H2020-ICT-25-2016-2017

HYbrid FLying rollIng with-snakE-aRm robot for contact inSpection

HYFLIERS

D4.3

Support platform and services

Contractual date of delivery	31 May 2022
Actual date of delivery	7 Oct 2022
Editor(s)	Ulrico Celentano (UOULU)
Author(s)	Marko Kauppinen, Miika Sikala, Timo Mäenpää, Ulrico Celentano (UOULU), Álvaro Caballero, Guillermo Heredia (USE), Marco Montes, Pedro J. Sanchez-Cuevas, Miguel A. Trujillo (CATEC)
Workpackage	WP4
Estimated person-months	27
Dissemination level	PU
Type	R
Version	1.0
Total number of pages	49

Abstract:

This document presents the final architecture of the ground station, its subsystems, and its realisation; the acquisition of inspection data from the ultrasound transducer thickness sensor, their processing into a database, and their visualisation; environment representation and path planning and navigation support for the hybrid robot with awareness of aerodynamic effects of obstacles.

Keywords:

Aerodynamic awareness. Autonomous motion. Battery management. Database. Data management. Data visualisation. Dynamic awareness. Hybrid robot. Navigation support. Operations support. Path planning. Pipe inspection. Robot operating system. Support platform. Unmanned aerial vehicle.

Executive summary

This deliverable extends the developments presented in the earlier deliverables [D4.1, D4.2]. More in particular: it provides the final architecture of the ground station and its subsystems; it presents the inspection data acquisition from the ultrasound sensor on the hybrid robot's satellite/arm to the thickness sensing subsystem and further to the inspection data processing into a relational database and their visualisation for the inspection engineer; it illustrates the outcomes on environment representation and path planning and navigation support for a hybrid robot with robotic arm and awareness of aerodynamic effects of obstacles found in oil and gas plants; it gives final considerations on the mobile support platform, including energy management.

Abbreviations

3D	three-dimensional
ADA	aerodynamics awareness
API	application programming interface
ARM	aerial robotic manipulator
ARS	aerial robotic system
BMS	battery management system
BRIEF	binary robust independent elementary feature
CRUD	create, read, update, delete
DA	dynamics awareness
DMBS	database management system
ENU	East-North-up
FOV	field of view
FSH	full screen height
GS	ground station
GSDB	ground station database
HMR	hybrid mobile robot
HR	hybrid robot (either HMR or HRA)
HRA	hybrid robot with arm
HTTP	hypertext transfer protocol
HYFLIERS	hybrid flying rolling with-snake-arm robot for contact inspection
IO	input and output
JSON	JavaScript object notation
$LiFePO_4$	lithium iron phosphate
LRM	long-reach manipulation
MP	motion planner
MSP	mobile support platform
ORB	oriented fast and rotated BRIEF
PC	personal computer
PCB	printed circuit board
PID	proportional-integral-derivative
PHP	PHP hypertext preprocessor
PWM	pulse-width modulation
REST	representational state transfer
RRT	rapidly-exploring random tree
SLAM	simultaneous localization and mapping
UART	universal asynchronous receiver-transmitter
UAV	unmanned aerial vehicle
UDP	user datagram protocol
UI	user interface
UT	ultrasound transducer

Wi-Fi	Wireless Fidelity (synonym for WLAN, wireless local area network)

Table of Contents

List of Figures

List of Tables

1. Introduction

This deliverable extends the developments presented in earlier WP4 deliverables [D4.1, D4.2]. This report starts in Section 2 by putting into the context of the WP4 objective and tasks the achievements reported in this deliverable D4.3. The rest of the report provides details on those. Correspondingly, Section 3 describes the final architecture for the mobile ground support platform with its subsystems. Further, Section 4 reports the developments on the inspection data management, detailing the research illustrated in D4.1. Additionally, Section 5 provides evidence of the work done in path planning and navigation support, extending and complementing what told in D4.1 and D4.2. Moreover, Section 6 shows more on the mobile support platform itself, outspreading the contents of D4.1 and D4.2. Finally, Section 7 provides conclusions about this report.

2. Support services

The main objective of WP4 concerns support services for the inspection operations of the HYFLIERS system. These support services can be divided into those supporting the inspection engineer in effectively performing the pipe inspection tasks, and those supporting the pilot and more generally the flight tasks in safely bringing the unmanned aerial vehicle in through the inspection points. This and the previous deliverables have covered the above as follows.

- Support to inspection engineer: measurement data management (D4.1), and inspection data acquisition, processing and visualisation (here in D4.3);
- Support to the pilot and flight tasks in general: path planning and navigation support (D4.1, D4.2 and here in D4.3), augmented reality applications (D4.2);
- Support platform for supporting functions: Mobile ground support platform and energy management (D4.1, D4.2 and here in D4.3).

3. Ground station subsystems (T4.3 part I)

As outlined in deliverable D4.1 [D4.1] and reminded shortly above, the ground station system has the role of supporting both the inspection engineer and the safety pilot in their respective tasks. To achieve this, the ground station communicates with the hybrid robot (HR), i.e., the unmanned aerial vehicle (flyer) and the ultrasound transducer (UT), with these two constituting the aerial segment of the system. On the ground segment, messaging is divided into the part relevant to the pilot and the one serving the inspection engineer, within the flight and the inspection subsystem, respectively. The former makes use of a pilot terminal, whereas the latter exploits an UT ground subsystem and a ground station (GS) subsystem including a database (DB) for the inspection data and a user interface (UI) for their visualisation. Communication between the aerial and the ground segment is obviously wireless, whereas communication between subsystems the respective aerial and ground subsystems is cabled, see Figure 1.

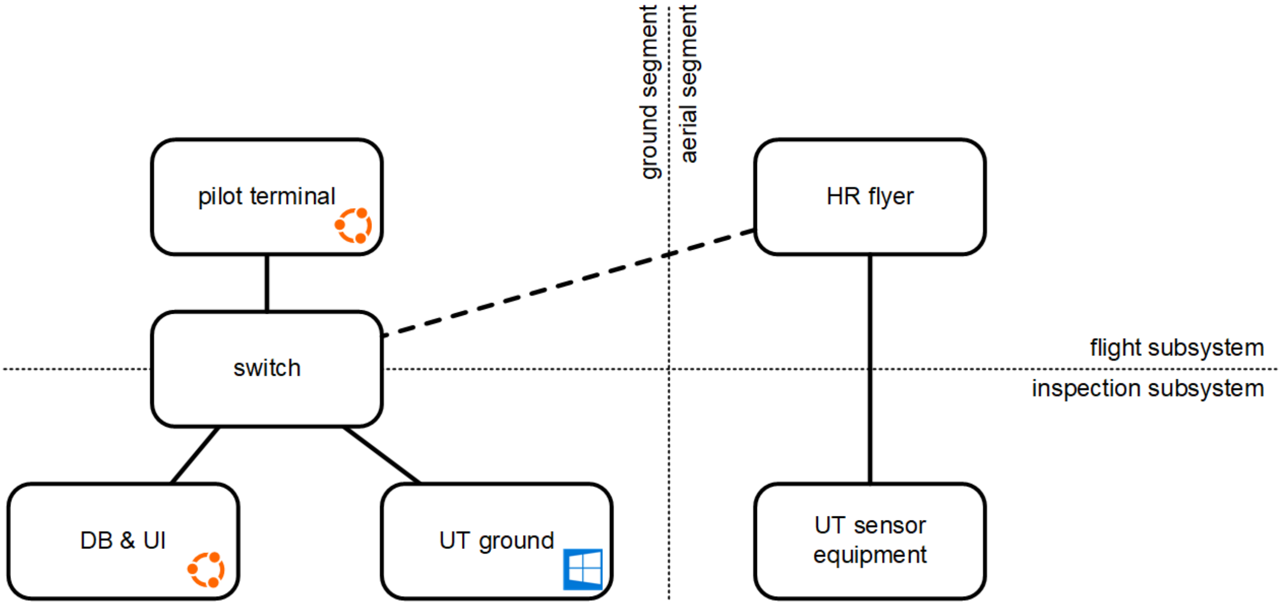


Figure 1. Architecture for the mobile support platform, from the ultrasound transducer (UT), through the hybrid robot (HR), to the mobile platform, including the UT ground subsystem, the database (DB), the operator's user interface (UI), and the user interface for the pilot. A solid line represents a wired connection, whereas a wireless link is depicted by a dashed line.

Section 4 focuses on the inspection subsystem and in particular it describes the inspection data acquisition, processing and visualisation. Correspondingly, Section 5 concentrates on the flight subsystem and it talks about path planning and navigation support. Finally, Section 6 reports about the mobile support platform realisation supporting the above supporting functionalities.

4. Inspection data acquisition, processing and visualisation (T4.2)

This section describes the advances in task T4.2 “T4.2: Measurement data management”. First a short subsection summarises the earlier work, then the outcomes of the most recent activities are described.

4.1. Introduction

The previous WP4 deliverable D4.1 described the architecture for the measurement data management service in pipe inspection operations, including the description of the communication and information flow between the hybrid robot and the device comprising the measurement sensor (i.e., the satellite or arm/tool) and the ground platform providing support to the inspection engineer and the safety pilot. In particular, D4.1 described:

- The inspection data service and its architecture, put into the context of oil and gas plants pipe inspection, with an emphasis on the measurement data from collection to delivery at end user.
- The data management environment with its hardware and software components, data types, involved interfaces, structure of the messages, and sequence of the messaging among the architectural elements, illustrated with the help of sequence diagrams.

4.2. Inspection data acquisition

Inspection data are generated by the UT sensor equipment onboard the satellite (in case of HMR) or the arm/tool (for HRA). Data travel through the HR further to the UT ground subsystem, see Figure 1 in Section 3.

The communication between the onboard UT equipment and the ground UT subsystem is over wireless UDP (user datagram protocol). The ground UT subsystem computer is Windows 10 64 bit.

The measurement file includes the samples of the A-scan signal as rows of measurements, see Table 1, where N is the number of samples of the A-scan signal, and M is the number of the A-scan signals. Each sample of the A-Scan signal is expressed in %FSH (Full Screen Height). More in particular, $M{=}Crawler\_Path\_Length(mm)Encoder\_Trigger\_Resolution(mm)$; A is the amplitude of the back wall echo in %FSH, T is the thickness of the pipe in mm, D is the distance of the encoder in mm.

Table 1. Output of the UT subsystem

A-scan signal	Amplitude (% FSH)	Thickness (mm)	Encoder position (mm)
S(1,1) S(1,N)	A(1)	T(1)	D(1)
…	…	…	…
S(M,1) S(M,N)	A(M)	T(M)	D(M)

Measurements are saved on a local disk with the following directory structure:

ROOT/{YYYYMMDD}/{PIPEno}/{PARTno}/{PIPEno}_{PARTno}_{PIPETYPE}_{SCANLINE}_{SCANID}.txt

where PIPETYPE is either STRAIGHT or ELBOW, and SCANLINE is TOP12, LAT3, BOT6, or LAT9, see Figure 2.

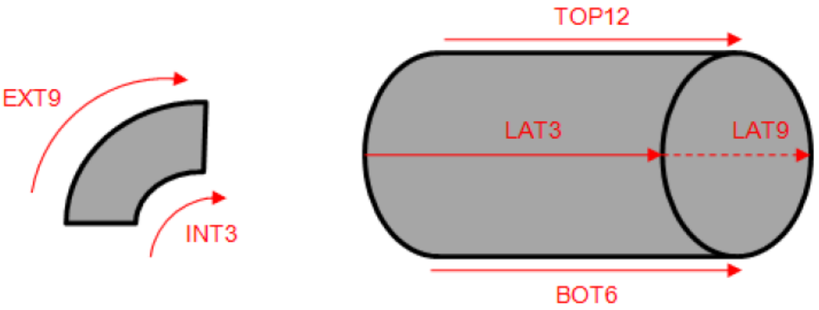
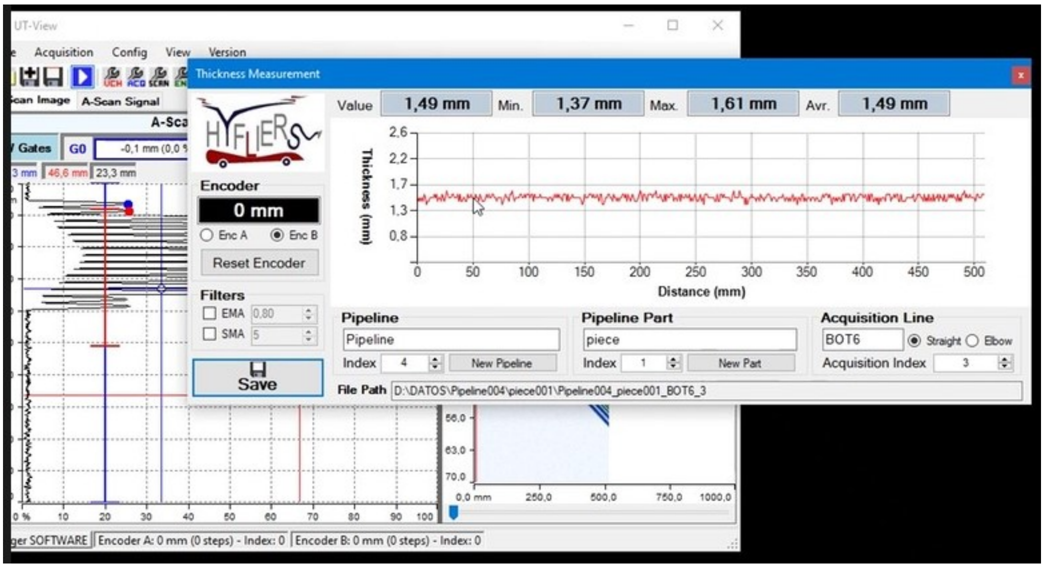


Figure 2. Crawler's paths on the pipe: on top (TOP; 12 clock position), bottom (BOT; i.e. 6 clock position), and lateral (LAT; 3 and 9 clock position) on a straight pipe, and external (EXT) and internal (INT) parts of an elbow.

