2 B_P K_0 \frac{\varepsilon \arctan(\frac{H_P}{\pi D_P}) - \alpha_0}{\pi A + K_0} \tag{5}$$

$$C_M = \frac{1}{8A} \pi^2 C_d \lambda \zeta^2 {B_P}^2 \tag{6}$$

where C_d is another dimensionless coefficient given by:

$$C_d = C_{fD} + \frac{\pi A {K_0}^2}{e} \frac{\left(\varepsilon \arctan(\frac{H_P}{\pi D_P}) - \alpha_0\right)^2}{(\pi A + K_0)^2} \tag{7}$$

More specifically, the parameter C_d represents the drag coefficient defined as:

$$C_d = \frac{2 F_d}{\rho u^2 S} \tag{8}$$

where F_d is the drag force, ρ is the mass density of the fluid, u is the flow velocity, and S is the reference area. Experimental values of the previous coefficients vary over the following ranges: A = 5–8 $(-)$, ε = 0.85–0.95 $(-)$, λ = 0.7–0.9 $(-)$, ζ = 0.4–0.7 $(-)$, e = 0.7–0.9 $(-)$, $C_{fD} = 0.015$ $(-)$, $\alpha_0 = -\pi/36 - 0$ (rad), and $K_0 = 6.11$ $(-)$. For the design of the UAV system developed in this study, the following values were chosen: $A = 5$ $(-)$, $\varepsilon = 0.85$

$(-)$, $\lambda = 0.75$ $(-)$, $\zeta = 0.5$ $(-)$, $e = 0.83$ $(-)$, $C_{fD} = 0.015$ $(-)$, $\alpha_0 = 0$ (rad), and $K_0 = 6.11$ $(-)$. The coefficient $A = D_p/c_p$ is the aspect ratio, where D_p is the propeller diameter and c_p is the blade average chord length. The coefficient C_{fD} is the zero-lift drag coefficient, which depends on the thickness of the blade, the Reynolds number, and the angle of attack. The parameter $e = 1.78\left(1 - 0.045A^{0.68}\right) - 0.64$ is the Oswald factor. The parameter λ is a correction coefficient, while the parameter ϵ is another correction factor that arises due to downwash. The parameter α_0 is the zero-lift angle of attack [48,49].

2.7. Choices of Motors and Propellers

Through an iterative process, based on the analysis of the mathematical models of the propellers, it was possible to choose a combination of propellers and motors that meets the required specifications. The delivery drone to be designed is a quadcopter. Thus, the total thrust required must be divided into four contributions generated by its four propellers. In addition, the motors must meet a thrust-to-weight ratio that is equal to two, which is a common design choice made for delivery drones. In the case of the UAV system designed in this investigation, the total mass is about 10 (kg). Consequently, the thrust force that each motor coupled to the propeller must offer must be more than 5 (kgf). For the reasons mentioned before, the chosen combination of motors and propellers consists of propellers having a diameter equal to 22 (inch) and a pitch equal to 7.4 (inch), and high-efficiency, low-KV motors with a maximum power output of 1400 (W). In summary, the parameters chosen for the electric motors are shown in Table 1.

Table 1. Parameters chosen for the electric motors.

KV	Motor Weight	Idle Current	Internal Resistance	Peak Current	Maximum Power
150 (RPM/V)	273 (g)	1 (A)	85 ± 5 (mΩ)	29.7/26.5 (A)	712.8/1272 (W)

3. UAV System Computer-Aided Design

3.1. Complete Geometric Model

The development of a virtual model of the delivery drone allows the design engineer to evaluate various aspects of the UAV system of interest without creating a physical prototype. In the case of the virtual prototype of the UAV system developed in this investigation, the CAD model was created in SOLIDWORKS, which is a three-dimensional design and drafting software that allows for the construction of complex geometric shapes. Every single component was designed to meet the design needs of the delivery drone. Then, the individual components were mated to create the finished assembly. The geometric analysis performed on the CAD model was challenging and absorbed a considerable amount of time because it was necessary to guarantee the correct functionality and disassembly of the single components forming the UAV system of interest [50,51]. The final result of this process is shown in Figure 2, while the single parts of the final assembly will be subsequently presented, thereby specifying the material and the design choices made.

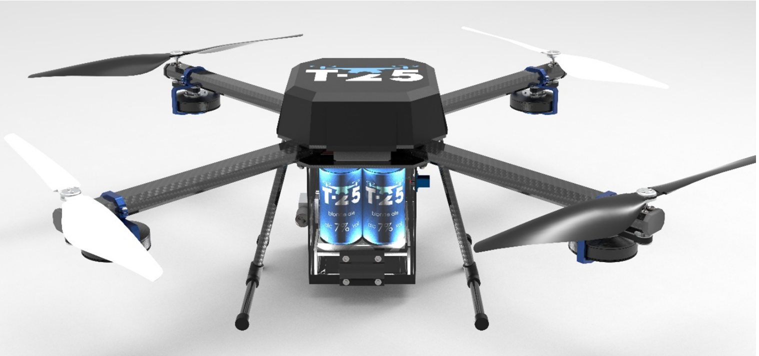


Figure 2. Final rendering of the proposed UAV system serving as a delivery drone.

As shown in Figure 2, the CAD model of the UAV system is composed of four support arms, four propeller shafts, four bearings, four motor supports, four DC motors controlling the propellers, four belt transmissions, a central frame, a beer crate, four rollers, four roller supports, a roller motor, an external case, landing gear, and a battery cover. All these mechanical components are described in detail below by following the criteria met for their geometric design.

3.2. Support Arms

The CAD model of the UAV system encompasses four support arms for properly distancing the blades of the propellers from the main body of the central chassis. Therefore, the length of each arm is a consequence of the propellers chosen, as discussed in detail in Section 2. In the proposed prototype, the frame is about one meter off from the central frame to avoid interference between the propellers during flight. The length of each arm is denoted with R and can be calculated as:

$$R = \frac{r_{\max}}{\sin(\frac{\theta}{2})} \tag{9}$$

being:

$$r_{\max} = 1.05 r_p \sim 1.2 r_p \tag{10}$$

where r_{max} is the maximum radius of a propeller, r_p represents the radius of the chosen propellers, and θ identifies the angle between the support arms. In particular, the geometric model of the mechanical system at hand was designed such that $\theta = \pi/2$ (rad), $r_p = 0.2794$ (m), $r_{max} = 0.3517$ (m), and $R = 0.4974$ (m). As shown in Figure 3, at the edge of each arm there is a slot for the shaft, which can be housed horizontally.

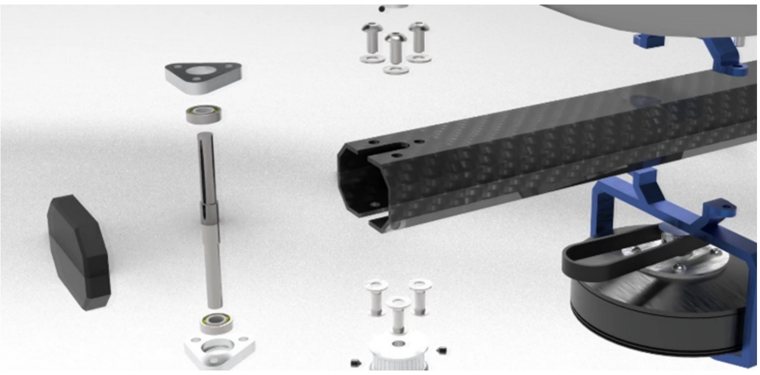


Figure 3. Components forming each support arm.

The material chosen for the support arm is carbon fiber to ensure lightness and strength. The configuration chosen for the arms is QuadX because it was experimentally verified that it is more efficient [52–54]. This configuration also makes access to the beers more manageable, which represent the payload of the delivery drone designed in this investigation. Otherwise, the arms would interfere with the opening of the case hatch. Additionally, the cross-sections of the support arms are designed to be octagonal, to create proper reference planes for positioning the motor and its support.

3.3. Propeller Shafts and Bearings

As can be seen in Figure 3, the shaft of each propeller has a larger central section that serves as a shoulder for the positioning of the bearings and their respective supports on both sides. By using the holes in the support, it is possible to mate the shaft to the arm, so that the bearings do not move. The matching between the shaft and the pulley, and the one between the shaft and the propellers, are ensured by two grains.

Axial ball bearings are used to support the axial thrusts generated by the propulsion system. In the case of the proposed design of the UAV system, the radial force is in fact negligible compared to the axial one. The bearings were inserted into custom-made aluminum housings, as represented in Figure 3. These triangular-shaped supports are designed to allow for a correct centering of the bearings, and to avoid disassembly caused by the thrusts of the propulsion system.

3.4. Motor Supports and Belt Transmissions

The UAV system is equipped with a gearbox on each arm. Thus, each electric motor is not located on the axis of each propeller. As a result, the motors require the design of proper mechanical supports to fasten them to the drone arms. These additional mechanical supports are represented in blue in Figure 3. For simplicity, the geometry of each additional mechanical support is identical to one of the arms, thereby guaranteeing a better matching. The material chosen for the motor supports is aluminum. These supports are designed for facilitating the installation of the motors, with their rotors directed toward the top. This also helps to reduce the elastic deflection of the propeller shafts. On the base of each support, there are four holes aligned with the corresponding holes already present on the motor. Four M4 screws are used to fasten the stator of each electric motor to the base of their mechanical supports, therewith leaving the rotor free to transmit motion.

The gearbox chosen to transmit the mechanical power from the electric motor to the propellers is a belt transmission shown in Figure 4.

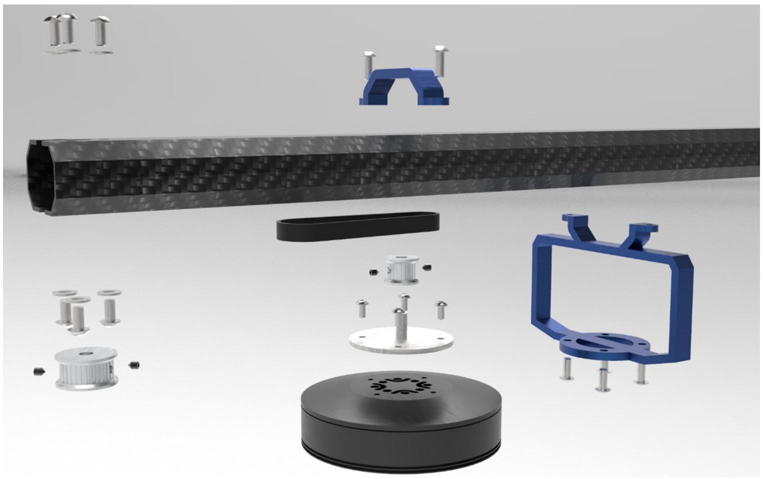


Figure 4. Arm disassembly showing the belt transmission.

This solution is more efficient and quieter than gears. The belt links two pulleys whose transmission ratio is 2:3.

3.5. Central Frame

As shown in Figure 5, the central frame of the UAV system serving as a delivery drone resembles the polygonal geometry of the arms.

There are recesses in the structure that reduce the weight of the frame without affecting its functionality, as well as its structural strength and integrity. The main function of the central chassis is to connect the four arms to the crate for transporting the beers representing the payload of the UAV system. The material chosen is carbon fiber for its lightness and rigidity, as mentioned previously. The space between the two plates of the frame is used for the positioning of the electronics, such as the flight controller and the ESCs. The electronics are fixed to the structure with cable ties.

3.6. Beer Crate

As mentioned before, the payload of the proposed UAV system serving as a delivery drone is represented by beverages such as beer. Beers are transported by a crate fastened to the structure of the UAV system, as shown in Figure 6.

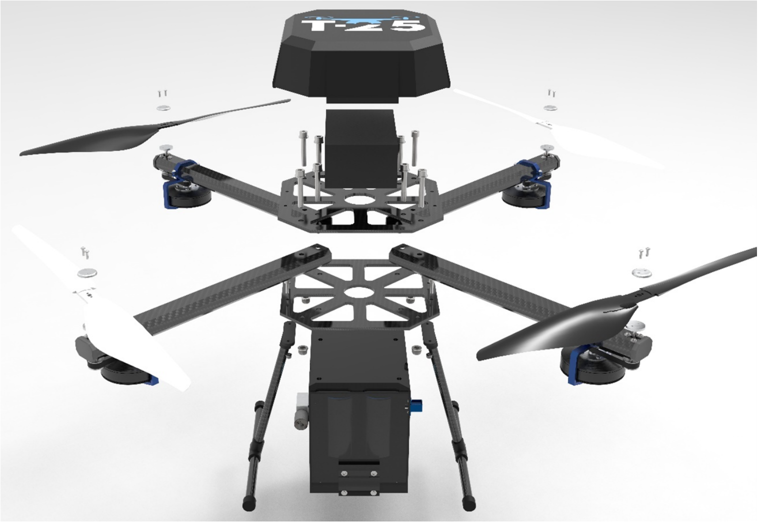


Figure 5. Drone central frame disassembly.

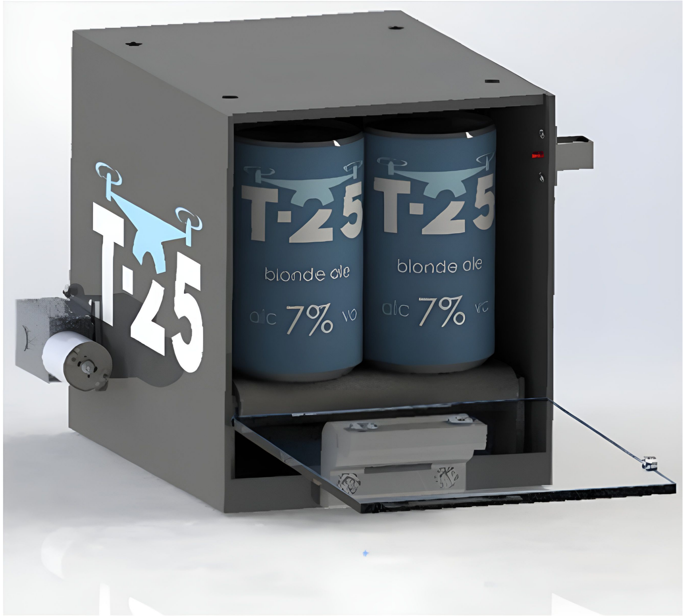


Figure 6. Beer crate.

Inside the crate, there is a conveyor belt on which the beers are placed. The delivery of the beers involves a series of operations. Once the quadcopter has landed, the customer can make payment via NFC. The signal is sent to the controller, which in turn opens the linear actuator that locks the door. The DC motor, housed outside the case, starts the conveyor belt rollers. The beers, owing to their own weight and because of the push of the rollers, open the door and slide on it. Thus, the customer can pick up the beers. When the beers are recovered, the torsional spring of the hinge closes the door again. This system helps to minimize the contact between the UAV and the customer, who might unintentionally damage the delivery drone.

3.7. Rollers, Roller Supports, and Roller Motor

Four rollers serve to support the conveyor belt transporting the beers representing the payload. The shaft of the rollers acts as a shoulder for the two bearings. The first bearing, the shaft, the second bearing, and a cap are inserted into the outside of the rollers by following this order, as shown in Figure 7.

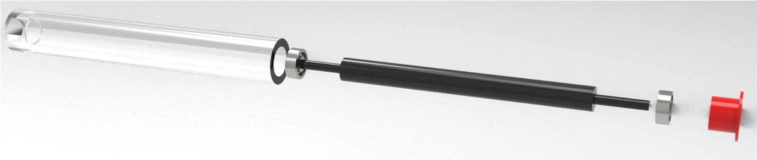


Figure 7. Rollers of the conveyor belt.

This assembly sequence serves to lock the bearings in place. There is a hole in the first roller for the insertion of the DC motor shaft.

The assembled rollers are positioned on special supports, as shown in Figure 8.

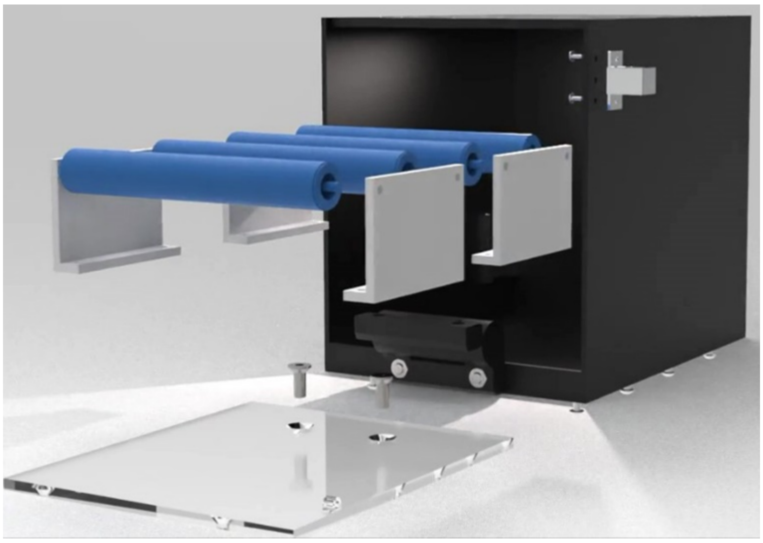


Figure 8. Roller supports.

The supports serve to facilitate the installation of the rollers inside the crate. The supports are L-shaped and have four holes. The holes on the lower surface allow for the fixing of the crate, while those on the upper surface are necessary for the placement of the rollers. The supports are necessary for positioning the rollers inside the crate. To guarantee an easy assembly and disassembly process, it was not possible to mount the rollers directly inside the crate because there would be no room for assembly. Because of the presence of the supports, the rollers are assembled outside and then fastened to the external crate.

As shown in Figure 9, the DC motor that controls the rotation of the rollers is attached to the outside of the case.

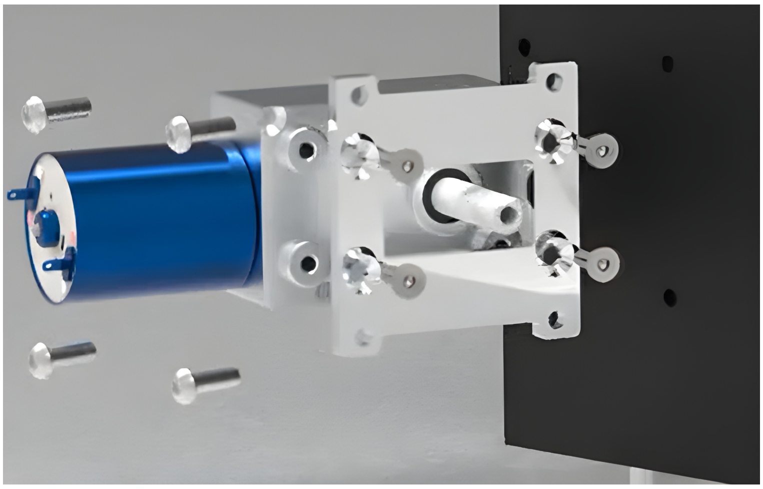


Figure 9. The DC motor commanding the rollers.

The process for fixing the case is facilitated by two aluminum plates, which are placed with one inside and one outside, and then secured with screws. To avoid interference problems, the motor is the first component to be assembled in the case. The plate represented in Figure 8 is the one on the outside. Another equal plate is collocated in the same position inside the case.

3.8. External Case, Landing Gear, and Battery Cover

The outer case is made of ABS plastic. Holes on the bottom surface of the case are required for fastening the rollers by supports. A plexiglass door closes the case. Plexiglass reduces friction between the beers and the door, thereby allowing them to slide without stopping. It is also an aesthetic choice because it lets the customer see the beers representing the payload of the delivery drone. The door hinge is equipped with a torsion spring that closes the door once the beers are picked up. To prevent the door from accidentally opening during flight, a linear actuator electronically locks it. This result is checked in a multibody simulation environment described in detail below in the manuscript.

The case of beers should not touch the ground because a certain angle is required for the beers to slide. For this purpose, the delivery drone is equipped with a simple landing gear composed of four cylindrical tubes connected by T-shaped supports. As shown in Figure 5, screws connect the landing gear to the central frame.

The battery is protected from the weather by a cover. The cover is made of plastic and is fixed to the frame with two flaps.

4. UAV System Multibody Modeling

4.1. Mathematical Background

In the case study in point, namely the kinematic and dynamic analysis of a UAV system working as a delivery drone, the laws of motion are provided as input functions to control the motion of the mechanical system at hand. Subsequently, both the processes of analysis and the synthesis of the control system are carried out. As shown below, the delivery drone is modeled as a multibody system composed only of rigid bodies endowed with a three-dimensional motion. While the differential equations considered in the mathematical modeling of the present problem take into account the dynamic behavior of the system under study, the additional algebraic equations appended to the set of differential equations serve to describe the effect of the mechanical joints and the presence of driving constraints imposed on the dynamical system to be controlled. Furthermore, a proper set of externally applied forces, torques, and moments is considered for the mathematical modeling of all the mechanical actions applied on the delivery drone modeled as a multibody system.

