A PX4 Integrated Framework for Modeling and Controlling Multicopters with Tiltable Rotors

Salvatore Marcellini, Jonathan Cacace and Vincenzo Lippiello

***Abstract*— This paper presents a general control framework for multicopters equipped with tiltable rotors (tilting multicopters). Differently from classical flat multicopters, tilting multicopters can be fully actuated systems able to decouple position and attitude control. The proposed framework has been transparently integrated into the widely used PX4 control stack, an open-source controller for ground and aerial systems, to fully exploit its high-level interfaces and functionalities and, at the same time, simplify the creation of new devices with tilting propellers. Simulation tools have been also added to the PX4 simulation framework, based on its Software-In-The-Loop (SITL) system and a set of simulated experiments in a dynamic robotic simulator have been carried out to demonstrate the effectiveness of this system. Moreover, to demonstrate the usability of the proposed framework, initial experiments with a real platform have been carried out. The proposed control framework is accessible at the following link: `https://github.com/prisma-lab/PX4_tilting_multicopters`**

I. INTRODUCTION

Over the last decades, the usage of Unmanned Aerial Vehicle (UAV) systems has massively grown. This growth is motivated by their maneuverability and massive modularity for the onboard sensing equipment (i.e. cameras, LIDARs, thermal cameras, etc...). Thanks to the improvements in the hardware devices and the strong aerial robotics community that provides new efficient control techniques, UAVs can nowadays be employed in industrial applications like industrial plants inspection [1]–[4], surveillance [5], [6], remote sensing and similar.

In recent years, aerial robots have been used also in that domains requiring interaction with the environment (see [7]). In this context, these robots take the name of Unmanned Aerial Manipulators (UAMs). The ability of a UAMs to reach high-altitude locations along with their capacity to stay in contact with the environment makes them attractive systems for industrial companies to perform inspection and maintenance operations. In this domain, one of the most efficient flight configurations for multicopters is represented by a special class of UAVs called tilting multicopter (see Fig. 1), where the motion of their floating base is obtained by modifying the orientation of its propellers [8]. The actuation capability of such platforms has been proven to be very suitable to enable safe aerial-physical interaction, obtaining better performance in flight and high disturbance rejection [9], [10]. Tiltrotors can be equipped with actively tiltable rotors (*tiltable* or *tilting*) and fixed tilted rotors (*tilted*). Tiltable drones [11] consist of a rigid base frame and multiple propellers attached to that with actuated servomotors. In this way, the propellers can be oriented towards different directions, enabling a fully actuated system. Differently, tilted drones [12] are equipped with multiple propellers oriented toward different directions and fixed to the robot's base frame. The proposed framework is focused on the modeling and control of tiltable multi-copter. Such devices are nowadays employed in more and more tasks [13]–[15]. However, the current platforms rely on custom hardware solutions and flight controllers also when they are implemented on open-source autopilots, like the well-known PixHawk board [16].

Fig. 1: Fully actuated tiltable coaxial octocopter

Among the different solutions present in the literature, PX4[1] [17] is one of the leading open-source autopilots for drones devoted to controlling a wide range of unmanned vehicles, like UAVs, ground rovers and underwater robots. Close to our domain, various rotary-wing platforms are supported by the PX4 control stack, namely single, tri, quad, hexa, octa copters. Vertical TakeOff and Landing (VToL) aircrafts [18] are supported as well. These systems consist of two different configurations: one to achieve vertical motion (during the takeoff and land) and one to plane during the flight. However, no tilting multirotor systems are currently supported[2]. Other control frameworks for autonomous aerial and ground mobile robots are the iNAV [19] and Arducopter [20] control stacks. The performance of these firmware has been compared in [12]. Despite the performance and supported functionalities of such autopilots being comparable, PX4 is becoming the leading research platform

The research leading to these results has been supported by the AERIAL-CORE project (Horizon 2020 Grant Agreement No. 871479). The authors are solely responsible for its content.

The authors are with the PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, Naples, 80125, Italy. Corresponding author's e-mail: `salvatore.marcellini@unina.it`.

[1] `https://px4.io/`

[2] At the time this paper is being written, the latest stable release of PX4 is the v1.13

for drones, from consumers to industrial applications. It has one of the biggest open-source communities and collaborates with several hardware manufacturers to assure compatibility between the software and the hardware, simplifying the development and the usage for the users.

In this context, the main contribution of this work concerns the development of a set of modules integrated into the PX4 flight stack to set up and control generic tilting multicopters. With this tool, developers can add new tilting rotor airframes without losing the high-level functionalities of the PX4 stack, like the interface with the remote controller and the ground control station, the safety checks and the interfaces with all the compatible external sensors and actuators (i.e. the landing gear, the GPS antennas, etc...). Moreover, simulation models of tiltable drones have been added to the Software-In-The-Loop (SITL) simulation environment of PX4. In particular, these models have been imported in the Gazebo simulator [21] to test the control framework in a physical, realistic environment.

We tested the proposed framework with two kinds of platforms: an H-shaped drone with one-direction tiltable rotors and a fully-actuated tiltable coaxial octarotor. The effectiveness of the proposed control system has been tested in simulation, in which a set of trajectories have been tracked. In addition, initial tests on a real platform have been carried out and here discussed.