SCANID is a running number distinguishing different executions of measurements over the same pipe and part. PIPEno and PARTno are entered manually by the operator (inspection engineer) using letters, numbers, dash and point. PIPEno and PARTno could be the *Functional Location* and the *Measuring Point Position*, according to the conventions provided by TotalEnergies and compatible with the reference system used in Chevron.



Value 1,49 mm Min. 1,37 mm Max. 1,61 mm Avr. 1,49 mm

Encoder 0 mm ○ Enc A ● Enc B

☐ EMA 0,80
☐ SMA 5

Pipeline: Pipeline, Index 4

Pipeline Part: piece, Index 1

Acquisition Line: BOT6 ● Straight ○ Elbow, Acquisition Index 3

File Path D:\DATOS\Pipeline004\piece001\Pipeline004_piece001_BOT6_3

Encoder A: 0 mm (0 steps) - Index: 0 | Encoder B: 0 mm (0 steps) - Index: 0

Figure 3. The UT view on the UT ground subsystem computer.

The GS Linux computer mounts the directory shared by the Windows computer. In this way, inspection data can be transferred to the GSDB.

4.3. Inspection data processing

Data are acquired on the GSDB subsystem for further processing and interaction with the operator through the dedicated UI. Following the procedure illustrated in Section 4.2, the UT sensor, set in position, sends the readings to the ground subsystem, and through it, to the GSDB subsystem, that stores them.

A Python script (UT_file_watcher.py) is responsible for the acquisition of the inspection data made available by the UT subsystem, by reacting to new files appearing in the shared directory (see Section

4.2) received as an argument. Another script (GSDB_api.py) provides an interface to the database (according to https://spacha.dev/gsdb-api).

The database [Sik2022] software is implemented as a representational state transfer (REST) application programming interface (API) using Laravel 8 (https://laravel.com/docs/8.x) and PHP framework, and it is designed to work on a web server such as Nginx (https://www.nginx.org/). A web-based realisation offers the flexibility of easily implementing the database locally or on the cloud.

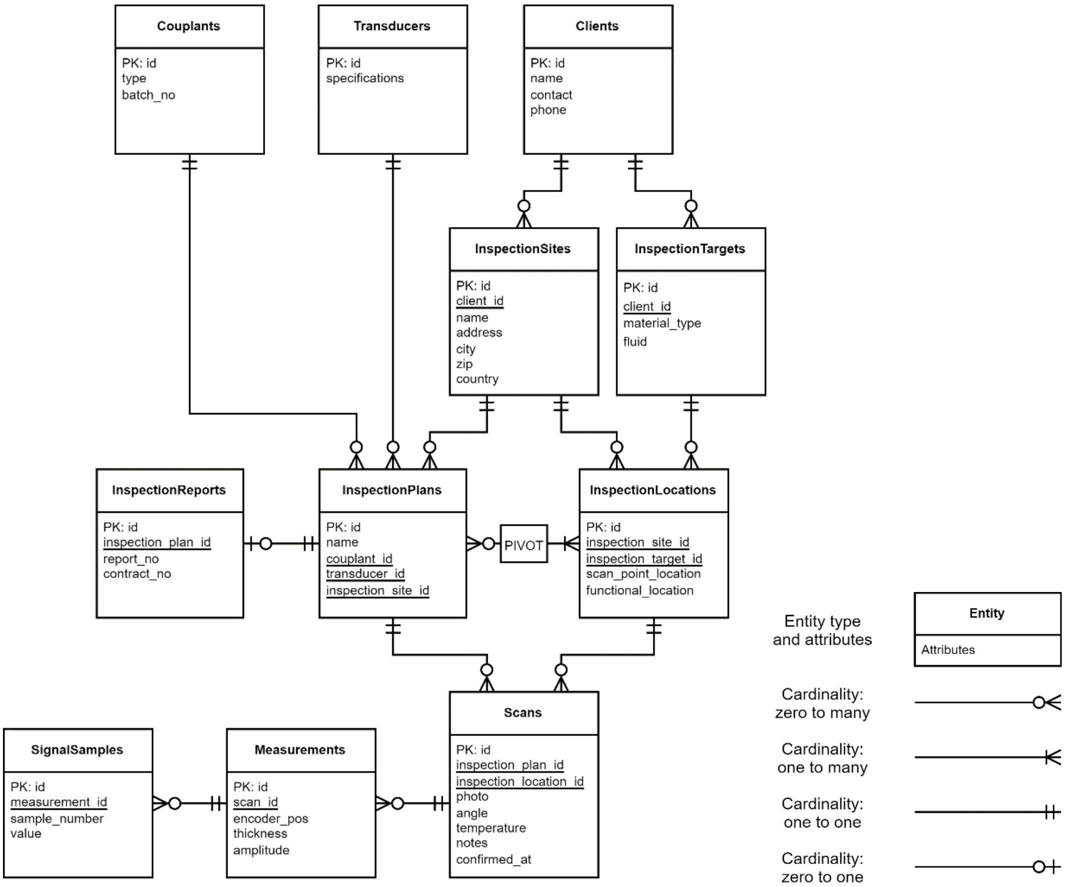


Figure 4. Relationships between the entities in the inspection database structure. At the bottom right of the figure is explained the convention (crow's foot notation) used in the main part of this illustration, that defines the cardinality of the relationships between entities. PK denotes the attribute that is used as primary key. Underlined attributes are references to primary keys of other entities.

The database stores all the relevant information (see D4.1) and it is realised as a relational database management system (DBMS). The entities implemented in the database are:

- *Client*: Represents a single organization utilizing the software.
- *InspectionSite*: Represents a physical site owned by a client.
- *InspectionLocation*: A single physical inspection location within a site.
- *InspectionTarget*: Describes physical and other properties of a pipe or other inspection target.
- *InspectionPlan*: Represents an inspection plan for a site. Contains a list of inspection locations.
- *Scan*: Contains a scan from single inspection location and certain angle on the pipe, possibly with a photo of the location.

- *Measurement*: Contains a signal and thickness measurement from single inspection location, angle, and encoder position.
- *SignalSample*: A single sample of the signal belonging to given measurement.
- *InspectionReport*: Represents an inspection report generated from a finished inspection plan.
- *Couplant*: A type of couplant used in inspection. Can affect the measurement results.
- *Transducer*: The ultrasonic transducer used in inspection.

The relationship between the above entities is depicted in Figure 4. Table 2 lists in more details the structure of the database.

Table 2. Database structure. All regular tables have ID as primary key and timestamps, as listed under the *common fields*.

field	type	length	extra
common fields			
id	BIGINT	20	Auto increment
created_at	TIMESTAMP		Nullable
updated_at	TIMESTAMP		Nullable
clients (table)			
name	VARCHAR	255	
contact	VARCHAR	255	
phone	VARCHAR	255	
couplants (table)			
type	VARCHAR	255	
batch_no	VARCHAR	255	
inspection_locations (table)			
inspection_site_id	INT	11	
inspection_target_id	INT	11	
scan_point_location	VARCHAR	255	
functional_location	VARCHAR	255	
inspection_plans (table)			
name	VARCHAR	255	
couplant_id	INT	11	
transducer_id	INT	11	
inspection_site_id	INT	11	
inspection_location_inspection_plan (pivot table)			
inspection_location_id	INT	11	
inspection_plan_id	INT	11	
inspection_reports (table)			
insepction_plan_id	INT	11	
report_no	VARCHAR	255	Nullable
contract_no	VARCHAR	255	
inspection_sites (table)			
client_id	INT	11	
name	VARCHAR	255	
address	VARCHAR	255	
city	VARCHAR	255	
zip	VARCHAR	255	
country	VARCHAR	255	

inspection_targets (table)			
client_id	INT	11	
material_type	VARCHAR	255	
fluid	VARCHAR	255	
scans (table)			
inspection_plan_id	INT	11	
inspection_location_id	INT	11	
photo	VARCHAR	255	Nullable
angle	DOUBLE	8;2	Nullable
temperature	DOUBLE	8;2	Nullable
notes	TEXT		Nullable
confirmed_at	TIMESTAMP		Nullable
measurements (table)			
scan_id	INT	11	
encoder_pos	DOUBLE	8;2	Nullable
thickness	DOUBLE	8;2	Nullable
amplitude	DOUBLE	8;2	Nullable
signal_samples (table)			
measurement_id	INT	11	
sample_number	INT	11	
value	DOUBLE	8;2	

Operations on the database are done via REST API, which is operated though CRUD (create, read, update, delete) operations with interactions represented by HTTP requests and responses containing JSON (JavaScript object notation) message bodies.

More in particular, the services are accessed in REST APIs using uniform resource identifiers (URI) and a fixed set of HTTP request methods (verbs), with each verb corresponding to an action towards a resource. Table 3 summarises the HTTP verbs and corresponding CRUD operations.

Table 3. HTTP verbs and corresponding CRUD operations used in RESTful API

HTTP verb	CRUD operation	Description
POST	Create	Create a new resource and return the ID of the resource
GET	Read	List all resources or get details of a specific one
PUT	Update/Replace	Update resource or replace an existing resource with given ID
PATCH	Update/Modify	Update some attributes of the resource with given ID
DELETE	Delete	Delete the resource with given ID

4.4. Inspection data visualisation

The operator’s UI, Figure 5, provides access to the data processed by the GSDB, Section 4.3.

