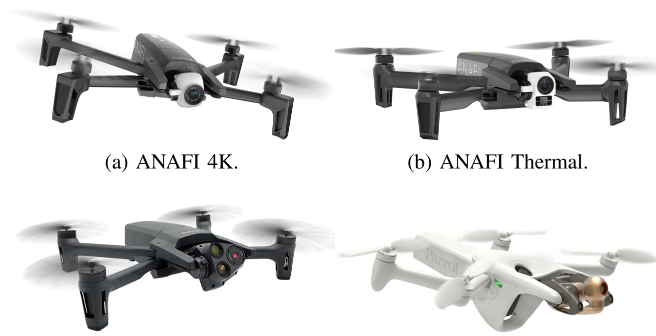
anafi_ros: from Off-the-Shelf Drones to Research Platforms

Andriy Sarabakha[1]

***Abstract*— The off-the-shelf drones are simple to operate and easy to maintain aerial systems. However, due to proprietary flight software, these drones usually do not provide any open-source interface which can enable them for autonomous flight in research or teaching. This work introduces a package for ROS1 and ROS2 for straightforward interfacing with off-the-shelf drones from the Parrot ANAFI family. The developed ROS package is hardware agnostic, allowing connecting seamlessly to all four supported drone models. This framework can connect with the same ease to a single drone or a team of drones from the same ground station. The developed package was intensively tested at the limits of the drones' capabilities and thoughtfully documented to facilitate its use by other research groups worldwide.**



(a) ANAFI 4K. (b) ANAFI Thermal.

(c) ANAFI USA. (d) ANAFI Ai.

Fig. 1. Parrot ANAFI family drones.

I. Introduction

As one of the fastest-growing fields in the aerospace industry, unmanned aerial vehicles (UAVs) can provide a cost-efficient solution to many time-consuming tasks, such as subterranean exploration [1], power-line inspection [2] and additive manufacturing [3]. Different applications require drones equipped with appropriate sensors, for example, a thermal camera for wildfire monitoring [4], a camera with a high optical zoom for aerial observation [5], or a stereo camera for visual odometry [6]. Custom-built drones can be designed and manufactured to meet specific needs and requirements, allowing for more flexibility in their functionality [7]. However, assembling and configuring custom-made drones require technical expertise and labour time.

Unlike custom-made drones, off-the-shelf drones are readily available and can be easily purchased from many retailers. This makes them an attractive option for individuals and organisations that want to use drones for a variety of purposes, such as teaching and research. Another advantage of using off-the-shelf drones is their reliability and durability since they are built to operate in a wide range of conditions.

While there are many advantages of using off-the-shelf drones, their main limitation is the possibility of autonomous deployments, making them less suitable for certain applications. To partially overcome this issue, proprietary software development kits (SDKs) were released by some drone manufacturers, such as DJI [8], Ryze [9], Parrot [10] and Bitcraze [11]. Still, only discontinued Parrot Bebop [12], and tiny-size Ryze Tello [13] and Bitcraze Crazyflie [14] have the interface to bridge them with the robot operating system (ROS). Nowadays, ROS has become a standard development environment for modern roboticists. The main advantage of ROS is the possibility for the modularization of software, making it easy to reuse and modify individual components. Besides, ROS provides a standard set of libraries, tools and conventions for communication between different parts of a robotic system. Moreover, ROS has a large and active community with many available resources. A huge variety of ROS packages is available for building the navigation stack for aerial robots, like visual-inertial localisation [15], environment perception [16], motion planning [17], model-based [18] and model-free [19] control.

This work introduces a bridge which allows a straightforward connection between the drones of the Parrot[*] ANAFI family (illustrated in Fig. 1) and both ROS1[†] and ROS2[‡]. Parrot ANAFI drones were chosen because each model in the family offers unique features making them suitable for various applications. A comprehensive comparison of the characteristics of Parrot ANAFI drones is provided. The developed ROS package is hardware agnostic, allowing connecting seamlessly to all supported drones. This framework also allows connecting to single or multiple drones from the same ground station. The developed package was intensively tested at the limits of the drones and thoughtfully documented to facilitate its use by other researchers.

This work is organised as follows. First, the technical details of the experimental platforms are provided in Section II. Next, Section III describes the structure of the developed framework. Then, Section IV provides experimental validation of the developed framework. Finally, Section V summarises this work with conclusions and future work.

This research was supported by the NTU Presidential Postdoctoral Fellowship (award number 021820-00001).

[1]Andriy Sarabakha is with the School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU), Singapore, 639798. andriy.sarabakha@ntu.edu.sg

[*]The authors have no relations or conflicts of interest with Parrot SA.

[†]The open-source code is available at https://github.com/andriyukr/anafi_ros/tree/ros1.

[‡]The open-source code is available at https://github.com/andriyukr/anafi_ros/tree/ros2.

II. Experimental Platforms

Let the world fixed frame be $\mathcal{F}_W = \{\vec{\mathbf{x}}_W, \vec{\mathbf{y}}_W, \vec{\mathbf{z}}_W\}$, and the drone body frame be $\mathcal{F}_D = \{\vec{\mathbf{x}}_D, \vec{\mathbf{y}}_D, \vec{\mathbf{z}}_D\}$. The origin of the body frame is located at the centre of mass (COM) of the UAV. The configuration with the corresponding reference frames is illustrated in Fig. 2.

The absolute position of UAV $\mathbf{p}_D^W = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is described by three Cartesian coordinates of its COM in $\mathcal{F}_W$. While the attitude of UAV $\boldsymbol{\theta}_D^W = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is described by three Euler's angles: roll ϕ, pitch θ and yaw ψ.

The time derivative of the position (x, y, z) gives the linear velocity of the UAV's COM expressed in $\mathcal{F}_W$:

$$\mathbf{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T, \tag{1}$$

and the velocity expressed in $\mathcal{F}_D$ is

$$\mathbf{v}_D = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T. \tag{2}$$

The relation between $\mathbf{v}$ and $\mathbf{v}_B$ is given by

$$\mathbf{v} = \mathbf{R}(\phi, \theta, \psi)\mathbf{v}_D, \tag{3}$$

in which $\mathbf{R}(\phi, \theta, \psi) \in \mathsf{SO}(3)$ is the rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$:

$$\mathbf{R}(\phi, \theta, \psi) = \begin{bmatrix} \mathrm{c}_\psi \mathrm{c}_\theta & \mathrm{c}_\psi \mathrm{s}_\phi \mathrm{s}_\theta - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi \mathrm{s}_\theta \\ \mathrm{c}_\theta \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi \mathrm{s}_\theta & \mathrm{c}_\phi \mathrm{s}_\psi \mathrm{s}_\theta - \mathrm{c}_\psi \mathrm{s}_\phi \\ -\mathrm{s}_\theta & \mathrm{c}_\theta \mathrm{s}_\phi & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix}, \tag{4}$$

in which $\mathrm{c}_\star$ and $\mathrm{s}_\star$ are $\cos(\star)$ and $\sin(\star)$, respectively.