The rest of the article is structured as follows. In Section II the motivations of our work are reported. In Section III a brief description of the tiltrotors considered in the evaluation case studies is reported. In Section IV the overall system architecture is explained. In Section V the user interface used to configure the tilting UAV is described. In Section VI a set of simulation case studies demonstrate the framework's effectiveness. Finally, in Section VII the conclusions of this work are reported.

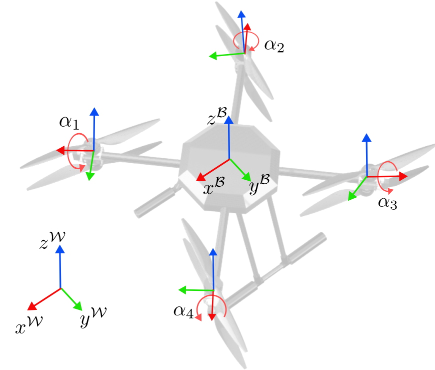


Fig. 2: Fully actuated tiltable coaxial octocopter

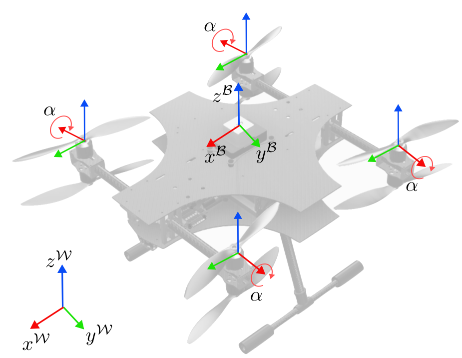


Fig. 3: H-shaped one-tilt coaxial octocopter

II. MOTIVATIONS

Conventional multicopters have been extensively used in recent years for a bunch of applications. These vehicles have six Degrees of Freedoms (DoFs) (i.e. three translations and three rotations) with four control inputs, making this system under-actuated. In particular, for flat multicopters, the two DoFs representing the rotations of the body frame around the x and y axes are coupled with the one of the position. As consequence, the orientation of the platform changes accordingly with the change of its position, causing disturbances to eventual sensors attached to the floating base, like cameras, LIDARs and similar. Differently, tilting multicopters have additional DoF thanks to the possibility to change the orientation of their propellers. In this way, considering the platform's total number of DoF, position-tracking tasks can be achieved without modifying the orientation of the floating base.

One of the first works concerning the modeling and control of a tiltable drone is presented in [22], where the dynamic model of a tiltable drone has been derived and its controller has been implemented and tested using numerical simulations, while in [23] the same model was experimentally evaluated with real-world tests. However, the drone used to validate the control approach was a pure research platform. Moreover, since that time different other tiltable or tilted drones have developed to reach industrial standards. Some examples of these devices are the APPELLIX [24], Texo Drone Survey Inspection platform [25], Ronik Inspectioneering UT device [26] and Voliro inspection drone [27] [28]. Despite the wide number of tiltable platforms that can be found in the literature, a common, open-source and configurable autopilot for such systems is not present yet.

III. TILTABLE DRONE CONFIGURATIONS

Even though the proposed framework is general and permits configuring any kind of tiltrotor, in this work we consider two types of tiltable drone configurations: the H-shape one-tilt and the fully actuated. The first one takes the form of a traditional H-shape multirotor, where the rotors are placed at the corners of the arms that form the shape of an H. The difference with the standard one consists of one or more servomotors, added to rotate the rotor-propeller groups around the arm's transversal axis (Fig. 3). Depending on the front axis orientation relative to the H shape, the drone is capable of moving forward and backward without changing its pitch angle, or left and right without

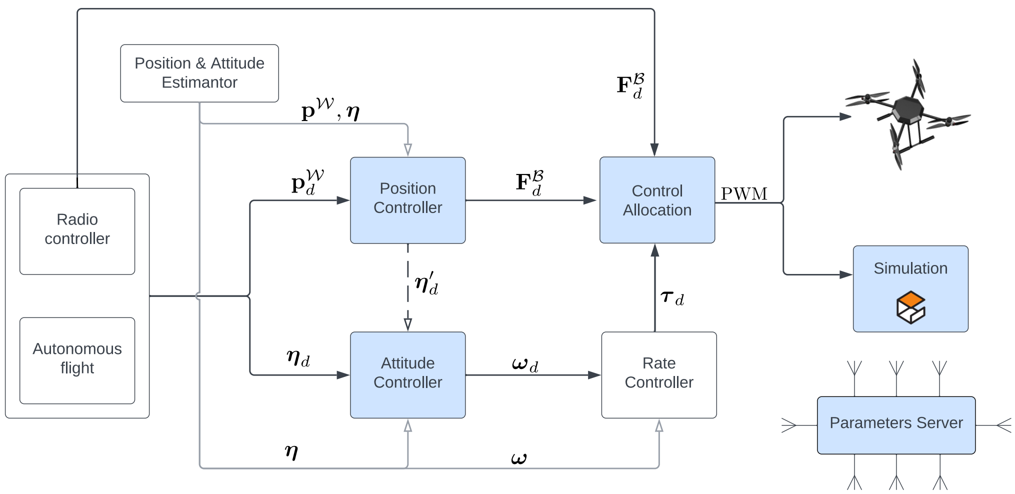


Fig. 4: Architecture of the customized version of PX4. The modified modules are highlighted with an azure background, while the unmodified modules have a white background.

changing its roll angle. This configuration can be useful in those applications where the drone needs to generate a desired force in only one direction, for example, pushing on a surface to perform some measurement with a specific sensor. The second configuration, instead, can take the form of several standard configurations, such as quadrotors, hexarotors, octorotors and so on. The difference here consists in the presence of one servomotor for each arm, with the rotation axis transversal to the arm (Fig. 2). Thanks to this configuration, the drone is able to move in all directions without changing its orientation and vice versa. The ability to perform more precise movements and to generate a force with any orientation while the drone maintains its attitude, makes these systems appealing for environment interaction tasks.