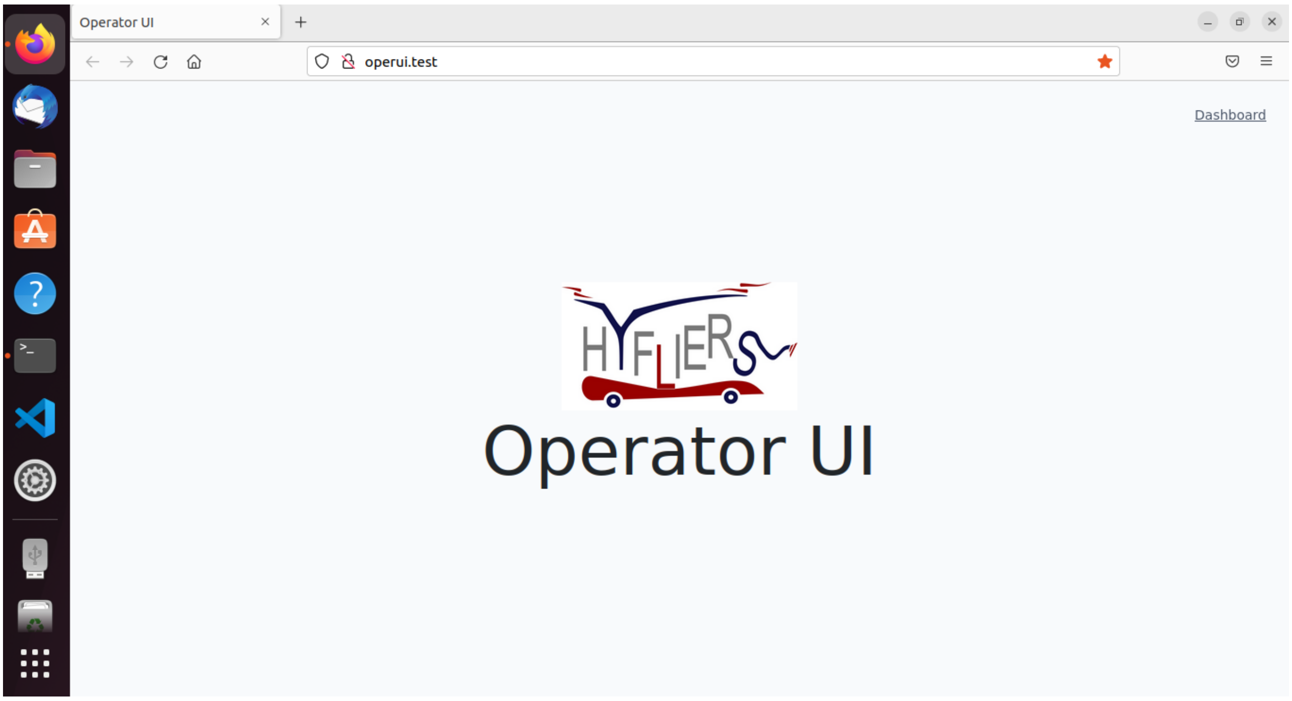
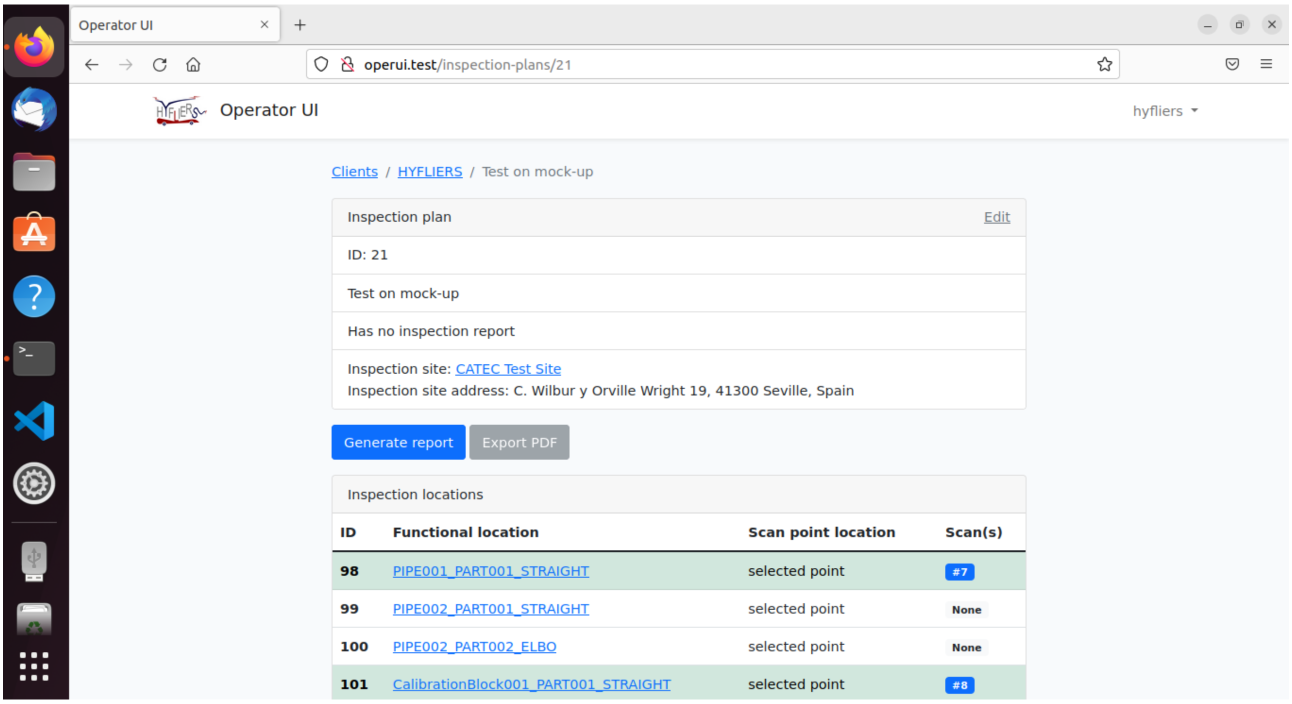


Figure 5. Welcome page of the Operator UI.

Figure 6 shows an example inspection plan before the measurements' completion (thus the mention of the missing inspection report). The UI reports the list of locations the HR and the UT must be brought to for inspection, identified by the logical coordinates represented by the *Functional Location* and the *Measuring Point Position*.



Clients / HYFLIERS / Test on mock-up

Inspection plan Edit

ID: 21

Test on mock-up

Has no inspection report

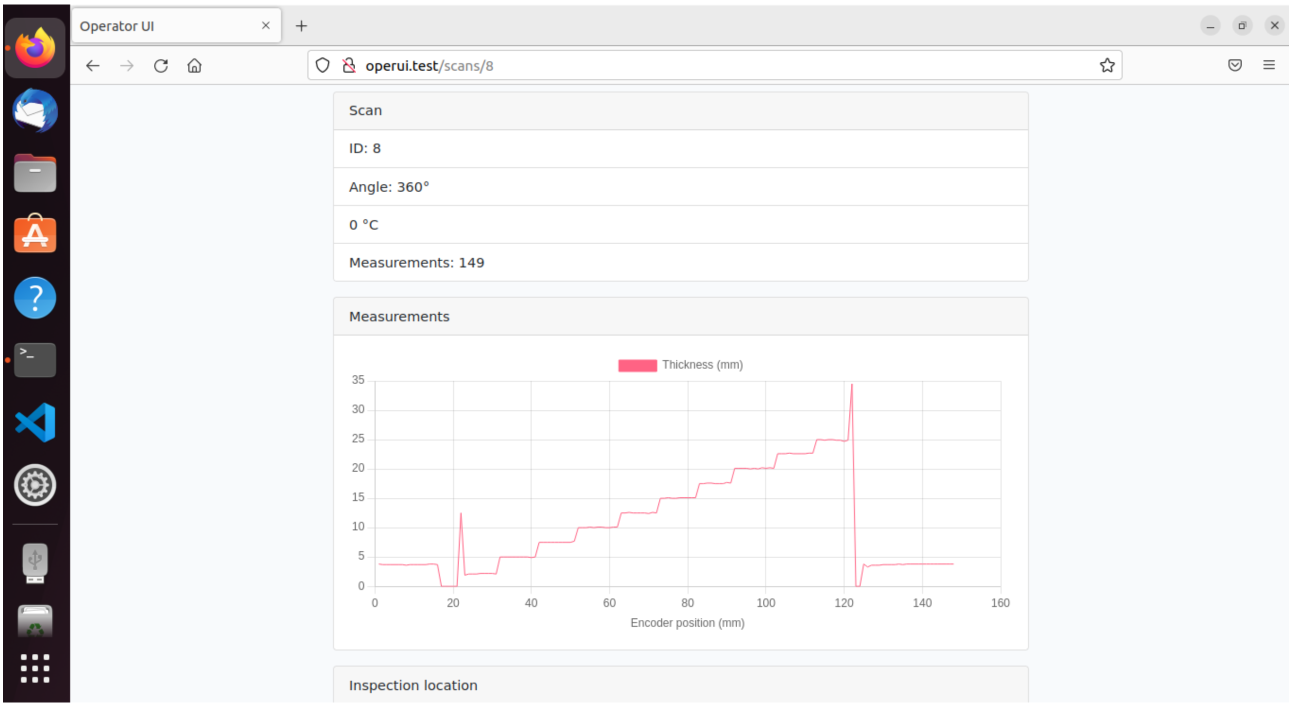
Inspection site: CATEC Test Site
Inspection site address: C. Wilbur y Orville Wright 19, 41300 Seville, Spain

Inspection locations

ID	Functional location	Scan point location	Scan(s)
98	PIPE001_PART001_STRAIGHT	selected point	#7
99	PIPE002_PART001_STRAIGHT	selected point	None
100	PIPE002_PART002_ELBO	selected point	None
101	CalibrationBlock001_PART001_STRAIGHT	selected point	#8

Figure 6. Example inspection plan.

As soon as inspection data are made available by the respective acquisition and processing subsystems, Sections 4.2 and 4.3, respectively, they are visualised on the interactive Operator UI, see an example given in Figure 7.



Scan

ID: 8

Angle: 360°

0 °C

Measurements: 149

Measurements

Inspection location

Figure 7. Visualisation of the measurement results on the operator’s UI. In this example, the calibration results are shown, cf. Figure 8.

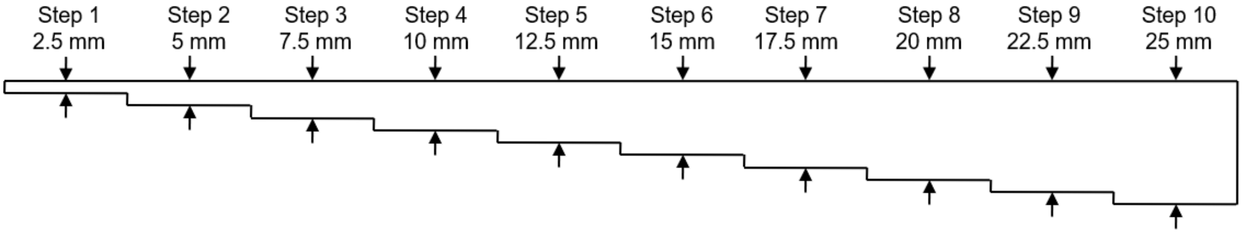


Figure 8. Calibration block and related calibration steps.

Once all the locations have been visited, the inspection engineer may generate the inspection report, through the button shown in Figure 6, so that it will be attached to the inspection plan.

5. Path planning and navigation support (T4.1 & T4.3)

This section presents the new advances developed within the framework of task T4.1 "3D navigation support" and task T4.3 "Mobile ground support" in the field of path planning and navigation support. For the sake of completeness, the following subsection starts summarising the contributions in previous deliverables D4.1 and D4.2 to put in context the new contributions. After that, a novel planning method with Aerodynamics Awareness is presented.

5.1. Introduction

Earlier deliverables D4.1 and D4.2 presented the design, implementation and validation of planning algorithms for the movement of the Hybrid Robot and its associated navigation support (obstacle detection, reactive behaviours, etc.). These algorithms [SuaEtal2020, CabEtal2022] paved the way for autonomous operation of the HR in its inspection tasks. The general overview of the three main research lines covered by those deliverables can be enumerated as follows:

- Generation of motion plans that allow autonomous motion along complete sequences of inspection points.
    - Consideration of the obstacles included in the maps of the industrial environments to generate trajectories free of collisions.
    - Consideration of the hybrid motion capabilities (flying, rolling) offered by the HR to generate optimal trajectories according to different indices (operation time, power consumption).
- Extensions of motion planner to guarantee safer trajectories for autonomous operation.
    - Consideration of dynamic behaviour of the system.
- Navigation support to enhance the level of autonomy offered by the system
    - Implementation of reactive behaviours in case the on-board sensors reveal the presence of obstacles not considered in the planning algorithms.

Deliverable D4.2 also presented a 3D (three-dimensional) augmented reality system to be part of the human operation supporting functions. This new type of interface for the hybrid robot operator will enrich the information available during the operation visually showing the status of the hybrid robot and its mission. This information will be provided to the operator in such a way that they do not have to change its field of view and hence, they can always maintain a look to the hybrid robot and its operation at any time.

This Section 5 extends the contributions in the second research line above by presenting a motion planning method that considers aerodynamic effects within the planning process [CabEtal2020]. The airflow generated by rotary-wing platforms is very influenced by the physical obstacles close to it. For this reason, the proximity of surfaces like the ground or a ceiling disturbs the natural motion of the air considerably, causing significant changes in the thrust generated by the rotors of the aerial platform. These aerodynamic phenomena, known respectively as ground effect and ceiling effect, can be dangerous for standard multi-rotors, provoking their destabilisation or even crashes. Aligned with this, the new planning method complements the Dynamics Awareness approach presented in deliverable D4.2. The new formulation implies that the expansion of the search tree will be based not only on the dynamics of the controlled system but also on its aerodynamical behaviour. To this end, a proper characterisation of the aerodynamic effects based on both theoretical and experimental considerations has been derived. This characterisation is considered within the trajectory generation process to discard states whose associated aerodynamic phenomena may provoke undesired collisions and to explore alternatives that lead to the most efficient trajectories within the area of safe operation.

This Section 5 also provides a complement to the activities for human operation supporting functions in the form of the assisted navigation with obstacle detection presented below.

5.1. HMR situational awareness and navigation support system

The HMR navigation support system consists of three complementary subsystems that aim to facilitate the operation of the robot and, at the same time, make it safer. These components are the obstacle detector, the environment representation and the assisted navigation mode.

