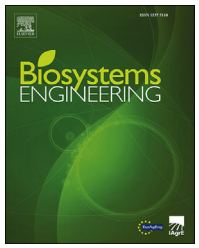
Cooperation of unmanned systems for agricultural applications: A case study in a vineyard

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110



Special Issue: Biosystems and Metrology

Research Paper

Cooperation of unmanned systems for agricultural applications: A case study in a vineyard

Martina Mammarella [a], *Lorenzo Comba* [b,a,*], *Alessandro Biglia* [b], *Fabrizio Dabbene* [a], *Paolo Gay* [b]

[a] *Institute of Electronics, Computer and Telecommunication Engineering – National Research Council of Italy– Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy*

[b] *Department of Agricultural, Forest and Food Sciences (DiSAFA) – Università degli Studi di Torino, Largo Paolo Braccini 2, 10095 Grugliasco (TO), Italy*

ARTICLE INFO

Article history:
Published online 31 December 2021

Keywords:
Precision farming
Unmanned vehicles cooperation
Unmanned aerial vehicles
Unmanned ground vehicles
Control logic
Path and mission planning

Fully-autonomous vehicles, both aerial and ground, could provide great benefits in the Agriculture 4.0 framework when operating within cooperative architectures, thanks to their ability to tackle difficult tasks, particularly within complex irregular and unstructured scenarios such as vineyards on sloped terrains. A decentralised multi-phase approach has been proposed as an alternative to more common cooperative schemes. When perennial crops are considered, it is advantageous to build a simplified geometrical (and georeferenced) crops model, which can be identified by using 3D point clouds acquired during a-priori explorative missions by unmanned aerial vehicles. This model can be used to plan the tasks to be performed within the crops by the in-field aerial and ground drones. In this companion paper, the proposed strategy is applied to a specific case study involving a vineyard on a sloped terrain, located in the Barolo region in Piedmont, Italy. Ad-hoc technologies and guidance, navigation and control algorithms were designed and implemented. The main objectives were to improve the autonomous driving capabilities of the drones involved and to automate the process of retrieving low-complexity maps from the data collected with preliminary remote sensing missions to make them available for the autonomous navigation by a quadrotor and an unmanned 4-wheel steering ground vehicle within the vine rows. Preliminary results highlight the benefits achievable by exploiting the tailored technologies selected and applied to improve each of the analysed mission phases.

1. Introduction

In the Agriculture 4.0 framework, great benefits can be achieved by allowing cooperation and collaborative action among unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs). Autonomous agricultural machines can not only lead to improvements in timeliness, but they can also represent valid alternatives to conventional machines, particularly when

* *Corresponding author*. Department of Agricultural, Forest and Food Sciences (DiSAFA) – Università degli Studi di Torino, Largo Paolo Braccini 2, 10095, Grugliasco (TO), Italy.
E-mail address: lorenzo.comba@unito.it (L. Comba).
https://doi.org/10.1016/j.biosystemseng.2021.12.010

Nomenclature

Abbreviations

ASM	Ackerman steering mechanism
CoM	Centre of mass
CTE	Cross track error
DWA	Dynamic window approach
FMU	Flight management unit
FW	Fixed-wing
GIS	Geographic information system
GNC	Guidance, navigation and control
GSD	Ground sample distance
ICR	Instantaneous centre of rotation
IMU	Inertial measurement unit
LMI	Linear matrix inequality
mRPI	Minimal robust positive invariant set
MPC	Model predictive control
RRT	Rapidly exploring random tree
RW	Rotary-wing
SFM	Structure from motion
SIL	Software in the loop
TRMPC	Tube based robust model predictive control
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
UV	Unmanned vehicle
WP	Waypoint
2WS	Two-wheels steering
4WS	Four-wheels steering

Symbols

A_K	Closed-loop
d	Distance among aircraft and i-th waypoint
d_e	Lateral deviation from UGV CoM and reference path
d_i	Distance among i-th node and final goal
d_{goal}	Distance among current node and final goal
d_O	Distance between sub-trajectory final node nearest obstacle
$d(t)$	Relative distance between the aircraft and the North-East coordinates
e_1, g_1	Vine rows end points
$e_{i\|k}$	Error deviation among actual and nominal state at time $k+i$
E_i	East coordinate of the i-th waypoint
E_{ref}	East coordinate of the reference point
E_{UAV}	East coordinate of the UAV current position
$F_{k-1,k}$	Triangular faces of the mesh between vertices V_{k-1} and V_k
F_z^B	Total vertical thrust
h	Altitude
$[H_N, h_N]$	Terminal constraints defined as linear inequalities
$[H_u, h_u]$	Input constraints defined as linear inequalities
$[H_x, h_x]$	State constraints defined as linear inequalities
h_{ref}	Reference altitude
J	Cost function
J_v	Velocity optimiser cost function
J_∞	Infinite horizon quadratic cost
κ	Curvature
K_s	Proportional gain for steering control
K_θ	Proportional gain for curvature control
$\mathcal{M}_k$	Triangular mesh
N	Prediction horizon
N_i	North coordinate of the i-th waypoint
N_{ref}	North coordinate of the reference point
N_{UAV}	North coordinate of the UAV current position
r_{ij}	Relative distance among ij-th wheel and the instantaneous ICR
$S_K(\infty)$	Minimal robust positive invariant set
u_k	Control input at time k
$u_{i\|k}$	Predicted values of the model input based on time k
$\mathbb{U}$	Input constraint set
v_F	Front velocity
v_{ij}	Linear velocity of the ij-wheel
v_R	Rear velocity
V_k	k-th vertex of the mesh
V_{ref}	Reference airspeed
$\mathbb{V}$	Nominal input constraint set
(x_{goal}, y_{goal})	Coordinates of the final goal
(x_G, y_G)	Coordinates of the UGV CoM with respect to the global frame
$x_{i\|k}$	Predicted values of the model state based on time k
x_k	State control at time k
$(x_{ref}, y_{ref}, \psi_{ref})$	Reference position coordinates and heading angle
$\mathbb{X}$	State constraint set
$\mathbb{X}_N$	Terminal constraint set
w_k	Persistent random disturbance at time k
$\mathbb{W}$	Disturbance set
$z_{i\|k}$	Nominal state at time k
$\mathbb{Z}$	Nominal state constraint set
α, β	Weight matrices
γ	Steering angle
γ_c	Control steering angle
δ_a	Aileron deflection
δ_e	Elevator deflection
δ_F	Front steering angle
$\delta_{F_{des}}$	Reference front steering angle
δ_R	Rear steering angle
$\delta_{R_{des}}$	Reference rear steering angle
θ	UGV heading angle
θ_e	Angular deviation of the UGV heading angle
ε	Slippage tolerance
ε_{max}	Maximum CTE performance index
ε_r	CTE performance index
ΔT	Throttle
ψ	Heading angle
ψ_{ref}	Reference heading angle
ψ_{UAV}	UAV heading angle
Ω_{ij}	Angular velocity of the ij-th wheel around the ICR
$[\varphi\ \vartheta\ \psi]^\top$	Euler angles (roll, pitch and yaw)
$\tau = [\tau_\varphi, \tau_\theta, \tau_\psi]$	Control torque components defined in the Body frame

operated within irregular and unstructured scenarios, such as some agricultural environments (Mammarella et al., 2020a). Indeed, besides their individual potential, UGVs and UAVs can tackle more difficult tasks by adopting operative schemes based on their cooperation. As discussed in the companion paper (Mammarella et al., 2022), the proposed cooperative scheme involves heterogeneous autonomous vehicles operated within a complex and unstructured scenario, such as vineyards on sloped terrains. The proposed cooperative solution goes beyond the classic standard scheme of employing parallel tasks. Indeed, it is based on a so-called multi-phase approach, where each unmanned vehicle (UV) agent is assigned a specific task whose successful completion is dependent, and at the same time instrumental, to the other agents' tasks in order to complete the global result in a precise and time-effective way. In the proposed cooperative framework, different unmanned aerial and ground vehicles are envisioned to perform a combination of remote sensing and in-field operations to map the selected area and later provide biopesticide distribution via heterogeneous autonomous machineries. The main aim is to fully automate not only the in-field operations performed by the UVs, by providing them with autonomous navigation capabilities, but also the preparatory phase.

In this framework, the exploitation of UVs within an agricultural scenario is optimised on three main levels: i) UV design; ii) mission planning; and iii) autonomous navigation software. The paper highlights how cooperation among drones could determine their successful operability as a complement, or may even represent a viable alternative to conventional machines. In particular, the main focus of this paper can be split into the following sub-tasks:

- *Offline definition of the optimal trajectory*, with the aim of reducing the computational demands required on board. This task is typically performed in two successive steps. The first consists of locating the reference waypoints by exploiting commercial software. Then, proper guidance algorithms are designed to enforce drone dynamics in order to obtain a feasible trajectory for the vehicle to follow, possibly coupled *online* with proper obstacle avoidance strategies.
- *Robust optimal control*, with the aim of obtaining controllers which are implementable on-board the different-but-similar UVs in the same fleet, despite the inherent differences due to the manufacturing process, thereby reducing the design phase. Moreover, robust controllers allow the tackling of external disturbance sources, such as wind turbulence or gusts, which could affect vehicle performance.
- *Low-complexity 3D maps*, to provide the required spatial description of the environment in which the drones are going to operate, reducing the size of required memory, without significant loss of relevant crop shape information. These maps allow georeferencing during in-field operations and computational compatibility with the UVs onboard computer.
- *Design of 4 wheels-steering (4WS) UGVs* to comply with terrain's constraints, in terms of: i) reduced turning radius; ii) improved steering flexibility (at low speed); iii) reduced vehicle side-slip angle, yaw rate and heeling angle; and iv) improved handling stability (at increasing velocities).
- *Cooperative definition of the optimal path for in-field operations* of both UGVs and rotary-wing (RW) UAVs to optimise the offline phase of the trajectory's design while tailoring the online control strategies according to the features and constraints of the vehicles.

As described in Mammarella et al. (2022), the multi-phase collaborative approach is split into three main layers: i) the *remote sensing phase*, performed by a (or a fleet of) fixed-wing (FW) UAV to collect information and aerial imagery from the selected field via onboard sensors and cameras; ii) the automatic *semantic interpretation of point cloud maps* to construct a low-complexity and computationally compatible georeferenced 3D map; and iii) *in-field operations* via (a fleet of) fully-autonomous UGVs and RW-UAVs, to perform complex tasks such as, e.g., scouting, spraying and shredding.

The goal of this paper is twofold: firstly, to demonstrate the effectiveness of the proposed multi-phase cooperative approach introduced in Mammarella et al. (2022); secondly, to validate the efficacy of the combination of algorithms and ad-hoc technologies designed and implemented for the autonomous driving of agricultural UVs by illustrating their application to a Barolo vineyard in the North of Italy, cultivated with the Nebbiolo vine variety. The selected scenario involves four parcels (see Fig. 1), for a total area of about 4 ha. The growing method is a vertical shoot position trellis system. In particular, Parcel F ([44.6150°–44.6157°] Lat, [7.9244°–7.9262°] Lon, Italy) was used below as a validation scenario.[1] In terms of technology and algorithms, an accurate selection among those currently available in the literature for the autonomous navigation of vehicles (not only in the agricultural field) and introduced in Mammarella et al. (2022) was performed. This selection process aimed at identifying the best combination of guidance, navigation and control (GNC) algorithms which could provide suitable performances in terms of low computational demands, reduced design time, and optimal tracking capabilities for fully-autonomous UVs, exploited in the innovative cooperative architecture proposed here. In particular, the proposed approach is based on a combination of existing algorithms, exploited here for the first time in a precision agriculture scenario, and ad-hoc design schemes, designed to comply with the peculiar features of the environment and the vehicles themselves, and here applied within the selected operative scenario to assess the achievable benefits and improvements compared to other schemes already available in the literature.

As anticipated, the proposed scenario envisions the cooperation between a FW-UAV dedicated to remote sensing tasks, which collects aerial imagery later automatically processed to obtain low-complexity, georeferenced 3D maps of the area of interest, and a fleet of UGVs and RW-UAVs for UVs-assisted in-field operations, e.g. scouting. Indeed, UGVs and multi-rotor UAVs can benefit from the former's (i.e. FW-UAV) essential and valuable information related to the crop and

[1] The presented results are limited to a single parcel due only to space limitations and for clarity. Indeed, the same method can easily be extended and applied to all the other parcels in the selected vineyard, as well as to different scenarios.

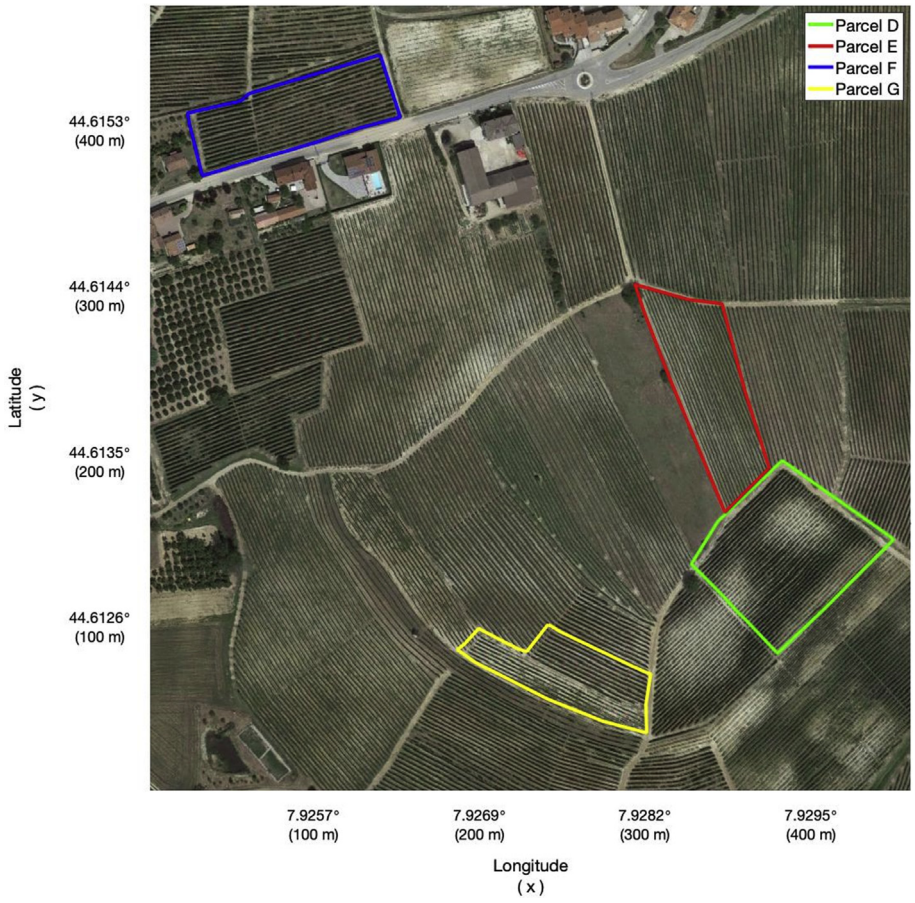


Fig. 1 – The selected scenario with identified parcels of interest (credit: Google Earth).

working environment. Moreover, for in-field operations, the heterogeneous cooperation among UAVs and UGVs makes it possible to benefit from each other's strengths, and to compensate for each other's weaknesses. Indeed, in the specific case considered, the load that aerial vehicles can carry is limited compared to ground-based vehicles. On the other hand, UGVs often have limited mobility compared to UAVs.