IV. PX4 Control Framework for Tiltable Multicopters

A. Mathematical Notation

The multirotor UAV coordinate systems are shown in Fig. 2-3, where $\mathcal{B} : (x^{\mathcal{B}}, y^{\mathcal{B}}, z^{\mathcal{B}})$ denotes the body fixed frame, attached to the UAV's center of mass, and $\mathcal{W} : (x^{\mathcal{W}}, y^{\mathcal{W}}, z^{\mathcal{W}})$ denotes the global world coordinate system. Let $\mathbf{p}^{\mathcal{W}} = [x^{\mathcal{W}}, y^{\mathcal{W}}, z^{\mathcal{W}}]^T \in \mathbb{R}^3$ be the position vector, $\boldsymbol{\eta} = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ be the attitude vector with ϕ_d, θ_d, ψ_d being the desired roll, pitch and yaw Euler angles, $\boldsymbol{\omega} = [\omega_\phi, \omega_\theta, \omega_\psi]^T \in \mathbb{R}^3$ be the angular rate vector, $\mathbf{f}^{\mathcal{B}} = [f_x^{\mathcal{B}}, f_y^{\mathcal{B}}, f_z^{\mathcal{B}}]^T \in \mathbb{R}^3$ be the force vector, $\boldsymbol{\tau}^{\mathcal{B}} = [\tau_\phi, \tau_\theta, \tau_\psi]^T \in \mathbb{R}^3$ be the torque vector. Following in this paper, with $(\cdot)_d$ will be indicated the desired quantities and the measured one without any subscript while, since the reference frame of the quantities presented above will not change, the apexes relative to the reference frame will be omitted for ease of writing.

B. Software Architecture

The software architecture of the proposed framework is integrated into the PX4 control stack[3] with some changes in the controller and allocation modules. A sketch of the proposed architecture is depicted in Fig. 4 where our main contribution is highlighted in boxes with azure background, integrated with the ones already available in the PX4 stack (boxes with white background).

Starting from the conceptualization of the PX4 architecture, the software modules reported in Fig. 4 can be logically divided into high-level and low-level modules. The high-level ones implement communication interfaces to command the drone in teleoperation or autonomous mode and reconstruct the state of the UAVs during the flight. Differently, the low-level modules implement the position, the attitude and the direct motor control to actuate the platform. A Pulse-Width Modulation (PWM) signal is generated for each motor of the platform, namely the motors for the propellers and the servomotors to tilt the propellers. This data is directly sent to the robot's actuator drivers when the firmware is run on a real system, and to the simulated models if the simulation layer of the PX4 software is enabled. Here, the simulation layer implements a SITL mechanism to bring the firmware architecture in a simulated UAV platform and environment to a local machine. Among the different tools, Gazebo dynamic simulator [21] is supported as well. In this context, specific simulation models suitable for the Gazebo simulator, representing the two tiltable multicopters tested in this work, have been implemented and deployed.

High-Level modules

Like a standard UAV, using the proposed framework a tilting multicopter can be controlled using different flight modes. Here, flight modes define how the autopilot can be commanded. For a matter of simplicity, we mainly discuss

[3]`https://docs.px4.io/v1.12/en/concept/architecture.html`

the autonomous and teleoperated control modes (i.e. *stabilized*, *position* and *offboard* modes). To interact with the aerial platform with a classical radio transmitter, the *Radio Controller* (RC) module can be used. Depending on the designated control mode, the RC can be used to specify the desired position ($\mathbf{p}_d^{\mathcal{W}}$) and attitude ($\boldsymbol{\eta}_d$) or directly specify the desired forces ($\mathbf{f}_d^{\mathcal{B}}$) acting on the platform and the attitude. Differently, the autonomous flight mode enables the control of the platform from a remote/onboard computer. In this case, the desired position and attitude (constrained to the total DoFs of the platform) are commanded.

Finally, to enable position and autonomous control, the state of the UAVs must be known. For this reason, the *Position & Attitude Estimator* module fuses different information sources like the GPS, optical flow, external pose estimation, IMU, and so on to calculate the position ($\mathbf{p}^{\mathcal{W}}$) and orientation ($\boldsymbol{\eta}$) of the platform during the flight.

It is worth to notice that these modules have not been modified with respect to the classical PX4 control stack. Meaning that the output of these modules has not been modified, while the way these data are processed by the low-level modules has been customized.