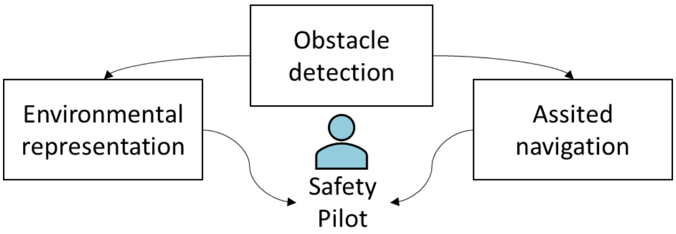


Figure 9. HMR navigation support system.

5.1.1. Obstacle detection

The first component of the HMR navigation support system is the obstacle detection. This module is responsible for detecting obstacles that may interfere with the robot's trajectory. To do this, we have limited the allowed movements of the HMR during the inspection operation and have installed three depth cameras on the robot. This limits the allowed movements of the HMR to flying forward, upward and downward, as these are the directions covered by the obstacle detection sensors.

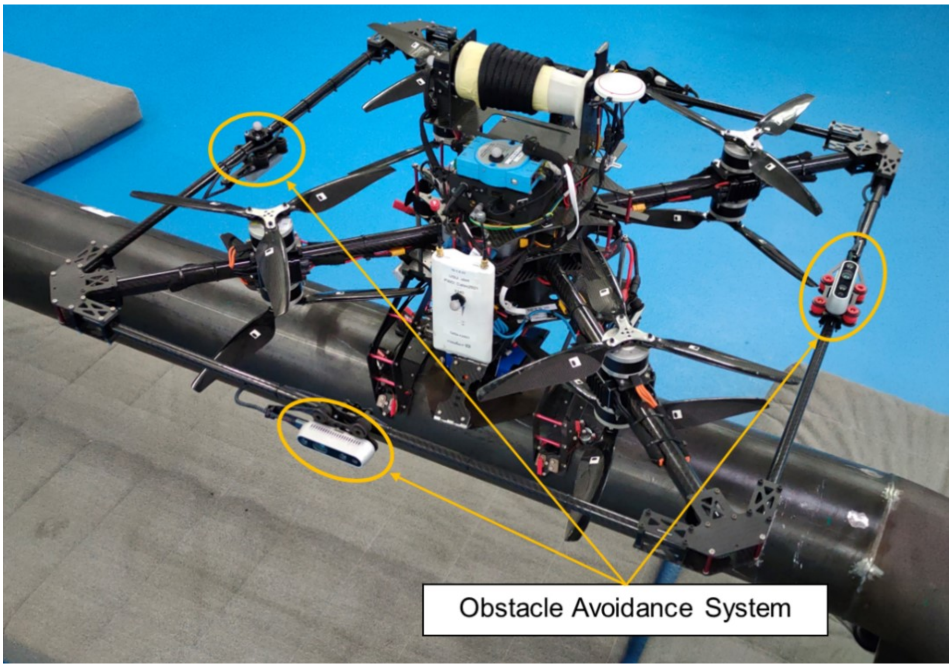


Figure 10. RealSense D435 position and orientation.

The information collected by the three cameras is combined by several transformations into a single point cloud. In addition, the position estimation provided by the localization system (see D2.2) is used to supplement the environment information needed by the mapping algorithm. This creates a 3D discrete occupancy grid of the environment. Then, based on the obstacle detector, the closest point of the robot to the obstacle is calculated using the K-d tree search algorithm from the PCL library. The information of the closest point is subsequently used by the assisted navigation node to enable or disable the robot's movement to avoid any potential collision of the robot with the environment.

5.1.2. Environment representation

As a complement to obstacle detection, the HMR has a representation of the environment that provides the pilot with a third-person view of the environment (see Figure 11). It is based on Rviz and uses the Octomap [HorEtal2013] to compute the occupancy grid.

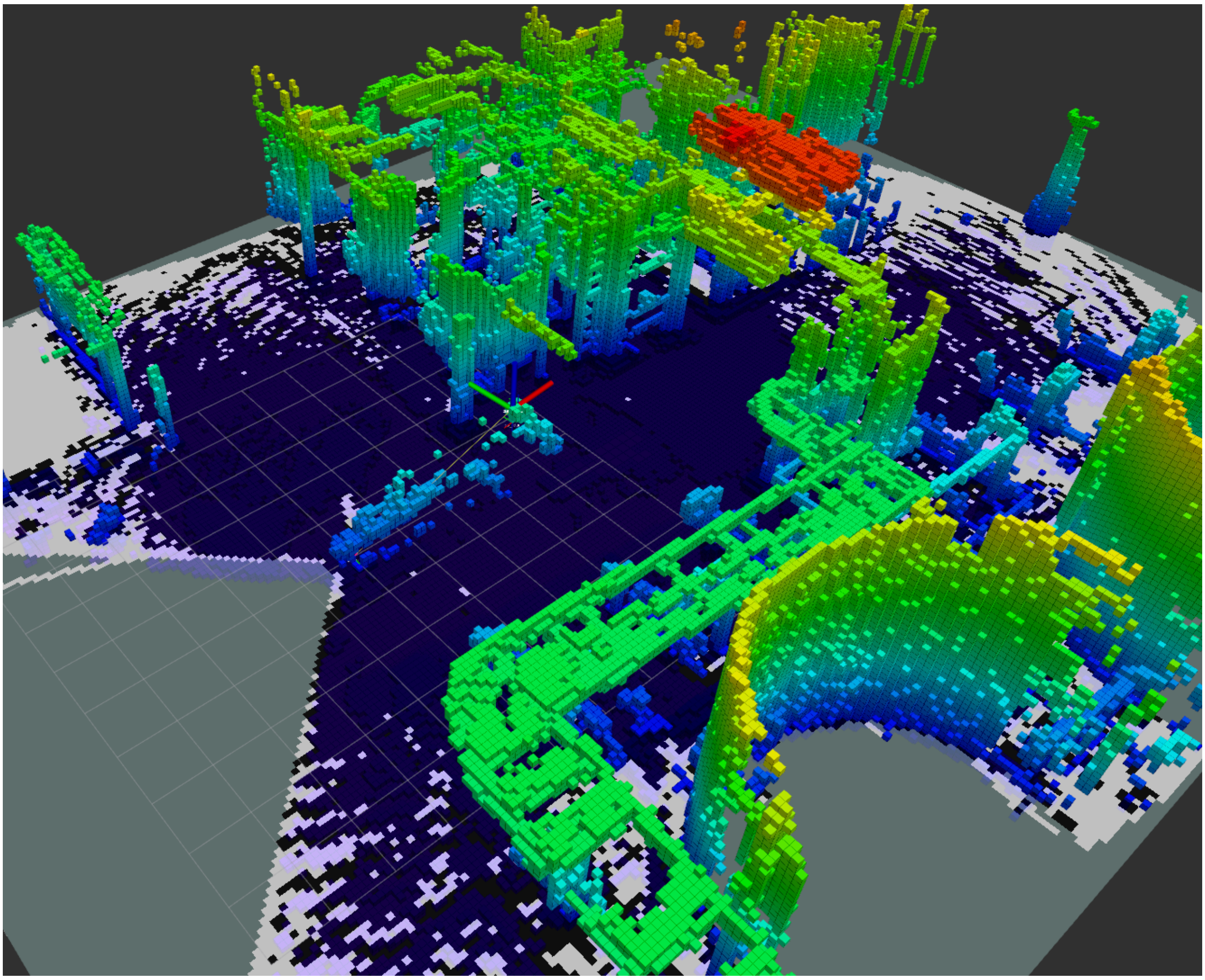


Figure 11. Environmental representation. Squared grids represent a occupied voxel, the colour makes reference to the height, the central reference system represents the location of the robot in real time.

5.1.3. Assisted navigation

Assisted navigation refers to the assisted flight mode of the state machine presented in D5.2. This flight mode is intended to be part of the navigation aid system because it facilitates the operation of the robot during flight. First, it allows the pilot to operate the robot through velocity commands. This means that the robot will maintain a static position in the air when not receiving any commands and will follow the velocity references when commanded by the sticks. Secondly, the robot is prevented from colliding with an obstacle in the environment. For this purpose, it will only accept commands from the pilot that respect the allowed movements taking into account the field of view (FOV) of the depth cameras (see section 5.1.1). In addition, if the system detects that the robot is approaching or moving against an obstacle, the system will notify the pilot and the HMR will stop when the closest point calculated by the obstacle detection reaches a predefined threshold.

5.2. Aerial robotic system for long-reach manipulation

This subsection introduces an Aerial Robotic System for Long-Reach Manipulation (ARS-LRM) which is used to illustrate the performance of the proposed planning method with Aerodynamics Awareness. The system description, the corresponding multi-body dynamical model and a distributed control approach are described hereinafter.

5.2.1. System description

As can be seen in Figure 12, the proposed ARS-LRM system consists of a multi-rotor and a robotic long-reach arm whose special integration with the aerial platform enables 360º rotation within the vertical plane that coincides with the centre of mass of the multi-rotor. This capability for multi-directional movement within the vertical plane widens significantly the inspection workspace of the robotic arm, which in turn extends the range of positions from which the aerial platform can perform the inspection task. The latter allows a better selection of the safest operation conditions for the aerial system. Additionally, the long-reach feature increments the safety distance between rotors and inspected areas. This last characteristic is even more relevant when the inspected areas include elements that can affect the rotor airflow.

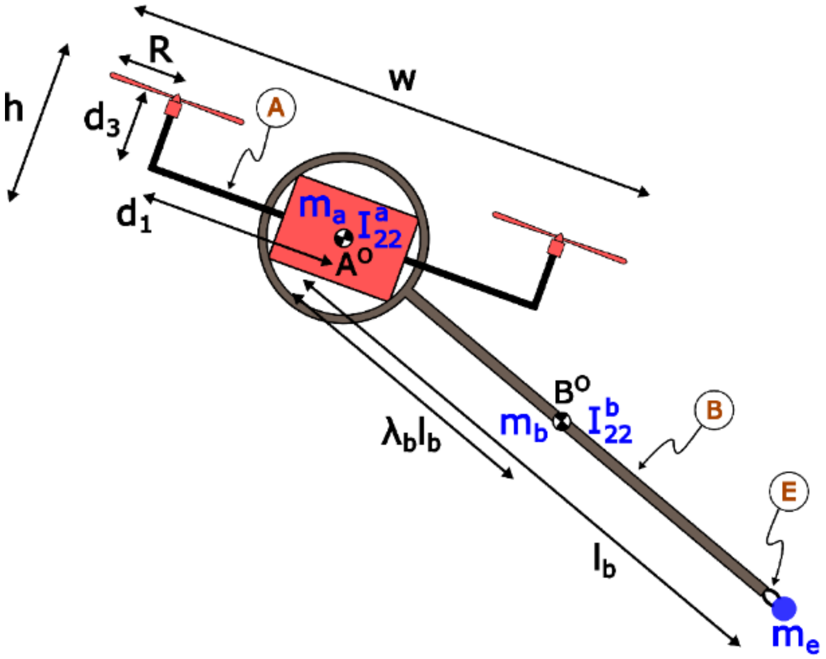


Figure 12. Aerial Robotic System for Long-Reach Manipulation (ARS-LRM). Geometry and mass distribution.

With the purpose of illustrating the planning method with Aerodynamics Awareness, a planar characterisation of the system has been adopted. This simplified approach eases the modelling and control derivations while maintaining the operation basis of the system. Following this assumption, the aerial platform A is characterised by a mass m_a, a principal moment of inertia I_{22}^{a} and dimensions $w \times h$. Concerning the location of the rotors, a quadcopter configuration with propellers of radius R can be selected. In this configuration, the rotors are symmetrically located with respect to the UAV centre of mass A^O at longitudinal and vertical distances d_1 and d_3 respectively. Additionally, the robotic arm B is assumed to be aligned with the UAV centre of mass. This long-reach manipulator has been considered as a rigid body of length l_b, mass m_b and principal moment of inertia I_{22}^{b}. Its centre of mass B^O is located at a distance $\lambda_b l_b$ ($\lambda_b \in [0,1]$) with respect to A^O. Finally, the end effector E has been treated in this simplified model as a punctual mass m_e located at the distal end of the robotic arm. The values of the parameters that have been previously defined are shown in Table 4.