4.2. Kinematic and Dynamic Analysis

The following is a description of the differential-algebraic equations of motion used to define the multibody model of the UAV system under consideration. As anticipated before, the fundamental approach employed for the mathematical modeling of the present UAV system serving as a delivery drone, which represents the case study of this investigation, is the Redundant Coordinate Formulation (RCF). Thus, although the dynamical system at hand has a number of degrees of freedom equal to $n_f = 6$, i.e., the three spatial translations and the three finite rotations of the chassis, its multibody model is based on a set of n_q redundant generalized coordinates, where $n_q > n_f$. More precisely, the Three-Dimensional Reference Point Coordinate Formulation with Euler Angles (3D-RPCF-EA) is the multibody formulation approach adopted in this work [55,56]. In the 3D-RPCF-EA, the Cartesian coordinates of a material point attached to each rigid body are employed as the translational coordinates, while the set of rotational coordinates used is represented by the Euler-Cardan angles based on the $X - Y - Z$ sequence. By doing so, assuming a space dimension equal to $d = 3$, and by denoting the generic rigid body with the superscript index i, the first angular displacement denoted with ϕ^i represents the system roll, the second angular displacement denoted with ϑ^i represents the system pitch, and the third angular displacement denoted with ψ^i represents the system yaw. The vector of rotational coordinates pertaining to the generic rigid body i is thereby donated with the vector $\boldsymbol{\theta}^i$ of dimensions $n_\theta = 3$. In this context, the rotation matrix $\mathbf{A}^i$ of a generic rigid body i is therefore a $d \times d$ matrix given by:

$$\mathbf{A}^i = \mathbf{A}_x^i \mathbf{A}_y^i \mathbf{A}_z^i \tag{11}$$

being:

$$\mathbf{A}_x^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{\phi^i} & -S_{\phi^i} \\ 0 & S_{\phi^i} & C_{\phi^i} \end{bmatrix}, \quad \mathbf{A}_y^i = \begin{bmatrix} C_{\vartheta^i} & 0 & S_{\vartheta^i} \\ 0 & 1 & 0 \\ -S_{\vartheta^i} & 0 & C_{\vartheta^i} \end{bmatrix}, \quad \mathbf{A}_z^i = \begin{bmatrix} C_{\psi^i} & -S_{\psi^i} & 0 \\ S_{\psi^i} & C_{\psi^i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

where the following abbreviations are used:

$$\begin{cases} C_{\phi^i} = \cos(\phi^i) \\ S_{\phi^i} = \sin(\phi^i) \end{cases}, \quad \begin{cases} C_{\vartheta^i} = \cos(\vartheta^i) \\ S_{\vartheta^i} = \sin(\vartheta^i) \end{cases}, \quad \begin{cases} C_{\psi^i} = \cos(\psi^i) \\ S_{\psi^i} = \sin(\psi^i) \end{cases} \tag{13}$$

Consequently, in the 3D-RPCF-EA, the number of generalized coordinates of each rigid body is given by $n_b = d + n_\theta = 6$. To this end, one can define the following generalized coordinate vector denoted with $\mathbf{q}^i$ of dimensions $n_b \times 1$ associated with the generic rigid body i:

$$\mathbf{q}^i = \begin{bmatrix} (\mathbf{R}^i)^T & \left(\boldsymbol{\theta}^i\right)^T \end{bmatrix}^T \tag{14}$$

being:

$$\mathbf{R}^i = \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix}, \quad \boldsymbol{\theta}^i = \begin{bmatrix} \phi^i \\ \vartheta^i \\ \psi^i \end{bmatrix} \tag{15}$$

where $\mathbf{R}^i$ denotes a vector of dimensions $d \times 1$ containing the translational coordinates of the generic body i, $\boldsymbol{\theta}^i$ denotes the vector of rotational coordinates of the generic body i having dimensions $n_\theta \times 1$, while x^i, y^i, and z^i respectively represent the longitudinal, lateral, and vertical linear displacements of the center of mass pertaining to the rigid body i, whereas ϕ^i, ϑ^i, and ψ^i respectively represent the roll, pitch, and yaw rotational displacements of the same body, as introduced before. The multibody model of the UAV system under study is composed of $N_b = 5$ rigid bodies, identified as follows:

1. Central frame: body $i = 0$.
2. Propeller I: body $i = 1$.
3. Propeller II: body $i = 2$.
4. Propeller III: body $i = 3$.
5. Propeller IV: body $i = 4$.

Where each rigid body is unambiguously labeled with the integer number i. By mathematically modeling the delivery drone as a multibody mechanical system composed of $N_b = 5$ rigid bodies deployed in a three-dimensional space, the resulting dynamical model is based on a set of redundant coordinates in number equal to $n_q = N_b n_b = 30$. As a result, it is apparent that the complete form of the multibody system generalized coordinate vector referred to as $\mathbf{q}$ of dimensions $n_q \times 1$ is given as follows:

$$\mathbf{q} = \begin{bmatrix} (\mathbf{q}^0)^T & (\mathbf{q}^1)^T & (\mathbf{q}^2)^T & (\mathbf{q}^3)^T & (\mathbf{q}^4)^T \end{bmatrix}^T \tag{16}$$

In the 3D-RPCF-EA, the mass matrix $\mathbf{M}^i$ of dimensions $n_b \times n_b$ and the inertia quadratic velocity vector $\mathbf{Q}_v^i$ of dimensions $n_b \times 1$ of each rigid body i respectively assume the following analytical forms:

$$\mathbf{M}^i = \begin{bmatrix} m^i\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \left(\bar{\mathbf{G}}^i\right)^T \bar{\mathbf{I}}_{G^i}^i \bar{\mathbf{G}}^i \end{bmatrix}, \quad \mathbf{Q}_v^i = \begin{bmatrix} \mathbf{0} \\ -\left(\bar{\mathbf{G}}^i\right)^T \left(\bar{\boldsymbol{\omega}}^i \times \left(\bar{\mathbf{I}}_{G^i}^i \bar{\boldsymbol{\omega}}^i\right) + \bar{\mathbf{I}}_{G^i}^i \dot{\bar{\mathbf{G}}}^i \dot{\boldsymbol{\theta}}^i\right) \end{bmatrix} \tag{17}$$

where the collocation of the reference point $\bar{O}^i$ of each body-fixed reference frame i is assumed to be coincident with the center of mass G^i of the same rigid body, m^i is the mass of the generic rigid body i, $\mathbf{I}$ denotes the $d \times d$ identity matrix, $\bar{\mathbf{I}}_{G^i}^i$ identifies the inertia matrix of the rigid body having dimensions $d \times d$ evaluated with respect to the body-fixed coordinate frame, $\bar{\boldsymbol{\omega}}^i$ is the angular velocity vector of the rigid body i having dimensions $d \times 1$ projected on its reference system, and $\bar{\mathbf{G}}^i$ is the transformation matrix having dimensions $d \times n_\theta$ that allows for determining the local angular velocity from the time derivative of the rigid body rotational coordinates. By selecting a proper body-fixed reference system that is the principal of inertia, the inertia matrix denoted with $\bar{\mathbf{I}}_{G^i}^i$ assumes the following simplified form:

$$\bar{\mathbf{I}}_{G^i}^i = \begin{bmatrix} I_{xx}^i & 0 & 0 \\ 0 & I_{yy}^i & 0 \\ 0 & 0 & I_{zz}^i \end{bmatrix} \tag{18}$$

where I_{xx}^i, I_{yy}^i, and I_{zz}^i represent the principal mass moments of inertia pertaining to the rigid body i. The conversion of the time derivative of the rotational coordinate vector $\dot{\boldsymbol{\theta}}^i$ into the local angular velocity vector $\bar{\boldsymbol{\omega}}^i$, as well as the conversion of the time derivative of the body generalized coordinate vector $\dot{\mathbf{q}}^i$ into the local angular velocity vector $\bar{\boldsymbol{\omega}}^i$, can be explicitly performed as follows:

$$\bar{\boldsymbol{\omega}}^i = \bar{\mathbf{G}}^i \dot{\boldsymbol{\theta}}^i = \bar{\mathbf{W}}^i \dot{\mathbf{q}}^i \tag{19}$$

where in the case of the $X - Y - Z$ sequence of Euler angles, the transformation matrices $\bar{\mathbf{G}}^i$ of dimensions $d \times n_\theta$ and $\bar{\mathbf{W}}^i$ of dimensions $d \times n_b$ are respectively given by:

$$\bar{\mathbf{G}}^i = \begin{bmatrix} C_{\vartheta^i}C_{\psi^i} & S_{\psi^i} & 0 \\ -C_{\vartheta^i}S_{\psi^i} & C_{\psi^i} & 0 \\ S_{\vartheta^i} & 0 & 1 \end{bmatrix}, \quad \bar{\mathbf{W}}^i = \begin{bmatrix} \mathbf{O} & \bar{\mathbf{G}}^i \end{bmatrix} \tag{20}$$

where $\mathbf{O}$ denotes a $d \times d$ zero matrix. For the UAV system modeled as a multibody mechanical system in this study, the inertial properties of each component, including the mass of the payload m^p attached to the chassis of the delivery drone, are reported in Table 2.

4.3. Mechanical Joints

The multibody system analyzed herein is mechanically held together by the presence of $N_{c,h} = 4$ holonomic algebraic constraints mathematically modeling the kinematic joints, which are identified as follows:

1. Revolute joint I: constraint $k = 1$.
2. Revolute joint II: constraint $k = 2$.
3. Revolute joint III: constraint $k = 3$.
4. Revolute joint IV: constraint $k = 4$.

Where each kinematic pair is unambiguously labeled with the integer number k. Thus, to guarantee the rotation imparted by the electric motor to each propeller in the computer simulation, four revolute joints are considered as the mechanical constraints inserted between the central frame of the delivery drone and the four propellers forming the quadcopter. By using the multibody approach to the analysis of articulated mechanical systems, revolute joints are modeled using holonomic bilateral constraint equations [57,58]. This type of joint constrains the relative motion between two rigid bodies by granting $n_f^k = 1$ relative degrees of freedom between its members, and therefore, it is mathematically described by $n_{c,h}^k = 5$ effective algebraic constraint equations in a three-dimensional space. The total number of holonomic algebraic equations is therefore equal to $n_{c,h}$ and is given by:

$$n_{c,h} = \sum_{k=1}^{N_{c,h}} n_{c,h}^k = 20 \tag{21}$$

The schematization of a generic revolute joint between two spatial rigid bodies is represented in Figure 10.

Table 2. Inertial properties of each rigid body of the delivery drone.

Descriptions	**Symbols**	**Data (units)**
Payload mass	m^p	2 (kg)
Central frame mass	m^0	8 (kg)
Central frame first moment of inertia	I_{xx}^0	0.27 (kg·m^2)
Central frame second moment of inertia	I_{yy}^0	0.26 (kg·m^2)
Central frame third moment of inertia	I_{zz}^0	0.42 (kg·m^2)
Propeller I mass	m^1	5.6×10^{-3} (kg)
Propeller I first moment of inertia	I_{xx}^1	9.8×10^{-6} (kg·m^2)
Propeller I second moment of inertia	I_{yy}^1	8.6×10^{-4} (kg·m^2)
Propeller I third moment of inertia	I_{zz}^1	8.6×10^{-4} (kg·m^2)
Propeller II mass	m^2	5.6×10^{-3} (kg)
Propeller II first moment of inertia	I_{xx}^2	9.8×10^{-6} (kg·m^2)
Propeller II second moment of inertia	I_{yy}^2	8.6×10^{-4} (kg·m^2)
Propeller II third moment of inertia	I_{zz}^2	8.6×10^{-4} (kg·m^2)
Propeller III mass	m^3	5.6×10^{-3} (kg)
Propeller III first moment of inertia	I_{xx}^3	9.8×10^{-6} (kg·m^2)
Propeller III second moment of inertia	I_{yy}^3	8.6×10^{-4} (kg·m^2)
Propeller III third moment of inertia	I_{zz}^3	8.6×10^{-4} (kg·m^2)
Propeller IV mass	m^4	5.6×10^{-3} (kg)
Propeller IV first moment of inertia	I_{xx}^4	9.8×10^{-6} (kg·m^2)
Propeller IV second moment of inertia	I_{yy}^4	8.6×10^{-4} (kg·m^2)
Propeller IV third moment of inertia	I_{zz}^4	8.6×10^{-4} (kg·m^2)

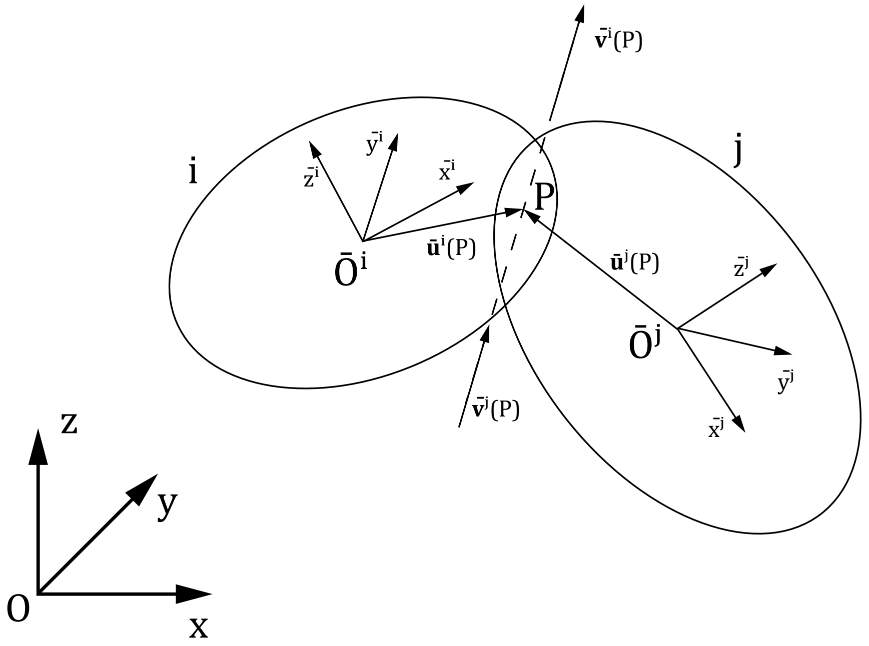


Figure 10. Mechanical scheme of a generic revolute joint.

Considering two generic rigid bodies denoted respectively by the integers i and j, and denoting by integer k the kinematic constraint associated with the generic revolute joint at hand, the corresponding constraint equations are embedded in the nonlinear constraint vector denoted with the vector $\mathbf{\Psi}^k$ of dimensions $n_{c,h}^k \times 1$. The precise mathematical expression of the holonomic algebraic equations representing a generic revolute joint labeled with k can be written in a general and compact vector form as follows:

$$\mathbf{\Psi}^k = \begin{bmatrix} \mathbf{r}^i(P) - \mathbf{r}^j(P) \\ \left(\mathbf{v}_1^i\right)^T \mathbf{v}_3^j \\ \left(\mathbf{v}_2^i\right)^T \mathbf{v}_3^j \end{bmatrix} = \mathbf{0} \tag{22}$$

where $\mathbf{r}^i(P)$ and $\mathbf{r}^j(P)$ respectively represent the global position vectors of dimensions $d \times 1$ of the collocation point P of the generic revolute joint k expressed, considering the kinematic representation of the rigid bodies i and j, while $\mathbf{v}_1^i$, $\mathbf{v}_2^i$, and $\mathbf{v}_3^i$ represent a triplet of unit vectors of dimensions $d \times 1$ forming a rigid triad attached to the revolute joint k modeled as belonging to the rigid body i, whereas $\mathbf{v}_1^j$, $\mathbf{v}_2^j$, and $\mathbf{v}_3^j$ represent a triplet of unit vectors of dimensions $d \times 1$ forming a rigid triad attached to the revolute joint k modeled as belonging to the rigid body j. By employing the multibody version of the fundamental formula of rigid kinematics, the absolute position vectors $\mathbf{r}^i(P)$ and $\mathbf{r}^j(P)$ are respectively defined as follows:

$$\mathbf{r}^i(P) = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}^i(P), \quad \mathbf{r}^j(P) = \mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}^j(P) \tag{23}$$

where $\bar{\mathbf{u}}^i(P)$ and $\bar{\mathbf{u}}^j(P)$ respectively represent the local position vectors of dimensions $d \times 1$ of the material point P associated with the body-fixed reference systems of the generic rigid bodies i and j connected by the revolute joint k. Additionally, the unit vectors $\bar{\mathbf{v}}^i$ and $\bar{\mathbf{v}}^j$ of dimensions $d \times 1$ represented in Figure 10 mathematically define the direction of the revolute joint axis by considering the local reference frames attached to the rigid bodies i and j. Thus, one can write:

$$\bar{\mathbf{v}}_3^i = \bar{\mathbf{v}}^i \quad \Rightarrow \quad \left.\begin{array}{c} \bar{\mathbf{v}}_1^i \\ \bar{\mathbf{v}}_2^i \end{array}\right\} = \mathrm{triad}\left(\bar{\mathbf{v}}_3^i\right) \quad \Rightarrow \quad \left\{\begin{array}{l} \mathbf{v}_1^i = \mathbf{A}^i \bar{\mathbf{v}}_1^i \\ \mathbf{v}_2^i = \mathbf{A}^i \bar{\mathbf{v}}_2^i \\ \mathbf{v}_3^i = \mathbf{A}^i \bar{\mathbf{v}}_3^i \end{array}\right. \tag{24}$$

and

$$\bar{\mathbf{v}}_3^j = \bar{\mathbf{v}}^j \quad \Rightarrow \quad \left.\begin{matrix} \bar{\mathbf{v}}_1^j \\ \bar{\mathbf{v}}_2^j \end{matrix}\right\} = \mathrm{triad}(\bar{\mathbf{v}}_3^j) \quad \Rightarrow \quad \left\{\begin{matrix} \mathbf{v}_1^j = \mathbf{A}^j\bar{\mathbf{v}}_1^j \\ \mathbf{v}_2^j = \mathbf{A}^j\bar{\mathbf{v}}_2^j \\ \mathbf{v}_3^j = \mathbf{A}^j\bar{\mathbf{v}}_3^j \end{matrix}\right. \tag{25}$$

where the function triad$(\mathbf{v})$ creates in output a right-handed triad of unit vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ of dimensions $d \times 1$ starting from the unit vector $\mathbf{v}$ of dimensions $d \times 1$ taken in input [59]. The Jacobian matrix of the constraint equations associated with the revolute joint for a spatial multibody system is denoted with the matrix $\boldsymbol{\Psi}_{\mathbf{q}^k}^k$ having dimensions $n_{c,h}^k \times 2n_b$, and it is defined as follows:

$$\boldsymbol{\Psi}_{\mathbf{q}^k}^k = \begin{bmatrix} \boldsymbol{\Psi}_{\mathbf{q}^i}^k & \boldsymbol{\Psi}_{\mathbf{q}^j}^k \end{bmatrix} = \begin{bmatrix} \mathbf{L}^i(P) & -\mathbf{L}^j(P) \\ \left(\mathbf{v}_3^j\right)^T\mathbf{D}_1^i & \left(\mathbf{v}_1^i\right)^T\mathbf{D}_3^j \\ \left(\mathbf{v}_3^j\right)^T\mathbf{D}_2^i & \left(\mathbf{v}_2^i\right)^T\mathbf{D}_3^j \end{bmatrix} \tag{26}$$

where $\mathbf{L}^i(P)$ and $\mathbf{L}^j(P)$ respectively represent the Jacobian matrices having dimensions $d \times n_b$ of the global position vectors $\mathbf{r}^i(P)$ and $\mathbf{r}^j(P)$ associated with the material point P, whereas $\mathbf{D}_1^i$, $\mathbf{D}_2^i$, and $\mathbf{D}_3^j$ respectively represent the Jacobian matrices having dimensions $d \times n_b$ of the global direction vectors $\mathbf{v}_1^i$, $\mathbf{v}_2^i$, and $\mathbf{v}_3^j$. For a generic rigid body i, the Jacobian matrices $\mathbf{L}^i(P)$ and $\mathbf{D}^i$ are respectively defined as follows:

$$\mathbf{L}^i(P) = \begin{bmatrix} \mathbf{I} & \mathbf{A}^i\left(\tilde{\bar{\mathbf{u}}}^i(P)\right)^T\bar{\mathbf{G}}^i \end{bmatrix}, \quad \mathbf{D}^i = \begin{bmatrix} \mathbf{O} & \mathbf{A}^i\left(\tilde{\bar{\mathbf{v}}}^i\right)^T\bar{\mathbf{G}}^i \end{bmatrix} \tag{27}$$

where $\tilde{\bar{\mathbf{u}}}^i(P)$ and $\tilde{\bar{\mathbf{v}}}^i$ respectively represent skew-symmetric matrices of dimensions $d \times d$ associated with the vectors $\bar{\mathbf{u}}^i(P)$ and $\bar{\mathbf{v}}^i$. The quadratic velocity vector of the constraint equations associated with the revolute joint for a spatial multibody system is denoted with the vector $\mathbf{Q}_{d,\Psi}^k$ of dimensions $n_{c,h}^k \times 1$ and it is defined as follows:

$$\mathbf{Q}_{d,\Psi}^k = \begin{bmatrix} -\dot{\mathbf{L}}^i\dot{\mathbf{q}}^i + \dot{\mathbf{L}}^j\dot{\mathbf{q}}^j \\ -\left(\mathbf{v}_3^j\right)^T\dot{\mathbf{D}}_1^i\dot{\mathbf{q}}^i - 2\left(\dot{\mathbf{v}}_1^i\right)^T\dot{\mathbf{v}}_3^j - \left(\mathbf{v}_1^i\right)^T\dot{\mathbf{D}}_3^j\dot{\mathbf{q}}^j \\ -\left(\mathbf{v}_3^j\right)^T\dot{\mathbf{D}}_2^i\dot{\mathbf{q}}^i - 2\left(\dot{\mathbf{v}}_2^i\right)^T\dot{\mathbf{v}}_3^j - \left(\mathbf{v}_2^i\right)^T\dot{\mathbf{D}}_3^j\dot{\mathbf{q}}^j \end{bmatrix} \tag{28}$$

The constraint quadratic velocity vector associated with the holonomic algebraic equations describing the revolute joints is a highly nonlinear vector that absorbs the terms that are quadratic with respect to the generalized velocities of the multibody system. This vector is of fundamental importance for the correct computer implementation of the index-one form of the equations of motion that include the set of holonomic constraints. For the UAV system modeled as a multibody mechanical system in this study, the geometric properties of each mechanical joint are reported in Table 3.