The time derivative of the attitude (ϕ, θ, ψ) gives the angular velocity expressed in $\mathcal{F}_W$:

$$\boldsymbol{\omega} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T, \tag{5}$$

and the angular velocity expressed in $\mathcal{F}_D$ is

$$\boldsymbol{\omega}_D = \begin{bmatrix} \omega_\phi & \omega_\theta & \omega_\psi \end{bmatrix}^T. \tag{6}$$

The relation between $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_D$ is given by

$$\boldsymbol{\omega} = \mathbf{T}\boldsymbol{\omega}_D, \tag{7}$$

in which $\mathbf{T}$ is the transformation matrix:

$$\mathbf{T} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}. \tag{8}$$

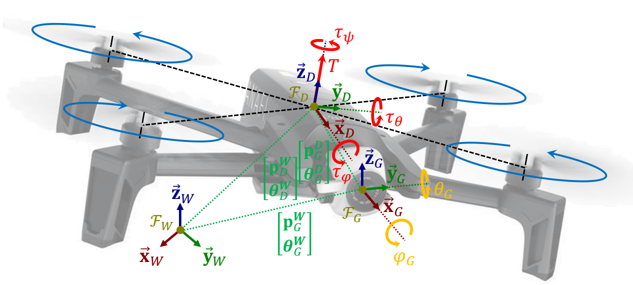


Fig. 2. Configuration of Parrot ANAFI with its reference frames.

The vector of control inputs $\mathbf{u}$ is considered as in [20]:

$$\mathbf{u} = \begin{bmatrix} T^* & \tau_\phi^* & \tau_\theta^* & \tau_\psi^* \end{bmatrix}^T, \tag{9}$$

where T^* is the reference thrust acting along $\vec{\mathbf{z}}_D$ axis, whereas τ_ϕ^*, τ_θ^* and τ_ψ^* are the reference moments acting around $\vec{\mathbf{x}}_D$, $\vec{\mathbf{y}}_D$ and $\vec{\mathbf{z}}_D$ axes, respectively.

For mobile robots equipped with a camera mounted on a gimbal, there is also a gimbal reference frame $\mathcal{F}_G = \{\vec{\mathbf{x}}_G, \vec{\mathbf{y}}_G, \vec{\mathbf{z}}_G\}$. The origin of the gimbal frame is located in front of the UAV's COM. The position of the camera in $\mathcal{F}_D$ is $\mathbf{p}_G^D = \begin{bmatrix} x_G & y_G & z_G \end{bmatrix}^T$, while the attitude of the gimbal in $\mathcal{F}_D$ is $\boldsymbol{\theta}_G^D = \begin{bmatrix} \phi_G & \theta_G & \psi_G \end{bmatrix}^T$. The rotation matrix of the gimbal in $\mathcal{F}_W$ can be calculated with

$$\mathbf{R}_G^W = \mathbf{R}(\phi, \theta, \psi)\mathbf{R}(\phi_G, \theta_G, \psi_G), \tag{10}$$

while the position of the camera in $\mathcal{F}_W$ can be obtained with

$$\mathbf{p}_G^W = \mathbf{p}_D^W + \mathbf{R}(\phi, \theta, \psi)\mathbf{p}_G^D. \tag{11}$$

A. ANAFI Drones

Parrot ANAFI drones are small, lightweight UAVs mainly designed for aerial photography and videography. They are equipped with high-resolution high-dynamic-range (HDR) cameras mounted on a 2-axis gimbal, allowing to capture smooth and stable footage from the air. These drones have a unique folding design, making them portable and easily deployable: it takes less than 1 min to unfold the drone, turn it on, connect to the remote controller and take off. Moreover, ANAFI drones have reasonably long flight duration thanks to the lithium polymer (LiPo) battery, which has a built-in USB-C port for hassle-free charging. The Parrot ANAFI family has four drone models: basic *ANAFI 4K*, *ANAFI Thermal* with a thermal camera, water- and dust-resistant *ANAFI USA* and *ANAFI Ai* with onboard computing and obstacle avoidance capabilities. These drones are illustrated in Fig. 1, while Table I summarises the properties of each model.

Remark 1: Since low-level stabilisation controllers as in [21] are included in ANAFI's autopilot as illustrated in Fig. 3, the virtual control inputs in (9) can be considered as:

$$\mathbf{u} = \begin{bmatrix} v_z^* & \phi^* & \theta^* & \omega_\psi^* \end{bmatrix}^T. \tag{12}$$

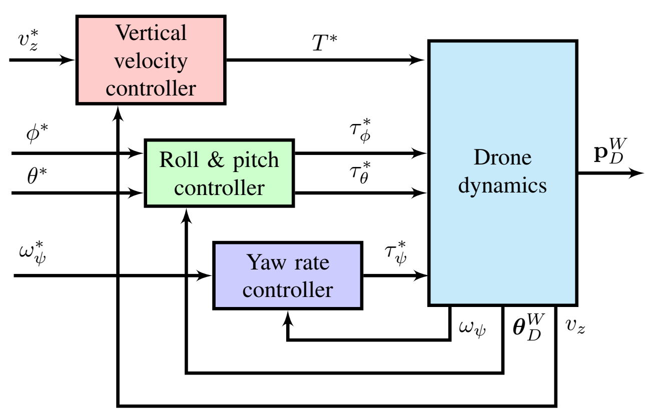


Fig. 3. Control scheme of Parrot ANAFI UAV.

TABLE I

TECHNICAL SPECIFICATIONS OF PARROT ANAFI DRONES.