Low-Level modules

The low-level modules of the system architecture cooperate to generate the control input for the real/simulated platform. In this context, the *Position Controller* module consists of a cascade of a proportional control action for the position error and a PID controller for the linear velocity error. In this module, the control law has not been changed, but its output is used differently. Considering a flat multicopter, the desired velocities in the horizontal plane are transformed into desired roll and pitch angles that are given as input to the attitude controller, due to the under-actuation of the system. Therefore, the outputs of the position controller for a flat multirotor are $\boldsymbol{\eta}'_d = [\phi_d, \theta_d]^T \in \mathbb{R}^2$ and the total thrust $f_d = f_z \in \mathbb{R}$. Since drones with tiltable rotors have more DoFs, they can also move in the horizontal plane without changing one or both angles mentioned above, depending on the tiltable configuration (Sec. III). For example, with an H-shaped one-tilt configuration as shown in Fig. 3, the drone can move forward and backwards tilting the rotors without changing its pitch angle. In that case the outputs of the position controller are $\boldsymbol{\eta}'_d = \phi_d \in \mathbb{R}$ and $\mathbf{f_d} = [f_x, f_z]^T \in \mathbb{R}^2$. Instead, with a fully actuated tiltable configuration (Fig. 2), position and attitude controllers are independent and the output of the position controller consists of the desired force vector $\mathbf{f}_d = [F_x, F_y, F_z]^T \in \mathbb{R}^3$ only.

The *Attitude Controller* module consists of a proportional control action on the attitude error. In the case of a standard multirotor, the roll and pitch desired angles come from the position controller, while the yaw angle reference is given as the desired state from the external inputs. Thus, the inputs of this module are $\eta_d = \psi_d \in \mathbb{R}$ and $\boldsymbol{\eta}'_d = [\phi_d, \theta_d]^T \in \mathbb{R}^2$. With tiltable drones, instead, the input vectors are $\boldsymbol{\eta_d} = [\phi_d, \psi_d] \in \mathbb{R}^2$ and $\eta'_d = \theta_d \in \mathbb{R}$, or $\boldsymbol{\eta_d} = [\theta_d, \psi_d] \in \mathbb{R}^2$ and $\eta'_d = \phi_d \in \mathbb{R}$ in the case of H-shaped one-tilt configurations and $\boldsymbol{\eta_d} = [\phi_d, \theta_d, \psi_d] \in \mathbb{R}^3$ in the case of fully actuated tiltable configuration. The output of this module is the desired angular rates vector for all the configurations. In the end, the *Rate Controller* module consists of a PID controller that gives as output the desired torque vector.

Finally, the *Control Allocation* module is where the desired forces and torques vectors are allocated to generate the desired angular velocities for the propellers, given as PWM signals to the motor drivers or the simulated multirotor. The input vectors are allocated through the allocation matrix as written in equation(1), where $\mathbf{n} \in \mathbb{R}^N$ is the vector of the propeller's angular velocities with $N \in \mathbb{R}$ being the number of rotors, $\mathbf{A} \in \mathbb{R}^{6\times N}$ is the allocation matrix that depends on the rotors configuration and parameters (Fig. 5), with $\mathbf{A}^\dagger$ being the Moore-Penrose inverse.

$$\mathbf{n}^2 = \mathbf{A}^\dagger \begin{bmatrix} \mathbf{f}_d \\ \boldsymbol{\tau}_d \end{bmatrix} \tag{1}$$

This module has been modified to take into account the new tiltable-rotors multirotors airframe family. For the H-shaped one-tilt configuration has been added the mapping of the desired angle on the servo motors, while for the fully actuated tiltable configuration has been changed the entire allocation matrix. More specifically, with the settings from the user interface (Fig. 5), the module generates a static allocation matrix $\mathbf{A}_{static} \in \mathbb{R}^{6\times 2N}$ [28] that allows to compute the forces vector $\mathbf{f}_{dec} = [f_{v_1}, f_{l_1}, ..., f_{v_N}, f_{l_N}]^T \in \mathbb{R}^{2N}$, with f_{v_i} and f_{l_i} for $i = 1, ..., N$ being the vertical and horizontal force of each rotor. At the end, from $\mathbf{f}_{dec}$ are computed the propeller's angular velocities and the desired angles of the servo motors.

Parameter server

To customize the tiltable drone configuration and tune its control, a new set of parameters is available with the proposed firmware. As described in Section V, the parameters related to the drone's configuration are used to specify the type of tiltable drone, the number of rotors and servomotors with their rotation limits. To tune the control of the drone, in addition to the already present control gains, we added few parameters related to the horizontal forces and desired body attitude bounds. More specifically, the parameter MC_MAX_FXY is used to limit the maximum and minimum value (expressed in Newton) of the forces that the drone can exert along x and y axes. The parameters introduced to limit the desired body attitude are MC_DES_PITCH_MAX, MC_DES_PITCH_MIN, MC_DES_ROLL_MAX and MC_DES_ROLL_MIN, which they represent the boundaries for the desired roll and pitch angles, expressed in degrees.