In Table 3, the scalar quantity R represents the length of each support arm measured from the center of mass of the central frame to the axis of the revolute joints associated with each propeller.

Table 3. Geometric properties of each kinematic joint of the delivery drone.

Kinematic Pair Number	**Kinematic Pair Type**	**First Body Number**	**Second Body Number**	**First Body Location Point and Joint Axis**	**Second Body Location Point and Joint Axis**
k	Name	i_k	j_k	$\begin{cases} \bar{\mathbf{u}}^{i_k} = \begin{bmatrix} \bar{x}^{i_k} & \bar{y}^{i_k} & \bar{z}^{i_k} \end{bmatrix}^T \\ \bar{\mathbf{v}}^{i_k} = \begin{bmatrix} \bar{v}_x^{i_k} & \bar{v}_y^{i_k} & \bar{v}_z^{i_k} \end{bmatrix}^T \end{cases}$	$\begin{cases} \bar{\mathbf{u}}^{j_k} = \begin{bmatrix} \bar{x}^{j_k} & \bar{y}^{j_k} & \bar{z}^{j_k} \end{bmatrix}^T \\ \bar{\mathbf{v}}^{j_k} = \begin{bmatrix} \bar{v}_x^{j_k} & \bar{v}_y^{j_k} & \bar{v}_z^{j_k} \end{bmatrix}^T \end{cases}$
1	Revolute	0	1	$\begin{cases} \bar{\mathbf{u}}^0 = \begin{bmatrix} \frac{\sqrt{2}}{2}R & -\frac{\sqrt{2}}{2}R & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$	$\begin{cases} \bar{\mathbf{u}}^1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^1 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$
2	Revolute	0	2	$\begin{cases} \bar{\mathbf{u}}^0 = \begin{bmatrix} \frac{\sqrt{2}}{2}R & \frac{\sqrt{2}}{2}R & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$	$\begin{cases} \bar{\mathbf{u}}^2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$
3	Revolute	0	3	$\begin{cases} \bar{\mathbf{u}}^0 = \begin{bmatrix} -\frac{\sqrt{2}}{2}R & \frac{\sqrt{2}}{2}R & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$	$\begin{cases} \bar{\mathbf{u}}^3 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$
4	Revolute	0	4	$\begin{cases} \bar{\mathbf{u}}^0 = \begin{bmatrix} -\frac{\sqrt{2}}{2}R & -\frac{\sqrt{2}}{2}R & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$	$\begin{cases} \bar{\mathbf{u}}^4 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \bar{\mathbf{v}}^4 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{cases}$

4.4. Driving Constraints

In the multibody model of the UAV system analyzed in this investigation, the presence of $N_{c,nh} = 4$ nonholonomic algebraic constraints is considered to mathematically describe the driving constraints associated with the imposed rotational motion of the propellers. The aforementioned driving constraints can be listed as follows:

1. Imposed angular velocity I: constraint $h = 1$.
2. Imposed angular velocity II: constraint $h = 2$.
3. Imposed angular velocity III: constraint $h = 3$.
4. Imposed angular velocity IV: constraint $h = 4$.

where each set of driving constraints is unambiguously labeled with the integer number h. Therefore, to properly guarantee an assigned angular rate of rotation imposed by the electric motor to each propeller, four driving constraints are considered in the dynamic analysis describing the kinematic conditions affecting the connection between the central frame of the delivery drone and the four propellers forming the quadcopter. Adopting a systematic multibody approach for the kinematic and dynamic analysis of articulated mechanical systems, the driving constraints describing the prescribed angular velocity, relatively imposed between the propellers and the central frame, are modeled as nonholonomic bilateral constraint equations. For this purpose, the mathematical formulation of this specific type of driving constraint is developed and proposed in this paper. First, since the present family of driving constraints is represented by a set of nonholonomic algebraic constraints, no degrees of freedom are removed via their introduction in the multibody model. Second, in a three-dimensional space, the imposition of a relative angular velocity between two rigid bodies is mathematically described by $n_{c,nh}^{h} = 3$ effective algebraic constraint equations. The total number of nonholonomic algebraic equations is therefore equal to $n_{c,nh}$, and is given by:

$$n_{c,nh} = \sum_{h=1}^{N_{c,nh}} n_{c,nh}^{h} = 12 \tag{29}$$

The schematization of the fundamental geometric quantities relative to a generic driving constraint, describing the imposed rotational motion between the propellers and the central frame of the delivery drone, is represented in Figure 11.

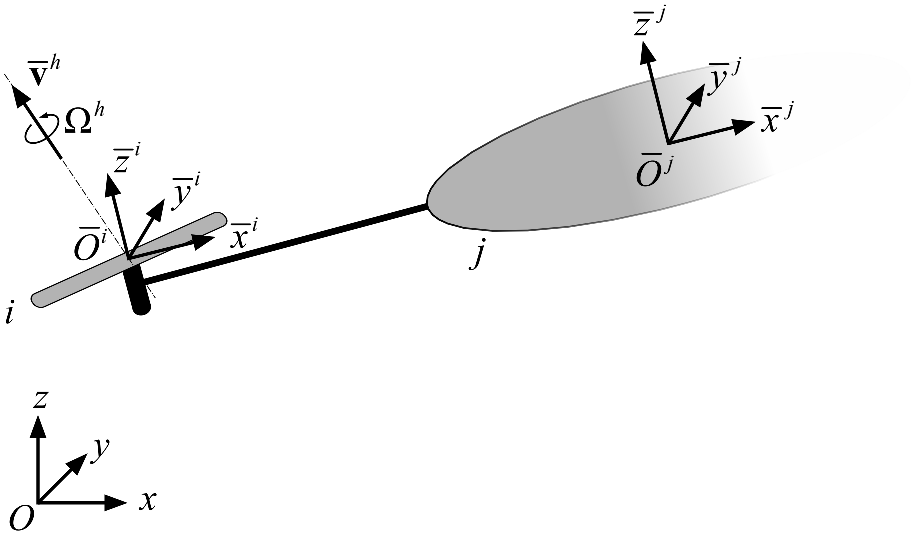


Figure 11. Mechanical scheme of a generic driving constraint.

Considering two generic rigid bodies denoted respectively by the integers i and j, the rotation matrices transforming the Cartesian coordinates of a generic material point attached to each rigid body from the body-fixed reference frame to the global reference frame are respectively denoted with $\mathbf{A}^i$ and $\mathbf{A}^j$. Denoting with $\mathbf{A}^{i,j}$ the rotation matrix that defines the orientation of an arbitrary body i with respect to a body-fixed reference system attached to the rigid body j, one can easily demonstrate the validity of the following matrix equation:

$$\mathbf{A}^i = \mathbf{A}^j\mathbf{A}^{i,j} \tag{30}$$

Starting from this fundamental kinematic relationship, the additive property of the angular velocity vectors can be readily proven [59,60], leading to the following fundamental vector equation:

$$\boldsymbol{\omega}^i = \boldsymbol{\omega}^j + \boldsymbol{\omega}^{i,j} \tag{31}$$

where $\boldsymbol{\omega}^i$ is the global angular velocity vector of the rigid body i expressed with respect to the inertial reference system, $\boldsymbol{\omega}^j$ is the global angular velocity vector of the rigid body j expressed with respect to the inertial reference system, and $\boldsymbol{\omega}^{i,j}$ is the global angular velocity vector of the rigid body i measured with respect to the rigid body j. It is important to note that the angular velocity vectors $\boldsymbol{\omega}^i$, $\boldsymbol{\omega}^j$, and $\boldsymbol{\omega}^{i,j}$ are all global vectors, that is, their algebraic components refer to the inertial reference system. More importantly, Equation (31) represents the fundamental building block for mathematically constructing the nonholonomic set of algebraic constraints describing the imposition of the relative angular velocity as an appropriate set of driving constraints. For this purpose, as already discussed in the manuscript, one can first express the angular velocity vectors $\boldsymbol{\omega}^i$ and $\boldsymbol{\omega}^j$ in the following general form:

$$\boldsymbol{\omega}^i = \mathbf{G}^i\dot{\boldsymbol{\theta}}^i = \mathbf{W}^i\dot{\mathbf{q}}^i, \quad \boldsymbol{\omega}^j = \mathbf{G}^j\dot{\boldsymbol{\theta}}^j = \mathbf{W}^j\dot{\mathbf{q}}^j \tag{32}$$

where:

$$\mathbf{G}^i = \mathbf{A}^i\bar{\mathbf{G}}^i, \quad \mathbf{W}^i = \mathbf{A}^i\bar{\mathbf{W}}^i, \quad \mathbf{G}^j = \mathbf{A}^j\bar{\mathbf{G}}^j, \quad \mathbf{W}^j = \mathbf{A}^j\bar{\mathbf{W}}^j \tag{33}$$

By knowing the geometric properties of the multibody system under study, the relative angular velocity vector $\boldsymbol{\omega}^{i,j}$ can be written as follows:

$$\boldsymbol{\omega}^{i,j} = \Omega^h\mathbf{v}^h = \Omega^h\mathbf{A}^i\bar{\mathbf{v}}^h = \Omega^h\mathbf{A}^j\mathbf{A}^{i,j}\bar{\mathbf{v}}^h = \Omega^h\mathbf{A}^j\bar{\mathbf{v}}^h \tag{34}$$

where Ω^h represents the magnitude of the relative angular velocity vector, $\mathbf{v}^h$ is the global axis vector of the relative angular velocity vector, and $\bar{\mathbf{v}}^h$ denotes the local axis vector of the relative angular velocity vector. Similarly, the relative angular acceleration vector is denoted with the vector $\boldsymbol{\alpha}^{i,j}$ of dimensions $d \times 1$ and can be mathematically defined as:

$$\boldsymbol{\alpha}^{i,j} = \dot{\Omega}^h\mathbf{v}^h = \dot{\Omega}^h\mathbf{A}^i\bar{\mathbf{v}}^h = \dot{\Omega}^h\mathbf{A}^j\mathbf{A}^{i,j}\bar{\mathbf{v}}^h = \dot{\Omega}^h\mathbf{A}^j\bar{\mathbf{v}}^h \tag{35}$$

One can therefore readily formulate the desired nonholonomic conditions as follows:

$$\boldsymbol{\omega}^i - \boldsymbol{\omega}^j = \boldsymbol{\omega}^{i,j} \quad \Leftrightarrow \quad \mathbf{W}^i\dot{\mathbf{q}}^i - \mathbf{W}^j\dot{\mathbf{q}}^j = \Omega^h\mathbf{A}^j\bar{\mathbf{v}}^h \quad \Leftrightarrow \quad \boldsymbol{\Theta}^h = \mathbf{0} \tag{36}$$

where the corresponding constraint equations are embedded in the nonlinear constraint vector denoted with the vector $\boldsymbol{\Theta}^h$ of dimensions $n^h_{c,nh} \times 1$. The complete mathematical expression of the nonholonomic algebraic equations describing a generic relative angular rotation associated with the set of driving constraints labeled with h can be written in a general and compact vector form as follows:

$$\boldsymbol{\Theta}^h = \mathbf{W}^i\dot{\mathbf{q}}^i - \mathbf{W}^j\dot{\mathbf{q}}^j - \Omega^h\mathbf{A}^j\bar{\mathbf{v}}^h = \mathbf{0} \tag{37}$$

Additionally, the Jacobian matrix of the constraint equations associated with the relative angular velocity imposed between two spatial rigid bodies is denoted with the matrix $\boldsymbol{\Theta}^h_{\dot{\mathbf{q}}^h}$ of dimensions $n^h_{c,nh} \times 2n_b$ and it is given by:

$$\boldsymbol{\Theta}^h_{\dot{\mathbf{q}}^h} = \begin{bmatrix} \boldsymbol{\Theta}^h_{\dot{\mathbf{q}}^i} & \boldsymbol{\Theta}^h_{\dot{\mathbf{q}}^j} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^i & -\mathbf{W}^j \end{bmatrix} \tag{38}$$

Finally, the quadratic velocity vector of the constraint equations associated with the relative angular velocity imposed between two spatial rigid bodies is denoted with the vector $\mathbf{Q}^h_{d,\Theta}$ of dimensions $n^h_{c,nh} \times 1$ and it is given by:

$$\mathbf{Q}^h_{d,\Theta} = \boldsymbol{\alpha}^{i,j} + \boldsymbol{\omega}^j \times \boldsymbol{\omega}^{i,j} - \dot{\mathbf{W}}^i\dot{\mathbf{q}}^i + \dot{\mathbf{W}}^j\dot{\mathbf{q}}^j \tag{39}$$

The constraint quadratic velocity vector associated with the nonholonomic algebraic equations describing the driving constraints is a highly nonlinear vector that absorbs the terms that are quadratic with respect to the generalized velocities of the multibody system. This vector is of paramount importance for the correct computer implementation of the index-one form of the equations of motion that include the set of nonholonomic constraints. For the UAV system modeled as a multibody mechanical system in this study, the geometric properties of each driving constraint are reported in Table 4.

Table 4. Geometric properties of each driving constraint of the delivery drone.

Driving Constraint Number	**Driving Constraint Type**	**First Body Number**	**Second Body Number**	**Relative Geometric Axis**
h	Name	i_h	j_h	$\bar{\mathbf{v}}^h = \begin{bmatrix} \bar{v}_x^h & \bar{v}_y^h & \bar{v}_z^h \end{bmatrix}^T$
1	Imposed relative angular velocity	1	0	$\bar{\mathbf{v}}^1 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
2	Imposed relative angular velocity	2	0	$\bar{\mathbf{v}}^2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
3	Imposed relative angular velocity	3	0	$\bar{\mathbf{v}}^3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
4	Imposed relative angular velocity	4	0	$\bar{\mathbf{v}}^4 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$

4.5. External Forces and Control Actions

The multibody model of the UAV system under study is subjected to $N_e = 4$ external forces, which are identified as follows:

1. Gravitational field: force $l = 1$.
2. Wind disturbance: force $l = 2$.
3. Contact force I: force $l = 3$.
4. Contact force II: force $l = 4$.

Where each external force is unambiguously labeled with the integer number l. For a generic external force identified by the integer number l, which is associated with the force vector $\mathbf{F}_e^l$ of dimensions $d \times 1$, expressed in the global reference system and applied to the material point P of the rigid body l, the corresponding generalized external force vector denoted with $\mathbf{Q}_e^l$ having dimensions $n_b \times 1$ can be obtained as:

$$\mathbf{Q}_e^l = \left(\mathbf{L}^l(P)\right)^T \mathbf{F}_e^l = \begin{bmatrix} \mathbf{F}_e^l \\ \left(\bar{\mathbf{G}}^l\right)^T \left(\bar{\mathbf{u}}^l(P) \times \bar{\mathbf{F}}_e^l\right) \end{bmatrix} \tag{40}$$

In the multibody model of the UAV system at hand, while both the total gravity force and the disturbance action of the wind are applied in the center of mass G^0 of the central frame, the first and second contact forces are respectively applied at the central tip points A^0 and B^0 of the right and left T-shaped legs of the landing gear attached to the chassis. On the other hand, in the multibody model of the UAV system developed in this work, a number of control actions equal to $N_u = 4$ are considered for monitoring and guiding the three-dimensional motion of the system. More specifically, the following set of control torques applied to the propellers is assumed:

1. Propeller torque I: $s = 1$.
2. Propeller torque II: $s = 2$.
3. Propeller torque III: $s = 3$.
4. Propeller torque IV: $s = 4$.

Where each control torque is unambiguously labeled with the integer number s. For a generic control torque identified by the integer number s, which is associated with the torque vector $\bar{\mathbf{T}}_u^s$ of dimensions $d \times 1$ expressed in the body-fixed reference frame of the rigid body s, the corresponding generalized control force vector denoted with $\mathbf{Q}_u^s$ having dimensions $n_b \times 1$ can be obtained as:

$$\mathbf{Q}_u^s = (\bar{\mathbf{W}}^s)^T \bar{\mathbf{T}}_u^s = \begin{bmatrix} \mathbf{0} \\ (\bar{\mathbf{G}}^s)^T \bar{\mathbf{T}}_u^s \end{bmatrix} \tag{41}$$

As discussed in the description of the propulsion system, there is an explicit nonlinear relationship between the magnitude of each control torque $\bar{\mathbf{T}}_u^s$, which is denoted with M_p^s and is applied to the generic propeller s, and the angular velocity imposed on the same propeller given by $\Omega^s = 2\pi N^s / 60$, where N^s indicates the revolutions per minute of the propeller labeled with the integer number s.