Within the proposed scenario, i.e. in-field UVs used for crop monitoring and management operations, the mission of heterogeneous systems of drones which are called to operate and cooperate, allows the vehicles to automatically follow the inter-row paths and, thus, the crop canopy. An effective example of such an application is spraying, where the UAVs and UGVs work within the vine rows simultaneously, in order to properly and efficiently distribute biopesticides on the crops, minimising spray drift to reduce wasted chemicals and to lower costs, via dedicated spraying systems.

It is important to highlight that the proposed multi-phase approach considers a scaled-down scenario and low-cost vehicles, since the goal is to validate the cooperative scheme (as well as the technology and algorithms therein) and to demonstrate its efficacy. Future large-scale validations may possibly envision larger commercial vehicles, compliant with the demands of the applications themselves.

In particular, to acquire the aerial imagery for the generation of the high-density 3D point clouds of the selected crops, the remotely piloted MH900 aircraft system, designed by

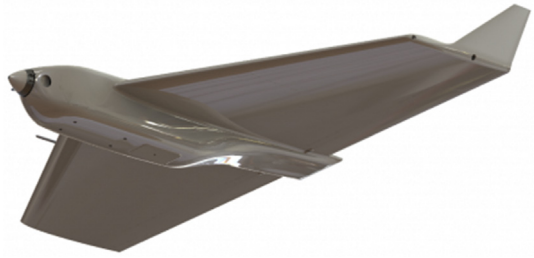


Fig. 2 – Rendering of the MH900 FW-UAV (credit:MAVTech).

MAVTech S.r.l., was considered. The MH900,[2] represented in Fig. 2, is a fixed-wing, tailless, integrated wing-body UAV, which guarantees an adequate aerodynamic efficiency providing a payload capability of up to 250 g, with good flight performances, such as mission range between 250 and 5000 m and an endurance of about 30 min. The cruise airspeed of the MH900 ranges from 12 m s^{-1} up to 15 m s^{-1} and it is able to tolerate winds of up to 11.1 m s^{-1} It has a wingspan of 900 mm and a weight of about 1.2 kg, including 250 g of payload. To

[2] The MH900 was developed by MAVTech Srl located in Bolzano (Italy) as a Technology Company of NOI Techpark Südtirol/Alto Adige. More details can be found at https://www.mavtech.eu/it/prodotti/mh900/.

Fig. 3 – The 4WS UGV designed and developed by the Department of Agricultural, Forest and Food Sciences (DiSAFA) of University of Turin, Italy.

perform the remote sensing mission, the MH900 was equipped with a Micasense© RedEdge multispectral camera (Micasense, 2015), for which the ground sampling distance (GSD), i.e. the distance between the pixel centers measured on the ground,[3] can be tailored according to the chosen flight altitude and overlapping specifications, as reported in Table II in Mascarello et al. (2017). A Pixhawk 4 autopilot[4] flight control system, which provides automatic stability and GNC capabilities, was installed on the vehicle and allows the evaluation of flight parameters either in real time or in post-flight mode. It is important to highlight that the MH900 represents only *one* of the possible commercial solutions available on the market in the low-cost mini-UAV category and able to provide the performance required for remote sensing missions. In this paper, this vehicle was preferred for its flexibility in terms of reconfigurability and for research purposes.

For the in-field operations performed by ground vehicles, the four-wheel steering electric UGV (see Fig. 3), developed by the Department of Agricultural, Forest and Food Sciences of the Università di Torino, was considered. The choice of exploiting a research robot instead of a commercial one was due to the lack of availability of commercial 4WS robots in the agricultural market. Although, there exist several fully-autonomous commercial vehicles that provide not only autonomous navigation capabilities that are also able to perform different in-field tasks within the crops such as the Kongskilde Robotti UGV,[5] the Greenbot,[6] the Moose[7] and the Husky[8] UGVs by ClearPathRobotics, the Dood UGV by Earthautomation,[9] and the OZ robot by Naio Technologies[10], none of them presents an independent steering scheme, which increases the manoeuvrability of the vehicle on tough terrains.

The DiSAFA 4WS-UGV has an intelligent traction system, with an Ackermann steering mechanism (ASM) on both the front and rear axles, plus an independent electric motor for each wheel. This enhances the traction performance (efficiency and mobility) on wet soils, and extends the battery life, which guarantees work sessions up to 8 h. The vehicle is 1.5 m long and 1 m wide with the front wheels mounted 0.75 m from the vehicle's centre of gravity and a steering range of 23°. The maximum speed of the wheel motors is around 8 rad s^{-1} whereas the maximum steering axis velocity is close to 2 rad s^{-1}. For autonomous navigation, the 4WS-UGV was equipped with: i) a Novatel OEM7600 GPS receiver; ii) a XSens MTI-10 series inertial measurement unit (IMU), consisting of accelerometer, gyroscope and magnetometer; and iii) four HC-SR04 ultrasonic sensors, two on each side of the UGV.

Last, the Q4T drone, developed by MAVTech s.r.l., was selected to operate within the vine rows (see Fig. 4). This quadrotor UAV is capable of carrying a widespread of sensors on board, thanks to its high flexibility. The available endurance can be modulated according to the payload/batteries ratio. It is very easy to access and to re-configure and it is equipped with a Pixhawk 2 autopilot. With a diagonal wheelbase of 620 mm, the Q4T has a net weight of around 2 kg, allowing it to reach a maximum of 5 kg of take-off weight. It is equipped with four T-Motor U5 V2.0 motors, each one providing around 2.5 kg of thrust when operated at 22.2 V, whereas the battery pack is housed inside a sliding case, inserted into the center of the main frame to properly guarantee the UAV's balance. In terms of sensors, the RW-UAV is equipped with: i) a GPS RTK navigation system for

[3] In an image with a 1-m GSD, adjacent pixels image locations are 1 m apart on the ground.
[4] Pixhawk® 4 is designed and developed by Holybro and Auterion, optimised to run the PX4 firmware.
[5] http://conpleks.com/robotech/new-automated.
[6] http://www.precisionmakers.com/greenbot/.
[7] https://clearpathrobotics.com/moose-ugv/.
[8] https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/.
[9] https://www.earthautomations.com/prodotto/.
[10] https://www.naio-technologies.com/en/oz/.

Fig. 4 – The Q4T quadrotor developed by MAVTech s.r.l. (credit: MAVTech).

the Q4T; ii) a Pixhawk 2 autopilot with a navigation system based on the PX4, including 3 sets of InvenSense IMU sensors for extra redundancy, which include accelerometers, gyroscopes and compass, and n. 2 redundant MS5611 barometers. The entire flight management unit (FMU) and the IMU are housed in a cube, located on top of the main board.

Despite the increasing availability on the market of a number of commercial vehicles, some of them designed ad-hoc to operate in specific agricultural scenarios, we noticed that there still are gaps that need to be addressed to improve vehicles performance when oriented towards full *autonomous navigation*, as highlighted also in Part I (Mammarella et al., 2022). For example, wind gusts or turbulence can have negative effects on the UAVs' stability, as they can disturb remote sensing tasks because of additional uncontrolled drone movements, leading to inaccurate measurements. Another aspect is related to safety issues, i.e. guaranteeing that the vehicles remain "close" to the planned/desired trajectory within a tolerance range defined by the mission requirements while ensuring collision avoidance. For all the aforementioned issues, which arise when looking for completely autonomous vehicles, it becomes essential to operate *directly* on the UVs' GNC features in order to improve and optimise the selected job/task. Hence, advanced GNC techniques, designed ad-hoc or inherited from other fields of application, are implemented for the fully-autonomous navigation of all the UVs involved in the multi-phase approach.

The paper is structured as follows. Section 2 is devoted to the theoretical definition of the selected guidance and control strategy for the FW-UAV flight software to be operated for the remote sensing phase. Moreover, the preliminary results obtained after applying them to the chosen vineyard are discussed. The application of the selected modelling frameworks, designed to retrieve crucial information regarding the spatial layout of the environment and, in particular, of the target crop are detailed in Section 3. These algorithms semantically interpret the 3D point clouds of the vineyards and generate low complexity 3D mesh vine row models. This information is exploited in Section 4, which also focuses on the presentation of the selected GNC strategies for both the 4WS-UGV and RW-UAV, as well as the corresponding

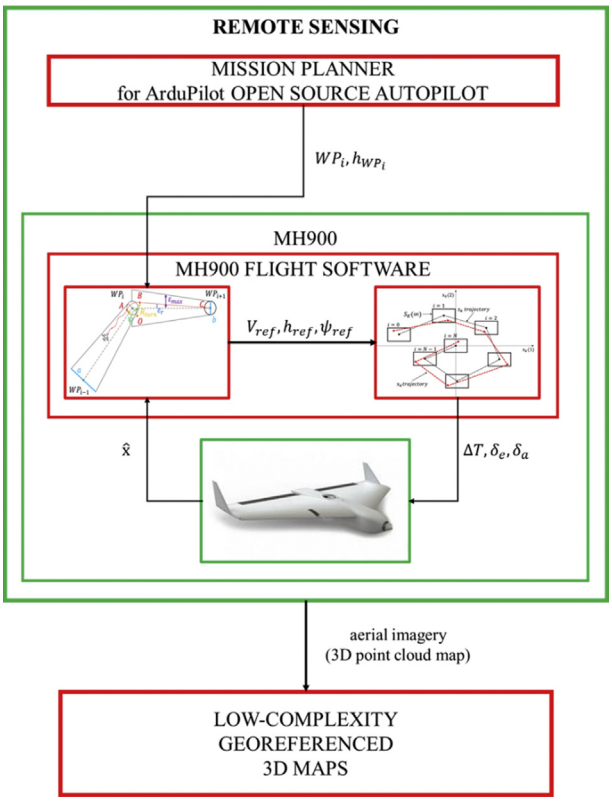


Fig. 5 – Detailed overview of the remote sensing mission phase including offline (red) and online (green) implementation steps. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

simulation results when operated within the vine rows. Conclusions and future developments end this work.

2. Remote sensing with the MH900

The remote sensing phase is devoted to the acquisition of aerial imagery by flying the MH900 FW-UAV, which is equipped with a multi-spectral camera, over the selected vineyard. In particular, the drone has to follow a snake-like path while maintaining a predefined, fixed relative altitude with respect to the terrain, to guarantee proper image acquisition.

As shown in Fig. 5, the first step consisted in defining the flying grid by assessing the location and proper distance of the target waypoints WP_i and related flying altitude, by considering a reference cruise speed of 12 m s^{-1}. The sequence of reference waypoints was obtained *offline*, by using the commercial ArduPilot Mission Planner[11] software, i.e. a full-featured ground station application for the ArduPilot open source autopilot project. These waypoints, which identify the grid pattern over the vineyard of interest, were fixed with respect to the vineyard's geographic information system (GIS) map, according to the terrain geometry, the picture resolution

[11] The ArduPilot Mission Planner is provided by the ArduPilot project, a corporation of numerous partners as listed at https://ardupilot.org/copter/docs/common-partners.html.

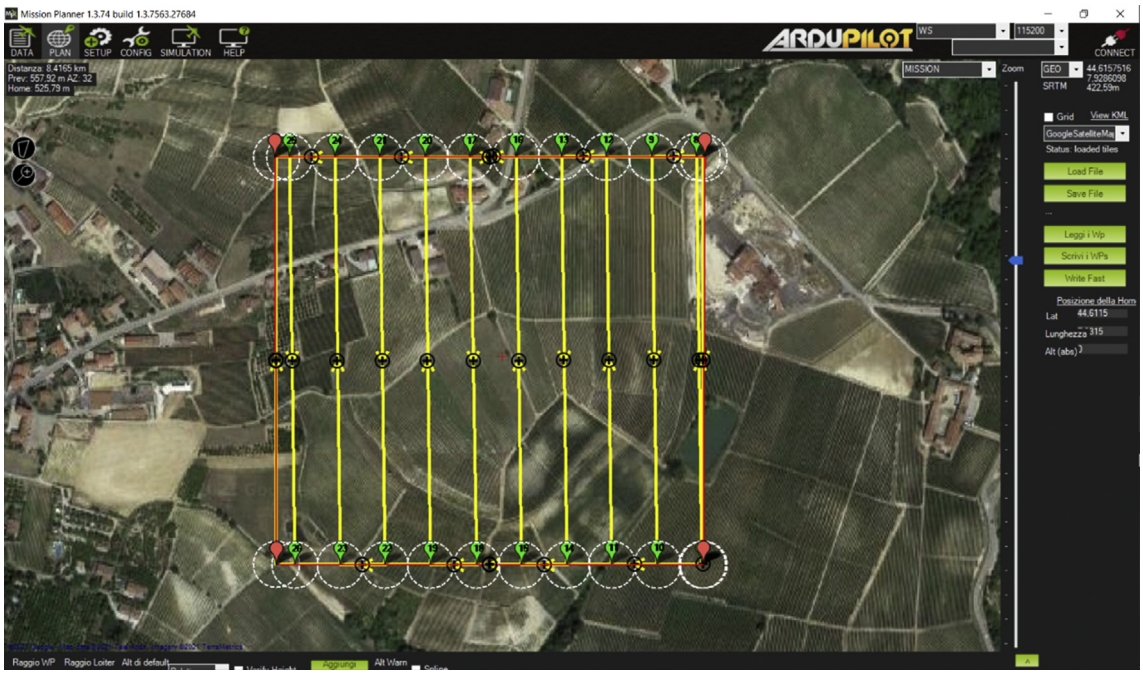


Fig. 6 – Waypoints planning with the ArduPilot Mission Planner.

required, and the average airspeed. In particular, in the case of significant changes in altitude between two successive waypoints (WPs), these terrain features were properly tackled by increasing the number of WPs, thus tightening the grid map and obtaining a better representation of the terrain itself.