V. User interface

To manage the firmware settings, the various parameters and the communication with the drone, PX4 users usually utilize the Graphic User Interface (GUI) QGroundControl (QGC)[4]. Being the proposed framework fully integrated

[4] http://qgroundcontrol.com/

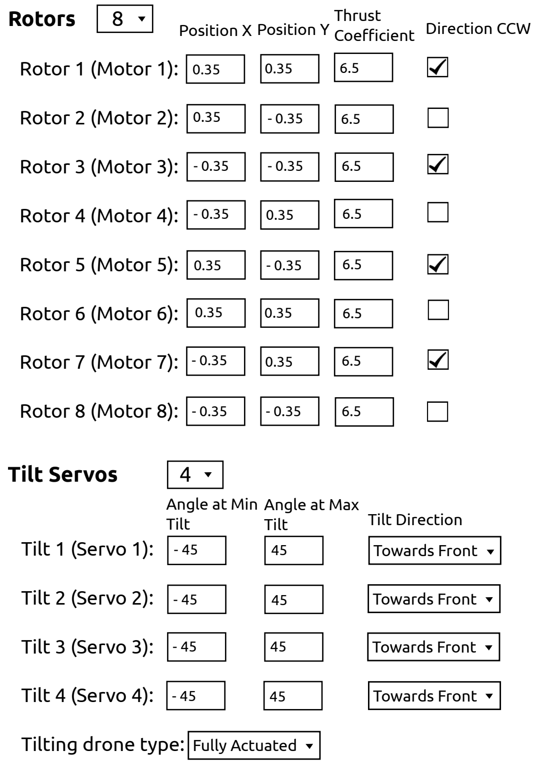


Fig. 5: QGC interface to set up a tiltable drone

with the original autopilot firmware, QGC can be used to set up the tiltable drones as well. Thanks to the *dynamic control allocation*[5], one of the new features introduced in the firmware, the user has the possibility to setup the geometry of its drone in the *actuators configuration* panel. Starting from that, we added some new parameters that are related to the configuration of a drone with tiltable propellers. As shown in Fig. 5, in addition to the rotors section, where it is possible to set the number of the rotors and their position with respect to the flight controller position, there is the one dedicated to the servomotors. This panel is used to set up the actuators that tilt the propellers and the tiltable drone configuration. The tilt direction parameter for the servomotors is used only for the H-shaped one-tilt configuration, to specify if the propellers rotate toward the front or the side of the drone. While, with the fully actuated configuration is not needed to specify the direction because is always considered that the servomotor rotation axis is transversal to the arm of the drone. The maximum and minimum angle values are used to set a limit for the servomotors. This value can be related to the mechanical limit of the servomotors or a safety limit that the user wants to impose, keeping in mind that it will influence also the flight performance.

VI. Case Studies

A set of experiments has been carried out to demonstrate the effectiveness of the control framework. The tests aimed at showing stabilization and trajectory tracking of the two airframes previously introduced (see Fig. 2-3). Two kinds of simulation tests were performed. A trajectory tracking test, in which the robot must follow a 3D planned trajectory and a hovering test, in which the orientation of the drone is changed while its position remains fixed. These tests have been performed in the *offboard flight mode*, meaning that the flight controller receives a desired trajectory from an external application. In this context, the trajectory has been planned with a fifth order polynomial function. As for the heading direction (namely, the yaw), the trajectory has been planned in order to follow the motion. The simulations have been performed using Gazebo simulator, while the communication between the external application and the emulated system autopilot is performed with the PX4-FastRTPS Bridge[6], to send the trajectory planned and executed in C++ using ROS2 as robotic middleware [29]. In the end, the effectiveness of the framework has been demonstrated also with initial experiments by teleoperating a fully actuated tilting multicopter.

A. Trajectory Tracking Test

This test has been performed with the two tilting multirotor platforms, the fully actuated and the H-shaped, respectively. In both cases, the drone was asked to takeoff (in about 10 s) and perform a circular motion. The reference (planned) trajectory and the actual performed one are reported in Fig. 6a and 6f. It is worth noting that the UAV with the fully actuated configuration is able to follow satisfactorily the desired trajectory (Fig. 6c), without changing its roll and pitch angles (Fig. 6d). Differently, the one-tilt H-shape drone must also modify its roll configuration in order to move along the y axis (Fig. 6h,6i). Finally, Figures 6e and 6j show the values of the rotor's angles computed from the proposed control framework to track the trajectory, for the two drone configurations, respectively.

B. Hovering Test

In the second test case, the goal was to demonstrate the functionality of the introduced configurations to decouple the translational DoFs from the rotational ones. The behavior of the simulated platforms in the Gazebo simulation environment during this test is depicted in Fig. 8, in which the rotation of the propellers is clear. For this reason, after the takeoff, the floating base is commanded to remain in the same position while changing the desired attitude. Results of this experiment performed with the fully actuated drone are reported. In particular, Fig. 7a shows the commanded and measured positions respectively, along with the platform attitude ($\boldsymbol{\eta}$), depicted in Fig. 7b in which the roll and pitch desired angles are reported. The attitude tracking error is depicted in Fig. 7c. It's worth noticing that this error is negligible, with a maximum error norm of 2 deg during the floating base rotation, while the drone changes its attitude to 20 degrees. Finally, Fig. 7d shows how the servomotors' angles change.