<table>
<tr><th colspan="2" rowspan="2">Parameter</th><th colspan="4">ANAFI</th></tr>
<tr><th>4K</th><th>Thermal</th><th>USA</th><th>Ai</th></tr>
<tr><td rowspan="11">Drone</td><td>Size folded[a] [mm]</td><td>244 × 67 × 65</td><td>218 × 69 × 64</td><td>252 × 104 × 82</td><td>304 × 130 × 118</td></tr>
<tr><td>Size unfolded[a] [mm]</td><td>242 × 315 × 65</td><td>242 × 315 × 64</td><td>303 × 398 × 84</td><td>378 × 498 × 118</td></tr>
<tr><td>Weight [g]</td><td>320</td><td>315</td><td>499</td><td>898</td></tr>
<tr><td>Maximum horizontal speed [m/s]</td><td colspan="2">15</td><td>14.7</td><td>16</td></tr>
<tr><td>Maximum vertical speed [m/s]</td><td colspan="4">4</td></tr>
<tr><td>Maximum wind resistance [m/s]</td><td colspan="2">13.9</td><td>14.7</td><td>12.7</td></tr>
<tr><td>Service ceiling [m]</td><td colspan="2">4500</td><td>6000</td><td>5000</td></tr>
<tr><td>Operating temperatures [°C]</td><td colspan="2">−10 to +40</td><td>−35 to +43</td><td>−10 to +40</td></tr>
<tr><td>Ingress protection</td><td colspan="2">✘</td><td>IP53[b]</td><td>IPX3[c]</td></tr>
<tr><td>Noise emission[d] [dB]</td><td>64</td><td>66</td><td>79</td><td>82</td></tr>
<tr><td>Slots</td><td colspan="3">MicroSD</td><td>MicroSD & SIM card</td></tr>
<tr><td colspan="2">Satellite navigation</td><td colspan="2">GPS & Glonass</td><td colspan="2">GPS, Glonass & Galileo</td></tr>
<tr><td rowspan="5">EO Camera</td><td>Sensor</td><td colspan="4">CMOS</td></tr>
<tr><td>Aperture</td><td colspan="3">f/2.4</td><td>f/2.0</td></tr>
<tr><td>ISO</td><td colspan="3">100 – 3200</td><td>50 – 6400</td></tr>
<tr><td>Shutter speed [s]</td><td colspan="3">1 – 1/10000</td><td>1/15 – 1/10000</td></tr>
<tr><td>Zoom [x]</td><td colspan="2">1 – 3</td><td>1 – 32</td><td>1 – 6</td></tr>
<tr><td rowspan="5">Video</td><td>Format</td><td colspan="3">MP4 (H.264)</td><td>MP4 (H.264, H.265)</td></tr>
<tr><td>Resolution</td><td>4K & FHD</td><td colspan="2">4K, FHD & HD</td><td>4K & FHD</td></tr>
<tr><td>Framerate [fps]</td><td>24 – 60</td><td>24 – 120</td><td>24 – 30</td><td>24 – 120</td></tr>
<tr><td>Horizontal field of view [°]</td><td colspan="3">69</td><td>68</td></tr>
<tr><td>Maximum video bandwidth [Mbps]</td><td colspan="2">100</td><td>5</td><td>200</td></tr>
<tr><td rowspan="3">Photo</td><td>Format</td><td colspan="4">JPEG & DNG (RAW)</td></tr>
<tr><td>Resolution [MP]</td><td colspan="3">21</td><td>48</td></tr>
<tr><td>Horizontal field of view [°]</td><td colspan="2">84</td><td>75</td><td>73</td></tr>
<tr><td rowspan="9">Thermal Camera</td><td>Sensor</td><td rowspan="9">✘</td><td>FLIR LEPTON 3.5</td><td>FLIR BOSON</td><td rowspan="9">✘</td></tr>
<tr><td>Resolution [pixels]</td><td>160 × 120</td><td>320 × 256</td></tr>
<tr><td>Temperature range [°C]</td><td>−10 to +400</td><td>−40 to +180</td></tr>
<tr><td>Thermal sensitivity [°C]</td><td colspan="2">0.05</td></tr>
<tr><td>Pixel pitch [μm]</td><td colspan="2">12</td></tr>
<tr><td>Horizontal field of view [°]</td><td>57</td><td>50</td></tr>
<tr><td>Photo format</td><td colspan="2">JPEG</td></tr>
<tr><td>Video format</td><td colspan="2">MP4 (H.264)</td></tr>
<tr><td>Video framerate [fps]</td><td colspan="2">9</td></tr>
<tr><td rowspan="3">Gimbal</td><td>Mechanical</td><td colspan="3">2-axis (roll, pitch)</td><td>3-axis</td></tr>
<tr><td>Electronic (EIS)</td><td colspan="4">3-axis</td></tr>
<tr><td>Tilt range[e] [°]</td><td colspan="3">from −90 to +90</td><td>from −116 to +176</td></tr>
<tr><td rowspan="7">Battery</td><td>Maximum flight time [min]</td><td>25</td><td>26</td><td colspan="2">32</td></tr>
<tr><td>Type</td><td colspan="2">LiPo (2 cells)</td><td colspan="2">LiPo (3 cells)</td></tr>
<tr><td>Capacity [mAh]</td><td colspan="2">2700</td><td>3400</td><td>6800</td></tr>
<tr><td>Voltage [V]</td><td colspan="2">7.6</td><td colspan="2">13.2</td></tr>
<tr><td>Weight [g]</td><td colspan="2">125</td><td>195</td><td>366</td></tr>
<tr><td>Maximum charging power [W]</td><td colspan="2">25</td><td>30</td><td>45</td></tr>
<tr><td>Charging port</td><td colspan="4">USB-C</td></tr>
<tr><td colspan="2">Controller</td><td colspan="3">Skycontroller 3</td><td>Skycontroller 4</td></tr>
<tr><td rowspan="3">Tools</td><td>Air SDK</td><td>✘</td><td>✘</td><td>✘</td><td>✔</td></tr>
<tr><td>Sphinx</td><td>✔</td><td>✘</td><td>✘</td><td>✔</td></tr>
<tr><td>anafi_ros</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
</table>

[a]length × width × height

[b]protected from limited dust ingress and water spray less than 60° from vertical

[c]protected from water spray less than 60° from vertical

[d]at 1 m

[e]from nadir to zenith

1) ANAFI 4K: This model (shown in Fig. 1a) is the first and basic drone of the ANAFI family. ANAFI 4K is one of the quietest drones in its class with a noise level of 64 dB at 1 m. With 25 min of flight time, the battery can be recharged via a USB-C cable in 90 min. The model of ANAFI 4K is available in Parrot's software-in-the-loop simulation environment – Sphinx.

2) ANAFI Thermal: This model (shown in Fig. 1b) is the upgrade of ANAFI 4K with a thermal camera. The optical unit of ANAFI Thermal combines the electro-optics with an infrared sensor, making it possible to identify temperatures between −10 °C and +400 °C. Thanks to the FLIR Lepton radiometric sensor, the absolute temperature of each pixel can be determined. The RGB image can be blended with thermal images. This enables to detection of hot spots with the thermal camera, while the RGB camera allows the viewing of important details. Despite the thermal camera, ANAFI Thermal is the smallest and lightest model of the family.

TABLE II

TECHNICAL SPECIFICATIONS OF PARROT SKYCONTROLLERS.

Parameter	**Skycontroller**	
	3	**4**
Size folded[a] [mm]	$94 \times 152 \times 72$	$147 \times 238 \times 55$
Size unfolded[a] [mm]	$153 \times 152 \times 116$	$147 \times 315 \times 55$
Weight [g]	386	606
Transmission system	Wi-Fi 802.11a/b/g/n	Wi-Fi 802.11a/b/g/n & 4G
Frequency used [GHz]	2.4, 5.8	2.4, 5
Maximum transmission distance [km]	4	∞
Video stream resolution	HD 720p	1080p
Battery capacity [mAh]	2500	3350
Battery life [h]	2.5	
Ports	USB-C (charge) & USB-A (connection)	USB-C (charge and connection) & micro-HDMI
Compatible mobile devices	screen size up to 6”	screen size up to 8”
Ingress protection	✘	IP5X[f]
Compatible drones	ANAFI 4K, ANAFI Thermal, ANAFI USA	ANAFI Ai
Support in anafi_ros	✔	✔

3) ANAFI USA: This model (shown in Fig. 1c) is the rescue-grade drone, featuring 32x zoom and thermal imaging capabilities to meet the demands of first responders and search-and-rescue teams. To achieve this, ANAFI USA is equipped with three front-mounted cameras: a thermal camera, 21 Mpx RGB wide-angle camera (for 1x to 5x zoom) and 21 Mpx RGB telephoto camera (for 5x to 32x zoom), which guarantees a continuous zoom. The 32x zoom allows seeing details as small as 1 cm from a distance of 50 m. The image stabilisation system of ANAFI USA ensures high-quality footage even at 15 m/s wind gust. Despite its compact design, ANAFI USA boasts a 32 min of flight time. ANAFI USA has IP53[b] ingress protection, offering water and dust resistance and making it suitable to fly in rainy conditions. ANAFI USA has a service ceiling of 6 km and can operate in temperatures between -35 °C and $+43$ °C. The body of ANAFI USA is mainly made of polyamide, reinforced with carbon fibre and streamlined using hollow glass beads. The data stored on ANAFI USA or sent through the networks are encrypted, and the drone is protected against malicious software modification attempts.