Another crucial requirement for the definition of the waypoints grid was provided by the sensor features. Indeed, considering a target GSD of 100 mm pixel^{-1}, the relative flight altitude was set to 150 m with respect to the terrain (see Mascarello et al., 2017), considering the Micasense© RedEdge multispectral camera, positioned in a nadiral orientation with a CMOS sensor of 1280 x 960 pixels, 4.8 × 3.6 mm dimension, 5.4 mm focal length, and altitude 150 m. In addition, to ensure an azimuthal overlap between two consecutive images, acquired with a 1 Hz frequency greater than 85%, the speed should not exceed 15 m s^{-1}. Nonetheless, the proposed strategy was also validated in simulation for different flight levels (see Fig. 12), corresponding to smaller GSD values (i.e. up to 41 mm pixel^{-1}). The obtained path was a typical snake-like mapping, as represented in Fig. 6, where parallel lines, i.e. transects, connect waypoints in an ordered sequence. This pattern made it possible to guarantee that the UAV properly captured a quantity of images that overlapped to the degree required for the mission and for the processing software.

As shown in Fig. 5, this set of WPs represent the main input of the guidance block, which is in charge of determining in *real-time* the reference airspeed V_{ref}, altitude h_{ref}, and heading ψ_{ref}, according to the current UAV's location and altitude, given the i-th waypoint position and altitude (i.e. $WP_i = (N_{WP_i}, E_{WP_i})$and h_{WP_i}). Then, to close the GNC loop, the controller assesses the optimal control action (in terms of throttle ΔT, elevator deflection δ_e and aileron deflection δ_a) to properly track the reference signals provided by the guidance while fulfilling system and actuation constraints, despite the presence of wind turbulence. Further details on the flight software are provided in Section 2.1, while the preliminary results obtained during the software-in-the-loop (SIL) validation are reported and described in Section 2.2.

2.1. *Guidance and control strategy for the MH900 flight software*

When a FW-UAV is involved in tasks such as remote sensing, surveillance or patrolling, it is typically required to track a predetermined path with high accuracy. Several approaches have been proposed in the literature, as surveyed e.g. by Sujit et al. (2014), to generate such paths, given the mission's objectives and constraints. Within this framework, the guidance segment assumes a relevant role for the accomplishment of the UAV's mission, providing a feasible trajectory to the controller. Different guidance algorithms can be implemented depending on the type of mission the UAV was designed for, as summarised in Mammarella et al. (2022). In this work, the approach proposed by Capello et al. (2013) was selected and applied. It consists in a simple but effective scheme, designed not only to be efficient but also computationally compatible with low-cost autopilots. The UAV is called to reach subsequent waypoints, while maintaining a fixed relative altitude with respect to the terrain. The starting point is represented by the sequence of target waypoints, in terms of North (N), East (E) and altitude (h) coordinates, which are fed to the guidance algorithm to obtain the real-time reference

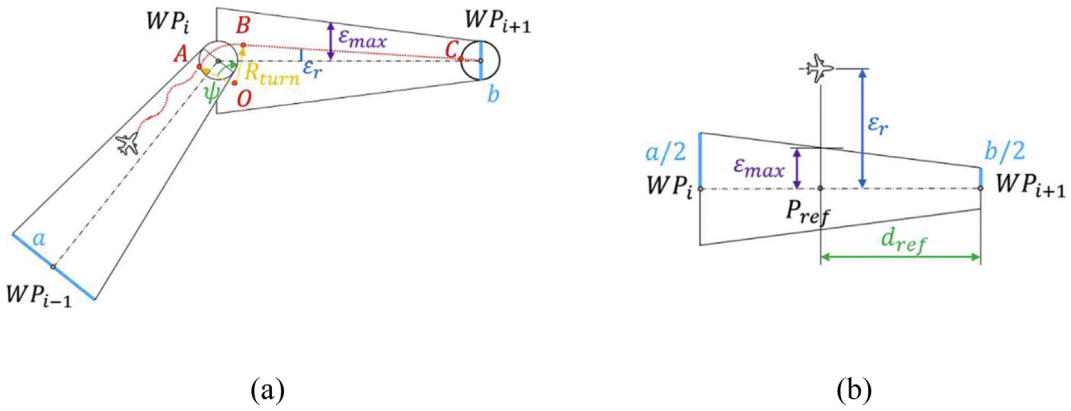


Fig. 7 – (a) Guidance phases and (b) definition of cross-track error (CTE) and reference distances (Capello et al. (2013)).

trajectory according to the UAV's current position and altitude. According to the assumptions reported in Capello et al. (2013), the guidance profile is divided into three phases (see also Fig. 7).

The first phase is the *waypoint approach*, represented by the red dotted line in Fig. 7a, during which the vehicle maintains fixed velocity at a pre-defined altitude while flying from waypoint WP_{i-1} towards waypoint WP_i. The waypoint is considered as reached when the UAV arrives at point A, flies into the imaginary circle centered at waypoint WP_i and starts the turning phase around waypoint WP_i in the direction of the next waypoint WP_{i+1}. During the second phase, identified by the dotted red arc between points A and B in Fig. 7a, the FW-UAV turns around waypoint WP_i with a velocity profile compliant with the turn rate constraint, function of the speed of the UAV and of the bank angle. This phase ends when the difference among the UAV's heading angle ψ_{UAV} and ψ is lower than a given threshold (5° in our case). Last, the third phase includes a straight flight from point B to point C, following the cross-track error (CTE) performance index $\varepsilon_r$requirement (see Fig. 7b), which is defined as follows

$$\varepsilon_r = \frac{|E_{UAV} - mN_{UAV} - (E_i - mN_i)|}{\sqrt{m^2+1}}, \tag{1}$$

where (E_{UAV}, N_{UAV}) are the East and North coordinates of the UAV's current position, (E_i, N_i) are the corresponding coordinates of waypoint WP_i, and m is defined as

$$m = \frac{E_{i+1} - E_i}{N_{i+1} - N_i}. \tag{2}$$

This means that the corrections on the heading angle are imposed only when the UAV's CTE is larger than an assigned value ε_{max} defined as

$$\varepsilon_{max} = \frac{\sqrt{(E_{i+1} - E_{ref})^2 + (N_{i+1} - N_{ref})^2}}{\sqrt{(E_{i+1} - E_i)^2 + (N_{i+1} - N_i)^2}}, \tag{3}$$

where (E_{ref}, N_{ref}) are the East and North coordinates of the reference point P_{ref} calculated as

$$E_{ref} = E_{UAV} + \varepsilon_r \psi_{ref}, \tag{4}$$

$$N_{ref} = N_{UAV} - \varepsilon_r \psi_{ref}. \tag{5}$$

As a result, if ε_r is smaller than ε_{max} at a certain time during the straight flight phase, no corrections of the aircraft heading angle are carried out, otherwise the new reference heading angle is the heading angle of the segment between the UAV's position and the next waypoint.

It should be noted that the trajectory resulting from the aforementioned guidance scheme is bi-dimensional, whereas in this selected case, a terrain-following strategy is also introduced due to the non-flat terrain profile. As first proposed in Mammarella et al. (2019), a ramp function is designed to follow the terrain's profile, by defining the time-varying reference altitude signal $h_{ref}(t)$ fed to the control scheme as

$$h_{ref}(t) = \frac{H_{i+1} - H_i}{\sqrt{(N_{i+1} - N_i)^2 + (E_{i+1} - E_i)^2}} \cdot d(t) + H_i, \tag{6}$$

where $d(t)$ is the time-varying relative distance among the aircraft, with the North-East coordinates given as $(N_{UAV}(t), E_{UAV}(t))$, and the i-th WP and $d(t)$ is defined as

$$d(t) = \sqrt{(N_{UAV}(t) - N_i)^2 + (E_{UAV}(t) - E_i)^2}. \tag{7}$$

The online definition of the optimal 3D trajectory, determined by the guidance algorithm, provides the reference signals to the longitudinal and lateral-directional control blocks, in terms of reference the velocity V_{ref}, altitude h_{ref} and heading angle ψ_{ref}.

Once the instantaneous reference trajectory is identified online, it is fed to the control block (see Fig. 5), which is in charge of tracking it, while, in addition, fulfilling mission, safety and mechanical constraints. Moreover, the control block also has to guarantee robustness against external disturbance sources and uncertainties due to unmodelled dynamics or to the manufacturing process. Indeed, these uncertainty sources could compromise the vehicle's stability and performance, which leads to the selection of a robust controller rather than the classic and simple control schemes typically implemented on UAV autopilots. In particular, a tube-based robust model predictive control (TRMPC) approach was selected for this work. This approach has already been tested for other applications, such as in Mammarella et al. (2017) for orbital spacecraft manoeuvres and in Mammarella and Capello (2020b) for remote sensing over a paddy field.

This choice was motivated by the two main features of this approach: i) the robustness against additive disturbances, which in the case of remote sensing missions could be represented by wind turbulence; and ii) the computational efficiency of a classic MPC, which also makes this scheme compatible with computationally limited autopilots such as those installed on board low-cost UAVs.

As detailed in Mayne and Rawlings (2009) and in Kouvaritakis and Cannon (2015), the TRMPC scheme is based on the concept of state trajectories *tubes*, each one representing an admissible disturbance sequence w over the observed time-window. The center of this tube corresponds to the nominal undisturbed trajectory, whose dynamics are the ones to be controlled and subject to tightened constraints with respect to the initial ones.

A discrete linear time-invariant system is considered

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{8}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ represent the state and the control input at time k, respectively, whereas $w_k \in \mathbb{R}^n$ is the persistent random disturbance, bound into a convex and compact set $\mathbb{W}$. It is assumed that the system is subject to hard constraints on both state x_k and input u_k, defined by convex polytopes, i.e. $\mathbb{X} = \{x \in \mathbb{R}^n | H_x x \leq h_x\}$ and $\mathbb{U} = \{u \in \mathbb{R}^m | H_u u \leq h_u\}$ respectively. The objective is to design a stabilising receding horizon control, which guarantees constraint satisfaction and minimises the infinite horizon quadratic cost

$$J_\infty = \sum_{i=0}^{\infty} x_i^\top Q x_i + u_i^\top R u_i. \tag{9}$$

However, predictive control strategies provide effective approximations of the optimal control law that can be computed efficiently and in real time, enabling MPC optimisation to be specified as a finite-dimensional problem. Hence, the optimisation problem can be defined as

$$\min_{\nu_k} \sum_{i=0}^{N-1} \left(x_{i|k}^\top Q x_{i|k} + u_{i|k}^\top R u_{i|k} \right) + x_{N|k}^\top P x_{N|k}, \tag{10}$$

$$\text{s.t. } x_{i+1|k} = Ax_{i|k} + Bu_{i|k}, \quad x_{0|k} = x_k,$$

$$x_{i|k} \in \mathbb{X}, \quad i \in [1, N-1],$$

$$u_{i|k} \in \mathbb{U}, \quad i \in [0, N-1],$$

$$x_{N|k} \in \mathbb{X}_N, \tag{11}$$

where $x_{i|k}$ and $u_{i|k}$ denote, over the prediction horizon N, the predicted values of the model state and input, respectively at time $k+i$, based on the information available at time k, $x_{N|k}^\top P x_{N|k}$ is the terminal cost that guarantees stability, and $\mathbb{X}_N$ is the (control invariant) terminal constraint set.