4.6. Differential-Algebraic Equations of Motion

The differential-algebraic form of the UAV system equations of motion modeled as a multibody system can be readily obtained through a standard and systematic assembly procedure. By doing so, the system mass matrix denoted with $\mathbf{M}$ and having dimensions $n_q \times n_q$ is given by:

$$\mathbf{M} = \mathop{\mathrm{A}}_{i=0}^{N_b-1} \mathbf{M}^i = \begin{bmatrix} \mathbf{M}^0 & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{M}^1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{M}^2 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M}^3 & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M}^4 \end{bmatrix} \tag{42}$$

where the symbol A stands for representing the standard assembly process employed in the field of multibody system dynamics. Similarly, the system inertia quadratic velocity vector denoted with $\mathbf{Q}_v$ and having dimensions $n_q \times 1$ can be readily assembled as follows:

$$\mathbf{Q}_v = \mathop{\mathrm{A}}_{i=0}^{N_b-1} \mathbf{Q}_v^i = \begin{bmatrix} \mathbf{Q}_v^0 \\ \mathbf{Q}_v^1 \\ \mathbf{Q}_v^2 \\ \mathbf{Q}_v^3 \\ \mathbf{Q}_v^4 \end{bmatrix} \tag{43}$$

The system generalized external force vector denoted with $\mathbf{Q}_e$ and having dimensions $n_q \times 1$ can be readily assembled as follows:

$$\mathbf{Q}_e = \mathop{\mathrm{A}}_{l=1}^{N_e} \mathbf{Q}_e^l = \begin{bmatrix} \mathbf{Q}_e^1 + \mathbf{Q}_e^2 + \mathbf{Q}_e^3 + \mathbf{Q}_e^4 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{44}$$

The system generalized control force vector denoted with $\mathbf{Q}_u$ and having dimensions $n_q \times 1$ can be readily assembled as follows:

$$\mathbf{Q}_u = \mathop{\mathrm{A}}_{s=1}^{N_u} \mathbf{Q}_u^s = \begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_u^1 \\ \mathbf{Q}_u^2 \\ \mathbf{Q}_u^3 \\ \mathbf{Q}_u^4 \end{bmatrix} \tag{45}$$

The system vector of holonomic algebraic constraints denoted with $\mathbf{\Psi}$ and having dimensions $n_{c,h} \times 1$ can be readily assembled as follows:

$$\mathbf{\Psi} = \mathop{\mathrm{A}}_{k=1}^{N_{c,h}} \mathbf{\Psi}^k = \begin{bmatrix} \mathbf{\Psi}^1 \\ \mathbf{\Psi}^2 \\ \mathbf{\Psi}^3 \\ \mathbf{\Psi}^4 \end{bmatrix} \tag{46}$$

The system vector of nonholonomic algebraic constraints denoted with $\boldsymbol{\Theta}$ and having dimensions $n_{c,nh} \times 1$ can be readily assembled as follows:

$$\boldsymbol{\Theta} = \mathop{\mathrm{A}}\limits_{h=1}^{N_{c,nh}} \boldsymbol{\Theta}^h = \begin{bmatrix} \boldsymbol{\Theta}^1 \\ \boldsymbol{\Theta}^2 \\ \boldsymbol{\Theta}^3 \\ \boldsymbol{\Theta}^4 \end{bmatrix} \tag{47}$$

The system Jacobian matrix relative to the set of holonomic algebraic equations denoted with $\boldsymbol{\Upsilon}_{\mathbf{q}}$ and having dimensions $n_{c,h} \times n_q$ can be readily assembled as follows:

$$\boldsymbol{\Upsilon}_{\mathbf{q}} = \mathop{\mathrm{A}}\limits_{k=1}^{N_{c,h}} \boldsymbol{\Upsilon}^k_{\mathbf{q}^k} = \begin{bmatrix} \boldsymbol{\Upsilon}^1_{\mathbf{q}^0} & \boldsymbol{\Upsilon}^1_{\mathbf{q}^1} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \boldsymbol{\Upsilon}^2_{\mathbf{q}^0} & \mathbf{O} & \boldsymbol{\Upsilon}^2_{\mathbf{q}^2} & \mathbf{O} & \mathbf{O} \\ \boldsymbol{\Upsilon}^3_{\mathbf{q}^0} & \mathbf{O} & \mathbf{O} & \boldsymbol{\Upsilon}^3_{\mathbf{q}^3} & \mathbf{O} \\ \boldsymbol{\Upsilon}^4_{\mathbf{q}^0} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \boldsymbol{\Upsilon}^4_{\mathbf{q}^4} \end{bmatrix} \tag{48}$$

The system Jacobian matrix relative to the set of nonholonomic algebraic equations denoted with $\boldsymbol{\Theta}_{\dot{\mathbf{q}}}$ and having dimensions $n_{c,nh} \times n_q$ can be readily assembled as follows:

$$\boldsymbol{\Theta}_{\dot{\mathbf{q}}} = \mathop{\mathrm{A}}\limits_{h=1}^{N_{c,nh}} \boldsymbol{\Theta}^h_{\dot{\mathbf{q}}^h} = \begin{bmatrix} \boldsymbol{\Theta}^1_{\dot{\mathbf{q}}^0} & \boldsymbol{\Theta}^1_{\dot{\mathbf{q}}^1} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \boldsymbol{\Theta}^2_{\dot{\mathbf{q}}^0} & \mathbf{O} & \boldsymbol{\Theta}^2_{\dot{\mathbf{q}}^2} & \mathbf{O} & \mathbf{O} \\ \boldsymbol{\Theta}^3_{\dot{\mathbf{q}}^0} & \mathbf{O} & \mathbf{O} & \boldsymbol{\Theta}^3_{\dot{\mathbf{q}}^3} & \mathbf{O} \\ \boldsymbol{\Theta}^4_{\dot{\mathbf{q}}^0} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \boldsymbol{\Theta}^4_{\dot{\mathbf{q}}^4} \end{bmatrix} \tag{49}$$

The system constraint quadratic velocity vector relative to the set of holonomic algebraic equations denoted with $\mathbf{Q}_{d,\Upsilon}$ and having dimensions $n_{c,h} \times 1$ can be readily assembled as follows:

$$\mathbf{Q}_{d,\Upsilon} = \mathop{\mathrm{A}}\limits_{k=1}^{N_{c,h}} \mathbf{Q}^k_{d,\Upsilon} = \begin{bmatrix} \mathbf{Q}^1_{d,\Upsilon} \\ \mathbf{Q}^2_{d,\Upsilon} \\ \mathbf{Q}^3_{d,\Upsilon} \\ \mathbf{Q}^4_{d,\Upsilon} \end{bmatrix} \tag{50}$$

The system constraint quadratic velocity vector relative to the set of nonholonomic algebraic equations denoted with $\mathbf{Q}_{d,\Theta}$ and having dimensions $n_{c,nh} \times 1$ can be readily assembled as follows:

$$\mathbf{Q}_{d,\Theta} = \mathop{\mathrm{A}}\limits_{h=1}^{N_{c,nh}} \mathbf{Q}^h_{d,\Theta} = \begin{bmatrix} \mathbf{Q}^1_{d,\Theta} \\ \mathbf{Q}^2_{d,\Theta} \\ \mathbf{Q}^3_{d,\Theta} \\ \mathbf{Q}^4_{d,\Theta} \end{bmatrix} \tag{51}$$

By employing the fundamental principles of classical mechanics and introducing two sets of Lagrange multipliers, namely a set of $n_{c,h}$ Lagrange multipliers grouped in the vector $\boldsymbol{\lambda}$ and associated with the holonomic algebraic constraints, as well as a set of $n_{c,nh}$ Lagrange multipliers grouped in the vector $\boldsymbol{\mu}$ and associated with the nonholonomic algebraic constraints, the differential-algebraic form of the multibody equations of motion describing the UAV system under consideration can be written as follows:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_v + \mathbf{Q}_e + \mathbf{Q}_u - \boldsymbol{\Upsilon}^T_{\mathbf{q}}\boldsymbol{\lambda} - \boldsymbol{\Theta}^T_{\dot{\mathbf{q}}}\boldsymbol{\mu} \\ \boldsymbol{\Upsilon} = \mathbf{0} \\ \boldsymbol{\Theta} = \mathbf{0} \end{cases} \tag{52}$$

From a mathematical point of view, this is a nonlinear set of index-three differential-algebraic equations, featuring holonomic and nonholonomic constraints, which is particularly challenging to solve analytically. Consequently, one needs to resort to proper numerical methods for obtaining an approximate solution to the problem at hand, which must be reliable, accurate, and physically consistent. To achieve this goal, two general approaches can be followed, although several viable computational approaches exist in the multibody literature [61]. The two fundamental methods suitable for numerically solving the general form of the multibody equations of motion are referred to as the Augmented Formulation (AF) and the Embedding Technique (ET) [59]. In the AF, the system equations of motion are properly transformed from the index-three form to the index-one form and are subsequently rearranged in a matrix form that allows for the explicit computation of the system generalized acceleration vector. In the ET, on the other hand, the system equations of motion are readily transformed from the index-three form to the index-one form, and subsequently, a proper velocity transformation matrix is used to obtain only the generalized acceleration vector corresponding to the system degrees of freedom. However, both the AF and the ET numerically suffer from the unwanted phenomenon of the drift of the algebraic constraints, and therefore, they require the use of an effective stabilization technique to enforce the constraint equations at the position and velocity levels [62–64]. The AF is the general multibody formulation approach employed in this research work for performing the dynamic analysis of the delivery drone. Therefore, the resulting index-one set of dynamic equations can be written as follows:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_v + \mathbf{Q}_e + \mathbf{Q}_u - \boldsymbol{\Psi}_{\mathbf{q}}^T\boldsymbol{\lambda} - \boldsymbol{\Theta}_{\dot{\mathbf{q}}}^T\boldsymbol{\mu} \\ \boldsymbol{\Psi}_{\mathbf{q}}\ddot{\mathbf{q}} = \mathbf{Q}_{d,\Psi} \\ \boldsymbol{\Theta}_{\dot{\mathbf{q}}}\ddot{\mathbf{q}} = \mathbf{Q}_{d,\Theta} \end{cases} \tag{53}$$

As mentioned before, a reliable constraint stabilization procedure is necessary for correctly solving the index-one form of the differential-algebraic multibody equations. To this end, the generalized coordinate partitioning method is adopted in this investigation [65,66].

4.7. Underactuated Dynamical System

Depending on the type, the number, and the collocation of the control actions, a controlled multibody mechanical system can be modeled as a fully actuated or an underactuated dynamical system [67,68]. In the case of a fully actuated multibody mechanical system, the number of independently manipulable control actions denoted with n_u is equal to the number of degrees of freedom of the system denoted with n_f, that is, $n_u = n_f$. Conversely, in the case of an underactuated multibody mechanical system, the number of control actions is less than the number of degrees of freedom, that is, $n_u < n_f$. The quadcopter designed in this research work has four motors that rotate the propellers at different velocities. The torques of the motors generate the thrust forces that allow the drone to move through space. Physically, the UAV system at hand is modeled as a rigid body with six degrees of freedom. However, the angular velocities of the propellers are the only parameters that can be arbitrarily controlled to guide the motion of the UAV system. As a result, the system under study turns out to be an underactuated multibody mechanical system because one has $n_f = 6$ and $n_u = 4$. Since the UAV system does not possess an actuator for each degree of freedom, some movements are not directly controllable. For example, in order to move the UAV system forward, the delivery drone must first rotate in the desired direction. In general, from a mathematical point of view, the underactuation characteristic of a multibody mechanical system can be identified and analyzed by observing the structure of the generalized control force vector denoted with $\mathbf{Q}_u$ that is used for controlling the system motion [69,70]. In mechanical engineering applications, this vector features the following general mathematical structure:

$$\mathbf{Q}_u = \mathbf{B}_u\mathbf{u} \tag{54}$$

where $\mathbf{u}$ is a vector of control actions having dimensions $n_u \times 1$ and $\mathbf{B}_u$ is the positioning matrix describing the collocation of the actuators that has dimensions $n_q \times n_u$. By analyzing the rank of the actuator positioning matrix relative to the multibody model of the delivery drone system under study, one obtains:

$$r_u = \mathrm{rank}(\mathbf{B}_u) = n_u < n_f \tag{55}$$

The rank r_u of the positioning matrix of the actuators is equal to the number of manipulable control actions n_u but it is lower than the degrees of freedom of the multibody system n_f. Therefore, the dynamical system at hand is an underactuated multibody mechanical system. As mentioned before, underactuated mechanical systems are dynamic systems that are particularly challenging to control since some movements or maneuvers cannot be implemented [71]. However, as discussed below in the paper, by properly defining the time laws of the control actions applied to the UAV system under study, as well as by cleverly combining together some movements involving the degrees of freedom of the drone system, it is possible to effectively achieve control of the system motion in realistic scenarios.

5. UAV System Control Development

5.1. Control Strategy

In this section, the fundamental aspects addressed in the development of the control system suitable for the UAV system considered as the case study are described. In general, controlling a dynamical system means properly choosing the actions to be implemented to modify and to improve the behavior of the mechanical system of interest [72]. By acting on the forces, moments, and torques that are manipulable as the input variables of the mechanical system, motion control can be achieved in output. The control strategy depends on the specific mechanical system under study. For the UAV system analyzed in this investigation, a combination of a feedforward (open-loop) plus feedback (closed-loop) control scheme is utilized for achieving the motion control, as discussed in detail below.

5.2. PID Controller

To achieve the attitude control and the trajectory tracking of the UAV system, the control method best suited to the case at hand is the closed-loop (feedback) control method. In this type of control, the control actions are based on the instant-by-instant variations of the state of the dynamical system. Although there are more advanced feedback control techniques, the one most widely used in industry is the Proportional-Integral-Derivative (PID) controller. Accordingly, the closed-loop control scheme employed in this work is represented in Figure 12.

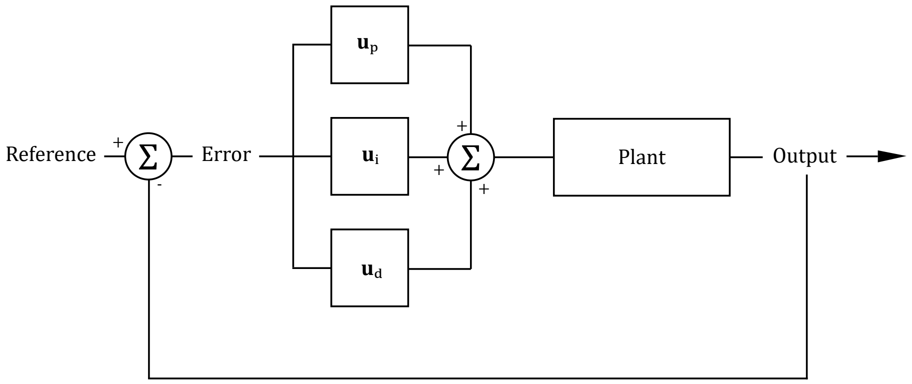


Figure 12. PID controller scheme.

As shown in Figure 12, the input of the PID controller is the error signal, which is obtained by comparing the actual time evolution of the dynamical system, measured through the sensors with the reference trajectory, which is planned offline in advance to meet the desired goals. In general, the total control action of a PID controller denoted with

$\mathbf{u}$ consists of three separate contributions: a contribution proportional to the error denoted with $\mathbf{u}_p$, a contribution proportional to the integral over time of the error denoted with $\mathbf{u}_i$, and a contribution that is proportional to the derivative of the error over time, denoted with $\mathbf{u}_d$. That is:

$$\mathbf{u} = \mathbf{u}_p + \mathbf{u}_i + \mathbf{u}_d \tag{56}$$

where:

$$\mathbf{u}_p = -\mathbf{K}_c\mathbf{e}, \quad \mathbf{u}_i = -\mathbf{H}_c\bar{\mathbf{e}}, \quad \mathbf{u}_d = -\mathbf{R}_c\dot{\mathbf{e}} \tag{57}$$

and

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_r, \quad \bar{\mathbf{e}} = \int_0^{\bar{t}} \mathbf{e}(\bar{t})d\bar{t}, \quad \dot{\mathbf{e}} = \frac{d\mathbf{e}}{dt} \tag{58}$$

with $\mathbf{q}$ and $\mathbf{q}_r$ being the actual and reference dynamical behaviors of the mechanical system, respectively, whereas $\mathbf{K}_c$, $\mathbf{H}_c$, and $\mathbf{R}_c$ identify the coefficient matrices of the proportional, integral, and derivative terms of the PID controller, respectively. In practical applications, the parameters of a given PID controller reduce often to scalar terms that are simply identified with the initial of their letters, namely, P, I, and D. Since the desired dynamical behavior denoted by the vector $\mathbf{q}_r$ is, in general, a function of time resulting from the preliminary phase of motion planning, the PID control strategy formulated herein can be inherently considered as a combination of a feedback plus feedforward control schemes. A strategy of this type is very robust as it seeks to minimize, by its nature, the deviation of the system behavior from the input instructions. However, the coefficient matrices of the PID controller must be properly tuned to obtain a good performance from the system guidance and the control of the motion.

In the virtual prototype of the UAV system, the PID controller provides as outputs the time laws and assigns them to the torques generated by the four motors. More specifically, in the computer simulation, the output of the PID controller is a torque, while in the real-world case, the actual controller returns voltage values that the motors and the propellers transform into propulsion forces. The physical transformation from the input voltage to the thrust force can be performed through the use of mathematical laws. For this purpose, the angular velocity of the propellers is proportional to the input voltage of the motors and can be expressed as follows:

$$N = K_V V \tag{59}$$

where V denotes the input voltage, N represents the revolutions per minute of the motor, and K_V is the constant that transforms the angular velocity of the motor into the input voltage. The mathematical laws that relate the angular velocities of the motors to the thrust force were discussed in detail in Section 2.

5.3. UAV System Control

In this subsection, the characteristics of the control scheme for the UAV system devised in this study are described. More precisely, to deliver the set of beers, the drone must follow a trajectory consisting of a vertical take-off phase, a horizontal movement, and an eventual yaw motion [73–75]. Consequently, the proposed control system simulates this trajectory using three separate PID controllers for the three movements, namely the vertical motion, the forward motion, and the yaw motion, as discussed in detail below.

5.3.1. Hovering and Vertical Motion

The first problem in the development of the control system is to make the delivery drone reach a predetermined altitude and keep it in balance. From a physical standpoint, in the center of mass of the drone acts the gravity force, while the four thrust forces, generated by the propulsion system, are applied to the edge of each arm. As a first approximation, one can consider the center of mass as being equidistant from the four electric motors. In the design phase, the authors tried to arrange the components symmetrically so that the propulsion forces, all equal in magnitude, and the gravity force do not generate moments in the hovering configuration. This peculiar behavior of the delivery drone is schematized in Figure 13.

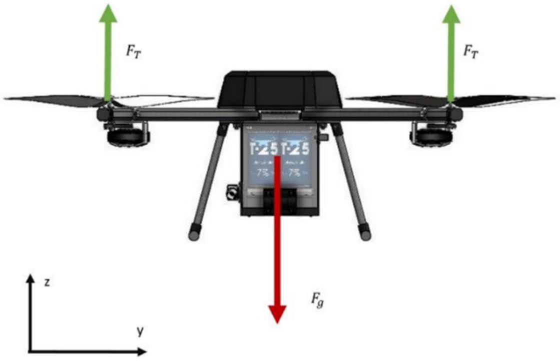


Figure 13. Hovering and vertical motion.

If the vector sum of the forces is positive along the z-axis, there is an acceleration that allows the quadcopter to lift [76]. In the opposite case, if the sum of the forces is negative along the z-axis, there is a decrease in altitude. Consequently, the hovering phase occurs when the sum of the forces is zero, as summarized in the equations reported below:

$$\begin{cases} \sum_{i=1}^{4} F_{T_i} + F_g > 0, & \text{ascending motion} \\ \sum_{i=1}^{4} F_{T_i} + F_g < 0, & \text{descending motion} \\ \sum_{i=1}^{4} F_{T_i} + F_g = 0, & \text{hovering} \end{cases} \tag{60}$$

A closed-loop PID controller attains the regulation of the vertical thrust. The values obtained in the PID adjustment are a consequence of an iterative process. In general, the overall performance of the system changes as the control parameters are modified according to the behavior reported in Table 5.

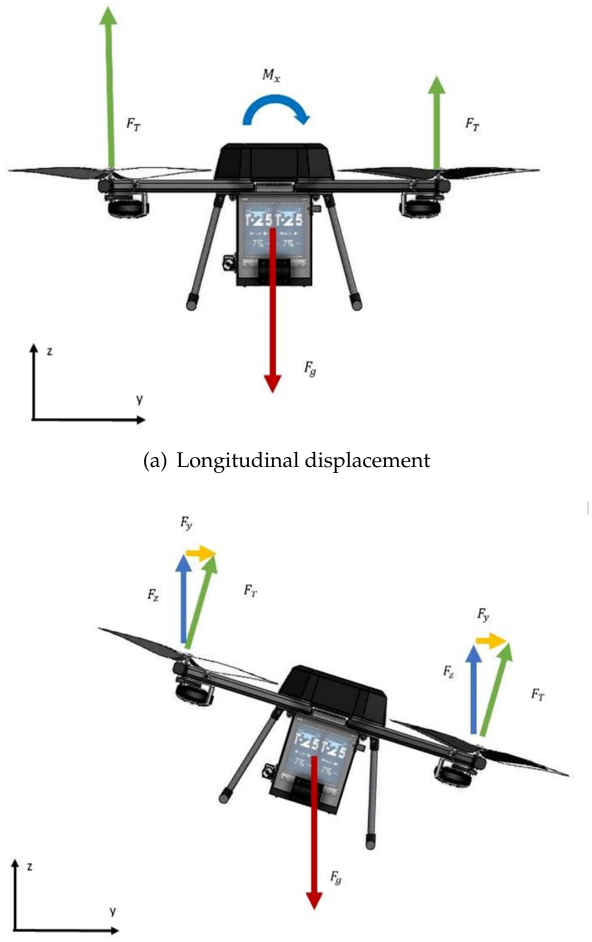
Table 5. PID value adjustments.