[5] https://docs.px4.io/main/en/concept/control_allocation.html

[6] https://dev.px4.io/v1.11_noredirect/en/middleware/micrortps.html

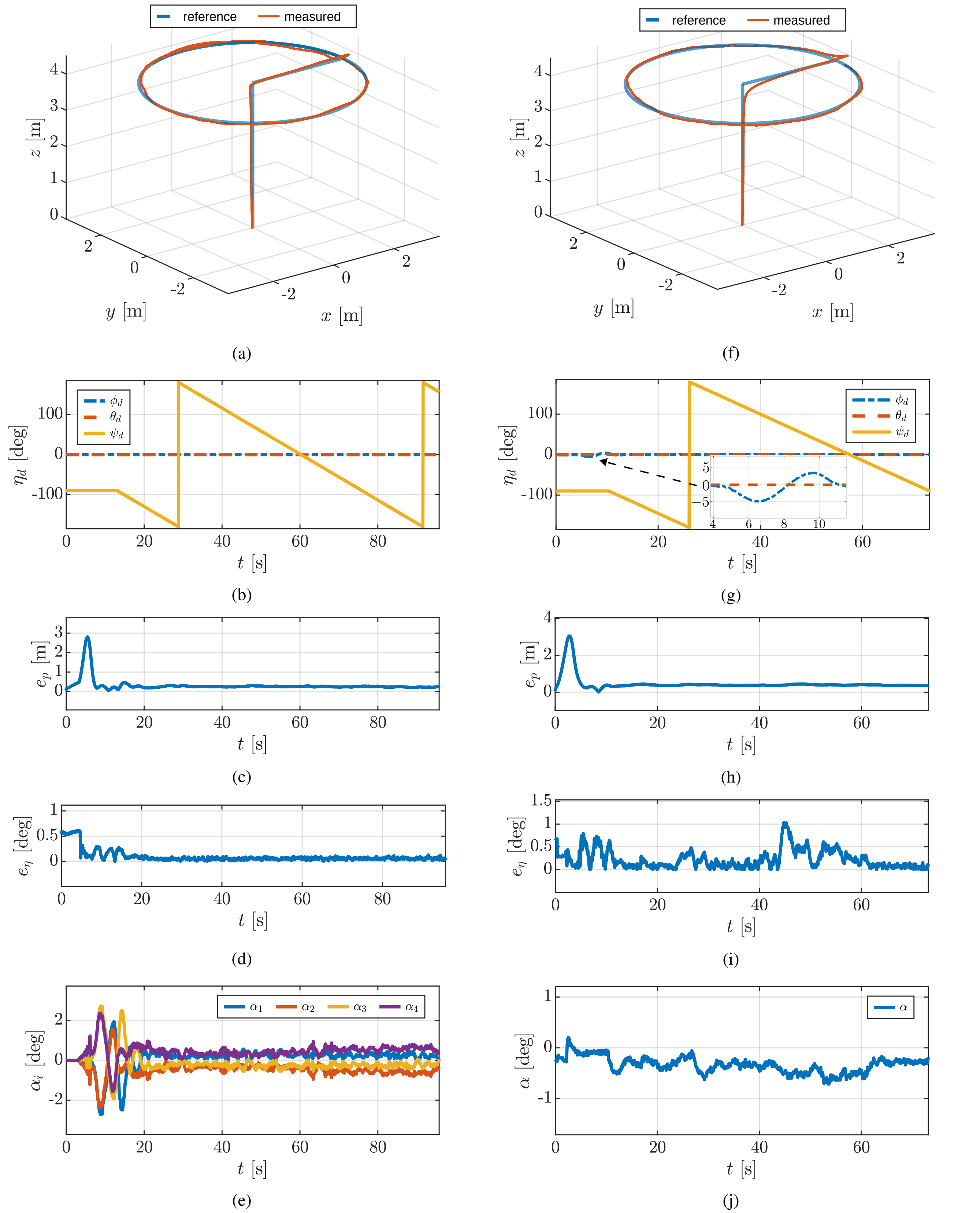
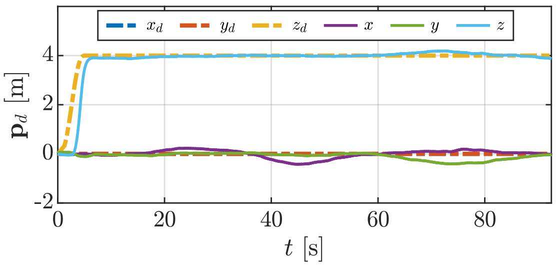
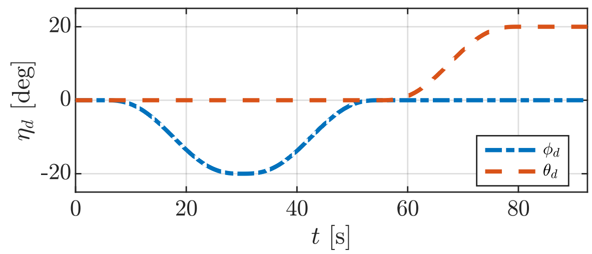


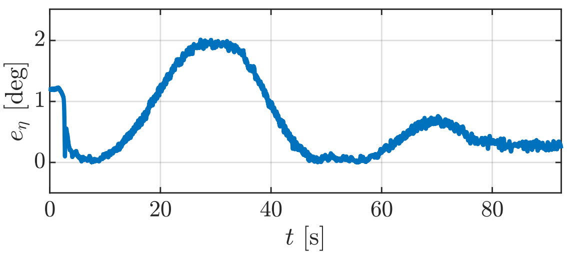
Fig. 6: Position trajectory tracking: in the left column there are the results obtained with the fully actuated tiltable configuration, and in the right column the results obtained with the H-shape one-tilt configuration. The plots represent respectively: (a)(f) the desired and measured 3D trajectory, (b)(g) the desired attitude expressed with Euler angles, (c)(h) the norm of the position error, (d)(i) the norm of the attitude error and (e)(j) the desired servomotor tilt angles