Due to the presence of disturbance w_k, the constraint satisfaction cannot be guaranteed. Hence, a different approach has to be considered. For the TRMPC scheme, the idea is to steer the uncertain trajectories to the nominal undisturbed one, controlling the "center" of the trajectory tube despite the effects due to the modelled disturbances (see Fig. 8). To separate the effects of the disturbance from the nominal dynamics over the predicted one, it is possible to assume that the predicted state $x_{i|k}$ is given by the sum of the nominal state $z_{i|k}$ and the error among the current state and the nominal one, i.e. $e_{i|k} = x_{i|k} - z_{i|k}$. Thus, exploiting the following time-varying feedback control law

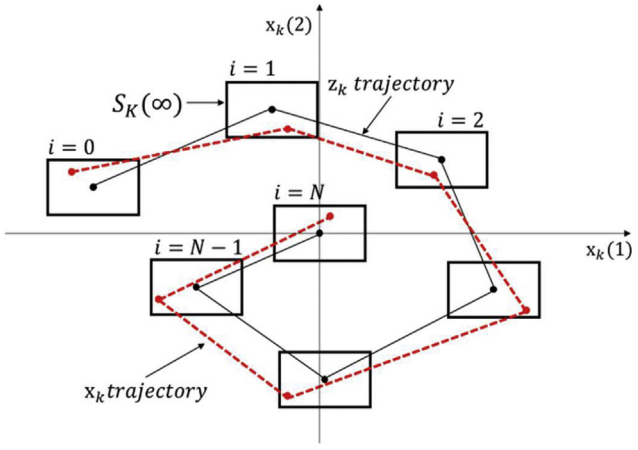


Fig. 8 – Outer-bounding tube representation at the k-th time step over a prediction horizon of N (Mammarella & Capello, 2020b).

$$u_{i|k} = v_{i|k} + K(x_{i|k} - z_{i|k}), \tag{12}$$

the predicted dynamics $x_{i+1|k} = Ax_{i|k} + Bu_{i|k} + w_{i|k}$ can be rewritten as follows

$$\begin{cases} z_{i+1|k} = Az_{i|k} + Bv_{i|k}, \\ e_{i+1|k} = A_K e_{i|k} + w_{i|k}, \end{cases} \tag{13}$$

where $A_K = A + BK$ is Shur stable. In this way, the effect of the disturbance is limited to the error dynamics, for which it is possible to define the minimal robust positive invariant (mRPI) set

$$S_K(\infty) \doteq \sum_{j=0}^{\infty} A_K^j \mathbb{W}. \tag{14}$$

For the definition of the mRPI set, the reader can refer to Blanchini and Miani (2008) and Kouvaritakis and Cannon (2015). The baseline TRMPC concept consists in controlling the nominal dynamics subject to tightened constraints in order to guarantee that each possible disturbed trajectory still satisfies the initial hard constraints. To properly design an outer approximation of the tightened state, input, and terminal constraint sets, i.e. $\mathbb{Z} \subseteq \mathbb{X} \ominus S_K(\infty)$, $\mathbb{V} \subseteq \mathbb{U} \ominus KS_K(\infty)$and $\mathbb{Z}_N \subseteq \mathbb{X}_N \ominus S_K(\infty)$ respectively, the approach proposed in Mayne and Rawlings (2009) was followed, obtaining

$$\mathbb{Z} \doteq \left\{ z \in \mathbb{R}^n \,\middle|\, H_x z \leq h_x - (1-\beta)^{-1} \varphi_N \right\}, \tag{15}$$

$$\mathbb{V} \doteq \left\{ v \in \mathbb{R}^m \,\middle|\, H_u v \leq h_u - (1-\beta)^{-1} K \varphi_N \right\}, \tag{16}$$

$$\mathbb{Z}_N \doteq \left\{ z \in \mathbb{R}^n \,\middle|\, H_N z \leq h_N - (1-\beta)^{-1} \varphi_N \right\}, \tag{17}$$

where β is a design parameter and φ_N is defined as

$$\varphi_N = w\left\{H_x \sum_{j=0}^{N-1} A_K^j \mathbb{W}\right\} \tag{18}$$

Once the tightened state, and the input and terminal constraint sets, i.e. $\mathbb{Z}$, $\mathbb{V}$ and $\mathbb{Z}_N$, respectively, have been defined, the finite horizon optimal quadratic cost can be defined for the nominal dynamics in terms of the nominal state $z_{i|k}$ and the nominal control input $v_{i|k}$ over the prediction horizon N as

$$\min_{v_k} \sum_{i=0}^{N-1}\left(z_{i|k}^T Q z_{i|k} + v_{i|k}^T R v_{i|k}\right) + z_{N|k}^T P z_{N|k} x200d;, \tag{19}$$

$$\text{s.t. } z_{i+1|k} = A z_{i|k} B v_{i|k}, z_{0|k} = x_k,$$

$$z_{i|k} \in \mathbb{Z}, \quad i \in [1, N-1],$$

$$v_{i|k} \in \mathbb{V}, \quad i \in [0, N-1],$$

$$z_{N|k} \in \mathbb{Z}_N. \tag{20}$$

The first control action $v_{0|k}^*$ of the optimal sequence v_k^*, solution of Eq. (16), represents the optimal control applied to the nominal system while the corresponding control on the uncertain system is defined according to the time-invariant control law expressed as $u_k = v_{0|k}^* + Kx$, with K being the stabilising feedback matrix, defined to guarantee the Schur stability of the closed-loop system. Indeed, to stabilise the system with respect to the parametric uncertainty q, ascribable for example to discrepancies between the mathematical model and the actual dynamics, neglected non-linearities and manufacturing processes, a linear matrix inequality (LMI) approach was applied to the definition of Schur stability of the closed-loop system. More details can be found in Mammarella et al. (2018).

2.2. Preliminary results

The remote sensing mission was planned according to the selected area, the vehicle, the operative features and the payload requirements. In this case, after the take-off phase, the MH900 needed to follow a snake-like path, preliminarily identified by 20 waypoints by the mission planner (see Fig, 6), that made it possible to cover the entire area while satisfying payload constraints (as anticipated in Section 2). Because of the non-flat profile of the terrain, additional waypoints were introduced directly in the planner to tighten the grid map and to improve the generation of the reference 3D trajectory according to the altitude variations along the path. Finally, a total of 50 WPs were identified, as represented by the green diamonds in Fig. 9.

In terms of operational constraints, the UAV was forced to maintain longitudinal and lateral-directional state deviations within a ±10% range, with respect to the reference values provided by the guidance algorithm,. This was despite the presence of a fixed-direction random wind turbulence of about 1 m s^{-1} in amplitude. Additional constraints are related to the actuation systems, which mechanically limit the throttle ΔT, elevator deflection δ_e and aileron deflection δ_a, due to variations of up to 20% compared to the trim condition.

The first SIL validation campaign, to test the proposed guidance and control strategy, was carried out by exploiting a MATLAB/Simulink simulator running on an Intel Core i7-7500U with a CPU @2.70 GHz, 16 GB of RAM and a 512 GB solid-state drive. The preliminary results are represented in Figs. 9–12, which compare the performances obtained with or without the effects of the wind turbulence. From Fig. 9, it is possible to observe that the UAV was able to track the reference trajectory along the area profile also in the case of wind turbulence affecting its performance (red line).

The efficacy of the GNC strategy was corroborated by the results represented in Fig. 10, where the real flight data in the absence (blue) or presence (red) of wind turbulence for altitude (h), heading angle (ψ) and roll angle (φ) are compared to the reference signals provided by the guidance algorithm. These simulation results proved the robustness and the efficacy of the controller since all the constraints are fulfilled, despite the significant impact of the wind disturbance, especially in the WP's proximity and during the turning phases, as highlighted in the zoom-in in Fig. 10. Figure 11 highlighted the control actions required by the MH900 to properly track the reference trajectory in terms of throttle, and elevator and aileron deflections. It can be observed that the main contribution was required during the turning phases or when the disturbance effect was stronger.

The choice of a flight altitude of 150 m implies a GSD of about 102 mm, when the Micasense RedEdge-M is considered (Mascarello et al., 2017). On the other hand, it is important to note that with more performing sensors or cameras, as e.g. the Sony RX-100 RGB camera, it is possible to obtain much better GSD levels (i.e. 41 mm $pixel^{-1}$), when considering the same flight altitude (see e.g. Gallo et al., 2019). Hence, when more precise (smaller) GSD levels are required, two possible solutions are at hand: i) when considering the Micasense RedEdge-M, the flight altitude can be reduced down to the one corresponding to the desired GSD; or ii) the UAV can be equipped with a different hardware in order to guarantee a better GSD while maintaining the same flight altitude (i.e. 150 m). In both cases, the efficacy of the proposed guidance and tracking control strategy, which was validated for the Micasense camera and the 150 m flight altitude, can easily be confirmed, as shown in Fig. 12 where the efficacy of the GNC scheme is validated for different flight profiles, each one characterised by a different flight altitude and, consequently, by a different GSD.

3. Aerial imagery elaboration and low-complexity maps

From the multispectral imagery acquired during the remote sensing phase, a dense 3D point cloud model of the case-study vineyard was generated, using the commercial structure from motion-based (SFM) software Agisoft Metashape.[12] Please note that only the spatial information from the dense 3D point cloud model was exploited in this study. Spectral content,

[12] Agisoft Metashape software is a product of Agisoft LLC company, located at St. Petersburg, Russia.

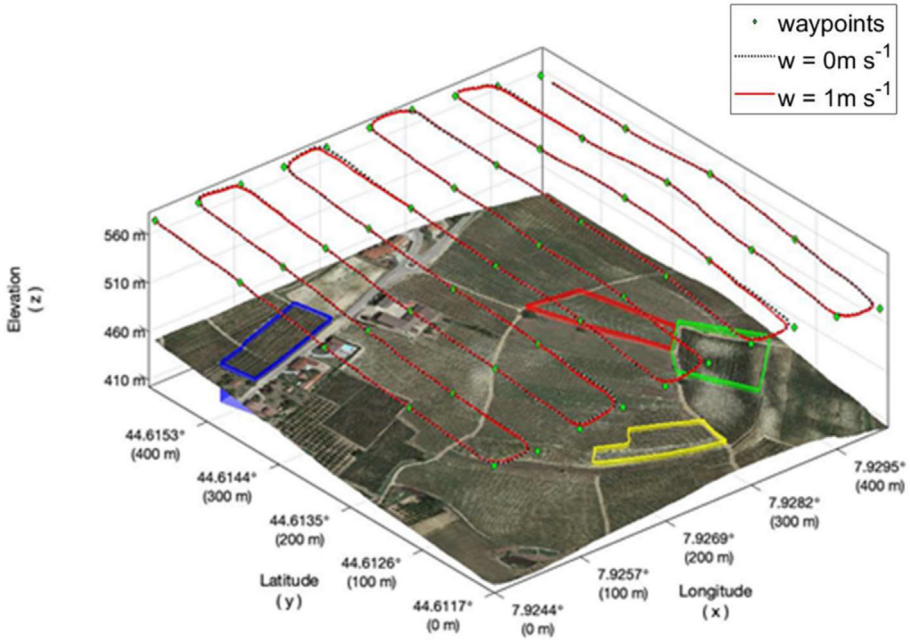


Fig. 9 – The 3D trajectory in the absence (black) or presence (red) of wind turbulence. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

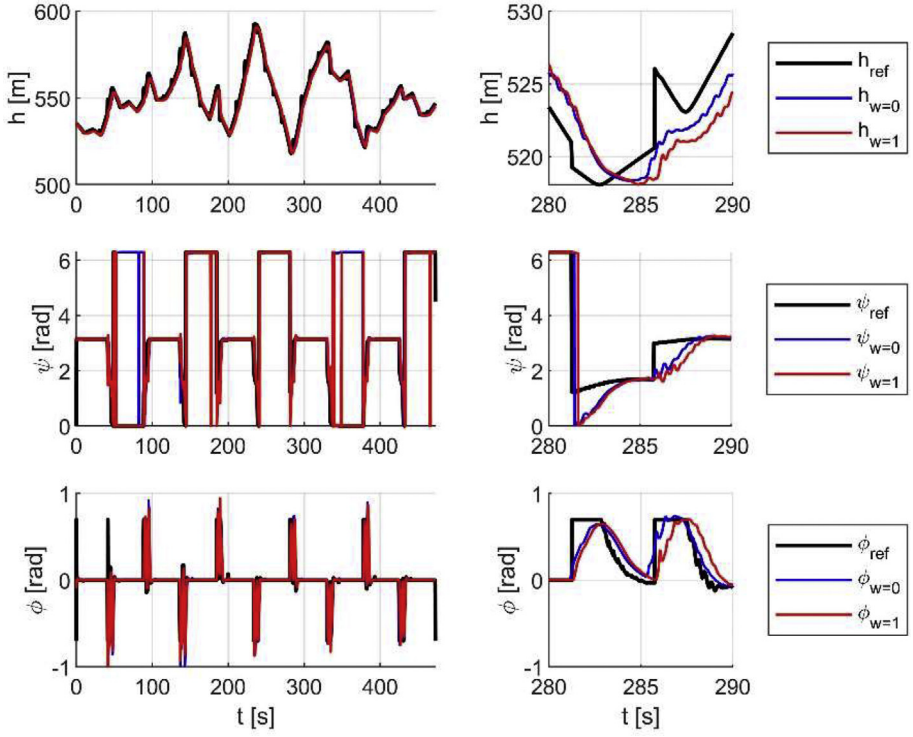


Fig. 10 – Reference signals (black) and real flight data in the absence (blue) or presence (red) of wind turbulence for altitude (h), heading angle (ψ) and roll angle (φ). On the right, zoom-in to highlight the differences between the two test cases. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

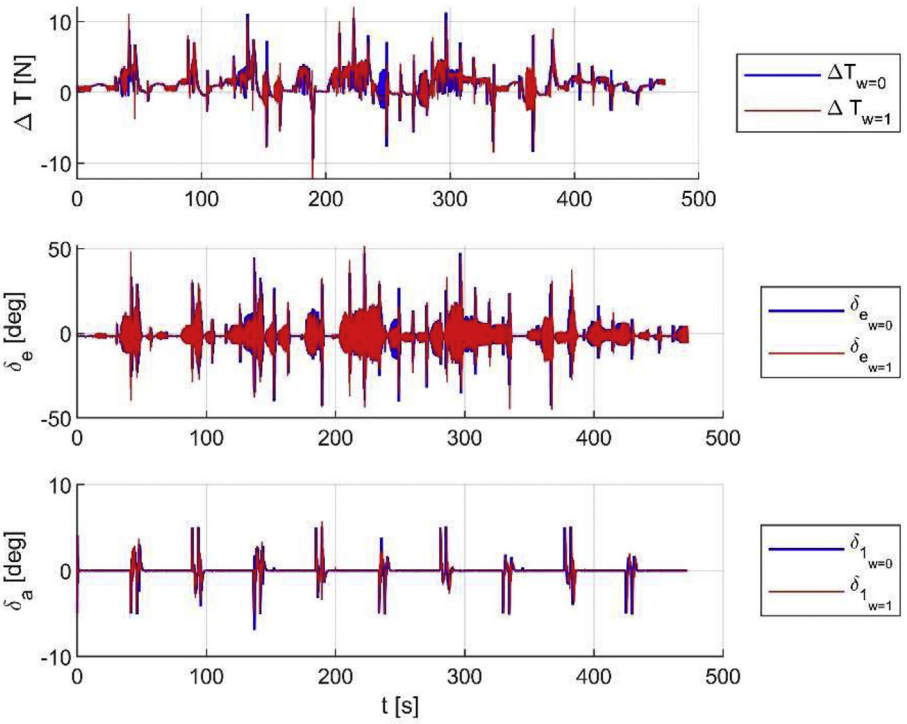


Fig. 11 – Control signals in the absence (blue) or presence (red) of wind turbulence in terms of throttle (ΔT), elevator deflection (δ_e) and aileron deflection (δ_a). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