Parameter	Rise Time	Overshoot	Settling Time	Error	Stability
P	decreases	increases	limited changes	decreases	get worse
I	decreases	increases	increases	could cancel	get worse
D	increases	decreases	decreases	no effect	improve

In the computer simulation of the UAV system analyzed in this work, the input provided is a ramp signal that increases the altitude of the drone, followed by a stationary signal that helps to maintain the final altitude. At the end of the hovering phase, the control signal gradually decreases to ensure a safe landing of the delivery drone. In an actual drone system, the height feedback signal comes from a special ultrasonic detector. This detector sends a sound wave toward the ground and measures the height, calculating the time it takes to return to the device. Alternatively, for higher altitudes, the height can be measured using a barometer that obtains it as a function of pressure variation. The feedback is subtracted from the input to obtain the error. In the computer simulation carried out using SIMSCAPE MULTIBODY, on the other hand, precise position measurements are provided by the mechanical joints.

5.3.2. Longitudinal and Lateral Motions

The quadcopter cannot perform a simple translational motion decoupled from a rotational motion because it does not have the necessary actuators [77,78]. For example, the delivery drone needs to perform a roll movement in the first step to move right. To perform the rotation, the motors must rotate at different angular rates. In particular, the motors collocated opposite to the chosen motion direction must accelerate. This gives the delivery drone the required net torque necessary to rotate around the longitudinal axis. The vertical component of the thrust force and the force of gravity balance each other, while the horizontal component of the thrust force serves as the propulsion force that allows for realizing the translation. This configuration is schematized in Figure 14. More specifically, the longitudinal displacement is schematically shown in Figure 14a, while the lateral displacement is schematically shown in Figure 14b.



(a) Longitudinal displacement

(b) Lateral displacement

Figure 14. Longitudinal and lateral motions.

Furthermore, a cascaded PID control system guides the forward–backward two-dimensional motion of the UAV system, as well as its longitudinal and lateral motions, as shown in Figure 15.

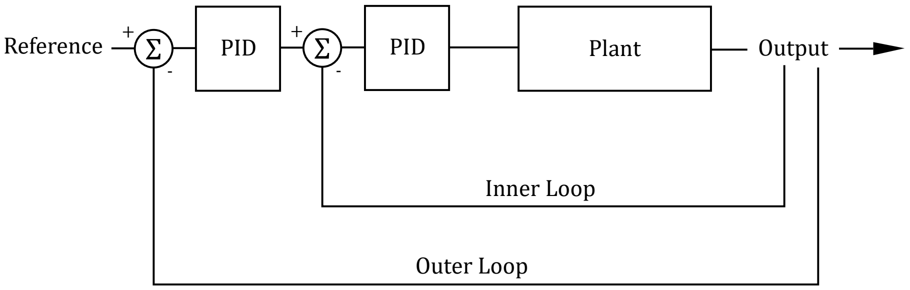


Figure 15. Cascaded PID controller.

In this type of feedback control scheme, there are two loops. The first one, called the outer loop, takes as a reference the required position and generates a signal input for the second loop, whereas the inner loop serves for realizing the attitude control of the delivery drone. The GPS signal feedback of a real drone is simulated by reading this information from the presence of proper mechanical joints in SIMSCAPE MULTIBODY. The trajectory is planned during the design phase, while the roll angle input measurement takes place during the flight using drone sensors such as gyroscopes and accelerometers. The difference between the outer loop output and the measurement from the sensors will be the input of the inner loop. As shown in detail below, an internal MATLAB tool, called PID Tuner, allows for simplifying the tuning of the parameters that characterize the two PID controllers.

5.3.3. Yaw Motion

The four propellers of the UAV system move two clockwise and two counterclockwise. They are alternated to avoid the generation of a mechanical moment around the relative vertical axis during roll and pitch operations. The fundamental static equation that describes the yaw motion of the delivery drone can be written as follows:

$$\sum_{i=1}^{4} \mathbf{M}_i = \mathbf{0} \tag{61}$$

where $\mathbf{M}_i$ represents the net torque produced by the propeller i. By changing the angular velocities of the propellers, the total mechanical moment around the z-axis will no longer be zero. Consequently, the drone will start to rotate on itself, as mathematically described by the following dynamic equation:

$$\sum_{i=1}^{4} \mathbf{M}_i + I_{zz}\dot{\boldsymbol{\omega}} = \mathbf{0} \tag{62}$$

where $\boldsymbol{\omega}$ denotes the global angular velocity vector of the delivery drone and I_{zz} identifies the vertical mass moment of inertia of the UAV system. The angular velocity of the propellers that rotate clockwise must be higher than those that rotate counterclockwise so that the drone rotates counterclockwise. Conversely, if the motors that rotate counterclockwise run faster than the others, the drone will rotate clockwise.

The electronic control of the yaw motion is always of the closed-loop PID type. The delivery drone follows the yaw angle reference provided in the design phase. The gyroscopes and accelerometers inboard the UAV system give a feedback signal to the system. The approach used in the simulation is to apply external forces to the axes of the propellers. The propellers will rotate because of the conversion of the force rates into angular velocity rates. The total mechanical moment about the vertical axis of the delivery drone is given by:

$$M_z = b_T(-F_{T_1} + F_{T_2} - F_{T_3} + F_{T_4}) \tag{63}$$

where b_T is the drag constant that is calculated empirically. The rate of thrust required for the yaw motion is added to one pair of motors moving in phase and subtracted from the other two. Because of this, the overall thrust force does not change compared to the case of no yaw motion. The rate of thrust for yawing augments the total thrust of the two motors rotating simultaneously and decreases the thrust of the other two. Therefore, the global thrust does not change compared to the no-yaw case.

5.4. PID Tuner

The MATLAB tool called PID Tuner is an extension included in the SIMULINK Control Design Toolbox. Based on a linearized approximation of the dynamical system of interest, also called the plant, it automatically provides the proportional, integral, and derivative values of the parameters of a PID controller. To this end, the PID Tuner considers the plant as the combination of all blocks between the output and input of the PID controller. The PID Tuner bases its initial design on the open-loop frequency response of the linearized system. However, the open-loop frequency-based model typically fails to properly simulate the outputs of a complex system. It is necessary to identify a new plant through the input and output data simulated by a proper MATLAB program that has been specifically designed for this purpose. This program has several presets, one of which is the step impulse, whose settings can be manually adjusted. At the end of the dynamical simulation, an output curve is obtained. The software proposes various linear models to approximate the output curve. They are transfer functions, in which parameters can be precisely adjusted either manually or automatically. Transfer functions are functions that model the dynamic behavior of a mechanical system in the frequency domain. The automatic estimation makes use of iterative search models, including the Gauss-Newton algorithm. The linearized model provides the software with a basis on which to adjust the PID parameters. It is possible to refine the response amplitude–time curve according to preference by adjusting the desired robustness and rise time. All the controlled models proposed by the software have an accuracy of greater than 90 percent. This means that the linearized model proposed by the MATLAB program corresponds to the real behavior in this percentage. In summary, the parameters of the control system developed for solving the regulation and tracking problems of the multibody model of the UAV system analyzed in this investigation are thoroughly described by the data reported in Table 6.

Table 6. Numerical values of the controller parameters in the ascending, descending, and hovering phases.

DISPLACEMENT	TYPE	LOOPS	P	I	D
LIFT	PID	Simple loop	28	22	18
YAW	PID	Simple loop	129.11	81.29	7.17
ROLL	CASCADED PID	Inner loop Outer loop	−3.93 0.52	−0.09 0.037	−15.68 0.74
PITCH	CASCADED PID	Inner loop outer loop	3.93 0.52	0.089 0.037	15.68 0.74

Table 6 contains the parameter values of the PID controllers employed in dynamic simulations as described in Sections 6 and 7. This table has four rows. The controlled movement is reported in each row, while on the columns, there are further specifications such as the type of PID controller, the type of loop, and the values of the controller coefficients (proportional, integrative, and derivative).

6. SIMSCAPE MULTIBODY Model Description

6.1. Computer Model of the Delivery Drone

This subsection covers the description of the SIMSCAPE MULTIBODY model of the UAV system considered as the case study of this investigation. An overview of the SIMSCAPE MULTIBODY model of the delivery drone is represented in Figure 16.

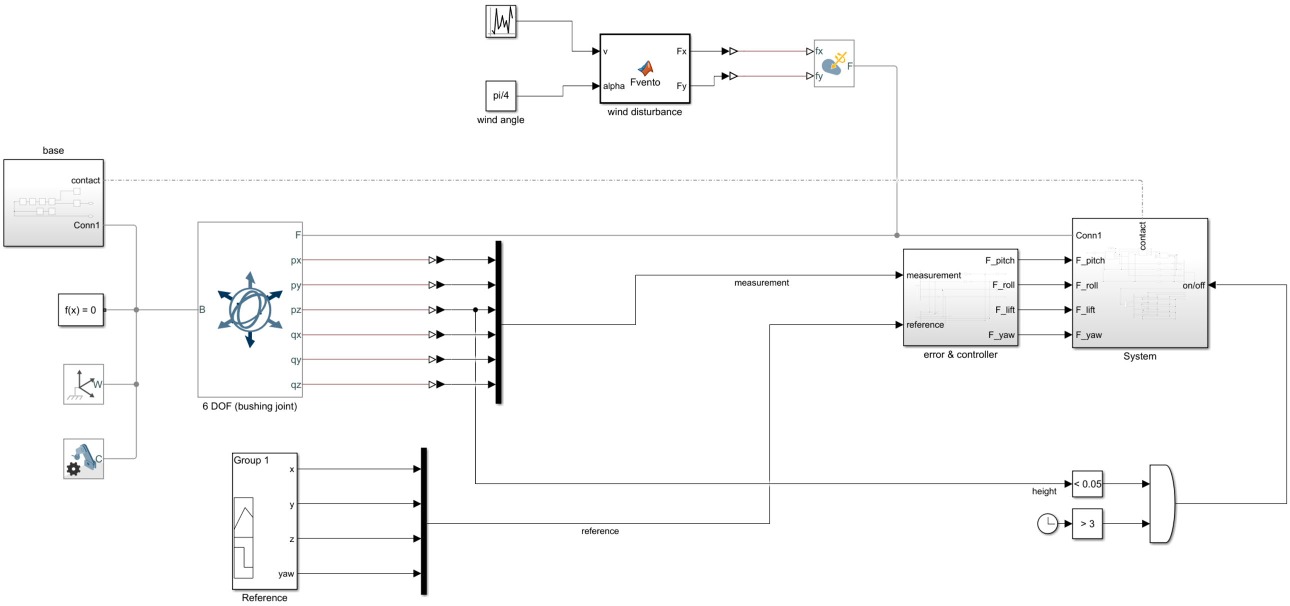


Figure 16. Delivery drone SIMSCAPE MULTIBODY model overview.

The computer implementation of the dynamical model in the SIMSCAPE MULTIBODY environment is described by analyzing one by one the individual subsystems and their peculiarities. The modeling of the system was completed as anticipated in MATLAB using the SIMSCAPE MULTIBODY software environment [79,80]. The various subsystems represented in Figure 16 are analyzed in detail below.

The CAD model developed using SOLIDWORKS was imported in SIMSCAPE MULTIBODY as a single rigid body to simplify the subsequent computer implementation of the dynamical simulations. Therefore, the complex model realized in a SOLIDWORKS environment was exported as a single ".SLDPRT" file. The four propellers were imported as four single ".SLDPRT" files. To model the six degrees of freedom of the drone in a three-dimensional space, a bushing joint without any force element was taken from the SIMSCAPE MULTIBODY library. This joint utilizes the Tait-Bryan angles commonly used in aviation to describe the rotation of a three-dimensional rigid body. As discussed in detail in Section 4, each propeller has a rotational degree of freedom, which was modeled in SIMSCAPE MULTIBODY with a revolute joint. The actuation forces acting on the frame are provided by a feedback control system and, in the case study under consideration, PID controllers were chosen, as discussed in detail in Section 5.

The dynamical model is provided with a reference input that describes the trajectory to be tracked. The input trajectory refers to the Cartesian coordinates of the drone center of mass and is provided with signal builders in the SIMULINK library. These blocks allow for the construction of complex time laws to guide the dynamic system. In particular, since the delivery drone is only endowed with four control actuators, four control inputs are supplied to the dynamical model to guide its motion: the vertical height to be reached (translation along the z-axis), the longitudinal and lateral positions with respect to a horizontal plane (translations along the x-axis and the y-axis), and a yaw angle with respect to the same reference plane (rotation about the z-axis). The control subsystem calculates the error concerning the measured position using the given input. The measured position is provided by the mechanical joint that simulates the sensors of a real drone. The PID controller takes in input the error and converts it into a control action, which is represented by a force quota that will be applied to the dynamic model, in correspondence with the axes of the propellers. As discussed in the paper, this is equivalent to applying control torques to the propellers or imposing the magnitudes of their relative angular velocities. In addition to the actuation forces, other external forces were considered to simulate real-world conditions by following the multibody model developed in Section 4. Primarily, these are the force of gravity, and secondly, a stochastic force that models the disturbance action of the wind.

As anticipated, the signal builder block available in SIMULINK allows for providing the reference inputs for the movement of the UAV system to the computer implementation of the system equations of motion described in Section 4. In particular, the altitude is a discontinuous function that has an ascending phase to set the desired altitude to be reached, a constant phase to achieve hovering, and a descending phase for landing [81–83]. The two-dimensional motion follows the same pattern as the height signal, except for the descending phase. After the hovering stage, the drone receives the displacement signal along one axis and reaches the first desired position. Then, the second signal is activated, thereby generating a displacement along the second axis. The two-dimensional displacement signals can also be combined. Once the two-dimensional movement is accomplished, the drone performs a turning maneuver around its axis.

In the proposed SIMSCAPE MULTIBODY model, the drone position is measured instant-by-instant using the bushing joint in six separate signals: three spatial displacements (longitudinal displacement, lateral displacement, and vertical displacement) and three finite rotations (roll angle, pitch angle, and yaw angle). As shown in Figure 17, these signals are connected to the measurement subsystem, which provides the error concerning the reference that takes the PID controller as input.

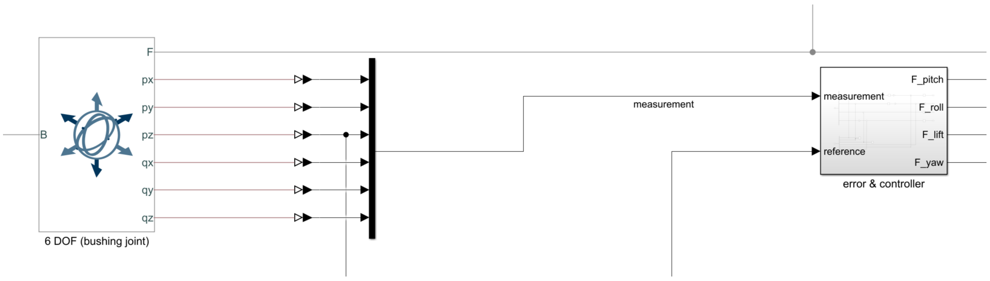


Figure 17. Measurement subsystem.

The error and controller subsystem shown in Figure 18 consists of four independent PID controllers, whose logic structures are described in detail in Section 5.

The first controller is the one that regulates altitude. It takes an error as the input and generates a proportional reaction. The error is given by a block ('Sum') that calculates the difference between the instantaneous values of the references and the measurements. This controller is based on a feedback loop. The second and third ones are two-dimensional displacement controllers, modeled with a cascaded PID controller. The system is analogous, in the case of the longitudinal and lateral displacements, along the x and y axes. Primarily, the outermost PID block receives the position-related error. This PID generates a signal that rotates the drone, creating a roll or pitch angle. The resulting angle is compared to the angular position, measured by the bushing joint. The resulting error between the two angles is used by the following PID to generate the force term required for the displacement. The fourth controller is the yaw control, which consists of a feedback controller in which an angular position is given as input. Similarly to the altitude PID controller, the error is measured as the difference between the law provided by the signal builder and the instantaneous position measured by the joint. Subsequently, a mathematical function block converts the force rate to the angular velocity of the propellers. The change in angular velocity generates a non-zero moment of inertia, thereby allowing for the turning motion to be achieved. The function that allows for the transformation from a force signal to an angular velocity signal also measures the output voltage, the revolutions per minute, and the absorbed power. The value of the multiplication constant is obtained from experimental data related to the propellers.

The drone subsystem represented in Figure 19 consists of the drone frame and the mathematical functions for the conversion of the force quotas in the rotation of the propellers.

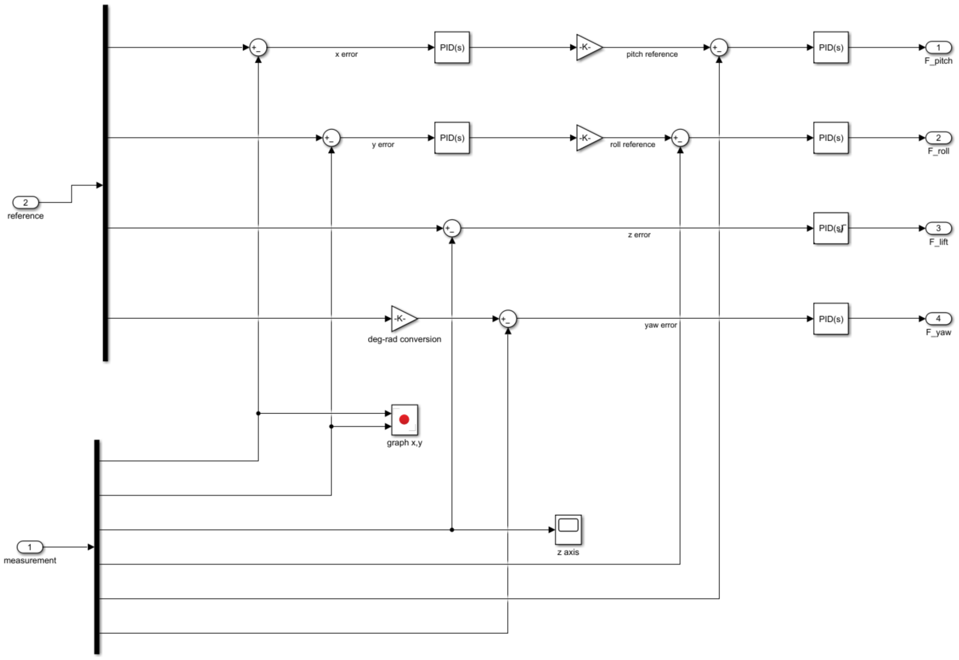


Figure 18. Error and controller subsystem.

As already discussed before, the motion of the delivery drone is generated by four forces actuated by the output of the PID controllers. To allow for proper motion, some blocks in SIMSCAPE MULTIBODY perform the sum/difference of the components of the forces to recreate the torques necessary for the motion of the UAV system. For instance, for the hovering maneuver, the thrust forces that the four motors must provide are all concordant and determined in this manner. Contrarily, in the case of the rolling maneuver, two of the four forces will be concordant, and the other two, discordant. In the SIMSCAPE MULTIBODY model, the thrust forces are applied to the frame at the propeller axes through an external force block. To avoid the power of the motors exceeding the constructive specifications imposed by the manufacturer, saturation blocks were used in order to limit the maximum force to an imposed value. The same idea was implemented to regulate the minimum thrust force, which cannot be negative.

In order to simulate the wind disturbances, an external force was included in the dynamic model implemented in SIMSCAPE MULTIBODY. The force varies with the wind velocity, which is randomly generated using a special function. This force is applied to the center of mass of the UAV system and is described by the following mathematical law:

$$F_v = \frac{1}{2} C_s S \rho v^2 \tag{64}$$

where $S \cong 0.06$ (m^2) represents the reference area of the drone opposing the current, $\rho = 1.22$ (kg/m^3) is the standard value of the air density, $C_s = 1(-)$ denotes the shape coefficient, and v identifies the relative wind velocity that results from the computer simulation performed in SIMSCAPE MULTIBODY. The force law described before was implemented in SIMSCAPE MULTIBODY through a user-defined mathematical function.

To ensure a proper landing, a plane on which the drone will come to rest at the end of the simulation was modeled in SIMSCAPE MULTIBODY as well. As represented in Figure 20, the brick solid 'Base1' was made rigid and modeled with a spatial contact force.