Table 4. ARS-LRM parameters

	Parameter	Value	Units
Mass and Inertia	m_a	5	kg
-	I_{22}^{a}	0.093	$kg \cdot m^2$
-	m_b	0.3	kg
-	λ_b	0.5	$-$
-	I_{22}^{b}	0.025	$kg \cdot m^2$
-	m_e	0.05	kg
Geometry	w	1.2	m
-	h	0.4	m
-	d_1	0.41	m
-	d_3	0.2	m
-	R	0.19	m
-	l_b	1	m

5.2.2. Modelling

As stated in [KonEtal2007], the dynamics of a multi-rotor is mainly determined by its mechanical model when it is not operating in the proximity of elements that can modify significantly the rotor airflow. This section embraces the same assumption to derive the basic structure of modelling (the characterisation of the aerodynamics will be covered later) and consequently the system will be described by means of a detailed mechanical model of the complete multi-body system. Kane's method has been used with this purpose since it holds some unique advantages when addressing multi-body robotic systems like the ARS-LRM. Of the latter, the most remarkable are the derivation of a compact model in first order differential equations that are uncoupled in the generalised speed derivatives as well as the easy computerisation and the computational efficiency of the resulting equations of motion.

The configuration variables selected as system generalised coordinates are the longitudinal q_1 and vertical q_3 positions of the UAV centre of mass A^O in the inertial reference frame N, the multi-rotor pitch angle q_5 and the joint angle of the robotic arm q_7 (see Figure 13). Generalised speeds u_i $(i = 1,3,5,7)$ are defined as:

$$ {}^{N}\boldsymbol{v}^{A^O} = u_1\boldsymbol{n_1} + u_3\boldsymbol{n_3} \qquad {}^{N}\boldsymbol{\omega}^{A} = u_5\boldsymbol{n_2} \qquad {}^{A}\boldsymbol{\omega}^{B} = u_7\boldsymbol{n_2} \tag{1} $$

where ${}^{N}\boldsymbol{v}^{A^{O}}$ is the velocity of the UAV centre of mass A^{O} with respect to the inertial reference frame N, ${}^{N}\boldsymbol{\omega}^{A}$ is the angular velocity of the aerial platform A with respect to the inertial reference frame N and ${}^{A}\boldsymbol{\omega}^{B}$ is the angular velocity of the robotic arm B with respect to the aerial platform A. Previous equations lead to the following kinematic differential equations:

$$[\dot{q}_1\ \ \dot{q}_3\ \ \dot{q}_5\ \ \dot{q}_7]^T = [u_1\ \ u_3\ \ u_5\ \ u_7]^T \tag{2}$$

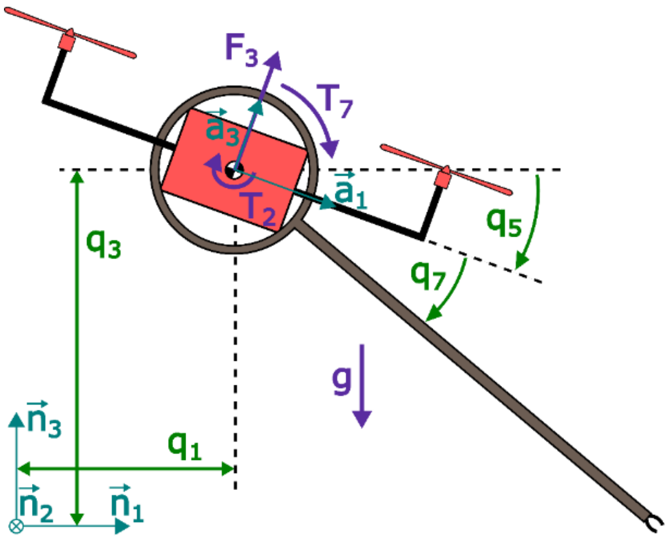


Figure 13. Configuration variables (green) and forces and torques applied to the ARS-LRM system (purple).

Regarding forces and torques exerted on the ARS-LRM system (see again Figure 13), the rotors generate a resultant lifting force $F_3\boldsymbol{a_3}$ applied to the multi-rotor centre of mass A^{O} as well as a torque $T_2\boldsymbol{a_2}$ applied to the aerial platform A. Additionally, the control action governing the manipulator is given by the torque applied to the arm joint $T_7\boldsymbol{a_2}$.

Application of Kane's method through MotionGenesis software [MotGen] leads to the following dynamic differential equations for translation and rotation:

$$\boldsymbol{A}\cdot[\dot{u}_1\ \ \dot{u}_3\ \ \dot{u}_5\ \ \dot{u}_7]^T = \boldsymbol{B} \tag{3}$$

where matrix $\boldsymbol{A}$ is:

$$\boldsymbol{A} = \begin{bmatrix} m_a + m_b + m_e & 0 & -l_b\xi_{m,1}s_{5,7} & -l_b\xi_{m,1}s_{5,7} \\ 0 & m_a + m_b + m_e & -l_b\xi_{m,1}c_{5,7} & -l_b\xi_{m,1}c_{5,7} \\ -l_b\xi_{m,1}s_{5,7} & -l_b\xi_{m,1}c_{5,7} & I_{22}^{a} + I_{22}^{b} + l_b^2\xi_{m,2} & I_{22}^{b} + l_b^2\xi_{m,2} \\ -l_b\xi_{m,1}s_{5,7} & -l_b\xi_{m,1}c_{5,7} & I_{22}^{b} + l_b^2\xi_{m,2} & I_{22}^{b} + l_b^2\xi_{m,2} \end{bmatrix} \tag{4}$$

matrix $\boldsymbol{B}$ is:

$$\boldsymbol{B} = \begin{bmatrix} F_3 s_5 + l_b \xi_{m,1} c_{5,7} (u_5 + u_7)^2 \\ F_3 c_5 - (m_a + m_b + m_e) g - l_b \xi_{m,1} s_{5,7} (u_5 + u_7)^2 \\ T_2 + l_b \xi_{m,1} c_{5,7} g \\ T_7 + l_b \xi_{m,1} c_{5,7} g \end{bmatrix} \tag{5}$$

and $\xi_{m,1} = m_e + \lambda_b m_b$, $\xi_{m,2} = m_e + \lambda_b^2 m_b$, $s_5 = \sin(q_5)$, $c_5 = \cos(q_5)$, $s_{5,7} = \sin(q_5 + q_7)$, $c_{5,7} = \cos(q_5 + q_7)$ and $g = 9.81\ m/s^2$ is the gravity acceleration.

5.2.3. Control

A distributed control scheme between the aerial platform and the robotic arm (see Figure 14) has been derived to provide the ARS-LRM system with the capability of tracking trajectories generated by the motion planning algorithms. The objective is the completion of the simulation environment that will allow the investigation of new planning strategies. A standard control structure that makes use of nonlinear control strategies based on model inversion has been selected for this purpose. This choice is considered representative of the average performance offered by the state-of-the-art controllers for this kind of systems.

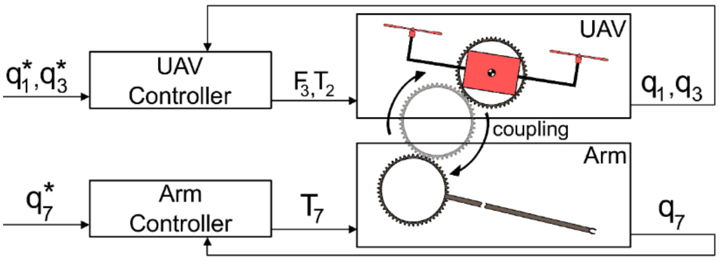


Figure 14. Distributed control scheme of the ARS-LRM system.

Regarding the multi-rotor, the control scheme is inspired by [KonEtal2007] and consists of linearising the system through model inversion and applying proportional-integral-derivative (PID) control laws to the resultant dynamics. In the model inversion mentioned above, the coupling action of the robotic arm is not considered and hence it will be treated as a perturbation that must be cancelled by the PID control laws. The underlying principle of control will be the adjustment of the multi-rotor lifting force vector in order to generate the translational accelerations required to reduce position errors. A general overview of the control scheme is shown in Figure 15 (left), where D_{13}^{-1}, K_5^{-1} and D_5^{-1} blocks represent, respectively, the inversions of the translational dynamics, rotational kinematics and rotational dynamics.

The control strategy selected for the robotic arm is again based on linearisation through model inversion and PID control, which yields a nonlinear control law capable of commanding the joint position of the manipulator within the allowed rotation rank of 360º. As in the previous case, the coupling action will not be considered in the model inversion and hence it will be treated as a perturbation that must be compensated by the control laws. The schematic representation of this approach is shown in Figure 15 (right) where D_7^{-1} represents the block in charge of inverting the dynamics of the arm and torque T_7 corresponds to the output signal of the controller.

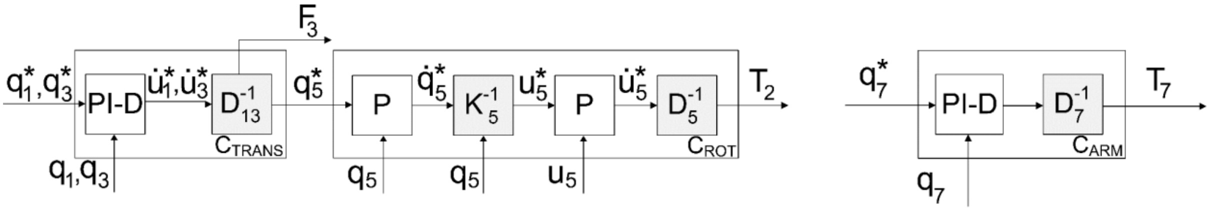


Figure 15. Block diagrams of the UAV controller (left) and the arm controller (right).

The parameters of the controllers have been tuned by means of the classic pole assignment method. The selected values constitute a trade-off that guarantees a proper dynamics range while the common mechanical limitations of this kind of systems are not overreached.

5.3. Characterisation of aerodynamic effects

The usage of aerial robotic systems for manipulation tasks usually requires that rotors operate in the proximity of surfaces that can affect their airflow. Working under such conditions leads to aerodynamic phenomena that can provoke destabilisations or even crashes. In order to avoid these dangerous flight conditions, the compensation of aerodynamic effects should be considered in the design of the controller [SanEtal2017] or taken into account in the trajectory generation process of the motion planner. This work adopts the second approach but firstly, a characterisation of these effects is required. With that purpose, a test-bench that allows the analysis of the thrust behaviour while rotors are operating close to surfaces has been designed.

The test-bench consists of an isostatic L-shaped structure that integrates a load cell in its lower endpoint and the rotor whose aerodynamics will be characterised in the distal end of the vertical part (see Figure 16). Additionally, a cylindrical joint γ allowing free rotation of the structure has been used to avoid any resultant force in the holder between the ground basis and the corner intersection of the L-shaped structure. An analysis of the force balance in this structure allows the calculation of the rotor thrust F_3 through the measurement of the reaction force F_{lc} applied to the load cell. The resultant equation of equilibrium is the following:

$$F_3 = \mathrm{F_{lc}} \frac{d_h}{d_v} \tag{6}$$

where d_h and d_v are the dimensions depicted in Figure 16. Regarding the electronics embedded in the structure, the test-bench integrates an Arduino Mega 2560 as data acquisition unit and includes sensors to measure the rotor speed as well as a load cell to collect the measurements of the reaction force F_{lc}. Moreover, the pulse-width modulation (PWM) signal commanded to the rotor is monitored at the output using an interruption-based algorithm that runs in the board. The Arduino Mega 2560 is connected through the serial port to a PC with Matlab that offers a graphical interface to process, display and save all the data collected by the sensors.

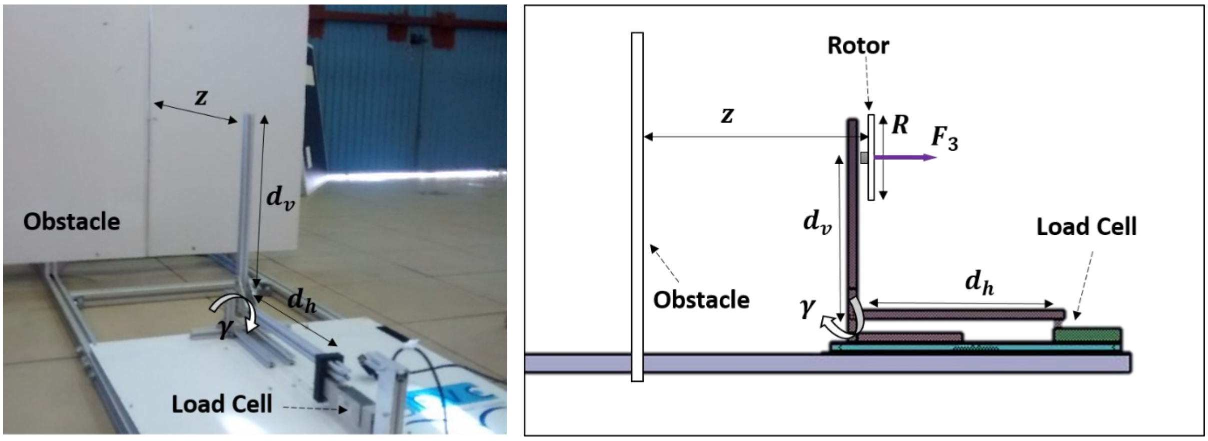


Figure 16. Test-bench used for the characterisation of the aerodynamic effects.