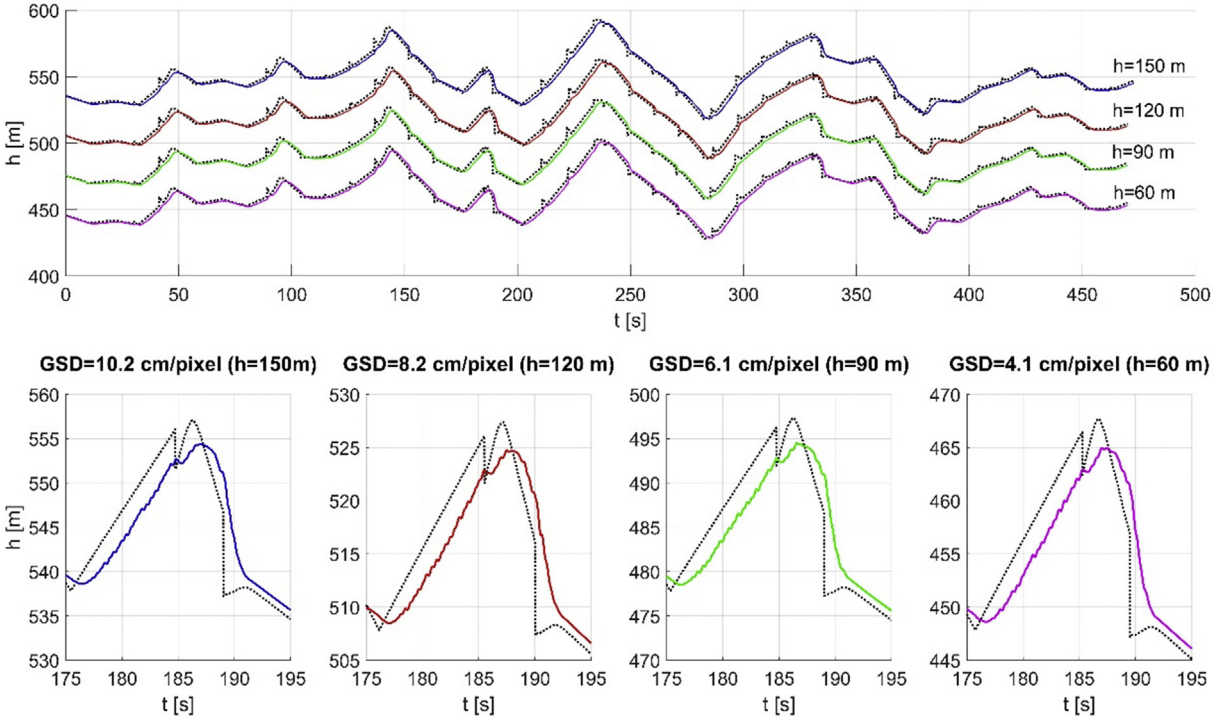


Fig. 12 – Flight profiles for different relative altitudes, each one corresponding to a different GSD.

which can be used to additionally asses the crop status, was not used. This dataset, which can be formally defined as a set S^{REF} of points represented by an array $p_i = [x_i, y_i, z_i]^T$, has a density of about 1500 points m^{-2} of modelled terrain surface (Fig. 13). In this phase, the dataset points are still not classified and are unordered. To retrieve valuable information about the

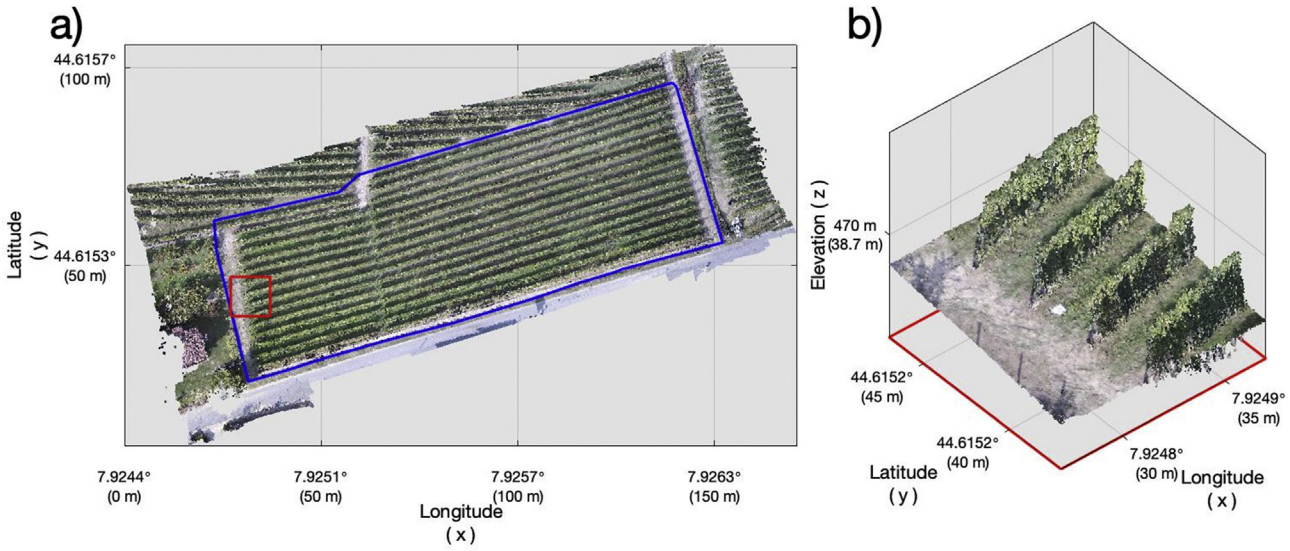


Fig. 13 – Dense point cloud of selected Parcel F (a), located in Barolo, Italy, and its enlargement (b). In detail, the low complexity model of the canopy is defined as a triangulated mesh.

agricultural environment and, in particular, about the crop (e.g. in terms of position, volume or spatial layout), this huge dataset has to be clustered, for example by detecting subsets of points representing the crop canopy. In this work, the two algorithms proposed in Comba et al. (2018) and Comba et al. (2020) were subsequently applied to the case-study vineyard model, in order to automatically detect the vineyard elements and to create a low complexity model of the scenario. This made it possible to obtain a simplified mesh suitable for the following cooperative definition of optimal paths for in-field operations. Indeed, the location of vine row end points, vine rows length, inter row space and, finally, vines canopy shape and volume are key specifications needed to properly perform the 3D path planning tasks. This pipeline, which is here fully automated and integrated, generated a light mesh model $\mathcal{M}_k$ representing the vine rows, without requiring any intervention by an operator.

$$\mathcal{M}_k = [\{V_{k-1}, V_k\}, F_{k-1,k}] \tag{21}$$

where V_{k-1} and V_k are the sets of mesh vertices, and $F_{k-1,k}$ is the set of triangular faces of the mesh between them, which are detected by solving the problem of minimum area c-gon containing a point set, as reported in detail in Comba et al. (2020).

For this work, the validation of the proposed method was limited to Parcel F, where the space between the vine plants and the inter-row space were about 0.9 and 2.5 m, respectively, and the variation in altitude was about 30 m. In the processed dataset, 16 fully represented vine rows were profitably detected, with an average length of 118 m (Fig. 14). The optimal light 3D mesh, representing each detected vine row, was generated by the algorithm as a trade-off between accuracy and data storage reduction. The dataset output of this process had a volume reduction of 98.6% with respect to the original 3D model generated by the SFM software, obtaining a file of only 4 Mb. This last aspect is also particularly relevant for its exploitation in real-time navigation, which requires shorter computational times. Graphical representations of the results are shown in Fig. 14.

3.1. *Unsupervised obstacle grid map definition*

Once a simple model of the detected vine rows was derived (Fig. 14), the same model was processed by a new unsupervised method, in order to derive an occupancy grid map, suitable for the subsequent path planning phases for the UVs in-field tasks.[13] To this aim, the information about the parcel boundaries previously retrieved was exploited to generate the set of all the spatial obstacles to be tackled. The proposed approach processed one vine row at a time, detecting a turning area at the end of the row, marked in yellow in Fig. 15b. Then, the grid of virtual static obstacles was embedded into the real one (vine row model, trees, etc).

Once the model $\mathcal{M}_k$ of the vine row closest to the mission starting point was selected, a virtual obstacle was generated from the closest vine row end point e_i in the form of a segment, tangent to the vine row (green lines), within the points e_i and g_i, obtained by the intersection of the segment and the parcel boundary. Since no information about which portion of the parcel boundary was closest to the considered vine row end e_i, the intersection of the segment with all the parcel boundaries was performed, and the closest one was then selected as suitable, i.e. g_i. This process was repeated at the end of each vine row, generating the occupancy grid map of the entire parcel (i.e. the blue grid in Fig. 15a), later used by the GNC suite for autonomous navigation within the vineyard.

[13] Information from low-complexity maps can be also exploited in autonomous navigation framework, as presented in Donati et al. (2021).

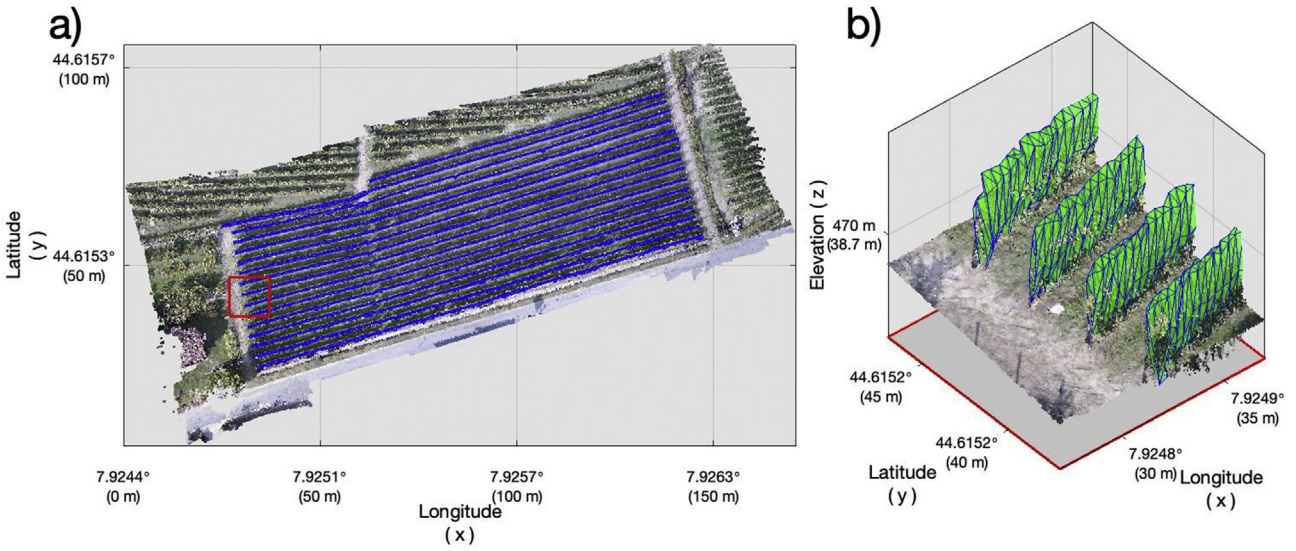


Fig. 14 – 3D triangulated light mesh representing the external envelope of the vine row canopies, automatically detected by the method in Comba et al. (2020), applied to the whole case study parcel (a) and its enlargement (b).

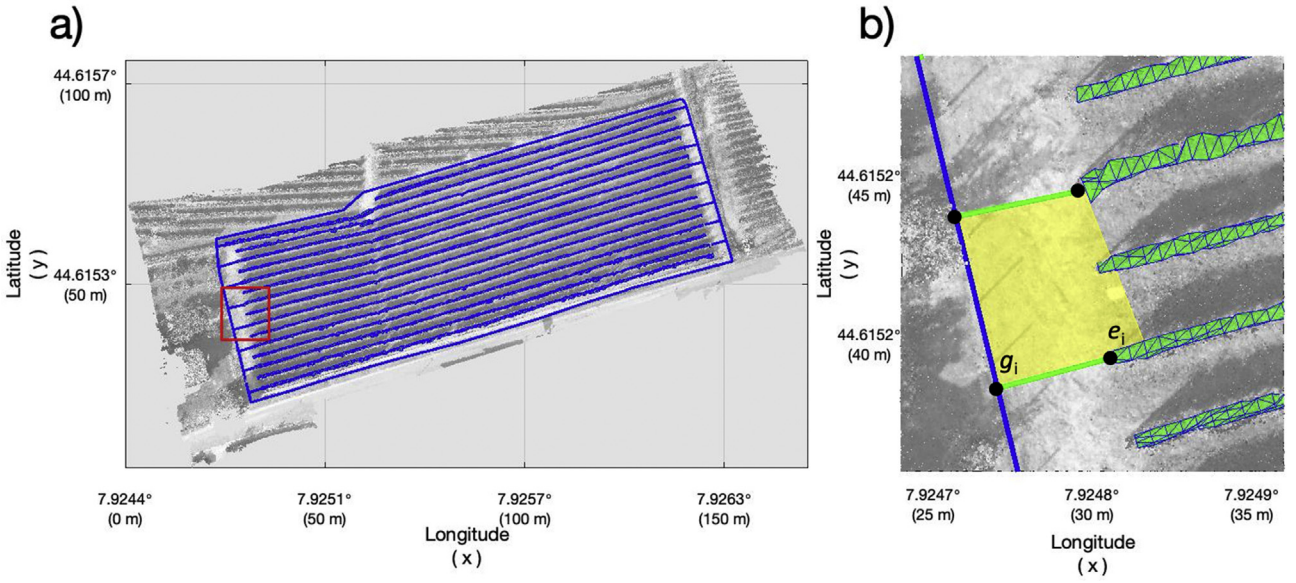


Fig. 15 – Obstacle grid map generated from the parcel boundaries and the vine row low complexity model of the whole case study parcel (a), and its enlargement (b).

4. In-field operations

In this section, the in-field operation tasks performed by the 4WS UGV and the quadrotor are described in detail, reporting the peculiarities of the GNC strategy for each UV called to operate within the vineyard rows and providing the preliminary simulation results. An overview of this phase is shown in Fig. 16.