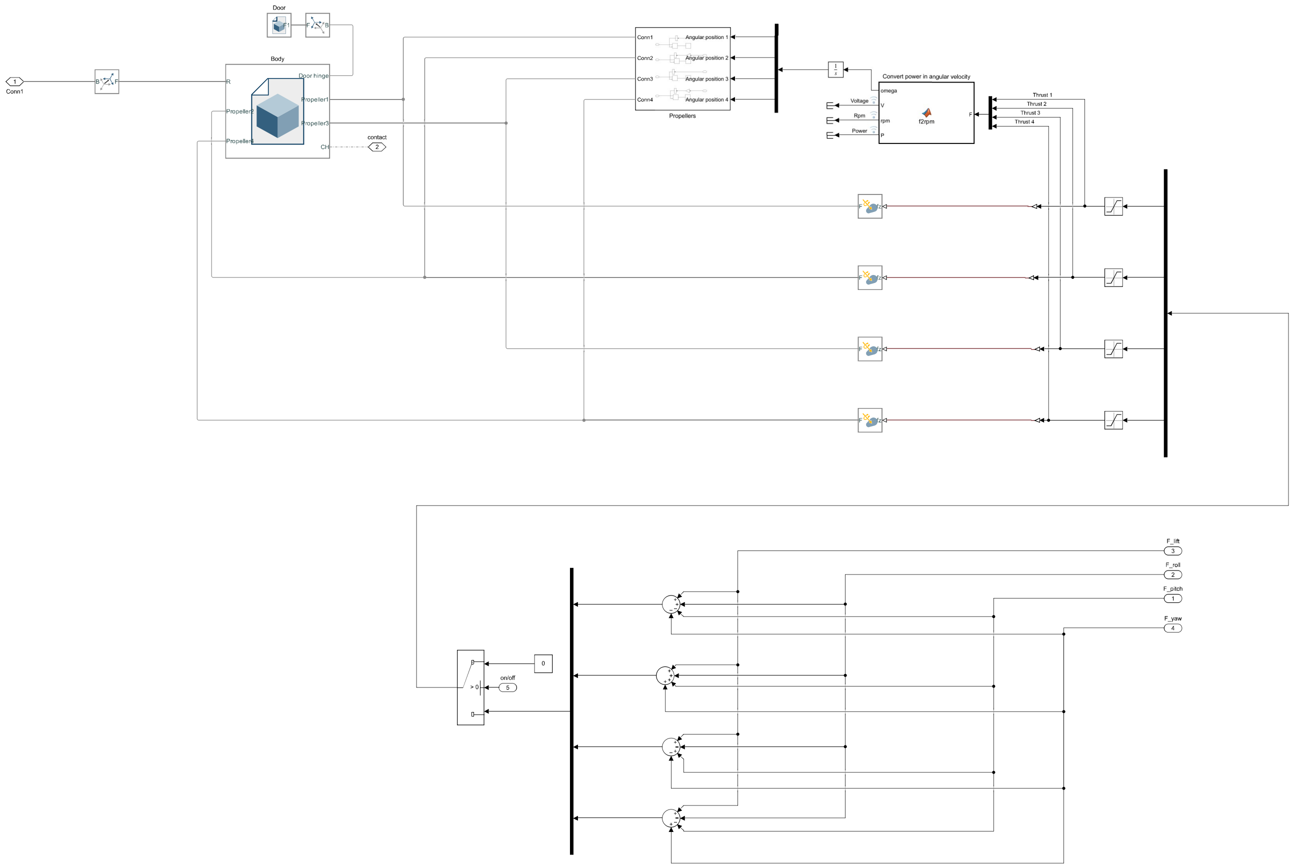


Figure 19. Drone subsystem.

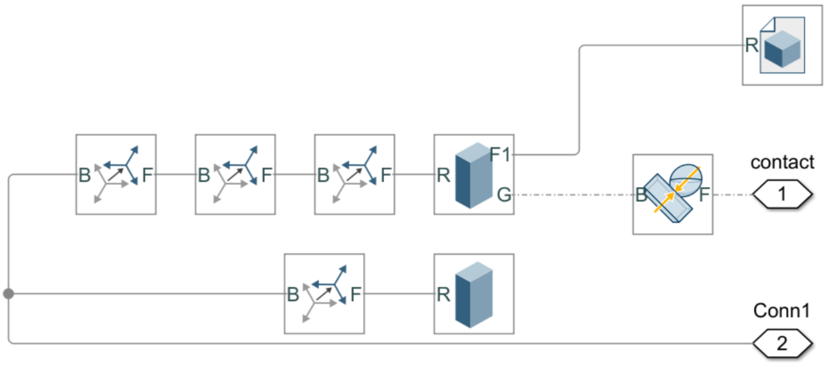


Figure 20. Landing base.

To dampen the residual kinetic energy accumulated during landing and due to the system inertia, a friction coefficient between the base and the landing gear 'Base2' was considered. The body geometry is recognized as solid due to the convex hull function, and a contact force is inserted between the two bodies to simulate a realistic collision. The friction model considered is of the Coulomb type with a static friction coefficient equal to $\mu_s = 0.5\ (-)$ and a dynamic friction coefficient equal to $\mu_d = 0.3\ (-)$. In SIMSCAPE

MULTIBODY, this friction model is referred to as the smooth stick-slip model.

To allow the engines to shut down on landing, an 'if clause' was inserted in the logic of the control system. Additionally, the condition for the propellers to stop rotating is that the height of the center of gravity relative to the global reference must be less than $h = 0.05$ (m).

6.2. Computer Model of the Beer Carrier

In this subsection, the fundamental elements necessary for performing a dynamical simulation of the unloading of the payload using SIMSCAPE MULTIBODY are described [84,85]. This operation takes place once the drone has landed. The SIMSCAPE MULTIBODY model of the beer carrier is represented in Figure 21.

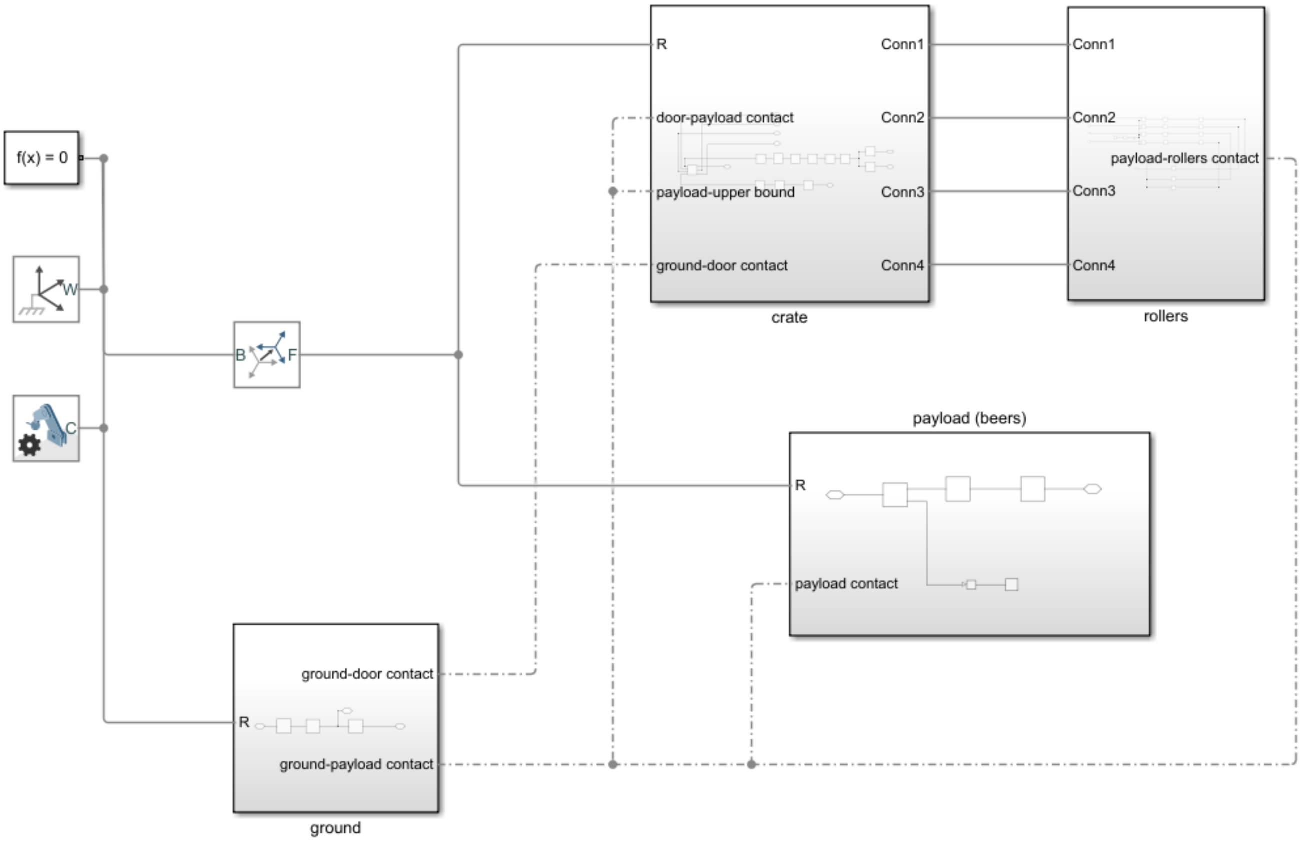


Figure 21. Beer Carrier SIMSCAPE MULTIBODY model.

The model that simulates the behavior of the payload consists of four subsystems. The first one, which is referred to as the external frame subsystem, is the outer case that contains the payload. The second one, which is referred to as the rollers subsystem, represents the rollers that allow its release. The third subsystem, which is referred to as the beer pack subsystem, models the six-pack beer. Finally, the fourth and last one, which is referred to as the ground subsystem, is responsible for representing the contact between the ground and the landing gear.

The structure of the box consists of a CAD model that schematizes its shape, developed using SOLIDWORKS. The hinge is connected to this external structure through a revolute joint, which allows only for one relative degree of freedom, and it is rigidly fixed to the door. In the SIMSCAPE MULTIBODY model, the revolute joint block simulates both the constraint and the internal mechanics of the hinge, which contains a torsional spring having an elastic stiffness coefficient given by $k_t = 0.043$ (N $\times$ m/rad). The data used for the physical parameters concerning the torsional spring refer to a product that is actually available in the market. The equilibrium configuration of the torsional spring ensures the proper automatic closure of the door immediately after the withdrawal of the beers. The interior of the container and the door are modeled as one-sided constraints. The contact with the parallelepiped representing the beers generates constraint reactions that are modeled through the 'Spatial Contact Forces' block available in SIMSCAPE MULTIBODY.

The rollers, connected to the case with appropriate revolute joints, are schematized as simple cylinders. The law of motion, provided in SIMSCAPE MULTIBODY as input via the 'Ramp' function, is a uniform rotary motion. The slope of the ramp represents the angular velocity of the rollers, expressed in revolutions per minute. The connections between the container and the rollers, which are in contact with the six-pack beer, are modeled in SIMSCAPE MULTIBODY as holonomic constraints. Additionally, the presence of friction is taken into account in the dynamical model to obtain realistic computer simulations. In fact, it is the friction between the rollers and the crate that allows for the motion of the payload.

A three-dimensional rigid body with six degrees of freedom is employed in SIMSCAPE MULTIBODY to represent the pack of beers. The model is a brick solid with dimensions corresponding to those of a real pack of beers. On the other hand, the ground is modeled in SIMSCAPE MULTIBODY through a brick solid with appropriate contact forces to simulate additional one-sided constraints.

7. Numerical Results and Discussion

7.1. Presentation of the Numerical Results

This subsection presents the results obtained from this research work. The goal was to create a virtual prototype of a UAV system with good functionality and features comparable to those on the market. The numerical results found in this work demonstrate the feasibility of the virtual prototype of the delivery drone designed in the present investigation. All the dynamical simulations mentioned in this section were carried out by employing the numerical values of the PID controller parameters reported in Table 6.

In the design and computer simulation phases, it was possible to evaluate the numerical results obtained to verify their consistency with the dynamical behavior of a real drone system having comparable characteristics and performance. From the design in the SOLIDWORKS environment, the main inertia matrix and the position of the center of mass of the entire UAV system were obtained. More specifically, the total mass of the delivery drone is equal to $m = 9.973$ (kg), whereas the total volume of the UAV system is given by $V = 6.278 \times 10^{-3}$ (m^3). Additionally, considering a local reference system placed in correspondence with the plane containing the lower frame of the UAV system, the local position vector of the center of mass of the delivery drone denoted with $\bar{\mathbf{u}}_G$ and its local inertia matrix denoted with $\bar{\mathbf{I}}_G$ are respectively given by:

$$\bar{\mathbf{u}}_G = \begin{bmatrix} \bar{x}_G \\ \bar{y}_G \\ \bar{z}_G \end{bmatrix} = \begin{bmatrix} 0.28 \times 10^{-3}\ (\mathrm{m}) \\ -17.28 \times 10^{-3}\ (\mathrm{m}) \\ 0.42 \times 10^{-3}\ (\mathrm{m}) \end{bmatrix} \tag{65}$$

and

$$\bar{\mathbf{I}}_G = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} 0.2321\ (\mathrm{kg} \cdot \mathrm{m}^2) & 0 & 0 \\ 0 & 0.2395\ (\mathrm{kg} \cdot \mathrm{m}^2) & 0 \\ 0 & 0 & 0.3663\ (\mathrm{kg} \cdot \mathrm{m}^2) \end{bmatrix} \tag{66}$$

The direction vectors of the principal axes of inertia are respectively denoted with $\bar{\mathbf{i}}_G$, $\bar{\mathbf{j}}_G$, and $\bar{\mathbf{k}}_G$. These vectors are given as follows:

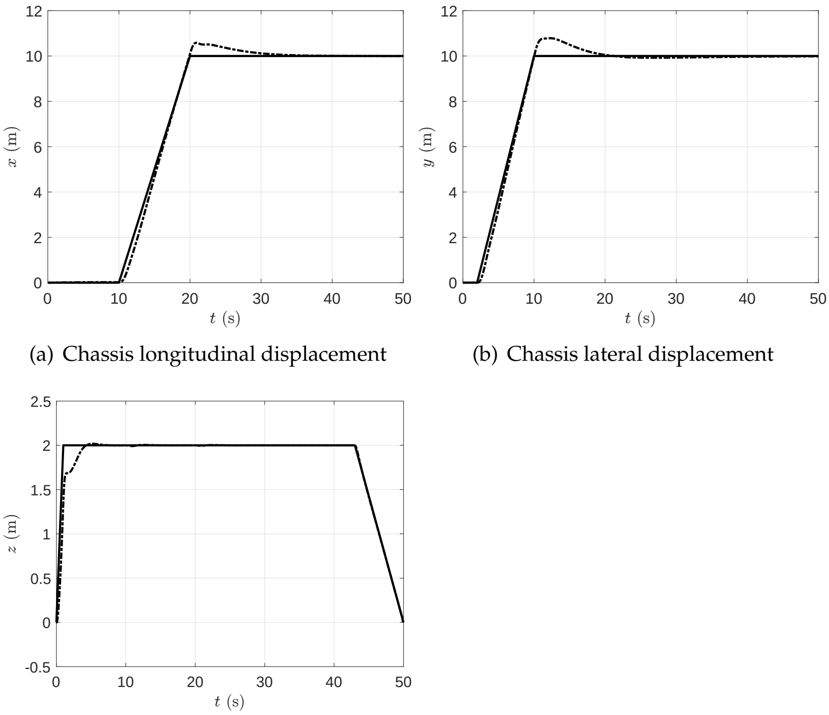
$$\bar{\mathbf{i}}_G = \begin{bmatrix} \bar{i}_{x,G} \\ \bar{i}_{y,G} \\ \bar{i}_{z,G} \end{bmatrix} = \begin{bmatrix} 1.00 \times 10^{-3}\ (\mathrm{m}) \\ 0.00 \times 10^{-3}\ (\mathrm{m}) \\ -0.05 \times 10^{-3}\ (\mathrm{m}) \end{bmatrix}, \quad \bar{\mathbf{j}}_G = \begin{bmatrix} \bar{j}_{x,G} \\ \bar{j}_{y,G} \\ \bar{j}_{z,G} \end{bmatrix} = \begin{bmatrix} -0.05 \times 10^{-3}\ (\mathrm{m}) \\ 0.00 \times 10^{-3}\ (\mathrm{m}) \\ -1.00 \times 10^{-3}\ (\mathrm{m}) \end{bmatrix} \tag{67}$$

and

$$\bar{\mathbf{k}}_G = \begin{bmatrix} \bar{k}_{x,G} \\ \bar{k}_{y,G} \\ \bar{k}_{z,G} \end{bmatrix} = \begin{bmatrix} 0.00 \times 10^{-3}\ (\mathrm{m}) \\ 1.00 \times 10^{-3}\ (\mathrm{m}) \\ 0.00 \times 10^{-3}\ (\mathrm{m}) \end{bmatrix} \tag{68}$$

Since the local position vector of the center of mass of the UAV system is close to the zero vector, the multibody formulation approach employed in Section 4 is mathematically justified by the actual geometric properties of the delivery drone.

In Figure 22, the time evolutions in the three-dimensional space of the UAV system translational coordinates obtained in MATLAB by simulating the SIMSCAPE MULTIBODY model are shown. In particular, Figure 22a,b is relative to the planar displacement of the delivery drone after the hovering maneuver, while Figure 22c represents the first phase of reaching vertical altitude.



(a) Chassis longitudinal displacement

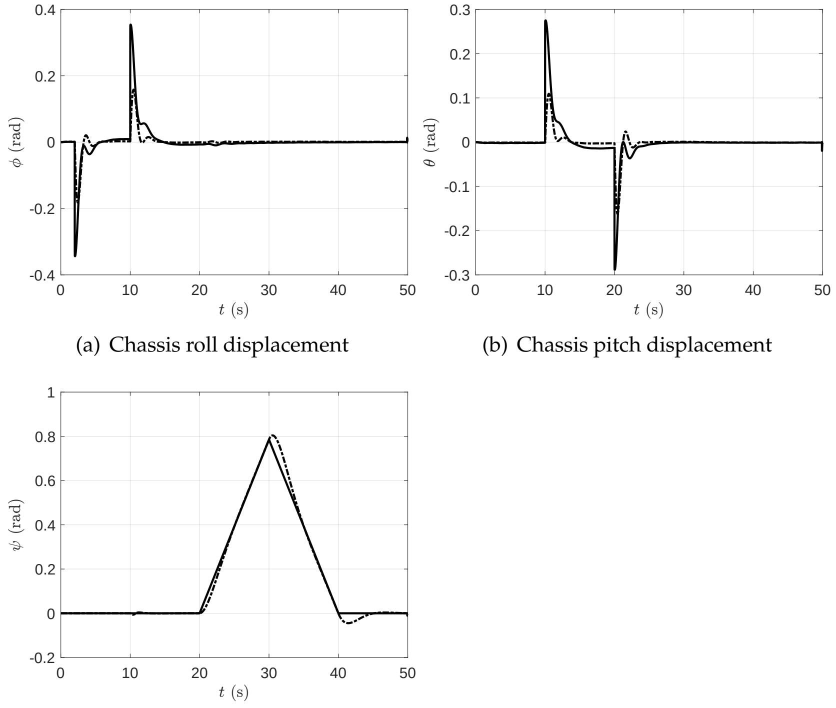
(b) Chassis lateral displacement

(c) Chassis vertical displacement

Figure 22. Longitudinal, lateral, and vertical (linear) displacements of the drone chassis. The solid lines (——) refer to the time-laws supplied in input from the MATLAB program. The dashed lines (- - -) refer to the measures detected by the sensors in the computer simulation, carried out using SIMSCAPE MULTIBODY.

In Figure 23, the time evolutions in the three-dimensional space of the UAV system rotational coordinates obtained in MATLAB by simulating the SIMSCAPE MULTIBODY model are shown. In particular, Figure 23a–c respectively represent the roll, pitch, and yaw angular displacements of the delivery drone.

In the MATLAB computational environment, the data relative to the number of revolutions of the single propellers, the values of the input voltage, and the power absorbed by the motors, obtained from the computer simulation performed using the SIMSCAPE MULTIBODY model, were plotted, thereby verifying the coherence with the tables reported by the producers of the components of actual drone systems found in the market. More specifically, the values of the physical quantities mentioned before are plotted as a function of time relative to the computer simulation. In Figure 24, the revolutions per minute of each propeller are represented as a function of time. To this end, Figure 24a–d respectively represent the revolutions per minute of the first, second, third, and fourth propellers.