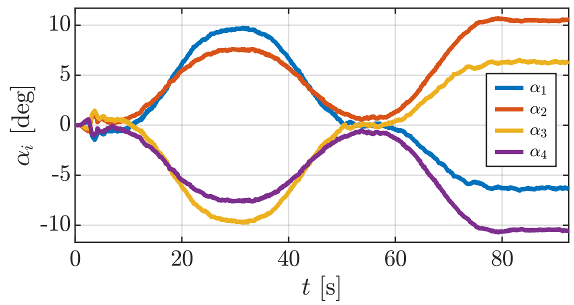
(a)



(b)



(c)



(d)

Fig. 7: Attitude tracking while hovering, with a fully actuated tilting configuration: (a) desired and measured position trajectory, (b) desired attitude trajectory represented with Euler angles, (c) norm of the attitude error, (d) desired servomotor tilt angles

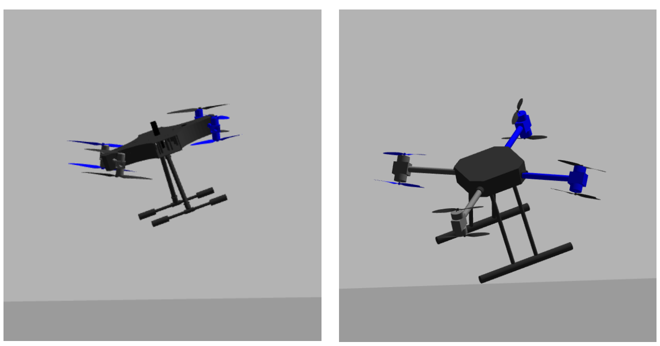


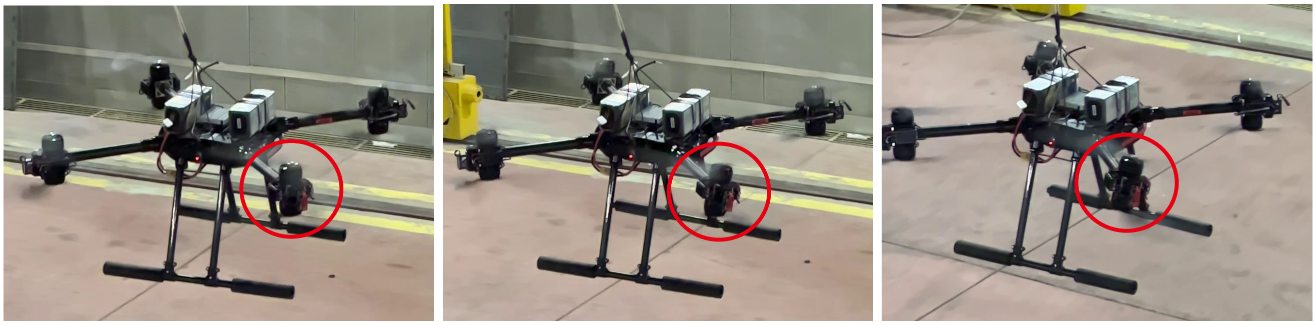
Fig. 8: Hovering with a desired attitude: on the left is shown the H-shape one-tilt configuration in hovering with a pitch angle of $-20\,\text{deg}$, on the right is shown the fully actuated tilting configuration in hovering with $20\,\text{deg}$ of roll and $-20\,\text{deg}$ of pitch angles

C. Preliminary Experiments

Initial experiments on a real system have been performed as well. In particular, a teleoperated flight in which a human operator drives a fully actuated platform has been carried out. Considering the system architecture described in Fig. 4, the operator uses the RC to directly generate the force vector ($\mathbf{f}_d^{\mathcal{B}}$) as input for the *allocation module*. In this context, the *stabilized flight mode* is used, since the position estimation of the UAV was not available. The platform utilized in this experiment (Fig. 9,1) consists of a coaxial octocopter commercial frame, with almost $1\,\text{m}$ of diameter and $11.5\,\text{Kg}$ of payload. It has been modified by adding eight T-Motor U7 V2.0 280rpm/V brushless motors coupled with 20" × 6" propellers, four high-speed Savox Monster SB-2292SG servomotors and a custom mechanism to make it fully actuated with tiltable rotors. It is equipped with the Holybro PixHawk 6C flight controller and is powered by two 6S-25C Li-Po batteries connected in parallel. It has also an Intel UP^2 companion computer, that will be used for future experiments. Despite the absence of the position feedback, this experiment allowed us to test the modified modules of the *control allocation*, *attitude controller*, and the *rate controller* showing a great stabilization of the platform during the hovering and random movements.

VII. Conclusions

Despite the numerous advantages that configurations with tiltable rotors offer, nowadays doesn't exist an open source firmware that allows one to control them. In this paper, we proposed a customization of the well-known firmware PX4 to introduce a new airframe family for drones with tiltable rotors. Two types of configurations have been added: the H-shaped one-tilt and the fully actuated. Both configurations have their advantages and disadvantages, considering that additional DoFs confer more versatility to the system, while on the other hand bringing more complexity in the control law and the dynamics. To validate the proper functioning of the firmware, we performed some experiments in the dynamic simulation environment Gazebo, testing the stabilization and trajectory tracking, for both position and attitude. The simulation results showed good performance of our customized firmware, giving us the confidence to perform



(a) (b) (c)

Fig. 9: Flight test with a fully actuated tilting drone. Inside the circle is highlighted the tilt of the propeller

tests also on real platforms. In fact, at the time this paper is being written, the firmware is being tested on two platforms with the configurations discussed above.