As mentioned earlier, the UVs operating in the field require the previously generated simplified 3D maps, as well as the mission and system requirements deriving from the mission planning phase, to properly plan their tasks. Indeed, the georeferenced information provided by the low-complexity map was fused with the data provided by the onboard navigation sensors (i.e. GPS and IMU) to improve the UVs localisation in real-time. On the other hand, the requirements and constraints deriving from the previous mission planning were enforced within the guidance and control suite and contributed to the optimal design of the algorithms themselves, performed *offline* and later uploaded on the onboard hardware.

4.1. *In-field operations with the 4WS UGV*

As anticipated in the previous sections, in the proposed mission concept, the 4WS-UGV is called to autonomously operate within the vine rows to perform in-field operations, such as scouting, bio-pesticide spraying, or shredding over

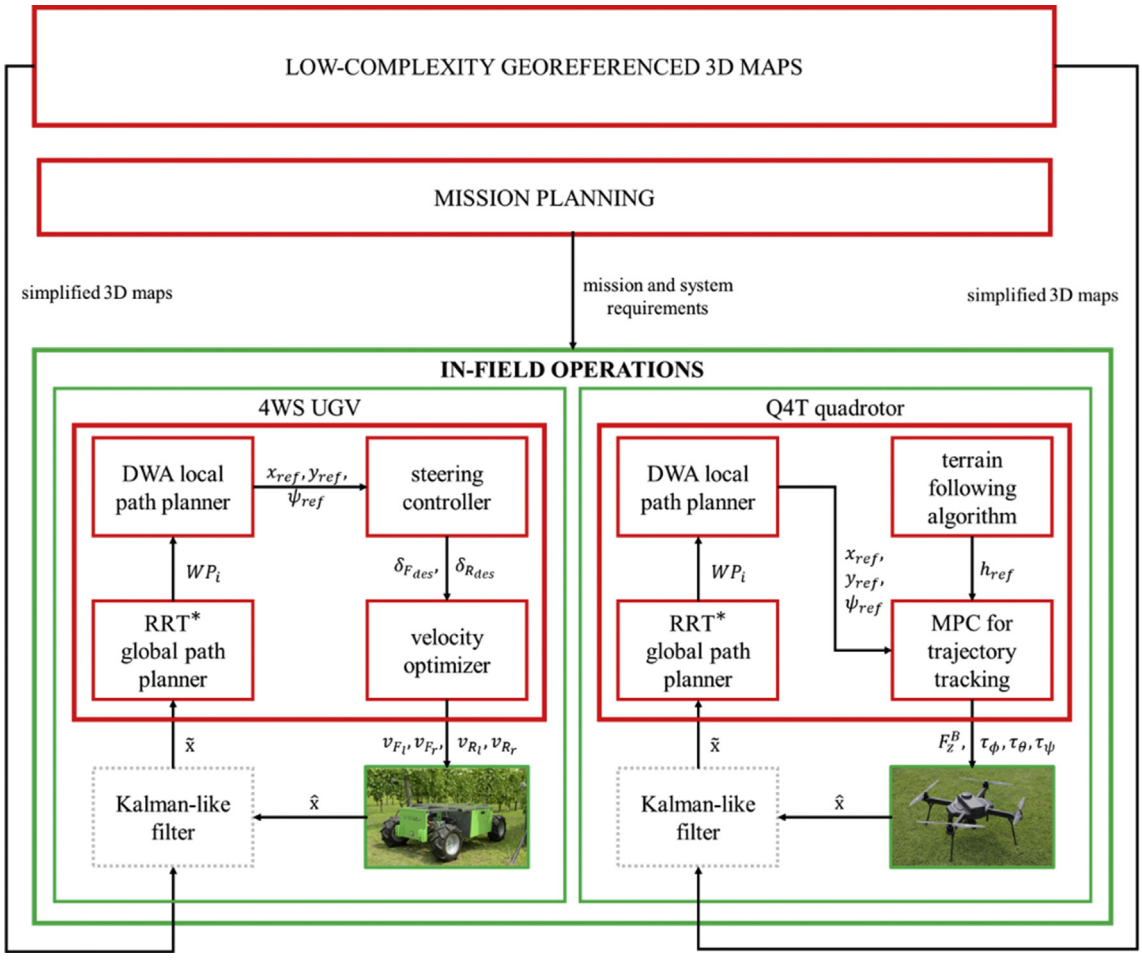


Fig. 16 – Detailed overview of the in-field operation mission phase performed by the 4WS UGV and the Q4T quadrotor, highlighting the offline (red) and online (green) implementation phases. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

sloped terrains. To improve the efficacy of the operation, autonomous driving is required, and an ad-hoc guidance strategy was selected and implemented, as detailed hereafter. Moreover, because of its four-wheel steering configuration, tailored control algorithms were conceived, which allowed to properly determine the optimal velocity profile for each wheel in order to guarantee no slippage.

4.1.1. *Guidance and control strategy for the 4WS UGV*

The classic approach for the path planning problem consists in splitting the guidance task into a global planning followed by a local planning (González et al., 2016). As defined in Kunchev et al. (2006), the global path planner is in charge of generating local goals (i.e. the waypoints) towards the final one, whereas the local path planner guarantees the smoothness and affordability of the reference trajectory that interconnects those goals. In this work, this philosophy for designing the guidance strategy of our 4WS UGV was embraced. In particular, an approach based on the so-called rapidly exploring random tree (RRT) was selected as the global path planner, which resulted particularly compatible with systems characterised by non-holonomic constraints and high degrees of freedom. This approach builds a tree by randomly choosing a node in the free space and finding the nearest node in the tree. Next, the planner expands this nearest node in the direction of the random node. The RRT approach has been extensively tested for automated vehicles, as in Dong et al. (2017). Because of the suboptimality of the solution, the so-called RRT *, first proposed in Karaman and Frazzoli (2011) to overcome the limitations of RRT, whose solutions are not asymptotically optimal, was applied for this work. The RRT * algorithm incrementally builds a search tree, providing anytime solutions, provably converging to an optimal solution, with minimal computational and memory requirements. Indeed, thanks to the addition of a heuristic rule, which forces the planner to choose the new points that are nearer to the goal, the RRT * turns out to be much more computationally efficient than the classic RRT method.

The starting point is to define the RRT approach in order to later explain how it was extended by Karaman and Frazzoli

(2011) to the RRT * scheme. The multi-step procedure can be summarised as follows:

a. The RRT first builds a tree using random samplings in the search space, starting from a given initial condition and expanding it (following the 2 wheels steering (2WS) model in our case) to find a path towards the final goal state.
b. During each iteration, a random state is selected from the configuration state and if the random sample lies in an obstacle-free region, then the algorithm searches for the nearest node in the tree according to a defined metric.
c. If the random state is accessible to the nearest node, then the tree is expanded by connecting these two nodes. Otherwise, it returns a new node by using a steering function, thus expanding the tree by connecting it with the nearest node. A Boolean collision checking process is performed to ensure a collision-free connection between these nodes.
d. The process continues until it reaches the maximum number of iterations, or the maximum simulation time is reached.

In the specific case of this work, for each state, three possible steering angles were provided to the RRT algorithm, i.e. $\delta = \left[-\frac{\pi}{6}, 0, \frac{\pi}{6}\right]$, so that the algorithm generated three different branches for the search space. Despite the reliable results, the process was too time consuming, especially because it was applied to a large area characterised by narrow corridors, as in the case of this example. Hence, a heuristic logic was added to the RRT scheme, which lead to the RRT * scheme, in which for each random node (x_i, y_i), the distance from the goal (x_{goal}, y_{goal}) was computed as

$$d_i = \sqrt{(x_i - x_{goal})^2 + \left(y_i - y_{goal}\right)^2}, \tag{22}$$

and this data was collected into a distance set $D = \{d_1, \ldots, d_n\}$ and later split into four classes, each related to a different quartile. In this way, the planner started looking for the starting points first in the 0–25% set and skipped to the next one, i.e. 25-50%, only if this set resulted empty, and so on until it found at least one point from which to generate new branches. With this approach, the nodes nearest to the goal have a larger chance to be selected as the root of a new branch. If all sets are empty, it means that the nodes would drive the vehicle to collide with obstacles and the problem is unsolvable.

Subsequently, to obtain an optimal trajectory, a guidance signal obtained from a so-called local planner, which is based on the former global planner and makes it possible to obtain a smooth and affordable path compliant with the non-holonomic constraints of the robot, must be overlapped with the first guidance signal provided by the RRT * scheme. For this work, the so-called dynamic-window approach (DWA), based on a receding-horizon scheme and described in Ögren and Leonard (2005), was exploited as the local path planner because of its capability to generate a smooth and optimised path for the robot. In robotics motion planning, DWA is an online collision avoidance strategy for mobile robots that, unlike other avoidance methods, is derived directly from the dynamics of the robot (Zhang et al., 2019), and is especially designed to deal with the constraints imposed by the limited velocities and accelerations of the robot (Fox et al., 1997). It consists of two main steps: i) firstly, generating a valid search space; and ii) second, selecting an optimal solution in the search space. In particular, in DWA, the search for the commands which control the robot is carried out directly in the velocity space. In this way, the kinematics of the UGV are directly incorporated into the method by reducing the search space to those velocities which are reachable under the non-holonomic constraints. Moreover, only the velocities which are safe (i.e. admissible), with respect to the obstacles, are considered. To check if the sub-trajectories generated by the admissible velocities (and steering angles) are collision free, each one is intersected with the circular safe-zone defined around the vehicle. Only if the intersection is empty, the corresponding trajectory can be defined safe. Then, an optimisation problem is solved in order to select the optimal velocities maximising a given objective function. In this case study, since each trajectory is defined by four parameters, i.e. considering the bicycle method these are the front v_F and rear v_R velocities and the corresponding steering angles δ_F and δ_R, the selected linear cost function is defined as

$$J(v_F, v_R, \delta_F, \delta_R) = \alpha d_O(v, \delta) + \beta d_{goal}(v, \delta) \tag{23}$$

where α and β are the weight matrices, d_O is the distance between the final node of the sub-trajectory and the nearest obstacle, and d_{goal} is the distance between the same node and the final goal. Then, exploiting a receding horizon approach, once the optimal sub-trajectory has been obtained, only the first step is implemented, and the procedure is iterated until the UGV reaches the final waypoint.

Once the optimal path was generated as previously described, it was necessary to design an ad-hoc control strategy to track the reference trajectory and to optimise the velocity of each wheel, minimising the slippage produced by the ASM. Going into the details, in this work, a two-level control strategy was considered, where a proportional steering controller was used to compute the desired steering angles for the front and rear wheels and later combined in a cascade with a velocity optimiser, based on a quadratic programming, to compute the optimal velocity profile for each wheel while enforcing non-holonomic constraints to drive the steering errors to zero.

Initially, the steering controller receives two inputs from the navigation system: i) the lateral deviation d_e of the vehicle center-of-mass (CoM) from the reference path; and ii) the angular deviation θ_e of the UGV heading angle from the

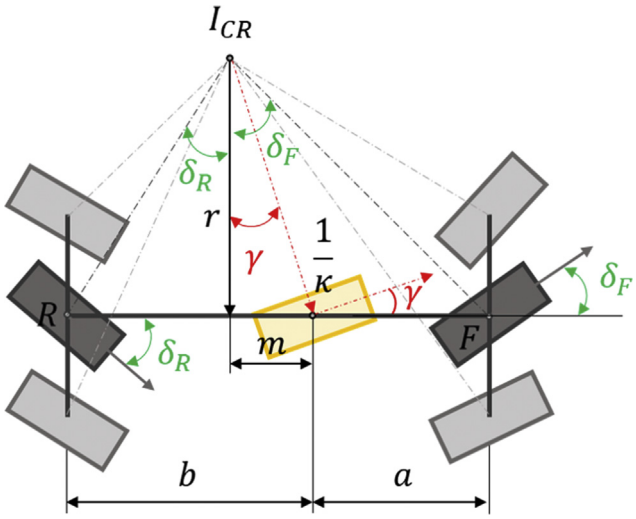


Fig. 17 – Acting on the virtual wheel applied to the CoM of the UGV (yellow one) with a steering command γ and a curvature command κ, it is possible to assess the desired steering angles for both front (δ_F) and rear (δ_R) axes. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

desired orientation. Hence, envisioning a virtual wheel located on the UGV's CoM, it is possible to resort to the steering command γ from the input d_e via a proportional law of the form

$$\gamma_c = -K_s d_e, \tag{24}$$

where K_s is an appropriate proportional gain. However, to correct the angular deviation θ_e it is not possible to act directly on the virtual wheel but one needs to act on the front and rear wheels, corresponding to the bicycle model (see Fig. 17), and to resort to the curvature parameter κ defined as

$$\kappa_c = -K_\theta \theta_e, \tag{25}$$

where K_θ is again a proportional tunable gain. Now, the reference values for the front and rear wheels steering angles, i.e. $\delta_{F_{des}}$ and $\delta_{R_{des}}$ respectively, can be recovered as a function of the curvature κ and steering angle γ with respect to the virtual wheel as follows

$$\begin{cases} \delta_{F_{des}} = arctan\dfrac{a\kappa + \sin\gamma}{\cos\gamma} \\ \delta_{R_{des}} = arctan\dfrac{b\kappa - \sin\gamma}{\cos\gamma} \end{cases}. \tag{26}$$

Next, to compute the commanded velocity profile for each wheel, it is necessary to enforce and fulfil the non-holonomic constraints defined as

$$\begin{cases} v_{Fj}\cos\delta_{Fj} - v_{Rj}\cos\delta_{Rj} = 0 \\ v_{il}\sin\delta_{il} - v_{ir}\cos\delta_{ir} = 0 \end{cases}, \tag{27}$$

where $i = F, R$ and $j = l, r$. So, first the relative distance between each wheel and the instantaneous center of rotation (ICR) is computed as

$$r_{ij} = \sqrt{(x_{ICR} - x_{ij})^2 + (y_{ICR} - y_{ij})^2}, \tag{28}$$

and it is possible to notice that the non-holonomic constraints are satisfied if each wheel maintains an angular velocity around the ICR

$$\Omega_{ij} = \frac{v_{ij}}{r_{ij}}, \tag{29}$$

equal to the UGV's angular velocity Ω_{des}. However, because of the ACMs and the non-holonomic constraints, it is necessary to resort to an optimisation problem based on a quadratic programming to obtain the optimal velocity profile that minimises slippage errors and satisfies the non-holonomic constraints. Hence, the following quadratic cost function can be defined

$$J_v = \|\alpha v - \beta\|^2, \tag{30}$$

where $v = [v_{F_l}, v_{F_r}, v_{R_l}, v_{R_r}]^\top$ is the design vector and (α, β) are defined depending on whether $\delta_{F_{act}}$ and $\delta_{R_{act}}$ are parallel or not. In the first case, it provides

$$\alpha = [1\ 1\ 1\ 1]^\top, \quad \beta = 4v_{des}, \tag{31}$$

otherwise

$$\alpha = \begin{bmatrix} \frac{1}{r_{F_l}} & \frac{1}{r_{F_r}} & \frac{1}{r_{R_l}} & \frac{1}{r_{R_r}} \end{bmatrix}^\top, \quad \beta = 4\Omega_{des}, \tag{32}$$

subject to the linear constraints of the form $Av \leq b$ with

$$A = \begin{bmatrix} \cos(\delta_{F_l}) & 0 & -\cos(\delta_{R_l}) & 0 \\ 0 & \cos(\delta_{F_r}) & 0 & -\cos(\delta_{R_r}) \\ \sin(\delta_{F_l}) & -\sin(\delta_{F_r}) & 0 & 0 \\ 0 & 0 & \sin(\delta_{R_l}) & -\sin(\delta_{R_r}) \end{bmatrix}, \quad b = [\varepsilon\ \varepsilon\ \varepsilon\ \varepsilon]^\top \tag{33}$$

defined according to Eq. (29) related to the non-holonomic constraints and where ε is an acceptable slippage tolerance (in this specific case $\varepsilon \in [10^{-3}, 10^{-2}]$).