4) ANAFI Ai: This model (shown in Fig. 1d) is the biggest and heaviest but most advanced of the ANAFI family. Anafi Ai is the first drone to use the 4G cellular network connection, in addition to Wi-Fi, as an alternative encrypted data link between the drone and the remote controller, theoretically enabling control at any distance. Besides a high-resolution 48 Mpx RGB camera with an ISO range of $50 - 6400$, ANAFI Ai is also equipped with a pair of multi-directional stereo cameras, which allow the computation of the occupancy grid to avoid obstacles automatically. Anafi Ai has a 3-axis gimbal differently from other Anafi models with 2-axis gimbals. The maximum horizontal speed of Anafi Ai is 16 m/s, thanks to the optimized aerodynamic performance of the vehicle. Anafi Ai’s 6800 mAh battery allows 32 min of flight time and can be recharged in 150 min. ANAFI Ai has IPX3[c] ingress protection, offering water resistance and making it suitable to fly in rainy conditions. ANAFI Ai can execute custom C++ and Python code onboard, thanks to Parrot’s Air SDK. Air SDK allows loading and running code directly on ANAFI Ai and accessing all sensors, connectivity interfaces and autopilot features. The model of ANAFI Ai is available in the Sphinx simulation environment.

[f] protected from limited dust ingress

B. Remote Controllers

Parrot ANAFI drones come with a handheld remote radio controller called Skycontroller, allowing the user to control the drone and access its various features. Skycontrollers feature a light and compact design with a conventional layout of sticks and buttons. Moreover, they have a built-in battery rechargeable through the USB-C port. The Parrot ANAFI family has two Skycontroller models: *Skycontroller 3* for ANAFI 4K, ANAFI Thermal and ANAFI USA, and *Skycontroller 4* for ANAFI Ai. The two versions of Skycontrollers are illustrated in Fig. 4, while Table II summarises the properties of each model.

1) Skycontroller 3: This model (shown in Fig. 4a) is a remote radio controller designed for Parrot ANAFI 4K, Thermal and USA drones. Skycontroller 3 has a maximum range of up to 4 km.

2) Skycontroller 4: This model (shown in Fig. 4b) is a remote 4G controller designed for Parrot ANAFI Ai. Skycontroller 4 has IP5X[f] ingress protection, offering dust resistance.

(a) Skycontroller 3. (b) Skycontroller 4.

Fig. 4. Parrot Skycontroller series remote controllers.

III. Developed Framework

The developed framework – `anafi_ros` – is a python-based ROS package which enables interfacing with all available Parrot ANAFI quadrocopters. Besides being compatible with all physical drones, `anafi_ros` can connect to virtual drones in Parrot's simulation environment – Sphinx. The developed `anafi_ros` is built on top of Parrot's official python SDK – Olympe – which provides a programming interface for Parrot ANAFI drones. The communication flow in `anafi_ros` allows connecting directly to the drones via Wi-Fi interfaces or through Skycontrollers via USB ports, which is highly recommended. The developed framework makes connecting multiple drones to the same ground station easy by automatically assigning a different virtual IP address to each connected Skycontroller and managing port forwarding. The main functionalities of `anafi_ros` include drone piloting, feedback of flight parameters from onboard sensors, gimbal control, drone state monitoring, video streaming from onboard cameras, picture capturing, video recording, file transferring between onboard storage and ground station, drone calibration and flight plan management.

Remark 2: For the complete list of subscribed and published topics, available services and parameters, please refer to Appendix.

The developed package is organised with several subelements to facilitate the development, as depicted in Fig. 5. In other words, each physical component, such as the drone itself, gimbal, camera, battery, connection link, storage device and remote controller, has a respective software element.

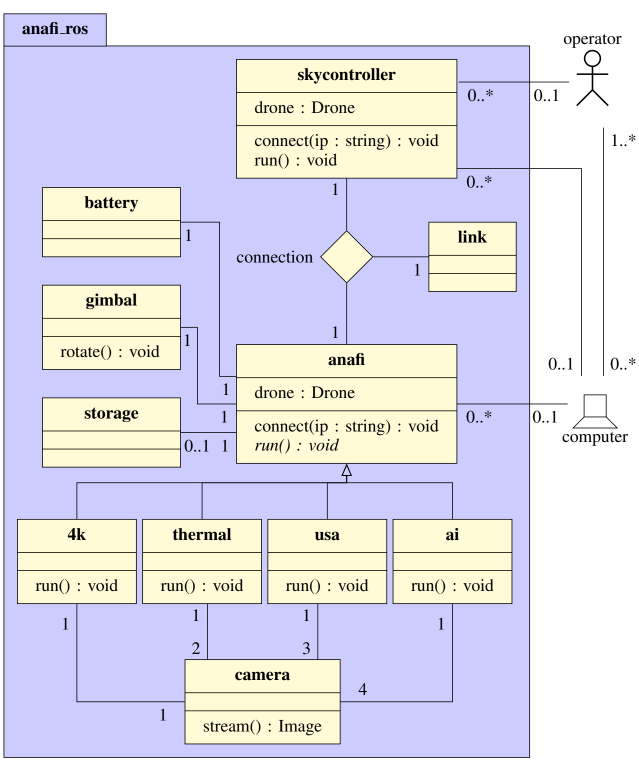


Fig. 5. UML diagram of the structure of `anafi_ros`.

1) Drone: The drone element is a core part of the package, which manages the connection to the drone, enables the control of the drone and provides feedback from the drone. This element allows piloting the drone in three modes: directly commanding values in (12), commanding relative displacements $\begin{bmatrix}\Delta x^* & \Delta y^* & \Delta z^* & \Delta\psi^*\end{bmatrix}^T$ or commanding world references $\begin{bmatrix}\lambda_x^* & \lambda_y^* & z^* & \psi^*\end{bmatrix}^T$, where λ_x^* and λ_y^* are the desired latitude and longitude, respectively. The drone element retrieves and publishes real-time information, such as the drone's attitude, altitude, speed and GPS location. It forwards to the drone the requests for arming, taking-off and landing. Drone class also allows bounding the altitude, distance, horizontal and vertical speed, pitch and roll angles, and attitude rates.

2) Gimbal: The gimbal element provides control and feedback on the camera's gimbal. This element controls the desired pitch ϕ_G^* and roll θ_G^* of the camera. It also provides the actual attitude $\boldsymbol{\theta}_G^D$ of the gimbal and allows for setting the maximum rotational speed of the gimbal.

3) Camera: The camera element provides essential capabilities for the camera, such as changing the zoom level, capturing pictures and recording videos. This element also publishes real-time video stream, camera calibration matrix and actual zoom level. It also allows setting the camera mode, image style and streaming mode and enabling HDR mode.

4) Battery: The battery element provides the battery status, such as the battery's level, health and voltage.

5) Link: The link element provides information on the connection to the drone, such as the link quality, signal strength and connection throughput.