As mentioned above, the experimental procedure is based on the measurements of the rotor thrust provided by the test-bench. These measurements will be collected in scenarios where an element (see the obstacle in Figure 16) is located close to the rotor (from above, below or laterally, depending on the particular phenomenon under study; that is, ground effect, ceiling effect or wall effect). The PWM input signal that governs the rotor speed will be the same in all tests, which in turn implies that the desired rotor thrust will also be the same. However, the measured rotor thrust will change according to the aerodynamic conditions. At the beginning of each experiment, the rotor will receive the constant PWM input during a period of time that guarantees a steady-state for the system. From this point on, the measurements registered by the load cell will be filtered conveniently to get the mean value of the measured rotor thrust. The final test result will be the data pair given by the mean value of the thrust and the distance to the element used to disturb the rotor airflow. Several tests are carried out for each distance under study. This makes it possible to obtain statistical characterisations like the standard deviations represented in Figure 17 using an error-bar format.

The experimental results that have been obtained to analyse the ground, ceiling and wall effects, together with their associated configurations in the test-bench, are shown in Figure 17. In these figures, the aerodynamic modification of system behaviour is given by τ, a ratio between the lifting force F_3 in the presence of aerodynamic effects (IAE) and the same force out of the presence of these effects (OAE):

$$\tau = \frac{F_{3,IAE}}{F_{3,OAE}} \tag{7}$$

The dependence between τ and the distance to the surface affecting the rotor airflow is expressed in terms of a non-dimensional ratio whose numerator is the distance x, z from rotor centre to obstacles (see Figure 17 above) and denominator is the rotor radius; that is, z/R for the ground and ceiling effects or x/R for the wall effect.

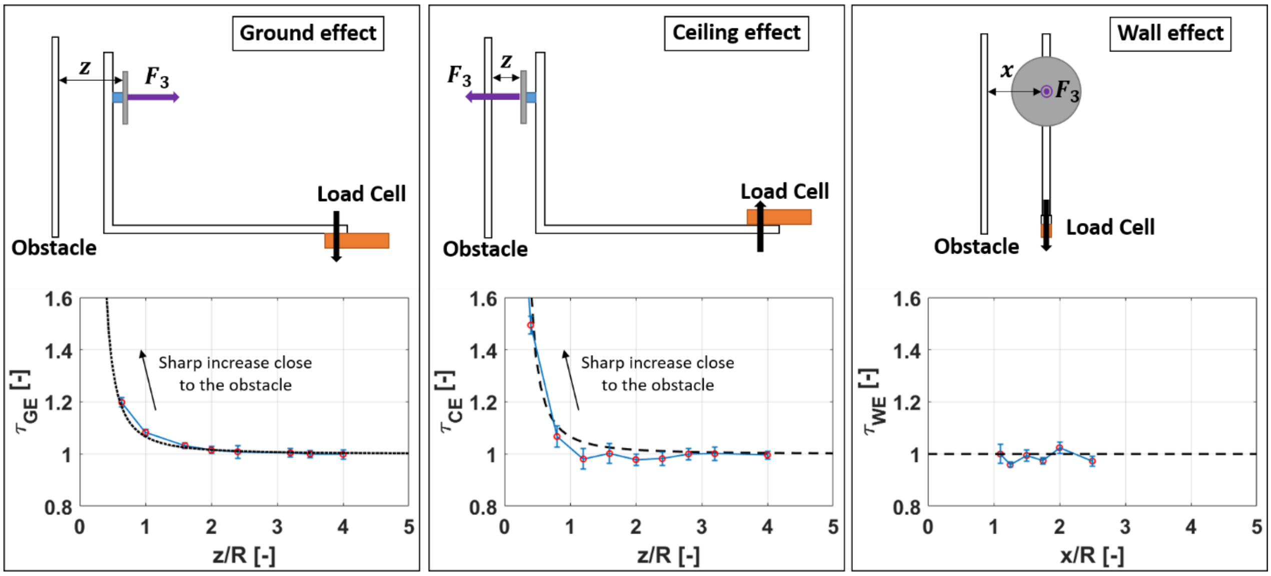


Figure 17. Characterisation of aerodynamic effects (below) and associated configuration of the test-bench (above): ground, ceiling and wall effects. Experimental results (red dots and blue error bars) and analytical models (dotted black curve for the theoretical model and dashed black curves for regression models).

The experimental results concerning the ground effect are consistent with the classical approach presented in [CheEtal1955] (dotted black curve in the left column of Figure 17). This coincidence validates the design of the test-bench as well as the experimental procedure. Furthermore, the classical approach previously mentioned is presented in Equation (8) and will be hereinafter adopted to model the ground effect. The derivation of this equation is based on the images method and the assumptions of the linearised potential aerodynamics.

$$\tau_{GE} = \frac{1}{1-\frac{1}{16}\left(\frac{R}{z}\right)^2} \quad (8)$$

Relative to the ceiling effect, there are no theoretical results in the literature. Alternatively, an experimental approach has been followed to derive an analytical expression:

$$\tau_{CE} = \frac{1}{1-\frac{1}{k_1}\left(\frac{R}{z+k_2}\right)^2} \quad (9)$$

where the coefficients $k_1 = 6.924$ and $k_2 = 3.782$ have been obtained by the least-square method when minimising the error with the experimental results. The resultant model has been depicted through the dashed black curve in the central column of Figure 17 for the ceiling effect. The experimental results show that this effect pulls the rotor towards the elements located above when the rotor is working closely. This effect may quickly lead to dangerous flight conditions because it tends to reduce abruptly the safety distance between the rotor and the element above.

Regarding the wall effect, the experimental results shown in the right column of Figure 17 have demonstrated that it does not have a significant influence on the system. The underlying explanation is that the natural rotor airflow streams from its upper part to its lower part and, therefore, an element located laterally does not produce such a big interference. Consequently, this effect will not be considered in lateral approximations to external elements.

The complete range of effects produced by the aerodynamic phenomena that have been studied in this subsection can be reflected in 3D maps. To this end, the numerical values corresponding to the different operation points will be computed using the models in Equations (8) and (9). One example of this graphical representation can be seen in Figure 18 for a rectangular-shaped obstacle. These maps will be used in the next subsection to implement the modifications of the motion planner that enable the Aerodynamics Awareness.

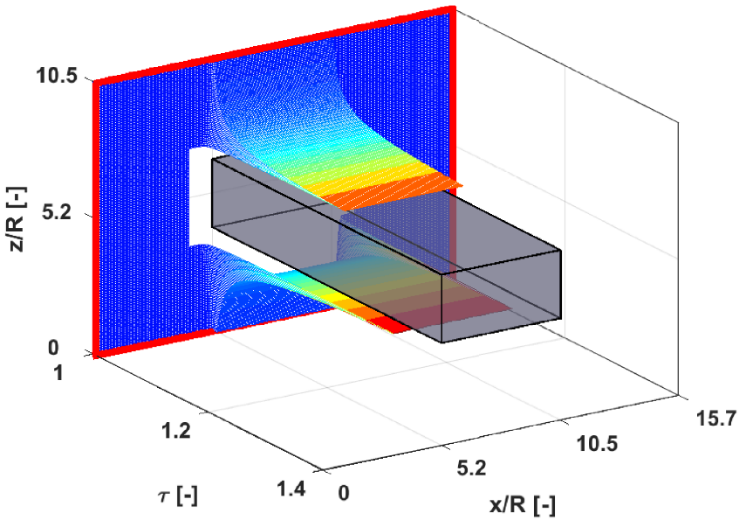


Figure 18. 3D map corresponding to the aerodynamic effects associated with a rectangular-shaped obstacle. The colour scale represents the magnitude of the ratio $\boldsymbol{\tau}$ of aerodynamic modification (from lower values in blue to higher values in red) while the white areas are associated with non-flyable regions due to the rotor geometry.

5.4. Motion planner with Dynamics and Aerodynamics Awareness

This subsection presents the motion planner (MP) with Aerodynamics Awareness (ADA) derived for the ARS-LRM system. The resultant algorithm will be referred to as MP-ARM-ADA. The acronym ARM stands for the more general classification of Aerial Robotic Manipulators and clarifies that the applicability of the algorithm extends beyond the specific ARS-LRM configuration proposed in this document. In general terms, the basis of the planning approach (MP-ARM) is built over an RRT* (optimal Rapidly-exploring Random Tree [KarEtal2011]) algorithm that has been optimised to exploit the advantages of the specific system under study. Over this optimisation of the basis method, two extensions have been developed to increase the safety of the system when operating in cluttered environments. Firstly, Dynamics Awareness (DA) can be applied to cope with the strong coupling between the aerial platform and the robotic arm. Secondly, Aerodynamics Awareness (ADA) for robust operation close to surfaces involving aerodynamic effects can then be applied. The following subsections develop properly both concepts.

5.4.1. Fundamentals of planner operation

The operation of the MP-ARM planner that constitutes the foundation for subsequent extensions (DA and ADA) is based on an integrated exploration of the search space for both subsystems of the ARS-LRM. To this end, the configuration variables of the aerial platform -with the exception of pitch angle q_5 that will be directly governed by the controller within a bounded range to guarantee smooth movements- and the robotic arm are considered jointly within the planning algorithm (see the configuration variables in Figure 13). This integrated strategy offers higher resolution in the planner operation since equivalent configurations in terms of final effector positions can be differentiated according to the positions of both the multi-rotor and the robotic arm. The latter makes it possible to achieve wider and safer operating conditions and constitutes a remarkable difference to previous contributions in the field of aerial manipulators where the planning problem was addressed in a decoupled way or only considering the position of the end effector.

Regarding the operation basis for the extension of the search tree, an RRT*-based algorithm (see Algorithm 1) that optimises energy and time performance has been developed. Although the proposed algorithm poses certain structural similarities to the well-known RRT* approach, most of the intermediate functionalities have been customised to deal with the aerial manipulator under study. For instance, Figure 19 illustrates how the two weighting parameters $p_{1,2}$ included in cost function $CF = f(p_{1,2})$ allow the prioritisation of certain movements in the ARS-LRM system. The complete set of these specific derivations have been properly described in a previous contribution [CabEtal2017].

Algorithm 1. Motion Planner for Aerial Robotic Manipulation (MP-ARM)

```
$\boldsymbol{Input}$: $map, param$
$\boldsymbol{Output}$: $trajectory$
$Tree \leftarrow INITIALIZATION(map, param)$
$\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ iter_{max}\ \boldsymbol{do}$
    $x_{rand} \leftarrow SAMPLE(\,)$
    $x_{nearest} \leftarrow NEAREST(Tree, x_{rand})$
    $x_{new} \leftarrow STEER(x_{nearest}, x_{rand})$
    $\boldsymbol{if}\ {\sim}COLLISION(x_{nearest}, x_{new}, map)\ \boldsymbol{then}$
        $x_{near} \leftarrow NEAR(Tree, x_{new})$
        $Tree \leftarrow ADD(x_{nearest}, x_{near}, x_{new})$
        $Tree \leftarrow REWIRE(x_{near}, x_{new})$
    $\boldsymbol{end\ if}$
$\boldsymbol{end\ for}$
$trajectory \leftarrow TRAJECTORY(Tree)$
```

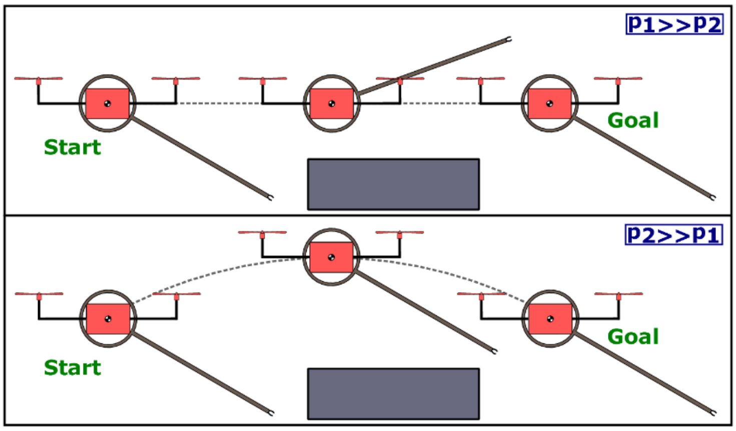


Figure 19. Effect of the weighting parameters $\boldsymbol{p_{1,2}}$ on the cost function $\boldsymbol{CF} = \boldsymbol{f}(\boldsymbol{p_{1,2}})$. $\boldsymbol{p_1} \gg \boldsymbol{p_2}$ prioritises trajectories in which the energy consumption of the UAV is minimum while $\boldsymbol{p_2} \gg \boldsymbol{p_1}$ prioritises trajectories in which the energy consumption of the manipulator is minimum.