4.1.2. *Preliminary results*

To define the mission objectives and requirements for the 4WS UGV, the obstacle grid map (described in Section 4.2) and the operational and system constraints, defined by the mission specifications and vehicle features, were combined to define the UGV's autonomous driving software setup. In particular, the occupancy map (blue lines and mesh in Fig. 18) was fed to the RRT* to identify the internal reference nodes (yellow diamonds), once the start node (green circle) and the goal node (red cross) were selected, as represented in Fig. 18.

Then, to ensure smoothness to the trajectory and to properly enforce the non-holonomic constraints, these target points were fed to the DWA local path planner, which returned the feasible reference trajectory (dotted black line) that the UGV must follow to properly perform its tasks while avoiding (static) obstacles (blue rectangles).[14] It is possible to observe that after the first turning phase, the reference trajectory was not perfectly aligned with the waypoints. This was

[14] For clarity and brevity, the simulation results were limited to the first two rows. Similar results were obtained when the entire vineyard was considered.

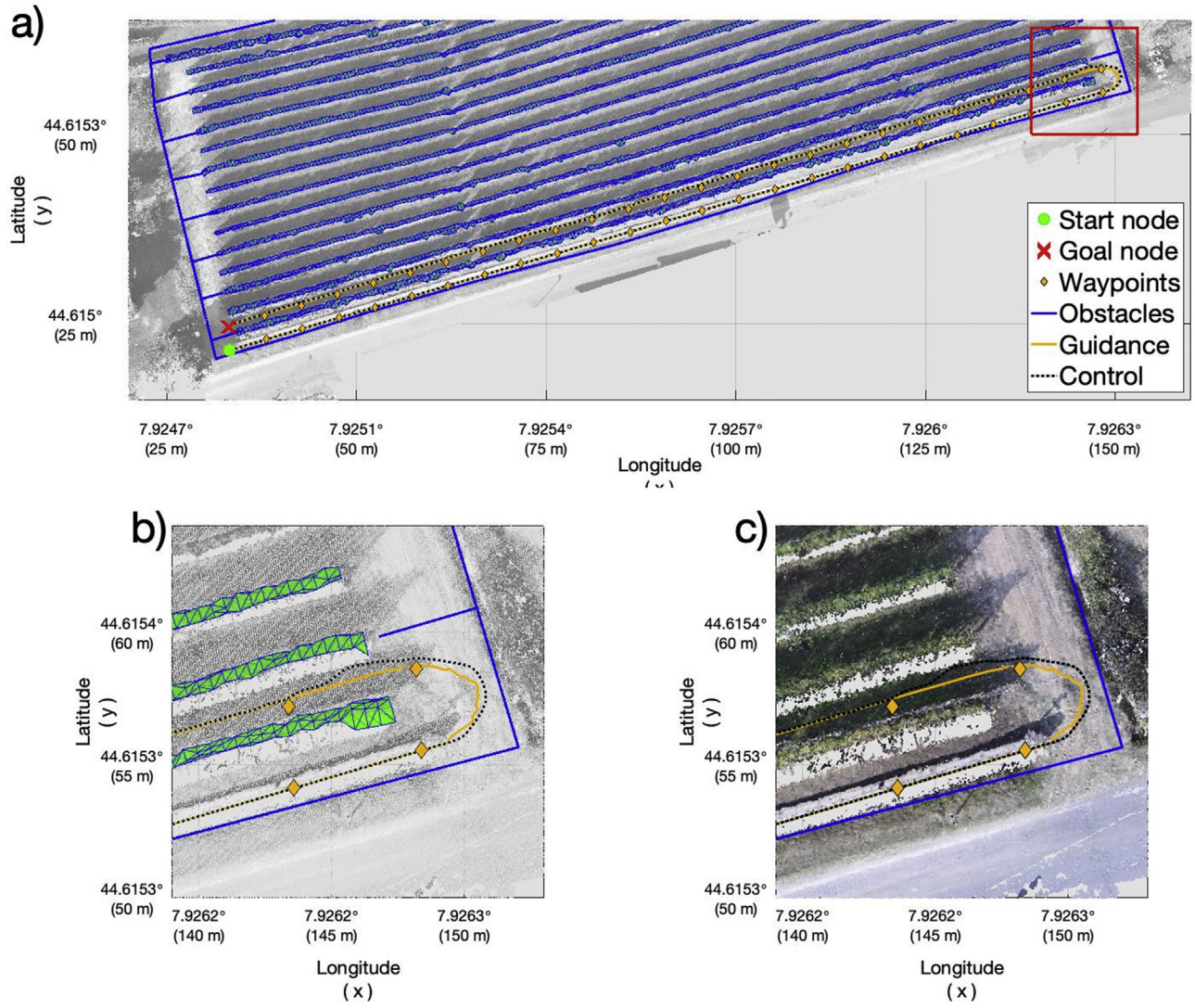


Fig. 18 – 2D trajectory of the 4WS UGV within the selected field (a) and zoom-in on the turning phase, with (b) and without (c) mesh.

due to the enforcement of non-holonomic constraints implemented by the DWA that generated the smooth trajectory, which remained close to the given target nodes (within a given tolerance) while obtaining a feasible solution.

The obtained reference trajectory was then used *online* by the controller, which defined in real-time the proper control action to be applied in terms of the angular velocity of the wheels, according to the UGV's current position and orientation. The resulting controlled trajectory is represented by the yellow line in Fig. 18. Also in this case, a MATLAB/Simulink simulator was created and used to test the GNC strategy, running the SIL tests on an Intel Core i7-7500U with a CPU @2.70 GHz, a 16 GB RAM and a 512 GB solid-state drive. Figure 18 shows an overview of the trajectory performed by the UGV along the vineyard rows overlapped with the low-complexity map (Fig. 18a), with a zoom-in on the turning phase (Fig. 18b and Fig. 18c), which was the most complex part of the mission since it was mandatory to guarantee no slippage and complete adherence to the terrain.

It is possible to notice that the vehicle was able to properly face this turning maneuver by remaining within the turning area while avoiding not only slippage (as also shown in Fig. 19b), but also the obstacles by remaining close to the reference trajectory. Indeed, it is possible to notice in Fig. 18b and Fig. 18c the adherence between the reference and the controlled trajectories, which highlights the effectiveness of the proposed cascade control scheme.

Last, in Fig. 19a, it is possible to observe the resulting behavior of the (CoM of the) vehicle in terms of position along the reference frame axes $(x_G, y_G)$and heading θ. On the other hand, in Fig. 19b it is possible to notice the comparison among the control actions required by the steering controller in terms of desired steering angles for the front $(\delta_F)$and rear (δ_R) axes (black lines) with respect to the actuated ones (red lines).Next,

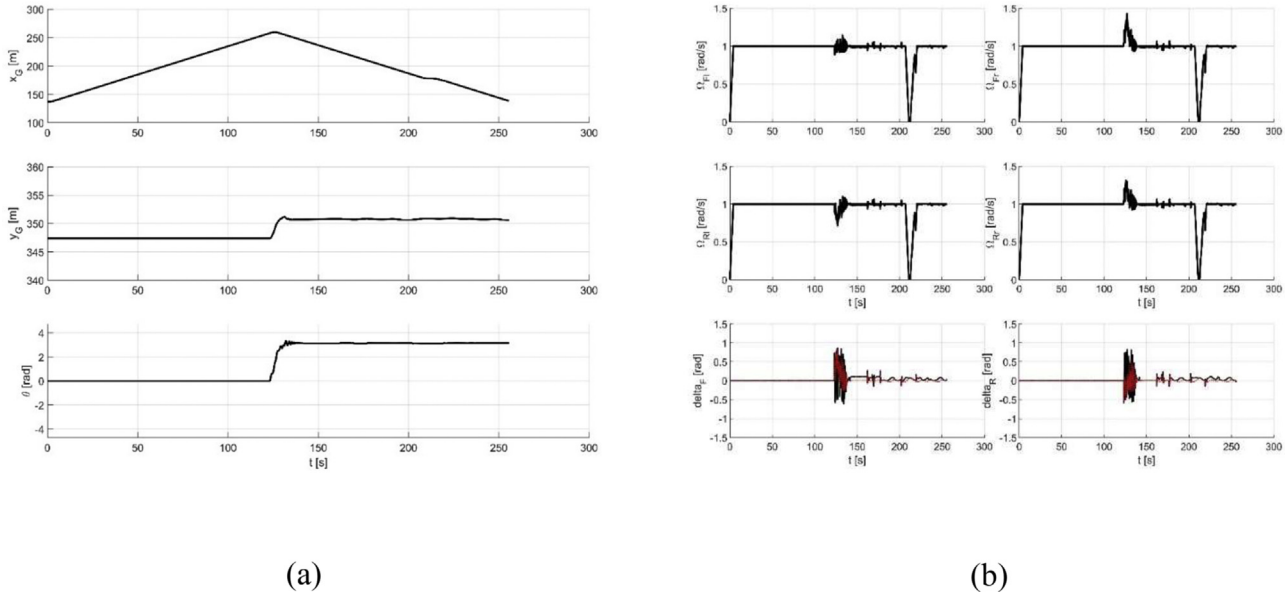


Fig. 19 – UGV position and orientation along the path (a) and corresponding control actions (b).

the optimal angular velocity profile for each wheel Ω_{ij}, defined by the velocity optimiser, is also shown.

4.2. *In-field operations with the Q4T quadrotor*

To ensure the fully-autonomous navigation of the RW-UAV while performing in-field operations, it is essential to provide optimal and efficient GNC capabilities to the RW-UAV. To this end, flyable GNC schemes must be implemented on board of the UAV autopilot to guarantee high efficiency and manoeuvrability in precision farming.[15] For this work, the guidance strategy adopted for the in-field operations via UAV is a combination of the ones adopted for the UGV, i.e. RRT* as the global path planner, DWA as the local planner, and the terrain following approach exploited for the FW-UAV, modified according to the flight requirements for this scenario, i.e. maintaining a relative altitude, with respect to the terrain model, of 1.5 m. To guarantee the fulfilment of the mission and of the system and safety requirements, despite the presence of external and internal disturbances and unmodelled uncertainty sources, and to properly track the reference trajectory, advanced control techniques were required, which combined robust optimisation and predictive control strategies while targeting reduced control effort and computational load without compromising drone endurance and performance. The class of control algorithms selected for this work involves model predictive control (MPC) schemes, because of their ability to explicitly handle constraints. The MPC philosophy can be simply described as follows: to predict future behaviour by using a system model, given the measurements or estimates of the current state of the system and a hypothetical future input trajectory or feedback control policy. For this work, a classic linear MPC (Mayne & Rawlings, 2009) was used, whose control design guidelines are not reported for brevity (since they retrace the ones of the TRMPC described in Section 4.1 for the FW-UAV). In the next section, the preliminary results obtained by exploiting the aforementioned GNC strategy are provided and described in detail.

4.2.1. *Preliminary results*

Within the proposed mission concept, the Q4T quadrotor was called to fly within the vineyard rows to perform a generic in-field task, e.g. scouting. To comply with the mission requirements, the RW-UAV had to fly about 1.5 m above the terrain while maintaining a cruise speed of 3 m s^{-1} and a minimum distance from the rows of about 1 m. As anticipated in the previous section, the reference trajectory was obtained by combining the 2D path $(x_{ref}, y_{ref}, \psi_{ref})$ designed for the UGV, starting from the occupancy grid map, together with a terrain following algorithm that, according to the simplified digital elevation model of the terrain, defines the reference altitude profile h_{ref}. This reference trajectory, designed *offline* and shown in Fig. 20 by a yellow line, was uploaded into the GNC block of the RW-UAV and it was used *online* as an input signal for the MPC controller, which had to guarantee the tracking of the reference signals, allowing at most a 10% deviation while enforcing actuation constraints on the available thrust F_z^B and torque $\tau = [\tau_\varphi, \tau_\theta, \tau_\psi]$ (see Q4T features described in § 1). The main difference with respect to the UGV's trajectory can be observed in Fig. 20b. Firstly, the take-off and landing phases need to be introduced.

Next, we envisioned the need to bring the UAV back to its initial location, once it reached the end of the selected area. To carry this out, an additional waypoint was added with the same latitude and longitude as the end of the vineyard but with a higher altitude (i.e. 3 m above the terrain). This allowed

[15] Details on nonlinear kinematical and dynamical models for quadrotors can be found in Sabatino (2015), Powers et al. (2015), and in Mammarella et al. (2022).