Future directions of this work concern the execution of further tests on real platforms, the test of different control approaches for drones with tiltable rotors to compare the performance with the already implemented PIDs cascade, and the implementation of a force or admittance/impedance control to exploit the ability of these configurations to exert a desired horizontal force.

REFERENCES

[1] J. Cacace, A. Finzi, V. Lippiello, G. Loianno, and D. Sanzone, “Aerial service vehicles for industrial inspection: Task decomposition and plan execution,” *Applied Intelligence*, vol. 42, p. 49–62, jan 2015.

[2] G. Aiello, F. Hopps, D. Santisi, and M. Venticinque, “The employment of unmanned aerial vehicles for analyzing and mitigating disaster risks in industrial sites,” *IEEE Transactions on Engineering Management*, vol. 67, no. 3, pp. 519–530, 2020.

[3] J. Cacace and et al., “Safe local aerial manipulation for the installation of devices on power lines: Aerial-core first year results and designs,” *Applied Sciences*, vol. 11, no. 13, 2021.

[4] T. Mao, K. Huang, X. Zeng, L. Ren, C. Wang, S. Li, M. Zhang, and Y. Chen, “Development of power transmission line defects diagnosis system for uav inspection based on binocular depth imaging technology,” in *2019 2nd International Conference on Electrical Materials and Power Equipment (ICEMPE)*, pp. 478–481, 2019.

[5] N. Dilshad, J. Hwang, J. Song, and N. Sung, “Applications and challenges in video surveillance via drone: A brief survey,” in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 728–732, 2020.

[6] K. S. Lee, M. Ovinis, T. Nagarajan, R. Seulin, and O. Morel, “Autonomous patrol and surveillance system using unmanned aerial vehicles,” in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, pp. 1291–1297, 2015.

[7] F. Ruggiero, V. Lippiello, and A. Ollero, “Aerial manipulation: A literature review,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[8] P. Zheng, X. Tan, B. B. Kocer, E. Yang, and M. Kovac, “Tiltdrone: A fully-actuated tilting quadrotor platform,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6845–6852, 2020.

[9] E. Cuniato, N. Lawrance, M. Tognon, and R. Siegwart, “Power-based safety layer for aerial vehicles in physical interaction using lyapunov exponents,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6774–6781, 2022.

[10] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, “Active interaction force control for contact-based inspection with a fully actuated aerial vehicle,” *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 709–722, 2021.

[11] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, “Design of multirotor aerial vehicles: A taxonomy based on input allocation,” *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 1015–1044, 2021.

[12] A. Franchi, R. Carli, D. Bicego, and M. Ryll, “Full-pose tracking control for aerial robotic systems with laterally bounded input force,” *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 534–541, 2018.

[13] W. Myeong and H. Myung, “Development of a wall-climbing drone capable of vertical soft landing using a tilt-rotor mechanism,” *IEEE Access*, vol. 7, pp. 4868–4879, 2018.

[14] W. Myeong, S. Jung, B. Yu, T. Chris, S. Song, and H. Myung, “Development of wall-climbing unmanned aerial vehicle system for micro-inspection of bridges,” in *Proceedings of the Workshop on The Future of Aerial Robotics: Challenges and Opportunities, International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada*, pp. 20–24, 2019.

[15] D. Lee, H. Seo, I. Jang, S. J. Lee, and H. J. Kim, “Aerial manipulator pushing a movable structure using a dob-based robust controller,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 723–730, 2020.

[16] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Pixhawk: A system for autonomous flight using onboard computer vision,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 2992–2997, 2011.

[17] L. Meier, D. Honegger, and M. Pollefeys, “PX4: A Node-Based Multithreaded Open Source Robotics Framework for Deeply Embedded Platforms,” *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, 2015.

[18] R. Olfati-Saber, “Global configuration stabilization for the vtol aircraft with strong input coupling,” *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1949–1952, 2002.

[19] “inav firmware.” [ttps://github.com/iNavFlight/inav/releases/tag/2.6.0. Accessed: 2022-12-01.

[20] “Ardupilot firmware.” [https://firmware.ardupilot.org]. Accessed: 2022-12-01.

[21] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *RotorS—A Modular Gazebo MAV Simulator Framework*, pp. 595–625. Cham: Springer International Publishing, 2016.

[22] M. Ryll, H. H. Bülthoff, and P. R. Giordano, “Modeling and control of a quadrotor uav with tilting propellers,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 4606–4613, 2012.

[23] M. Ryll, H. H. Bülthoff, and P. R. Giordano, “First flight tests for a quadrotor uav with tilting propellers,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 295–302, 2013.

[24] “Appellix.” [www.apellix.com]. Accessed: 2020-01-19.

[25] “Texo drone service.” [www.texodroneservices.co.uk/blog/56]. Accessed: 2020-01-19.

[26] “Ronik inspectioneering.” [http://www.inspectioneering.eu/]. Accessed: 2020-01-19.

[27] “Voliro inspection.” [https://www.voliro.com/inspection]. Accessed: 2020-01-19.

[28] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski, “The voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle,” *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 34–44, 2018.

[29] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming - Second Edition: Design, Build, and Simulate Complex Robots Using the Robot Operating System*. Packt Publishing, 2nd ed., 2018.