6) Storage: The storage element provides the available memory on the microSD cards if installed. This element also allows downloading media (photos and videos) from the storage device and formatting it.

7) Controller: The controller element provides an alternative to connect to the drone via the remote controller. This element reads and publishes the state of the sticks (gaz/yaw and pitch/roll), triggers (gimbal tilt and camera zoom) and buttons (return to home, centre camera and reset zoom). The remote controller also streams its real-time attitude.

A. Complementary Packages

A complimentary ROS package – `anafi_autonomy`[i] – was developed on top of `anafi_ros` to enable safe navigation of ANAFI drones by adding some high-level capabilities, like position and velocity control. Besides, other open-source ROS packages are available for building the navigation stack for aerial robots, like visual-inertial localisation [15][ii], environment perception [16][iii], motion planning [17][iv], model-based [18][v] and model-free [19][vi] control.

[i] https://github.com/andriyukr/anafi_autonomy
[ii] https://github.com/raulmur/ORB_SLAM2
[iii] https://github.com/robot-perception-group/AirPose
[iv] https://github.com/HKUST-Aerial-Robotics/Fast-Planner
[v] https://github.com/uzh-rpg/data_driven_mpc
[vi] https://github.com/andriyukr/controllers

IV. Experimental Validation

To verify the declared characteristics and validate the developed package, we tested the drone's flight characteristics, gimbal response and camera capabilities separately.

A. Drone

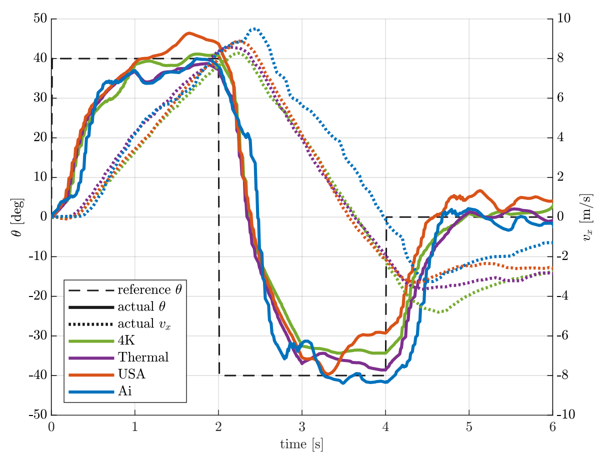
The drones were pushed to their limits by commanding the maximum control inputs to verify the drones' capabilities and validate the developed package. The tests were performed in an open field on a windless day.

For the pitch tracking response, first, the commanded pitch was set to $40°$ for 2 s, then, it was reversed to $-40°$ for 2 s and, finally, set to $0°$. As can be observed from Fig. 6a, all drones were able to achieve the desired pitch while reaching speeds above 8 m/s and stabilise at around $0°$ in the end. ANAFI 4K and Thermal had smoother behaviour, while ANAFI USA had a more aggressive response.

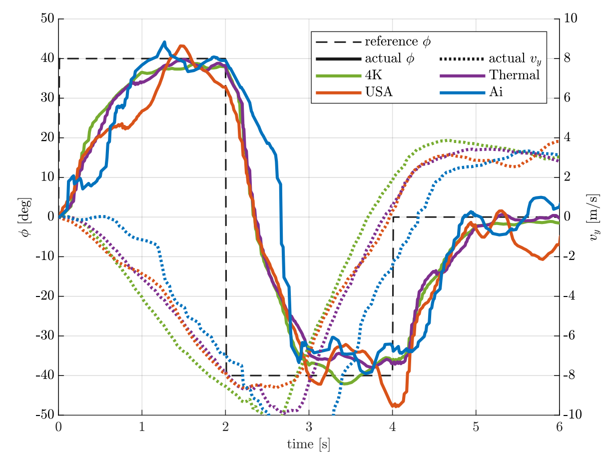
Similarly, for the roll tracking response, first, the commanded roll was set to $40°$ for 2 s, then, it was reversed to $-40°$ for 2 s and, finally, set to $0°$. As can be observed from Fig. 6b, all drones were able to achieve the desired roll while reaching speeds of above 10 m/s and stabilise at around $0°$ in the end. ANAFI 4K and Thermal still had a stable response, while ANAFI USA and Ai had more twitching behaviour.

For the vertical velocity tracking response, first, the commanded velocity was set to 4 m/s for 2 s, then, it was reversed to -4 m/s for 2 s and, finally, set to 0 m/s. As can be observed from Fig. 6c, all drones have an initial delay between 100ms and 200ms but later were able to achieve the desired climbing and descend speeds, reaching the altitude of almost 7 m in 2 s, and stabilise at around 0 m/s in the end. All drones had smooth and stable behaviour.

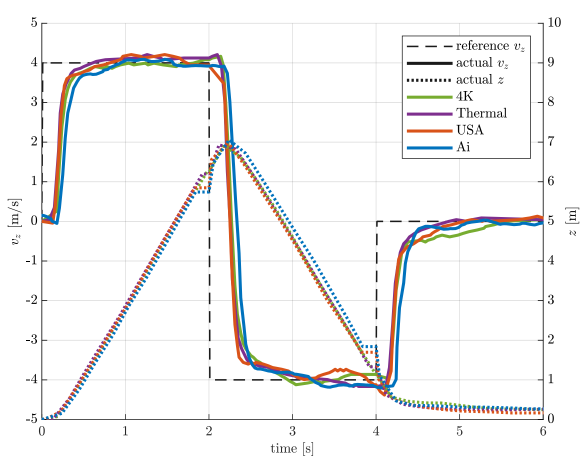
For the yaw rate tracking response, first, the commanded yaw rate was set to $200°/\text{s}$ for 2 s, then, it was reversed to $-200°/\text{s}$ for 2 s and, finally, set to $0°/\text{s}$. As shown in Fig. 6d, all drones were able to achieve the desired yaw rate, making a $360°$ turn in less than 2.5 s, and stabilise at around $0°/\text{s}$ in the end. Similarly, to the velocity tracking, ANAFI 4K, Thermal and USA have a response delay of approximately 100ms; while, for ANAFI Ai, it is approximately 200ms. After the transient phase, all drones had stable spins at the desired yaw rate.



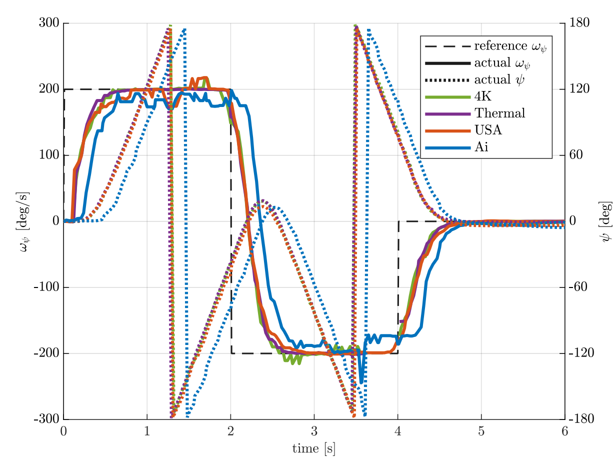
(a) Pitch tracking.



(b) Roll tracking.