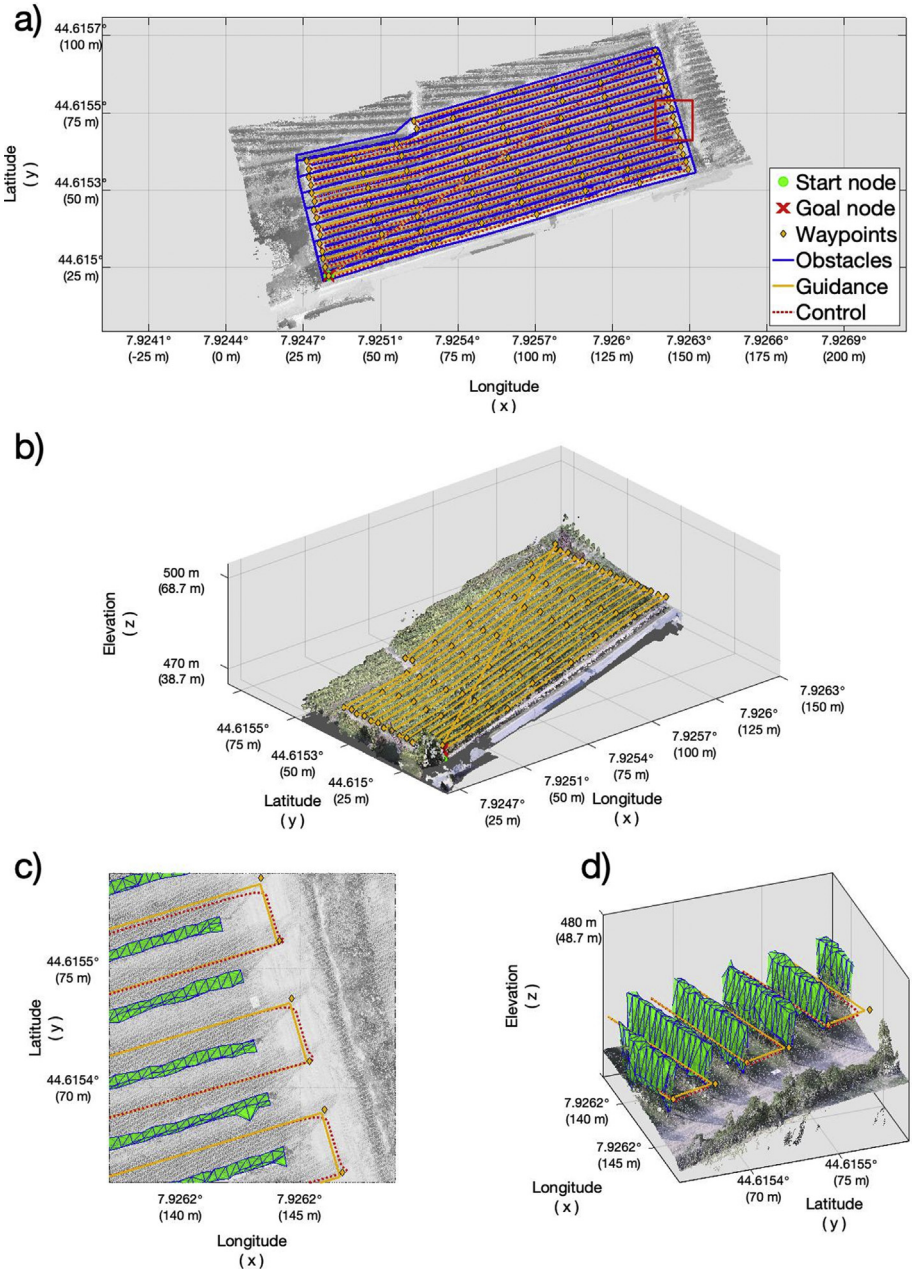
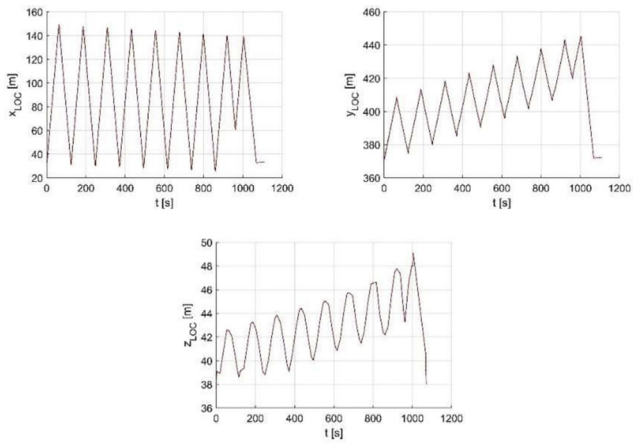


Fig. 20 – 2D (a) and 3D (b) quadrotor within rows and zoom-in over the turning phases (c)–(d).

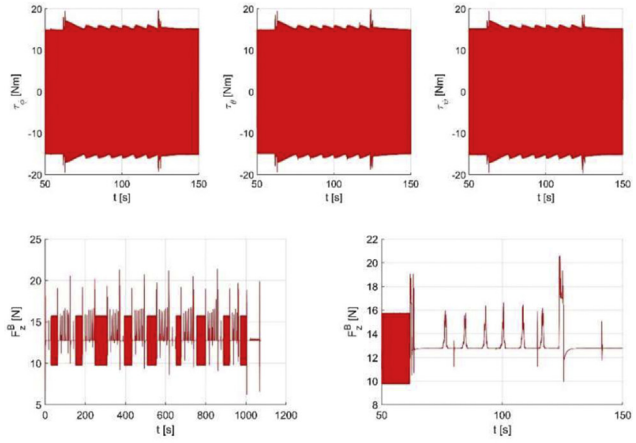
the UAV to diagonally overfly the entire vineyard and to return home without colliding with any vine row.

The preliminary simulation results are represented in Figs. 20 and 21. For the simulations, a MATLAB/Simulink simulator was used, running on an Intel Core i7-7500U with a CPU @2.70 GHz, a 16 GB RAM and a 512 GB solid-state drive. In Fig. 20a and Fig. 20b respectively, the complete 2D and 3D flight tracks performed by the UAV are represented within the selected parcel also during turning phases, highlighting the accuracy obtained with the selected guidance and control strategy when flying within the rows whilst maintaining a safe distance from the (static) obstacles. This latter aspect can be better observed in the 2D and 3D zoom-ins in Fig. 20c and Fig. 20d, respectively, where it is also possible to notice the effective tracking within the vine rows, represented by the blue mesh.

The comparison between the reference signals (fed to the controller from the guidance block) and the recorded tracks are depicted in Fig. 21(a), highlighting the efficacy of the control scheme to properly track the given trajectory while fulfilling the state deviation constraints. Moreover, it is possible to observe that during the turning phases, the system was able to properly track the position signal. In Fig. 21(b), the control



(a)



(b)

Fig. 21 – (a) Reference state signals (dashed black line) compared with real flown data (red line) and (b) control actions in terms of thrust F_z^B and torque components τ_i. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

actions are reported in terms of total thrust F_z^B (commanded to all four rotors) and control torques with respect to the x/roll (τ_φ), y/pitch (τ_θ), and z/yaw (τ_ψ) directions. Again, it is possible to observe that, in addition to the actuation constraint satisfaction, a higher control demand was required during the turning phase (as highlighted in the zoom-in in the time range $t \in [50, 150]$ s).

5. Conclusions

This companion paper to Mammarella et al. (2022) proposed an operating scheme based on the cooperation among heterogeneous UVs to conduct agricultural operations. This multi-phase approach is aimed at optimising the cooperation among the UVs in an agricultural scenario, maximising the information gathered from them, which can then be exploited by the other UVs. In particular, our attention was focused on those technologies that make it possible to comply with the terrain's, to exploit fleets of drones providing robust control, to minimise the development of autonomous driving software by the use of offline and co-shared optimal trajectory definition, and complying with onboard computational capabilities, such as the generation of low-complexity georeferenced maps. In this part of our work, the multi-phase approach introduced in our companion paper (Mammarella et al., 2022), was applied to a specific scenario where commercial vehiclesand a lab prototype were operated. Preliminary results highlighted the benefits that can be achieved by exploiting tailored technologies and algorithms that are selected and applied to improve the analysed mission phases.

Funding

This research was partially funded by the project "New technical and operative solutions for the use of drones in Agriculture 4.0" (PRIN 2017; Prot. 2017S559BB).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the Azienda Agricola G.D. Vajra for hosting the experiments and MAVTech Srl for driving the UAV-camera system.

REFERENCES

Blanchini, F., & Miani, S. (2008). *Set-theoretic methods in control*. Boston: Springer (Birkhäuser).

Capello, E., Guglieri, G., & Quagliotti, F. (2013). A waypoint-based guidance algorithm for mini UAVs. *IFAC Proceedings Volumes*, 46(30), 120–125. https://doi.org/10.3182/20131120-3-FR-4045.00005 (IFAC-PapersOnline).

Comba, L., Biglia, A., Ricauda Aimonino, D., & Gay, P. (2018). Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. *Computers and Electronics in Agriculture*, 155, 84–95. https://doi.org/10.1016/j.compag.2018.10.005

Comba, L., Zaman, S., Biglia, A., Ricauda Aimonino, D., Dabbene, F., & Gay, P. (2020). Semantic interpretation and complexity reduction of 3D point clouds of vineyards. *Biosystems Engineering*, 197, 216–230. https://doi.org/10.1016/j.biosystemseng.2020.05.013

Donati, C., Mammarella, M., Comba, L., Biglia, A., Dabbene, F., & Gay, P. (2021). Improving agricultural drone localization using

georeferenced low-complexity maps. In *2021 IEEE international workshop on metrology for agriculture and forestry (MetroAgriFor)* (pp. 372–377). https://doi.org/10.1109/MetroAgriFor52389.2021.9628607

Dong, Y., Zhang, Y., & Ai, J. (2017). Experimental test of unmanned ground vehicle delivering goods using RRT path planning algorithm. *Unmanned Systems, 5*(1), 45–57. https://doi.org/10.1142/S2301385017500042

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine, 4*(1), 23–33. https://doi.org/10.1109/100.580977

Gallo, R., Ristorto, G., Bojeri, A., Zorzi, N., Rinaldi, M. F., Sauli, G., & Mazzetto, F. (2019). Design a Web Platform to manage environmental monitoring information to be used in multicriteria evaluations of Green Infrastructures. *IOP Conference Series: Earth and Environmental Science, 275*(1). https://doi.org/10.1088/1755-1315/275/1/012005

González, D., Pérez, J., Milanés, V., & Nashashibi, F. (2016). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems, 17*(4), 1135–1145. https://doi.org/10.1109/TITS.2015.2498841

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research, 30*(7), 846–894. https://doi.org/10.1177/0278364911406761

Kouvaritakis, B., & Cannon, M. (2015). Model predictive control: Classical, robust and stochastic. In *Advanced textbooks in control and signal processing*. Springer.

Kunchev, V., Jain, L., Ivancevic, V., & Finn, A. (2006). Path planning and obstacle avoidance for autonomous mobile robots: A review. *Lecture Notes in Computer Science, 4252*, 537–544. https://doi.org/10.1007/11893004_70. LNAI-II.

Mammarella, M., & Capello, E. (2020b). Tube-based robust MPC processor-in-the-loop validation for fixed-wing UAVs. *Journal of Intelligent and Robotic Systems, 100*, 239–258. https://doi.org/10.1007/s10846-020-01172-6

Mammarella, M., Capello, E., Park, H., Guglieri, G., & Romano, M. (2017). Spacecraft proximity operations via tube-based robust model predictive control with additive disturbances. In *68th international astronautical congress*. Adelaide, Australia.

Mammarella, M., Capello, E., Park, H., Guglieri, G., & Romano, M. (2018). Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance. *Aerospace Science and Technology, 77*, 585–594. https://doi.org/10.1016/j.ast.2018.04.009

Mammarella, M., Comba, L., Biglia, A., Dabbene, F., & Gay, P. (2020a). Cooperative agricultural operations of aerial and ground unmanned vehicles. *IEEE International Workshop on Metrology for Agriculture and Forestry*, 224–229. https://doi.org/10.1109/MetroAgriFor50201.2020.9277573

Mammarella, M., Comba, L., Biglia, A., Dabbene, F., & Gay, P. (2022). Cooperation of unmanned systems for agricultural applications: A theoretical framework. *Biosystem Engineering*. https://doi.org/10.1016/j.biosystemseng.2021.11.008. In press.

Mammarella, M., Ristorto, G., Capello, E., Bloise, N., Guglieri, G., & Dabbene, F. (2019). Waypoint tracking via tube-based robust model predictive control for crop monitoring with fixed-wing UAVs. In *2019 IEEE international workshop on metrology for agriculture and forestry* (pp. 19–24). https://doi.org/10.1109/MetroAgriFor.2019.8909260

Mascarello, L. N., Quagliotti, F., & Ristorto, G. (2017). A feasibility study of an harmless tiltrotor for smart farming applications. In *2017 international conference on unmanned aircraft systems* (pp. 1631–1639). https://doi.org/10.1109/ICUAS.2017.7991374

Mayne, D. Q., & Rawlings, J. B. (2009). *Model predictive control: Theory and design*. Nob Hill Publishing.

Micasense. (2015). Micasense RedEdge© 3 multispectral camera - user manual. *Rev, 6*.

Ögren, P., & Leonard, N. E. (2005). A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics, 21*(2), 188–195. https://doi.org/10.1109/TRO.2004.838008

Powers, C., Mellinger, D., & Kumar, V. (2015). Quadrotor kinematics and dynamics. In *Handbook of unmanned aerial vehicles* (pp. 307–328). https://doi.org/10.1007/978-90-481-9707-1_71

Sabatino, F. (2015). *Quadrotor control: Modeling, nonlinear control design, and simulation*. EES Examensarbete / Master Thesis.

Sujit, P. B., Saripalli, S., & Sousa, J. B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Systems, 34*(1), 42–59. https://doi.org/10.1109/MCS.2013.2287568

Zhang, F., Li, N., Xue, T., Zhu, Y., Yuan, R., & Fu, Y. (2019). An improved dynamic window approach integrated global path planning. In *2019 IEEE international conference on robotics and biomimetics* (pp. 2873–2878). https://doi.org/10.1109/ROBIO49542.2019.8961684