5.4.2. Dynamics Awareness

The basic approach presented in the previous subsection guarantees planned trajectories that are collision free as well as efficient in terms of execution time and energy consumption. However, the strong dynamical coupling that exists in the ARS-LRM system between the multi-rotor and the robotic arm requires further attention since it provokes considerable differences between planned and executed trajectories. This is a consequence of the controller limitations to compensate the perturbations associated with the aforementioned couplings. The collision risk introduced by these differences is especially critical for cluttered environments like industrial sites. Figure 20 illustrates an example of the undesired dynamic effects that can arise in such scenarios.

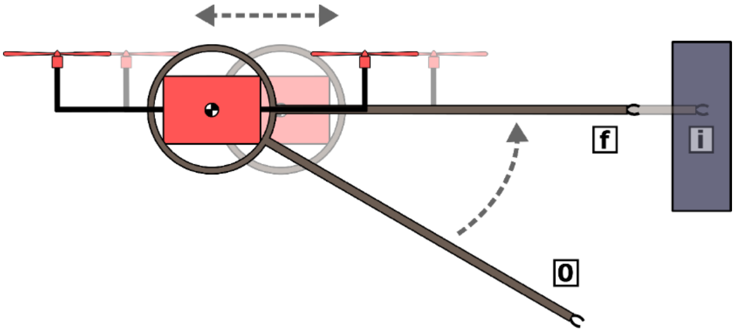


Figure 20 ARS-LRM system moving the robotic arm from initial position **0** to final position $\boldsymbol{f}$. The UAV oscillation produced in the intermediate states of this manoeuvre (shaded intermediate position $\boldsymbol{i}$) provokes a collision with the obstacle.

In order to overcome this undesired influence of the dynamical coupling, the inclusion of Dynamics Awareness (DA) in the MP-ARM planning algorithm is required. To this end, the expansion of the search tree is based on the behaviour of the controlled system, which means that collision detection is calculated through closed-loop simulations of the controlled system (green solid line in Figure 21) instead of simply using geometrical interpolation between states (blue dashed line in Figure 21). This guarantees that the resultant planned trajectories are both free of obstacles and compatible with the dynamic constraints of the ARS-LRM system. The approach considered for the dynamical analysis of the tree extension is based on a root-to-candidate validation. Thus, not only the dynamical feasibility of the new possible branch reaching the candidate node is analysed, but also the complete path from the tree root. For more details about this extension of the basis algorithm, please refer to previous work [CabEtal2018].

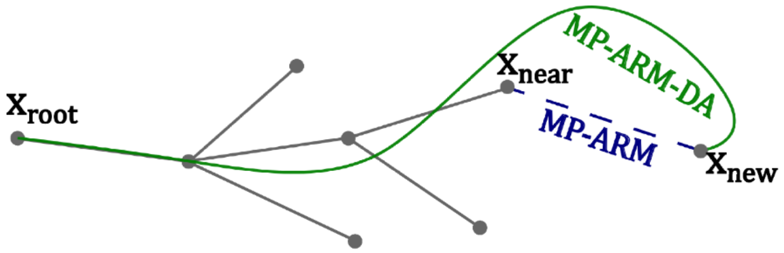


Figure 21. Operation basis of the MP-ARM-DA algorithm.

5.4.3. Aerodynamics Awareness

When the ARS-LRM system moves autonomously in cluttered environments, the Dynamics Awareness becomes an essential feature for safe operation. However, this mechanism could not be enough for robust obstacle avoidance in some scenarios. As stated in Section 5.3, aerodynamic phenomena such as the ground effect or the ceiling effect associated with the operation of a multi-rotor in the proximity of surfaces can modify its aerodynamic forces and, as a consequence, might provoke destabilisations or even crashes. For this reason, in order to avoid these unsafe conditions, this section proposes the inclusion of Aerodynamics Awareness (ADA) within the motion planning process.

Similar to the Dynamics Awareness approach, the principle of operation of the Aerodynamics Awareness mechanism is based on ensuring collision-free trajectories through the closed-loop simulation of the controlled ARS-LRM system. However, in this new approach, not only have the dynamics of the system been considered for the expansion of the search tree, but also its aerodynamics. Figure 22 (left) schematises the operation basis of the enhanced algorithm (MP-ARM-ADA) in contrast with the MP-ARM and MP-ARM-DA versions. In this figure, the difference between the trajectories computed by MP-ARM-DA and MP-ARM-ADA is the additional safety distance with respect to the grey surface that Aerodynamics Awareness suggests. Otherwise, the controlled system would not be able to compensate for the ceiling effect and might provoke a collision. Additionally, Figure 22 (right) depicts the complete closed-loop scheme required to perform the simulations that give support to the Aerodynamics Awareness concept. As can be seen, the MP-ARM-ADA algorithm makes use of an aerodynamic model whose input is given by the control signals commanded by the controller ($F_{3,OAE}$,$T_{2,OAE}$) in terms of the corresponding PWM signals and whose output is the force and torque that actually govern the movement of the system as a consequence of aerodynamic effects ($F_{3,IAE}$,$T_{2,IAE}$).

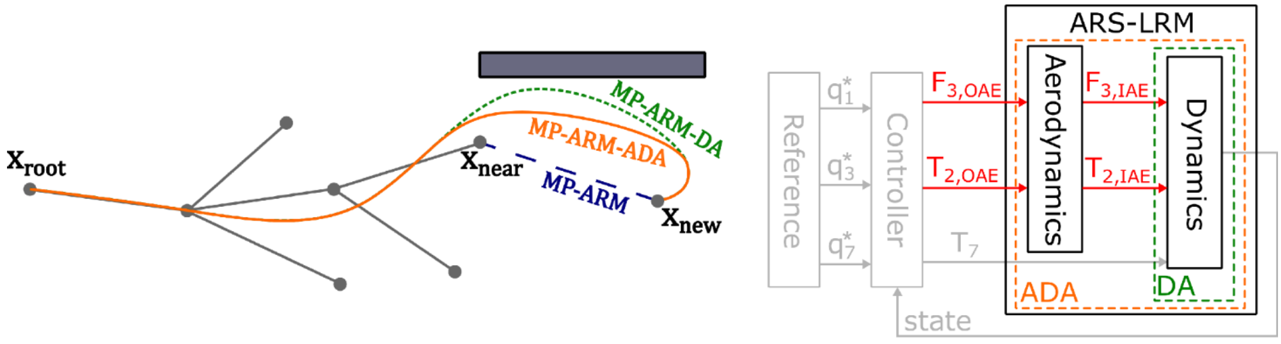


Figure 22. Operation basis of the MP-ARM-ADA algorithm. The terms highlighted in red correspond to the variables employed to model the Aerodynamics Awareness.

In order to derive the block *Aerodynamics* introduced in Figure 22 (right), the force and torque commanded by the controller ($F_{3,OAE}$,$T_{2,OAE}$) must be broken down into the contributions of the different rotors. This will pave the way for the application of the equations derived in Section 5.3 to characterise the aerodynamic effects of an individual rotor. The aforementioned decomposition can be implemented as follows:

$$\begin{aligned} F_{3,i} &= F_{3,i}^{L} + F_{3,i}^{R} \\ T_{2,i} &= d_1\left(F_{3,i}^{L} - F_{3,i}^{R}\right) \qquad (i = IAE, OAE) \end{aligned} \tag{10}$$

where superscripts L, R make reference to left and right rotors in Figure 13. Additionally, the longitudinal q_1^L, q_1^R and vertical q_3^L, q_3^R position of each rotor is also required to calculate the value of the aerodynamic phenomena at a certain operation point:

$$\begin{aligned} q_1^L &= q_1 - d_1\cos(q_5) + d_3\sin(q_5) \qquad & q_1^R &= q_1 + d_1\cos(q_5) + d_3\sin(q_5) \\ q_3^L &= q_3 + d_1\sin(q_5) + d_3\cos(q_5) \qquad & q_3^R &= q_3 - d_1\sin(q_5) + d_3\cos(q_5) \end{aligned} \tag{11}$$

Using these positions as entry point in the 3D maps computed in Section 5.3, the aerodynamic ratio τ between the actual thrust that governs the platform movement (IAE) and the thrust commanded by the controller (OAE) can be obtained. Consequently, the lifting force of each rotor in presence of aerodynamic effects is given by:

$$F_{3,IAE}^{j} = \tau\left(q_1^j, q_3^j\right)F_{3,OAE}^{j} \qquad (j = L, R) \tag{12}$$

The combined usage of Equations 10 and 12 allows the final calculation of the total force and torque $F_{3,IAE}$,$T_{2,IAE}$ that are really exerted upon the platform.

5.5. Application scenario: inspection of structural elements in pipe arrays

This subsection proposes the inspection of structural elements in pipe arrays as use case to evaluate the motion planner presented above. To this end, the simulated environment given by Figure 23 (left) will be assumed here. In this scenario, the ARS-LRM system has to perform a visual inspection of the junction point between the pipe array and one of the pillars that bear its weight at height (marked with a red point in the figure). For that, a visual camera will be integrated as end effector in the long-reach arm of the ARS-LRM system.

The inspection plan will be the following. The aerial system will start the operation from the top part of the pipe array and will navigate around this until reaching the position where the inspection point is reachable for the end effector. This operation is schematised in Figure 23 (right), where the solid red line denotes the operational limits and the dashed black line the safety margins with respect to the obstacles whose violation would be considered as a collision.

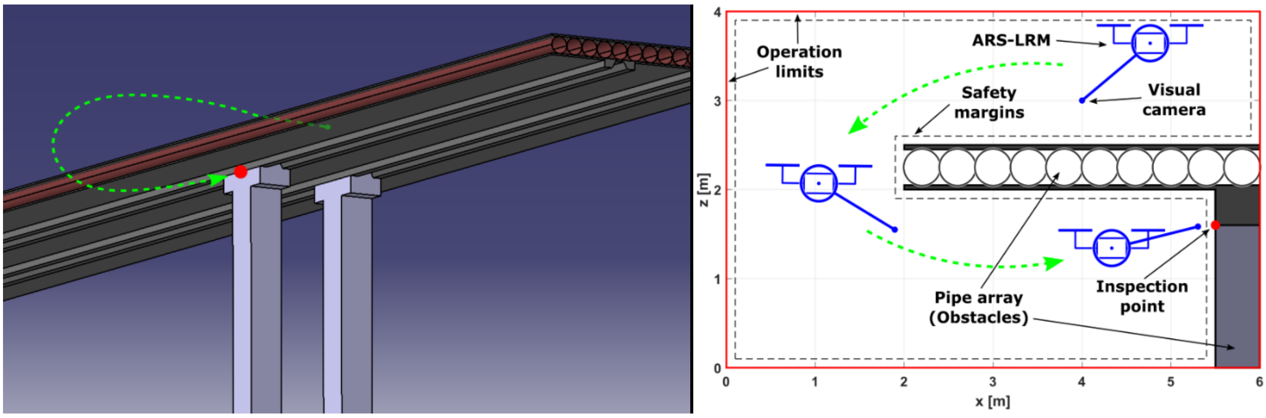


Figure 23. Application scenario given by the inspection of structural elements in pipe arrays.

The aerodynamic challenges of this scenario arise because the most efficient trajectories for the inspection task under consideration demand that the ARS-LRM system flies close to the surfaces of the pipe array. The latter makes this scenario suitable to establish a first validation of the MP-ARM-ADA algorithm. The expectation is that the resulting trajectories are as efficient as possible in terms of energy consumption, but without endangering the integrity of the aerial platform because of the aerodynamic phenomena. In order to characterise these effects, the approach presented in Section 5.3 has been followed to compute the 3D map of aerodynamic effects corresponding to the application scenario proposed in this subsection (see Figure 24).