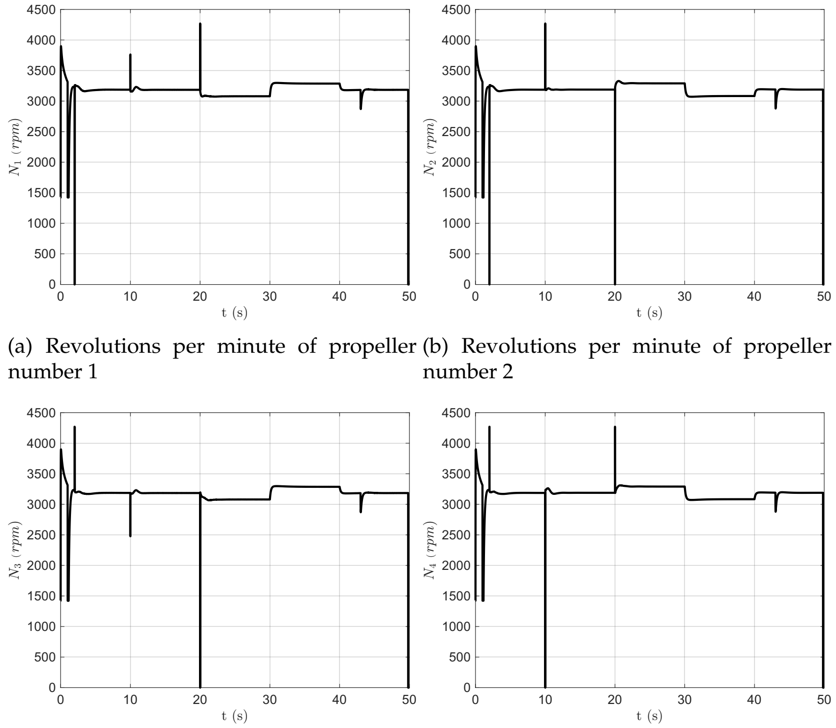


(a) Chassis roll displacement

(b) Chassis pitch displacement

(c) Chassis yaw displacement

Figure 23. Roll, pitch, and yaw (angular) displacements of the drone chassis. The solid lines (——) refer to the time-laws supplied in input from the MATLAB program. The dashed lines (- - -) refer to the measures detected by the sensors in the computer simulation, carried out using SIMSCAPE MULTIBODY.



(a) Revolutions per minute of propeller number 1

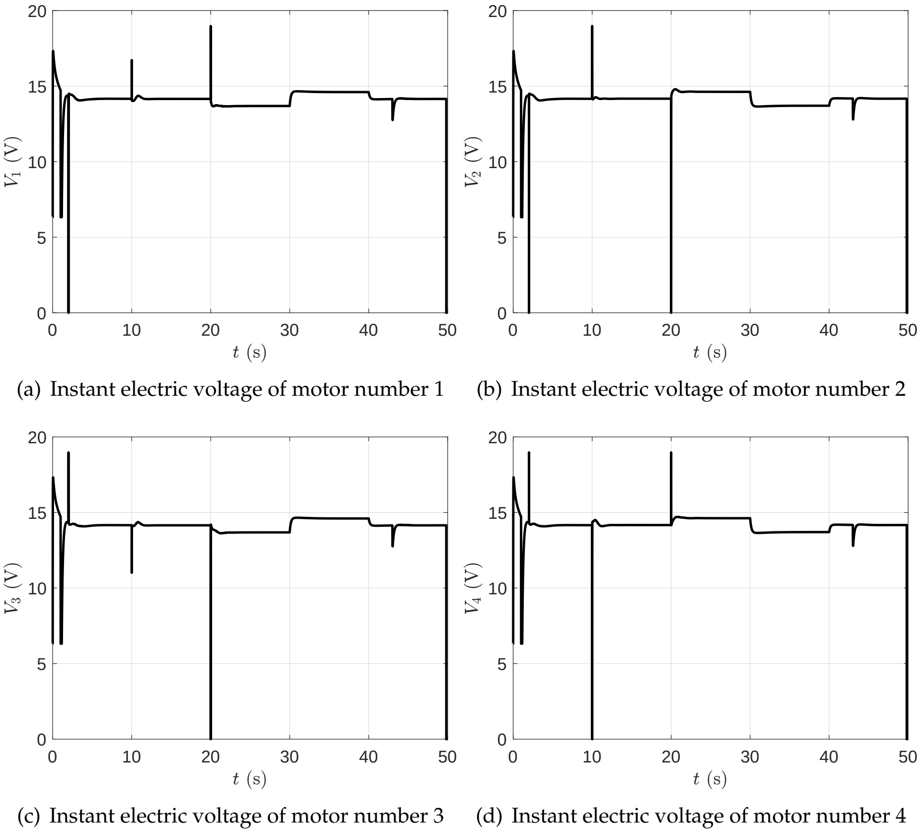
(b) Revolutions per minute of propeller number 2

(c) Revolutions per minute of propeller number 3

(d) Revolutions per minute of propeller number 4

Figure 24. Revolutions per minute of each propeller, plotted as a function of time.

In Figure 25, the instant voltage of each motor is represented as a function of time. To this end, Figure 25a–d respectively represents the voltage values of the first, second, third, and fourth electric motors.



(a) Instant electric voltage of motor number 1 (b) Instant electric voltage of motor number 2

(c) Instant electric voltage of motor number 3 (d) Instant electric voltage of motor number 4

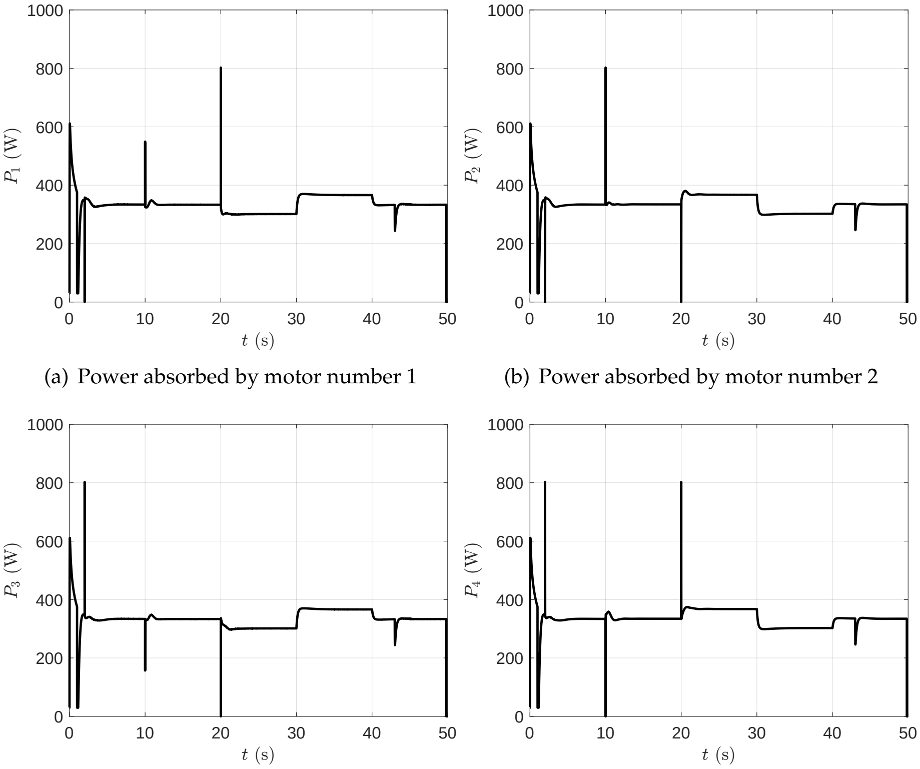
Figure 25. Instant electric voltage of each motor, plotted as a function of time.

In Figure 26, the power absorbed by each motor is represented as a function of time. To this end, Figure 26a–d respectively represent the power values of the first, second, third, and fourth electric motors.

Through the simulation performed in the SIMSCAPE MULTIBODY environment, it was then possible to evaluate the performance of the quadcopter designed in this investigation. A summary of the most relevant numerical results found is represented in Figure 27.

Figure 27a shows the curve that relates the variation of the revolutions per minute with the propulsive thrust generated by the propellers, expressed in (kgf). The values refer to the propulsive thrust generated by a single propeller. Another fundamental parameter to be observed is the torque variation as a function of the number of revolutions of a single propeller, as shown in Figure 27b. This parameter is useful for designing the mechanical transmission system. The last plot represented in Figure 27c indicates the power absorbed by each engine, which is useful for making evaluations regarding the propulsion system used.

In Figure 28, the time evolutions in the three-dimensional space of the beer crate representing the drone payload and obtained in MATLAB by simulating the SIMSCAPE MULTIBODY model are shown. In particular, Figure 28a–c respectively represent the longitudinal, vertical, and angular displacements of the beer crate that performs a planar motion.



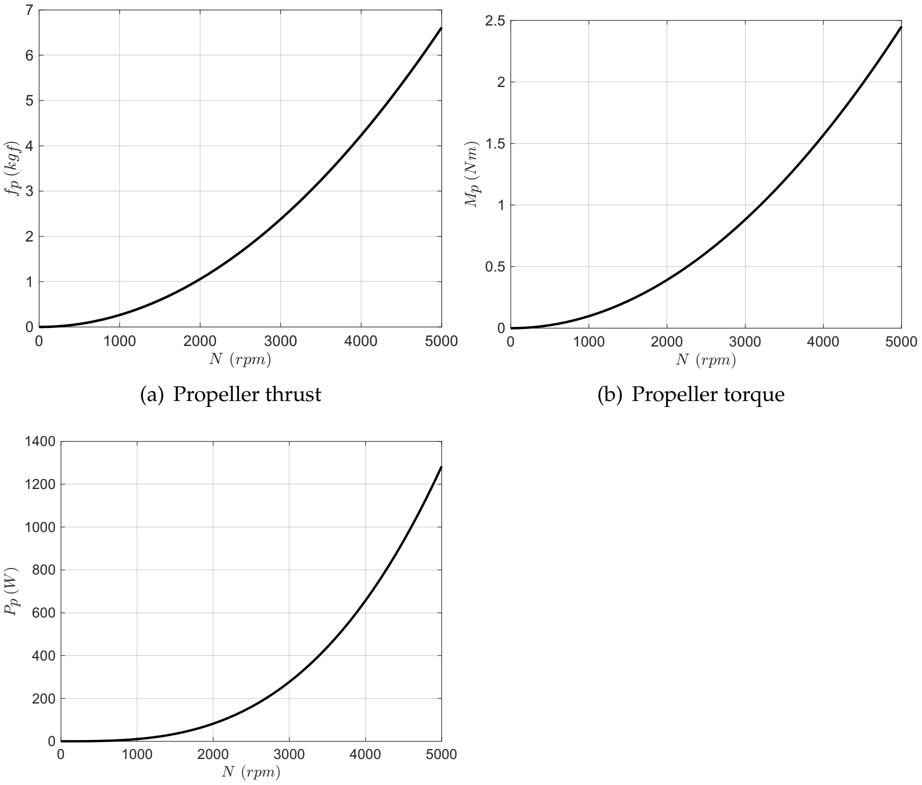
(a) Power absorbed by motor number 1

(b) Power absorbed by motor number 2

(c) Power absorbed by motor number 3

(d) Power absorbed by motor number 4

Figure 26. Power absorbed by each motor, plotted as a function of time.

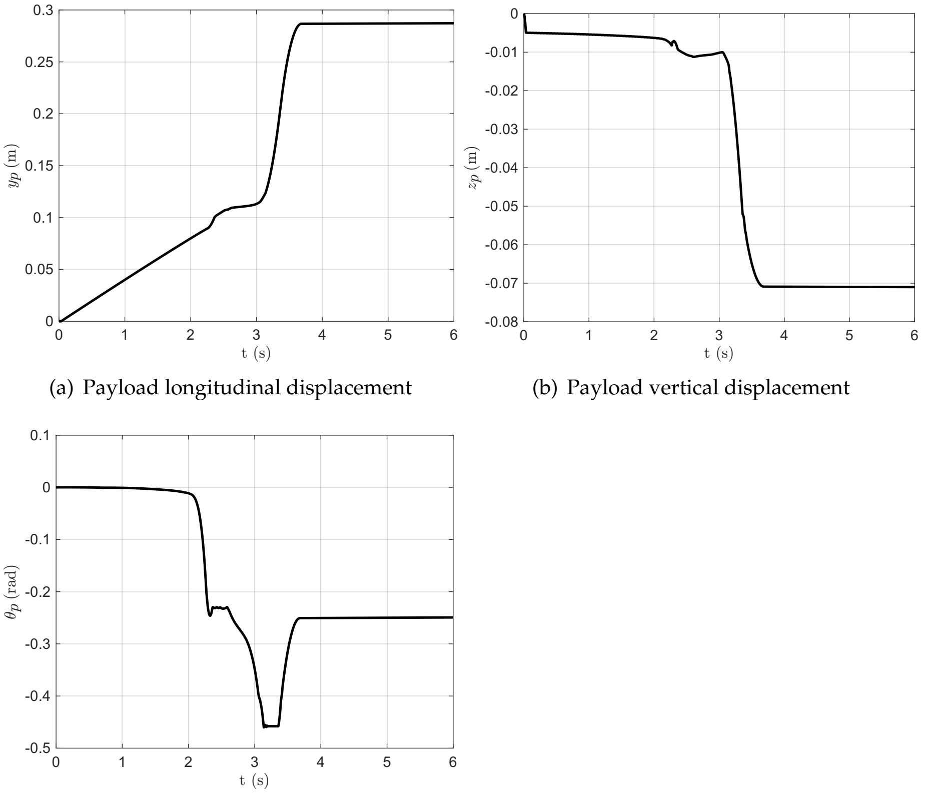


(a) Propeller thrust

(b) Propeller torque

(c) Propeller power

Figure 27. Thrust, torque, and power of each propeller, plotted as a function of time.



(a) Payload longitudinal displacement

(b) Payload vertical displacement

(c) Payload angular displacement

Figure 28. Longitudinal, vertical, and angular displacements of the beer crate representing the drone payload performing a planar motion obtained from the computer simulation, carried out using SIMSCAPE MULTIBODY.

7.2. Discussion of the Numerical Results

In this subsection, a brief discussion about the quality of the numerical results found in this investigation, and the challenges addressed in this study is provided. A performance index based on Root-Mean-Squared Error (RMSE) is introduced to quantitatively compare the trajectory designed in MATLAB, and is provided as input with the output data measured in the simulation carried out in SIMSCAPE MULTIBODY. The index is calculated with the equation given below:

$$RMSE = \sqrt{\frac{1}{N_d} \sum_{i=1}^{N} \left(\xi_i - \bar{\xi}_i\right)^2} \tag{69}$$

where N_d are the number of data resulting from the time discretization, ξ_i is the reference value, and $\bar{\xi}_i$ are the output data. The computed values of the RMSE between the reference (linear and angular) displacements and the resulting (linear and angular) displacements of the delivery drone found in the dynamical simulations are reported in Table 7.

Through the numerical results obtained using the SIMSCAPE MULTIBODY model and processed in the MATLAB computational environment, it was possible to verify the feasibility of the virtual prototype of the delivery drone designed in this work. During the design phase of the three-dimensional model, particular attention was paid to the position occupied by the center of mass of the UAV system to allow for the mathematical simplifications in the multibody approach described in Section 4. Through the results found using SOLIDWORKS and SIMSCAPE MULTIBODY, it was verified that the center of mass was in a central position, in line with the plane containing the four arms of the drone. This is necessary to ensure high stability during flight and to allow for smoother movements by the quadcopter. The results related to the mass of the UAV system were useful for designing the propulsion system, and for verifying the realization of components that are not directly available in the market. The data obtained from the multibody model indicating the power, the revolutions per minute, and the input voltages are consistent with the chosen propulsion system. The entire control system allows the drone to perform the maneuvers imposed by the time law with a high degree of robustness and accuracy. As demonstrated by the resulting dynamical behavior of the UAV system, optimal responses can be obtained, even with varying external disturbances. In conclusion, the numerical results found using MATLAB and SIMSCAPE MULTIBODY are completely satisfying for the goal of this investigation.

Table 7. Calculation of the RMSE for comparing desired trajectory with the output detected by the sensors in the computer simulation, carried out in SIMSCAPE MULTIBODY.

Symbol	Data (Units)
RMSE_x	0.2373 (m)
RMSE_y	0.2700 (m)
RMSE_z	0.0981 (m)
RMSE_ϕ	0.0334 (rad)
RMSE_θ	0.0277 (rad)
RMSE_ψ	0.0220 (rad)

In this work, the study of the realization of a virtual prototype of the UAV system serving as a delivery drone and its subsequent computer implementation required several simplifying assumptions. First, it must be considered that the drone is a multiple-inputs multiple-outputs (MIMO) system. In the context of the computer simulations, however, it was considered as a set of several single-input single-output (SISO) systems in order to divide and to control the problems of each movement (altitude, roll, pitch, and turn). Another issue related to the development of the control system is that the simulation is based on a simplification of the behaviors of brushless motors commonly used in actual drones. While brushed motors can be controlled by a simple potentiometer that adjusts the voltage, a brushless motor requires a specific controller that makes use of a Pulse-Width Modulation (PWM) system. In the dynamical simulations, the electric motors are treated as simply brushed motors whose angular velocity depends only on the input voltage. These simplifications are useful for the dynamical simulations, but they make impossible an eventual implementation on a real model without the opportune modifications. These important issues will be addressed and solved in future investigations.

To summarize the principal outcomes of this work, Table 8 contains the main performance parameters of the delivery drone, including hovering time, payload capacity, flight range, maximum altitude, the maximum angle of inclination, and the maximum forward velocity.

Table 8. Estimated performance data of the quadcopter designed in this study.

Parameters	Data	Units
Hovering time	19.3	min
Payload	9.9	kg
Flying range	12.1	km
Maximum takeoff altitude	5.3	km
Maximum tilt angle	58.9	°
Maximum forward velocity	20.7	m/s

The performance data reported in Table 8 were obtained through special-purpose software using an online simulator called Flight Evaluation, which estimated the main features of the delivery drone based on its specifications. This software is freely available at https://flyeval.com, accessed on 3 February 2023.

As far as the operational range of the delivery drone is concerned, for a standard UAV system, the Radio Controlled (RC) transmitter and receiver operate on a radio frequency of 2.4 (GHz), which offers several advantages, including a reduced interference with other transmitters, a lower power consumption, and a smaller occupancy volume. Typical RC transmitters can provide a range of up to 1000 (m). The short control wavelength allows for shorter transmit and receive antennas compared to other frequency ranges. However, a disadvantage of this frequency range is its susceptibility to obstacles, which can cause a low efficiency of the control signal. While a single channel allows for controlling only one specific operation, quadcopters similar to the one designed in this study require at least four channels for throttle, yaw, pitch, and roll control. Furthermore, additional channels may be needed for auxiliary operations, such as for opening the door of the payload.

8. Summary, Conclusions, and Future Developments

The purpose of this research work was to design and analyze a quadcopter serving as a delivery drone capable of performing a set of chosen operations. In the development of the UAV system considered as the case study of this investigation, it was chosen to design a system capable of transporting a pack of beers autonomously, thereby carrying an approximately two kilograms payload. To this end, the delivery drone is based on the classic design of a quadcopter, with the addition of a special box needed to transport the beverages representing the payload. The design of the structure of the delivery drone is based on the use of components that can be easily manufactured or that are readily found in the market. The main structure of the delivery drone is made of a carbon frame having four arms, a set of aluminum supports, and an ABS box. To simplify the final design, all these mechanical components are bolted using common structural screws. The central structure necessary to transport the load weighs about ten kilograms, a weight that is in line with commercial drones. The choices of propellers, motors, and batteries were based on a simplified propulsion model developed in this work. Additionally, the feasibility of the geometric model was verified through 3D CAD drawings.

A simplified multibody model was developed, starting from the CAD geometry, in order to be able to dynamically simulate the behavior of the proposed UAV system serving as a delivery drone. As for the design of the control system, it was chosen to simulate the behavior of the drone by using SIMSCAPE MULTIBODY, which is an additional software available in the MATLAB simulation environment. This computer program allows for accurately simulating, through appropriate blocks, the behaviors of multibody models subject to internal and external forcing actions. The implementation of the multibody model is carried out by importing simplified geometric models constrained by specific kinematic pairs, which represent the mechanical joints that connect the components of the mechanical system at hand. The virtual prototype of the delivery drone is subjected to external forces such as gravity and wind force. The motion control of the delivery drone is achieved through the use of a system of simple PID controllers that are arranged in a cascade architecture. Once a time law is set for all the segments of the trajectory to be followed by the UAV system, the behavior of the delivery drone is reproduced through numerical experiments. The trajectory tracking was found to be very robust, even in the presence of a variable forcing disturbance due to the wind. In addition to the flight simulation, it was also chosen to test the automated movement of the beers contained in the crate, thereby reproducing the delivery of the actual product representing the payload. In this final scenario, in virtue of a proper choice of the motor and by leveraging the computer simulations of the constraints, it was demonstrated through numerical experiments that the payload can be correctly unloaded in a way that is consistent with the predictions.