(c) Vertical speed tracking.



(d) Yaw rate tracking.

Fig. 6. Piloting response.

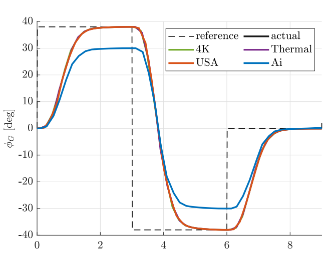
B. Gimbal

All ANAFI drones have an active gimbal on which the main cameras are mounted. The gimbal can adjust its roll and pitch in orientation or angular velocity modes. Fig. 7 shows the responses of the gimbal in two modes for two controlled axis between the gimbal's operational limits. It is possible to observe that ANAFI Ai has the fastest response but a limited range compared to other ANAFI drones.

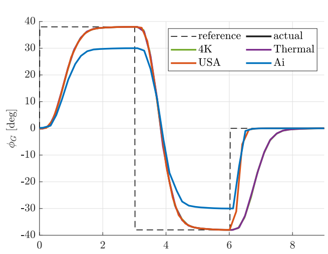
Remark 3: ANAFI 4K, Thermal and USA are equipped with similar gimbals, so their response is almost identical.

C. Camera

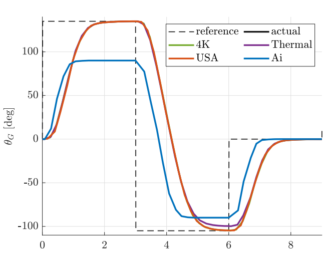
The main difference between ANAFI drones is the set of cameras they are equipped with, as summarized in Fig. 8. The developed package allows switching online between streams from all available cameras. ANAFI 4K has one RGB front-mounted camera, which can stream live 1280×720 px images, shown in Fig. 8a. ANAFI Thermal, besides the same RGB camera as ANAFI 4K, also has a thermal camera, which can stream live 960×720 px images, shown in Fig. 8b. ANAFI USA has three front-mounted cameras: a thermal camera and two RGB wide-angle and telephoto cameras, which can stream high-details images, shown in Fig. 8c, where the road sign highlighted in red in Fig. 8a is zoomed in. ANAFI Ai, besides a high-resolution RGB camera, which can stream live 1920×1080 px images, also has a pair of frontal stereo cameras, which allow the computation of 3D environment information, like the 176×90 px disparity map, shown in Fig. 8d, where the palm leaves in the proximity are detected. Besides streaming live the video feed, all drones can shoot pictures, record videos and store them at maximum resolution on the memory card. In addition, `anafi_ros` allows downloading the stored media from the drone.



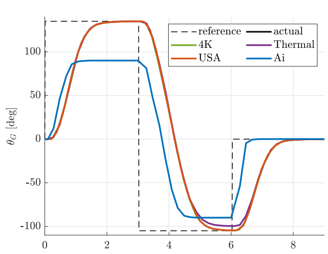
(a) Roll tracking in orientation mode.



(b) Roll tracking in velocity mode.



(c) Pitch tracking in orientation mode.

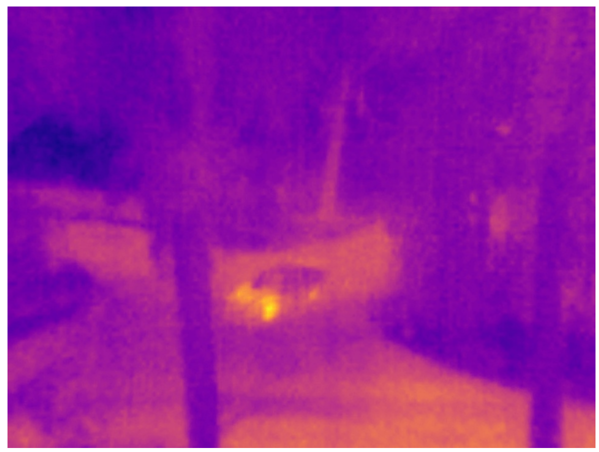


(d) Pitch tracking in velocity mode.

Fig. 7. Gimbal response.

Remark 4: All ANAFI drones also have a down-facing grey-scale global shutter 320×240 px camera for optical flow. However, this video stream is not accessible yet.

(a) RGB image from ANAFI 4K.



(b) Thermal image from the ANAFI Thermal.

(c) 32x zoomed RGB image from ANAFI USA.

(d) Disparity map image from ANAFI Ai.

Fig. 8. Camera features of different Anafi drones.

V. Conclusions

This work introduces a ROS1 and ROS2 package – `anafi_ros` – for simple interfacing with the drones from the Parrot ANAFI family. The developed ROS package is hardware agnostic, allowing connecting seamlessly to four supported models. The developed package was intensively tested on the drones at maximum roll and pitch angles of $\pm 40°$, corresponding to the horizontal speeds above ± 10 m/s, maximum vertical speed of ± 4 m/s and maximum yaw rate of $\pm 200°$/s. All drone models demonstrated satisfactory performance and stable response. We hope the developed framework will provide new opportunities for further applications of aerial robots.

Appendix

A. Subscribed topics

<topic name> (`<message type>`): <topic description>

- **camera/command** (`CameraCommand`): camera zoom commands
- **drone/command** (`PilotingCommand`): drone piloting commands
- **drone/moveby** (`MoveByCommand`): move the drone by the given displacement and rotate by the given angle
- **drone/moveto** (`MoveToCommand`): move the drone to the specified location
- **gimbal/command** (`GimbalCommand`): gimbal attitude commands

B. Published topics

<topic name> (`<message type>`, < frequency>) ∈ {`<set of values>`} / [`<range of values>`]: <topic description> [<measurement units>]

- **battery/health** (`UInt8`, 1 Hz) ∈ [`0`: bad, `100`: good]: battery health [%]
- **battery/percentage** (`UInt8`, 30 Hz) ∈ [`0`: empty, `100`: full]: battery level [%]
- **battery/voltage** (`Float32`, 1 Hz): battery voltage [V]
- **camera/awb_b_gain** (`Float32`, 30 Hz): camera automatic white balance (AWB) blue gain
- **camera/awb_r_gain** (`Float32`, 30 Hz): camera automatic white balance (AWB) red gain
- **camera/camera_info** (`CameraInfo`, 30 Hz): main camera's info
- **camera/exposure_time** (`Float32`, 30 Hz): exposure time of the main camera [s]
- **camera/image** (`Image`, 30 Hz): image from the main front camera
- **camera/hfov** (`Float32`, 30 Hz): camera's horizontal field of view [°]
- **camera/iso_gain** (`UInt16`, 30 Hz): camera's sensitivity gain
- **camera/vfov** (`Float32`, 30 Hz): camera's vertical field of view [°]
- **camera/zoom** (`Float32`, 5 Hz): camera zoom level [x]
- **drone/altitude** (`Float32`, 30 Hz) > `0.0`: drone's ground distance [m]
- **drone/altitude_above_to** (`Float32`, 5 Hz): drone's ground distance above the take-off point [m]
- **drone/attitude** (`QuaternionStamped`, 30 Hz): drone's attitude in north-west-up frame
- **drone/gps/fix** (`Bool`, 1 Hz) ∈ {`true`: GPS is fixed, `false`: GPS is not fixed}
- **drone/gps/location** (`NavSatFix`, 1 Hz): drone's GPS location
- **drone/gps/satellites** (`UInt8`): number of GPS satellites
- **drone/rpy** (`Vector3Stamped`, 30 Hz): drone's roll, pitch and yaw in north-west-up frame [°]
- **drone/speed** (`Vector3Stamped`, 30 Hz): drone's speed in body frame [m/s]
- **drone/state** (`String`, 30 Hz) ∈ {'CONNECTING', 'LANDED', 'TAKINGOFF', 'HOVERING', 'FLYING', 'LANDING', 'EMERGENCY', 'DISCONNECTED', ...}: drone's state
- **gimbal/attitude/absolute** (`QuaternionStamped`, 5 Hz): gimbal's attitude in north-west-up frame
- **home/location** (`PointStamped`): home location
- **link/quality** (`UInt8`, 30 Hz) ∈ [`0`: bad, `5`: good]: link quality
- **skycontroller/attitude** (`QuaternionStamped`, 20 Hz): SkyController's attitude in north-west-up frame
- **skycontroller/command** (`SkycontrollerCommand`, 100 Hz): command from SkyController
- **skycontroller/rpy** (`Vector3Stamped`, 20 Hz): SkyController's attitude in north-west-up frame [°]
- **storage/available** (`UInt64`): available storage space [B]
- **time** (`Time`, 30 Hz): drone's local time

C. Services

<service name> (`<service type>`): <service description>

- **camera/photo/stop** (`Photo`): stop photo capture
- **camera/photo/take** (`Photo`): take a photo
- **camera/recording/start** (`Recording`): start video recording
- **camera/recording/stop** (`Recording`): stop video recording
- **camera/reset** (`Trigger`): reset zoom level
- **drone/arm** (`SetBool`): {`true`: arm the drone; `false`: disarm the drone}
- **drone/calibrate** (`Trigger`): start drone's magnetometer calibration process
- **drone/emergency** (`Trigger`): cut out the motors
- **drone/halt** (`Trigger`): halt and start hovering
- **drone/land** (`Trigger`): take-off the drone
- **drone/reboot** (`Trigger`): reboot the drone
- **drone/rth** (`Trigger`): return home
- **drone/takeoff** (`Trigger`): land the drone
- **flightplan/pause** (`Trigger`): pause the flight plan

- **flightplan/start** (`FlightPlan`): start the flight plan based on the Mavlink file existing on the drone
- **flightplan/stop** (`Trigger`): stop the flight plan
- **flightplan/upload** (`FlightPlan`): upload the Mavlink file to the drone
- **gimbal/calibrate** (`Trigger`): start gimbal calibration
- **gimbal/reset** (`Trigger`): reset the reference orientation of the gimbal
- **home/navigate** (`SetBool`): {`true`: start return home; `false`: stop return home} trigger navigate home
- **home/set** (`Location`): set the custom home location
- **skycontroller/offboard** (`SetBool`): {`true`: switch to offboard control; `false`: switch to manual control} change control mode
- **storage/download** (`SetBool`): {`true`: delete media after download; `false`: otherwise} download media from the drone
- **storage/format** (`Trigger`): format removable storage

D. Parameters

<parameter name> (<parameter type>) := <default value> ∈ {<set of values>} / [<range of values>]: <parameter description> [<measurement units>]

- **camera/autorecord** (`bool`) := `false` ∈ {`true`: enabled; `false`: disabled}: auto record at take-off
- **camera/ev_compensation** (`int`) := `9` ∈ {`0`: −3.00; `3`: −2.00; `6`: −1.00; `9`: 0.00; `12`: 1.00; `15`: 2.00; `18`: 3.00}: camera exposure compensation [EV]
- **camera/hdr** (`bool`) := `true` ∈ {`true`: enabled; `false`: disabled}: high dynamic range (HDR) mode
- **camera/max_zoom_speed** (`float`) := `10.0` ∈ [`0.1`, `10.0`]: maximum zoom speed [tan(°) /s]
- **camera/mode** (`int`) := `0` ∈ {`0`: camera in recording mode; `1`: camera in photo mode}: camera mode
- **camera/relative** (`bool`) := `false` ∈ {`true`: commands relative to the camera pitch; `false`: otherwise}
- **camera/rendering** (`int`) := `0` ∈ {`0`: visible; `1`: thermal; `2`: blended}: thermal image rendering mode (`1` and `2` supported only by ANAFI Thermal and ANAFI USA)
- **camera/streaming** (`int`) := `0` ∈ {`0`: minimize latency with average reliability (best for piloting); `1`: maximize reliability with an average latency; `2`: maximize reliability using a frame-rate decimation}: streaming mode
- **camera/style** (`int`) := `0` ∈ {`0`: natural look; `1`: flat and desaturated images, best for post-processing; `2`: intense – bright colors, warm shade, high contrast; `3`: pastel – soft colors, cold shade, low contrast}: images style
- **drone/banked_turn** (`bool`) := `true` ∈ {`true`: enabled; `false`: disabled}: banked turn
- **drone/max_altitude** (`float`) := `2.0` ∈ [`0.5`, `4000.0`]: maximum altitude [m]
- **drone/max_distance** (`float`) := `10.0` ∈ [`10.0`, `4000.0`]: maximum distance [m]
- **drone/max_horizontal_speed** (`float`) := `1.0` ∈ [`0.1`, `15.0`]: maximum horizontal speed [m/s]
- **drone/max_pitch_roll** (`float`) := `10.0` ∈ [`1.0`, `40.0`]: maximum pitch and roll angle [°]
- **drone/max_pitch_roll_rate** (`float`) := `200.0` ∈ [`40.0`, `300.0`]: maximum pitch and roll rotation speed [°/s]
- **drone/max_vertical_speed** (`float`) := `1.0` ∈ [`0.1`, `4.0`]: maximum vertical speed [m/s]
- **drone/max_yaw_rate** (`float`) := `180.0` ∈ [`3.0`, `200.0`]: maximum yaw rotation speed [°/s]
- **drone/model** (`string`) := ∈ {‘4k’, ‘thermal’, ‘usa’, ‘ai’, ‘unknown’}: drone’s model
- **gimbal/max_speed** (`float`) := `180.0` ∈ [`1.0`, `180.0`]: maximum gimbal speed [°/s]
- **home/autotrigger** (`bool`) := `true` ∈ {`true`: enabled; `false`: disabled}: auto trigger return-to-home
- **home/ending_behavior** (`int`) := `1` ∈ {`0`: land; `1`: hover}: return-to-home ending behavior
- **home/min_altitude** (`float`) := `20.0` ∈ [`20.0`, `100.0`]: return-to-home minimum altitude [m]
- **home/precise** (`bool`) := `true` ∈ {`true`: enabled; `false`: disabled}: precise return-to-home
- **home/type** (`int`) := `4` ∈ {`1`: take-off location; `3`: user-set custom location; `4`: pilot location}: home type for return-to-home
- **storage/download_folder** (`string`) := “ ∼/Pictures/Anafi”: path to the download folder