From an analytical standpoint, one of the most important contributions of this paper has been the development of a multibody model of the UAV system serving as a delivery drone. The mechanical model based on the multibody approach proposed in this investigation is quite general, and therefore, it can be used for modeling and simulating the dynamic behaviors of a large variety of quadcopters. More importantly, the paper proposes a novel method for the mathematical modeling of the imposed relative angular velocity between the propellers and the support arms of the UAV system. As shown in the manuscript, this important analytical result is achieved through the mathematical formulation of a proper set of algebraic constraints of nonholonomic nature, that is, a nonlinear system of algebraic equations depending both on the generalized coordinates and the generalized velocities of the quadcopter modeled as a multibody system. Standing the close relation between the thrust force and rotation rate of the propellers, the formulation of nonholonomic algebraic constraint equations mentioned before was also found to be fundamental for the motion control of the delivery drone.

In today's world, there is a significant push for the automation of logistics for industrial and civil applications. To the center of the reflectors, there is the employment of flying means for autonomous control. In recent years, some of the sectors that have seen exponential growth from this point of view are multimedia (aerial shots and filming), rescue, agriculture (both for monitoring and for the distribution of pesticides), geological surveys, and parcel delivery. The biggest challenges expected in the future will be those related to the noise of the devices and their coexistence. Among the most innovative solutions to this problem, there are hybrids between flying devices, and bipedal or wheeled robots. In this vein, this paper shows the workflow necessary to solve simple logistical problems through the construction of a flying device. The UAV system designed in the present work is adaptable to multiple situations, as it is based on a modular design. For example, one could eliminate the product container representing the payload, and replace it with different equipment, such as cameras, luggage racks, or rescue tools. In addition, it was shown how a simple control system could be implemented, to highlight the potential of modern software in the development and simulation of autonomous systems. In future work, the approach employed in this investigation will be used for the computer-aided design and the virtual prototyping of autonomous machines.

Author Contributions: This research paper was principally devised and developed by the first author (C.M.P.). Great support in the development of the paper was provided by the second, third, and fourth authors (M.D.G., E.B.O., and L.S.). The detailed review carried out by the fifth author (A.N.) considerably improved the quality of the work. The manuscript was written with the contributions of all authors. All authors discussed the results, reviewed the methodology, and approved the final version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wan, K.; Gao, X.; Hu, Z.; Wu, G. Robust motion control for UAV in dynamic uncertain environments using deep reinforcement learning. *Remote Sens.* **2020**, *12*, 640. [CrossRef]
2. Lopez-Rivera, M.; Cortes-Villada, A.; Giraldo, E. Optimal multivariable control design based on a fuzzy model for an unmanned aerial vehicle. *IAENG Int. J. Comput. Sci.* **2021**, *48*, 316–321.
3. Hu, B.; Wang, J. Deep learning based hand gesture recognition and UAV flight controls. *Int. J. Autom. Comput.* **2020**, *17*, 17–29. [CrossRef]
4. Zhang, Y.; Zhang, Y.; Yu, Z. Path following control for UAV using deep reinforcement learning approach. *Guid. Navig. Control* **2021**, *1*, 2150005. [CrossRef]
5. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]

6. Azam, M.A.; Mittelmann, H.D.; Ragi, S. UAV formation shape control via decentralized markov decision processes. *Algorithms* **2021**, *14*, 91. [CrossRef]
7. Bai, D.; Tian, J.; Li, D.; Li, S. Multiple UAVs Tracking for Moving Ground Target. *Eng. Lett.* **2022**, *30*, 829–834.
8. Belmonte, L.M.; Morales, R.; Fernández-Caballero, A. Computer vision in autonomous unmanned aerial vehicles—A systematic mapping study. *Appl. Sci.* **2019**, *9*, 3196. [CrossRef]
9. Hu, J.; Zhang, H.; Li, Z.; Zhao, C.; Xu, Z.; Pan, Q. Object traversing by monocular UAV in outdoor environment. *Asian J. Control* **2021**, *23*, 2766–2775. [CrossRef]
10. Cifuentes-Molano, M.F.; Hernandez, B.S.; Giraldo, E. Comparison of Different Control Techniques on a Bipedal Robot of 6 Degrees of Freedom. *IAENG Int. J. Appl. Math.* **2021**, *51*, 1–7.
11. Gu, S.; Chen, J.; Tian, Q. An implicit asynchronous variational integrator for flexible multibody dynamics. *Comput. Methods Appl. Mech. Eng.* **2022**, *401*, 115660. [CrossRef]
12. Wang, K.; Tian, Q. A nonsmooth method for spatial frictional contact dynamics of flexible multibody systems with large deformation. *Int. J. Numer. Methods Eng.* **2023**, *124*, 752–779. [CrossRef]
13. Shabana, A.A. Integration of computer-aided design and analysis: Application to multibody vehicle systems. *Int. J. Veh. Perform.* **2019**, *5*, 300–327. [CrossRef]
14. Citarella, R.; Armentani, E.; Caputo, F.; Naddeo, A. FEM and BEM analysis of a human mandible with added temporomandibular joints. *Open Mech. Eng. J.* **2012**, *6*, 100–114. [CrossRef]
15. Cappetti, N.; Naddeo, A.; Naddeo, F.; Solitro, G. Finite elements/Taguchi method based procedure for the identification of the geometrical parameters significantly affecting the biomechanical behavior of a lumbar disc. *Comput. Methods Biomech. Biomed. Eng.* **2016**, *19*, 1278–1285. [CrossRef] [PubMed]
16. Muscat, M.; Cammarata, A.; Maddio, P.D.; Sinatra, R. Design and development of a towfish to monitor marine pollution. *Euro-Mediterr. J. Environ. Integr.* **2018**, *3*, 1–12. [CrossRef]
17. Tanev, T.; Cammarata, A.; Marano, D.; Sinatra, R. Elastostatic model of a new hybrid minimally-invasive-surgery robot. In Proceedings of the 14th IFToMM World Congress, Taipei, Taiwan, 25–30 October 2015.
18. De Simone, M.C.; Celenta, G.; Rivera, Z.B.; Guida, D. Mechanism Design for a Low-Cost Automatic Breathing Applications for Developing Countries. In Proceedings of the International Conference "New Technologies, Development and Applications", Sarajevo, Bosnia and Herzegovina, 23–25 June 2022; pp. 345–352.
19. Mangoni, D.; Soldati, A. Model-based simulation of dynamic behaviour of electric powertrains and their limitation induced by battery current saturation. *Int. J. Veh. Perform.* **2021**, *7*, 156–169. [CrossRef]
20. Cammarata, A.; Lacagnina, M.; Sinatra, R. Closed-form solutions for the inverse kinematics of the Agile Eye with constraint errors on the revolute joint axes. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 317–322.
21. Tasora, A.; Mangoni, D.; Benatti, S.; Garziera, R. Solving variational inequalities and cone complementarity problems in nonsmooth dynamics using the alternating direction method of multipliers. *Int. J. Numer. Methods Eng.* **2021**, *122*, 4093–4113. [CrossRef]
22. Huang, Z.; Xi, F.; Huang, T.; Dai, J.S.; Sinatra, R. Lower-mobility parallel robots: Theory and applications. *Adv. Mech. Eng.* **2010**, *2*, 927930. [CrossRef]
23. Cammarata, A.; Sinatra, R.; Rigano, A.; Lombardo, M.; Maddio, P.D. Design of a large deployable reflector opening system. *Machines* **2020**, *8*, 7. [CrossRef]
24. Kant, R.; Nigam, V.K.; Kumar, R. Design and Analysis of Unmanned Aerial Vehicle (Uav) Using Solidworks 2016 Edition. *Int. Res. J. Eng. Technol* **2019**, *6*, 5132–5138.
25. Ballous, K.A.; Khalifa, A.N.; Abdulwadood, A.; Al-Shabi, M.; Assad, M.E.H. Medical kit: Emergency drone. In Proceedings of the Unmanned Systems Technology XXII. International Society for Optics and Photonics, Online Only, 27 April–9 May 2020; Volume 11425, p. 114250V.
26. Zhafri, Z.; Effendi, M.; Rosli, M.F. Optimization of assembly process and environmental impact using DFMA and sustainable design analysis: Case study of drone. *Aip Conf. Proc.* **2018**, *2030*, 020074.
27. Meenakshipriya, B.; Raja, N.B.; Faroshkhan, S.; Matris, S. Design and fabrication of 3D printed QuadDrone for altitude measurement. *Int. J. Aerosp. Syst. Sci. Eng.* **2021**, *1*, 85–94. [CrossRef]
28. Benito, J.A.; Glez-de Rivera, G.; Garrido, J.; Ponticelli, R. Design considerations of a small UAV platform carrying medium payloads. In Proceedings of the Design of Circuits and Integrated Systems, Madrid, Spain, 26–28 November 2014; pp. 1–6.
29. Tnunay, H.; Abdurrohman, M.Q.; Nugroho, Y.; Inovan, R.; Cahyadi, A.; Yamamoto, Y. Auto-tuning quadcopter using Loop Shaping. In Proceedings of the 2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Jakarta, Indonesia, 19–21 November 2013; pp. 111–115.
30. De Simone, M.C.; Russo, S.; Rivera, Z.B.; Guida, D. Multibody model of a UAV in presence of wind fields. In Proceedings of the 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), Prague, Czech Republic, 20–22 May 2017; pp. 83–88.
31. Hai-long, F.Q.d.P. The application of cascade PID control in UAV attitude control. *Microcomput. Inf.* **2009**, *25*, 9–10.
32. Amir, M.Y.; Abbass, V. Modeling of quadrotor helicopter dynamics. In Proceedings of the 2008 International Conference on Smart Manufacturing Application, Goyangi, Republic of Korea, 9–11 April 2008; pp. 100–105.

33. Razinkova, A.; Gaponov, I.; Cho, H.C. Adaptive control over quadcopter UAV under disturbances. In Proceedings of the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Gyeonggi-do, Republic of Korea, 22–25 October 2014; pp. 386–390.
34. Fernando, H.; De Silva, A.; De Zoysa, M.; Dilshan, K.; Munasinghe, S. Modelling, simulation and implementation of a quadrotor UAV. In Proceedings of the 2013 IEEE 8th International Conference on Industrial and Information Systems, Peradeniya, Sri Lanka, 17–20 December 2013; pp. 207–212.
35. Pessen, D.W. A new look at PID-controller tuning. *J. Dyn. Sys. Meas. Control* **1994**, *116*, 553–557. [CrossRef]
36. Azeemi, N.Z. Cooperative Trajectory and Launch Power Optimization of UAV Deployed in Cross-Platform Battlefields. *Int. Assoc. Eng. Eng. Lett.* **2021**, *29*, 57–68.
37. De Simone, M.C.; Guida, D. Control design for an under-actuated UAV model. *FME Trans.* **2018**, *46*, 443–452.
38. Du, H.; Wang, W.; Xu, C.; Xiao, R.; Sun, C. Real-time onboard 3D state estimation of an unmanned aerial vehicle in multi-environments using multi-sensor data fusion. *Sensors* **2020**, *20*, 919. [CrossRef] [PubMed]
39. Malgaca, L.; Lök, Ş.İ. Measurement and modeling of a flexible manipulator for vibration control using five-segment S-curve motion. *Trans. Inst. Meas. Control* **2022**, *44*, 1545–1556. [CrossRef]
40. Guo, A.; Zhou, Z.; Zhu, X.; Bai, F. Low-cost sensors state estimation algorithm for a small hand-launched Solar-powered UAV. *Sensors* **2019**, *19*, 4627. [CrossRef] [PubMed]
41. Pan, Y.; Dai, W.; Xiong, Y.; Xiang, S.; Mikkola, A. Tree-topology-oriented modeling for the real-time simulation of sedan vehicle dynamics using independent coordinates and the rod-removal technique. *Mech. Mach. Theory* **2020**, *143*, 103626. [CrossRef]
42. Villecco, F.; Pellegrino, A. Entropic measure of epistemic uncertainties in multibody system models by axiomatic design. *Entropy* **2017**, *19*, 291. [CrossRef]
43. Villecco, F. On the evaluation of errors in the virtual design of mechanical systems. *Machines* **2018**, *6*, 36. [CrossRef]
44. De Simone, M.C.; Veneziano, S.; Guida, D. Design of a Non-Back-Drivable Screw Jack Mechanism for the Hitch Lifting Arms of Electric-Powered Tractors. *Actuators* **2022**, *11*, 358. [CrossRef]
45. Kaiser, I.; Strano, S.; Terzo, M.; Tordela, C. Anti-yaw damping monitoring of railway secondary suspension through a nonlinear constrained approach integrated with a randomly variable wheel-rail interaction. *Mech. Syst. Signal Process.* **2021**, *146*, 107040. [CrossRef]
46. Bettega, J.; Richiedei, D. Trajectory tracking in an underactuated, non-minimum phase two-link multibody system through model predictive control with embedded reference dynamics. *Mech. Mach. Theory* **2023**, *180*, 105165. [CrossRef]
47. Kaiser, I.; Strano, S.; Terzo, M.; Tordela, C. Estimation of the railway equivalent conicity under different contact adhesion levels and with no wheelset sensorization. *Veh. Syst. Dyn.* **2023**, *61*, 19–37. [CrossRef]
48. Quan, Q. *Introduction to Multicopter Design and Control*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 150–160.
49. Torenbeek, E.; Wittenberg, H. *Flight Physics: Essentials of Aeronautical Disciplines and Technology, with Historical Notes*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
50. Quan, Q.; Dai, X.; Wang, S. *Multicopter Design and Control Practice: A Series Experiments Based on MATLAB and Pixhawk*; Springer Nature: Berlin/Heidelberg, Germany, 2020.
51. Hassani, H.; Mansouri, A.; Ahaitouf, A. Mechanical modeling, control and simulation of a quadrotor UAV. In Proceedings of the International Conference on Electronic Engineering and Renewable Energy, Saidia, Morocco, 13–15 April 2020; pp. 441–449.
52. Ononiwu, G.; Onojo, O.; Ozioko, O.; Nosiri, O. Quadcopter design for payload delivery. *J. Comput. Commun.* **2016**, *4*, 1–12. [CrossRef]
53. Patel, K.; Barve, J. Modeling, simulation and control study for the quad-copter UAV. In Proceedings of the 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, India, 15–17 December 2014; pp. 1–6.
54. Luukkonen, T. Modelling and control of quadcopter. *Indep. Res. Proj. Appl. Math. Espoo* **2011**, *22*, 22.
55. Bai, Q.; Shehata, M.; Nada, A. Review study of using Euler angles and Euler parameters in multibody modeling of spatial holonomic and non-holonomic systems. *Int. J. Dyn. Control* **2022**, *10*, 1707–1725. [CrossRef]
56. Pappalardo, C.M.; Guida, D. On the Lagrange multipliers of the intrinsic constraint equations of rigid multibody mechanical systems. *Arch. Appl. Mech.* **2018**, *88*, 419–451. [CrossRef]
57. Guida, R.; De Simone, M.; Dašić, P.; Guida, D. Modeling techniques for kinematic analysis of a six-axis robotic arm. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Oradea, Romania, 30–31 May 2019; Volume 568, p. 012115.
58. Yilmaz, Z.; Yilmaz, O.; Bingul, Z. Design, analysis and simulation of a 6-DOF serial manipulator. *Kocaeli J. Sci. Eng.* **2020**, *3*, 9–15. [CrossRef]
59. Shabana, A.A. *Dynamics of Multibody Systems*; Cambridge University Press: Cambridge, UK, 2003.
60. Shabana, A.A. *Computational Dynamics*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
61. Pappalardo, C.M.; Guida, D. A comparative study of the principal methods for the analytical formulation and the numerical solution of the equations of motion of rigid multibody systems. *Arch. Appl. Mech.* **2018**, *88*, 2153–2177. [CrossRef]
62. Blajer, W. Methods for constraint violation suppression in the numerical simulation of constrained multibody systems–A comparative study. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1568–1576. [CrossRef]
63. Flores, P.; Machado, M.; Seabra, E.; Tavares da Silva, M. A parametric study on the Baumgarte stabilization method for forward dynamics of constrained multibody systems. *J. Comput. Nonlinear Dyn.* **2011**, *6*, 011019. [CrossRef]

64. Pappalardo, C.M.; Guida, D. Dynamic analysis of planar rigid multibody systems modeled using natural absolute coordinates. *Appl. Comput. Mech.* **2018**, *12*, 73–110. [CrossRef]
65. Marques, F.; Souto, A.P.; Flores, P. On the constraints violation in forward dynamics of multibody systems. *Multibody Syst. Dyn.* **2017**, *39*, 385–419. [CrossRef]
66. Flores, P.; Nikravesh, P.E. Comparison of different methods to control constraints violation in forward multibody dynamics. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013; Volume 55966, p. V07AT10A028.
67. Seifried, R. *Dynamics of Underactuated Multibody Systems*; Springer: Berlin/Heidelberg, Germany, 2014.
68. Guida, D.; Pappalardo, C.M. Forward and inverse dynamics of nonholonomic mechanical systems. *Meccanica* **2014**, *49*, 1547–1559. [CrossRef]
69. Fantoni, I.; Lozano, R.; Lozano, R. *Non-Linear Control for Underactuated Mechanical Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2002.
70. Pappalardo, C.M.; Guida, D. On the dynamics and control of underactuated nonholonomic mechanical systems and applications to mobile robots. *Arch. Appl. Mech.* **2019**, *89*, 669–698. [CrossRef]
71. Seifried, R. Two approaches for feedforward control and optimal design of underactuated multibody systems. *Multibody Syst. Dyn.* **2012**, *27*, 75–93. [CrossRef]
72. Offermann, A.; Castillo, P.; De Miras, J. Nonlinear model and control validation of a tilting quadcopter. In Proceedings of the 2020 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphaël, France, 15–18 September 2020; pp. 50–55.
73. Abdelhay, S.; Zakriti, A. Modeling of a quadcopter trajectory tracking system using PID controller. *Procedia Manuf.* **2019**, *32*, 564–571. [CrossRef]
74. Almaged, M.; Khather, S.I.; Abdulla, A.I. Design of a discrete PID controller based on identification data for a simscape buck boost converter model. *Int. J. Power Electron. Drive Syst.* **2019**, *10*, 1797. [CrossRef]
75. Azar, A.T.; Ammar, H.H.; Ibrahim, Z.F.; Ibrahim, H.A.; Mohamed, N.A.; Taha, M.A. Implementation of PID controller with PSO tuning for autonomous vehicle. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 26–28 October 2019; pp. 288–299.
76. Abdalla, M.; Al-Baradie, S. Real Time Optimal Tuning of Quadcopter Attitude Controller Using Particle Swarm Optimization. *J. Eng. Technol. Sci* **2020**, *52*, 745–764. [CrossRef]
77. Qian, L.; Liu, H.H. Path-following control of a quadrotor UAV with a cable-suspended payload under wind disturbances. *IEEE Trans. Ind. Electron.* **2019**, *67*, 2021–2029. [CrossRef]
78. Kose, O.; Oktay, T. Dynamic modeling and simulation of quadrotor for different flight conditions. *Eur. J. Sci. Technol.* **2019**, *15*, 132–142.
79. Mohapatra, S.; Srivastava, R.; Khera, R. Implementation of a two wheel self-balanced robot using MATLAB Simscape Multibody. In Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019; pp. 1–3.
80. Usman, M. Quadcopter Modelling and Control With MATLAB/Simulink Implementation. Bachelor's Thesis, LAB University of Applied Sciences, Lahti, Finland, 2020.
81. Vamsi, D.S.; Tanoj, T.S.; Krishna, U.M.; Nithya, M. Performance Analysis of PID controller for Path Planning of a Quadcopter. In Proceedings of the 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC), Chennai, India, 21–23 August 2019; pp. 116–121.
82. Zouaoui, S.; Mohamed, E.; Kouider, B. Easy tracking of UAV using PID controller. *Period. Polytech. Transp. Eng.* **2019**, *47*, 171–177. [CrossRef]
83. Zeng, Y.; Zhang, R. Energy-efficient UAV communication with trajectory optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [CrossRef]
84. Budnyaev, V.A.; Filippov, I.F.; Vertegel, V.V.; Dudnikov, S.Y. Simulink-based Quadcopter Control System Model. In Proceedings of the 2020 1st International Conference Problems of Informatics, Electronics, and Radio Engineering (PIERE), Novosibirsk, Russia, 10–11 December 2020; pp. 246–250.
85. Ji, J.; Han, J.-N.; Meng, L.-F. Modeling and simulation of quadrotor UAV based on Simscape. *J. Meas. Sci. Instrum.* **2020**, *11*, 169–176.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.