E. Custom messages

- **`CameraCommand`**
  - `Header` **header**: header of the message
  - `uint8` **mode** ∈ {`0`: level; `1`: velocity}: control mode
  - `float32` **zoom**: zoom command [x] / [x/s]
- **`GimbalCommand`**
  - `Header` **header**: header of the message
  - `uint8` **mode** ∈ {`0`: position; `1`: velocity}: control mode
  - `uint8` **frame** ∈ {`0`: none; `1`: relative; `2`: absolute}: gimbal’s frame of reference
  - `float32` **roll**: roll command [°] / [°/s]
  - `float32` **pitch**: pitch command [°] / [°/s]
  - `float32` **yaw**: pitch command [°] / [°/s]
- **`MoveByCommand`**
  - `Header` **header**: header of the message
  - `float32` **dx**: x displacement [m]
  - `float32` **dy**: y displacement [m]
  - `float32` **dz**: z displacement [m]
  - `float32` **dyaw**: yaw displacement [°]
- **`MoveToCommand`**
  - `Header` **header**: header of the message
  - `float64` **latitude**: latitude [°]
  - `float64` **longitude**: longitude [°]
  - `float64` **altitude**: altitude [m]
  - `float32` **heading**: heading w.r.t. North [°]
  - `uint8` **orientation_mode** ∈ {`0`: none; `1`: to target; `2`: heading start; `3`: heading during}: orientation mode

- **`PilotingCommand`**
  - `Header` **header**: header of the message
  - `float32` **roll**: roll angle [°]
  - `float32` **pitch**: pitch angle [°]
  - `float32` **yaw**: yaw rate [°/s]
  - `float32` **gaz**: vertical velocity [m/s]
- **`SkycontrollerCommand`**
  - `Header` **header**: header of the message
  - `int8` **x** ∈ [`-100`, `100`]: x-axis [%]
  - `int8` **y** ∈ [`-100`, `100`]: y-axis [%]
  - `int8` **z** ∈ [`-100`, `100`]: z-axis [%]
  - `int8` **yaw** ∈ [`-100`, `100`]: yaw-axis [%]
  - `int8` **camera** ∈ [`-100`, `100`]: camera-axis [%]
  - `int8` **zoom** ∈ [`-100`, `100`]: zoom-axis [%]
  - `bool` **return_home** ∈ {`true`: pressed; `false`: not pressed}: return-to-home (front top) button
  - `bool` **takeoff_land** ∈ {`true`: pressed; `false`: not pressed}: take-off/land (front bottom) button
  - `bool` **reset_camera** ∈ {`true`: pressed; `false`: not pressed}: reset camera (back left) button
  - `bool` **reset_zoom** ∈ {`true`: pressed; `false`: not pressed}: reset zoom (back right) button

F. *Custom services*

- **`FlightPlan`**
  - `string` **file**: path to the flight plan file on local computer
  - `string` **uid**: flight plan UID in drone's directory
- **`Location`**
  - `float64` **latitude**: latitude [°]
  - `float64` **longitude**: longitude [°]
  - `float64` **altitude**: altitude [m]
- **`PilotedPOI`**
  - `float64` **latitude**: latitude to look at [°]
  - `float64` **longitude**: longitude to look at [°]
  - `float64` **altitude**: altitude to look at [m]
  - `bool` **locked_gimbal** ∈ {`true`: gimbal is locked on the point of interest, `false`: gimbal is freely controllable}: gimbal is locked
- **`Photo`** → `string` **media_id**: media id
  - `uint8` **mode** ∈ {`0`: single shot; `1`: bracketing – burst of frames with a different exposure; `2`: burst of frames; `3`: time-lapse – frames at a regular time interval; `4`: GPS-lapse – frames at a regular GPS position interval}: photo mode
  - `uint8` **photo_format** ∈ {`0`: full resolution, not dewarped; `1`: rectilinear projection, dewarped}: photo format
  - `uint8` **file_format** ∈ {`0`: jpeg; `1`: dng; `2`: jpeg and dng}: file format
- **`Recording`** → `string` **media_id**: media id
  - `uint8` **mode** ∈ {`0`: standard; `1`: hyperlapse; `2`: slow motion; `3`: high-framerate}: video recording mode

Acknowledgment

The author thanks the support team of Parrot for their assistance.

References

[1] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, and K. Alexis, "CERBERUS in the DARPA Subterranean Challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[2] A. Suarez, R. Salmoral, A. Garofano-Soldado, G. Heredia, and A. Ollero, "Aerial Device Delivery for Power Line Inspection and Maintenance," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 30–38.

[3] K. Zhang, P. Chermprayong, F. Xiao, D. Tzoumanikas, B. Dams, S. Kay, B. Kocer, A. Burns, L. Orr, C. Choi, D. Darekar, W. Li, S. Hirschmann, V. Soana, S. Ngah, S. Sareh, A. Choubey, L. Margheri, V. Pawar, and M. Kovac, "Aerial additive manufacturing with multiple autonomous robots," *Nature*, vol. 609, pp. 709–717, 09 2022.

[4] M. M. Valero, S. Verstockt, B. Butler, D. Jimenez, O. Rios, C. Mata, L. Queen, E. Pastor, and E. Planas, "Thermal Infrared Video Stabilization for Aerial Monitoring of Active Wildfires," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2817–2832, 2021.

[5] L. Chen, Z. Fang, and Y. Fu, "Consistency-Aware Map Generation at Multiple Zoom Levels Using Aerial Image," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 5953–5966, 2022.

[6] H. X. Pham, A. Sarabakha, M. Odnoshyvkin, and E. Kayacan, "PencilNet: Zero-Shot Sim-to-Real Transfer Learning for Robust Gate Perception in Autonomous Drone Racing," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 847–11 854, 2022.

[7] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, p. eabl6259, 2022.

[8] "DJI Developer," 2023. [Online]. Available: https://developer.dji.com

[9] "Tello-Python," 2019. [Online]. Available: https://github.com/dji-sdk/Tello-Python

[10] "Parrot Drone SDK," 2023. [Online]. Available: https://developer.parrot.com/docs/index.html

[11] "SDL Bitcraze," 2023. [Online]. Available: https://www.bitcraze.io/documentation/repository/aideck-gap8-examples/master/getting-started/sdk/

[12] "bebop_autonomy," 2018. [Online]. Available: https://github.com/AutonomyLab/bebop_autonomy

[13] "tello_ros," 2022. [Online]. Available: https://github.com/clydemcqueen/tello_ros

[14] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3299–3304.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[16] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bülthoff, M. J. Black, and A. Ahmad, "Active Perception Based Formation Control for Multiple Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4491–4498, 2019.

[17] B. Zhou, J. Pan, F. Gao, and S. Shen, "RAPTOR: Robust and Perception-Aware Trajectory Replanning for Quadrotor Fast Flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.

[18] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-Driven MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[19] A. Sarabakha and E. Kayacan, "Online Deep Fuzzy Learning for Control of Nonlinear Systems Using Expert Knowledge," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 7, pp. 1492–1503, 2020.

[20] A. Sarabakha and E. Kayacan, "Y6 Tricopter Autonomous Evacuation in an Indoor Environment Using Q-Learning Algorithm," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 5992–5997.

[21] V. Mistler, A. Benallegue, and N. M'Sirdi, "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback," in *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*, 2001, pp. 586–593.