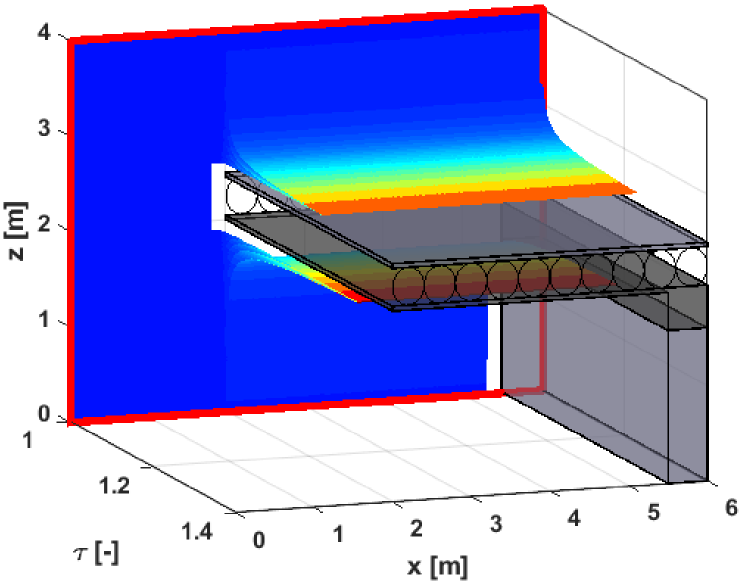


Figure 24. 3D map of aerodynamic effects for the application scenario.

5.6. Validation results

This subsection analyses the simulations corresponding to the application of the presented motion planners to the inspection scenario previously described. This validation analysis has been structured around two main phases. As a first step, the performance of the MP-ARM and MP-ARM-DA algorithms has been evaluated. This evaluation has revealed their limitations in the inspection scenario under consideration because of their lack of Aerodynamics Awareness. As a second step, a complete analysis of the MP-ARM-ADA performance has also been addressed. The analysis of these results allows concluding that the Aerodynamics Awareness is required to generate both efficient and safe trajectories when the ARS-LRM system operates close to elements that can modify significantly the rotor airflow.

In general terms, the validation tests have been carried out following the same sequence. Firstly, the planning algorithm under analysis has been executed to generate the motion plan. Then, the resultant plan has been provided to the controlled ARS-LRM system. The objective is therefore to analyse not only the planned trajectories, but also the closed-loop simulations of the controlled system when following the commanded trajectories. In fact, the latter will be considered the most representative information to establish the performance of the different algorithms.

The simulation work has been performed in a Matlab-Simulink framework that provides the graphical evolution of the system variables. This graphical output will be used throughout this subsection to illustrate the obtained results. Additionally, intuitive snapshot diagrams have also been included. In these diagrams, the dashed and dotted coloured lines represent, respectively, the planned (light colours) and the simulated (dark colours) movements of the multi-rotor centre of mass as well as the positioning of the end effector. In contrast, the snapshots themselves only cover some intermediate configurations to illustrate intuitively the qualitative behaviour of the ARS-LRM system. These representative configurations are time-ordered by identifying labels that go from the initial position (label 0) to the final position (label f) passing through a set of intermediate positions (labels i_1, i_2, i_3, i_4).

5.6.1. Results using the MP-ARM algorithm

Figure 25 and Figure 26 show the closed-loop simulation results when the controlled ARS-LRM system (dark blue) tracks the trajectory planned with the MP-ARM motion planner (light blue). Whilst the planned trajectory can be considered efficient and collision-free, the attempt to execute such optimal trajectory provokes serious inconveniences due to the aerodynamic phenomena. More precisely, since the controlled ARS-LRM system is commanded to navigate close to the bottom surface of the pipe array (the shortest trajectory to the final inspection pose shown in Figure 25), this implies flying in presence of the associated ceiling effect. The latter produces disturbances that the controller is not able to reject. Consequently, the system becomes unstable until finally the rear rotors collide with the pipe array.

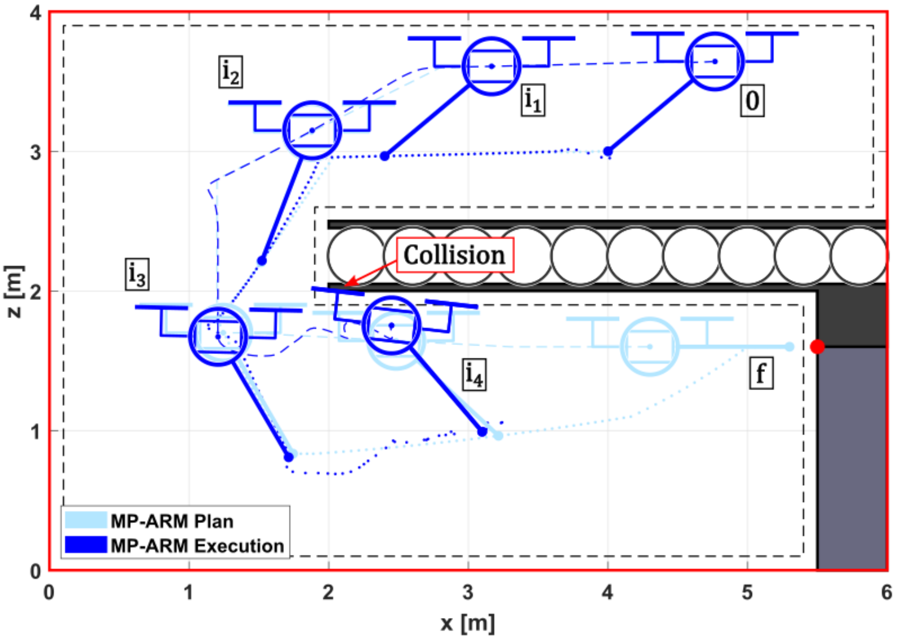


Figure 25. Snapshot diagram corresponding to the closed-loop execution (dark blue) of the trajectory planned with the MP-ARM algorithm (light blue).

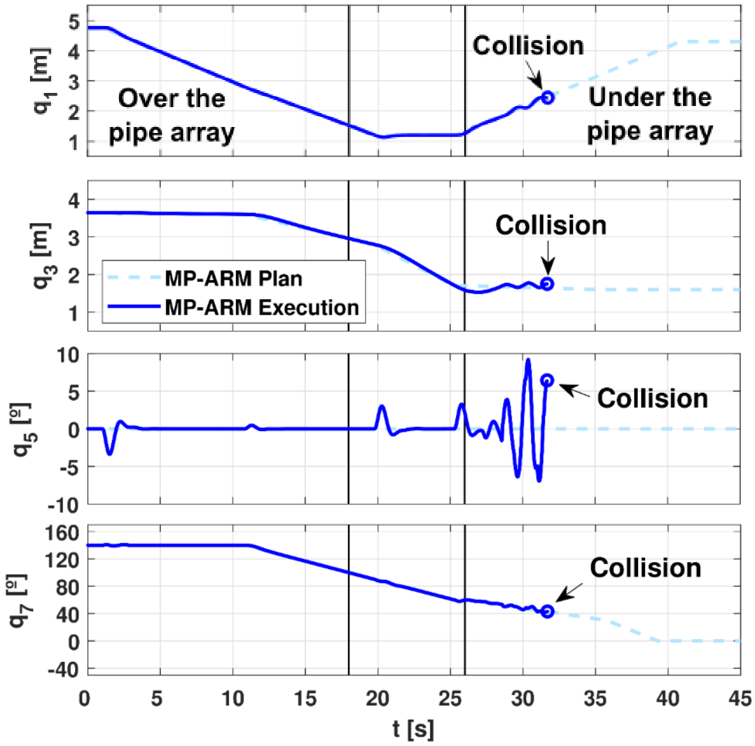


Figure 26. Evolution along time of the trajectory planned with the MP-ARM algorithm (dashed light blue) and executed by the controlled ARS-LRM system (solid dark blue).

5.6.2. Results using the MP-ARM-DA algorithm

As in the previous case, Figure 27 and Figure 28 depict the simulation results when the controlled ARS-LRM system (dark green) tracks a trajectory generated by the MP-ARM-DA motion planner (light green). Although in this case the planner takes into account the dynamics of the system within the planning process, the disturbances associated with the ceiling effect lead again the ARS-LRM system to collide with the pipe array in the closed-loop simulation. The only remarkable difference is that the collision instant is delayed. This is a consequence of the addition of Dynamics Awareness since the planner is now aware of the collision risk produced by the states close to the safety margins (overshooting associated with these states that can be observed in the dynamic simulations). As a result, the MP-ARM-DA planner discards those states. However, this improvement is not enough to operate safely in this application scenario since the collision eventually takes place.

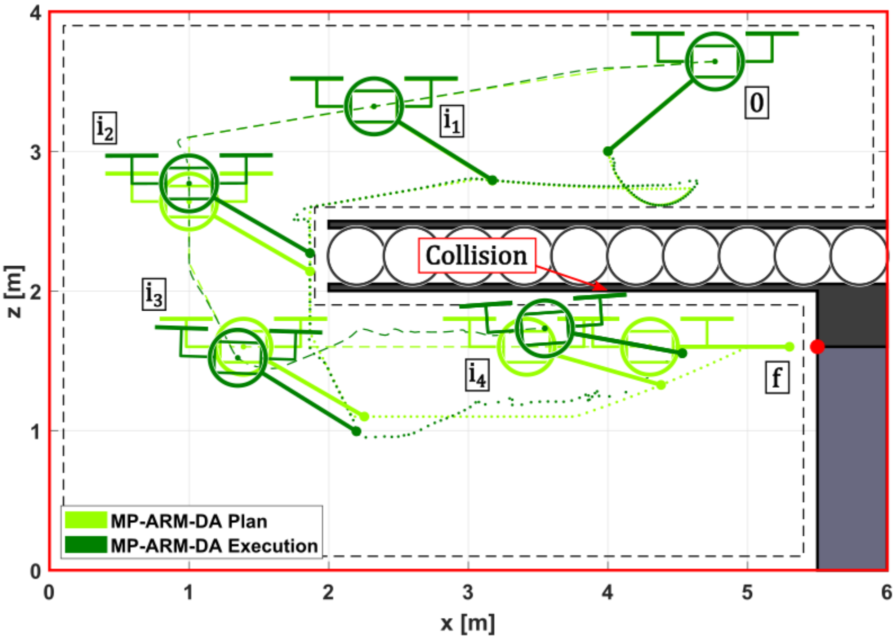


Figure 27. Snapshot diagram corresponding to the closed-loop execution (dark green) of the trajectory planned with the MP-ARM-DA algorithm (light green).

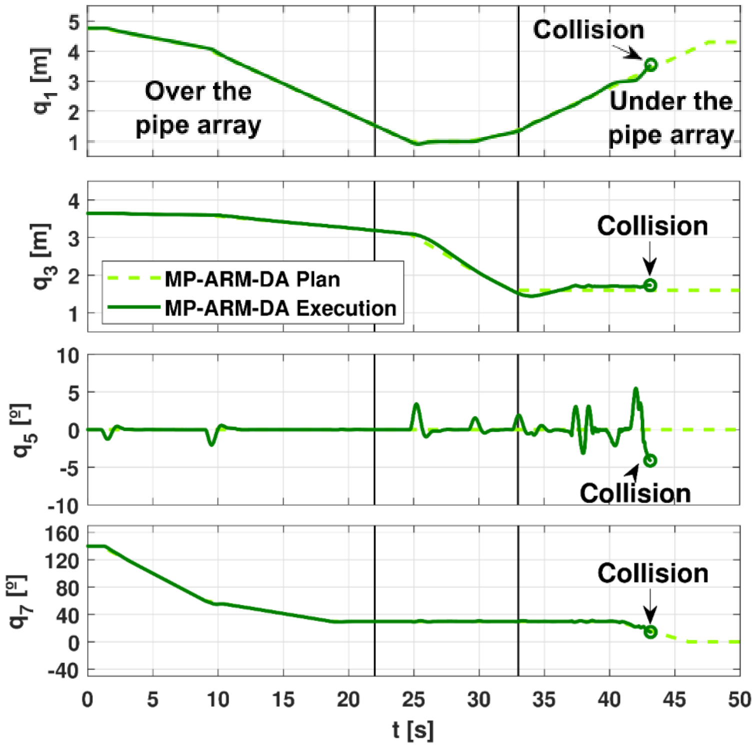


Figure 28. Evolution along time of the trajectory planned with the MP-ARM-DA algorithm (dashed light green) and executed by the controlled ARS-LRM system (solid dark green).

5.6.3. Results using the MP-ARM-ADA algorithm

In order to conclude with the simulation sequence, Figure 29 and Figure 30 show the results when a trajectory generated by the MP-ARM-ADA motion planner (light orange) is commanded to the controlled ARS-LRM system (dark orange). In contrast to MP-ARM and MP-ARM-DA, the enhanced MP-ARM-ADA algorithm avoids navigating through the hazardous area close to the pipe array. Thanks to the Aerodynamics Awareness, the planner discards these problematic states during the planning process and explores alternatives that lead to the most efficient trajectories but within the area of safe operation. To this end, the MP-ARM-ADA algorithm also takes advantage of the long-reach capabilities of the ARS-LRM system. They make it possible for the aerial platform to reach the inspection point from positions that maintain a wider safety margin with respect to the elements causing the aerodynamic phenomena.

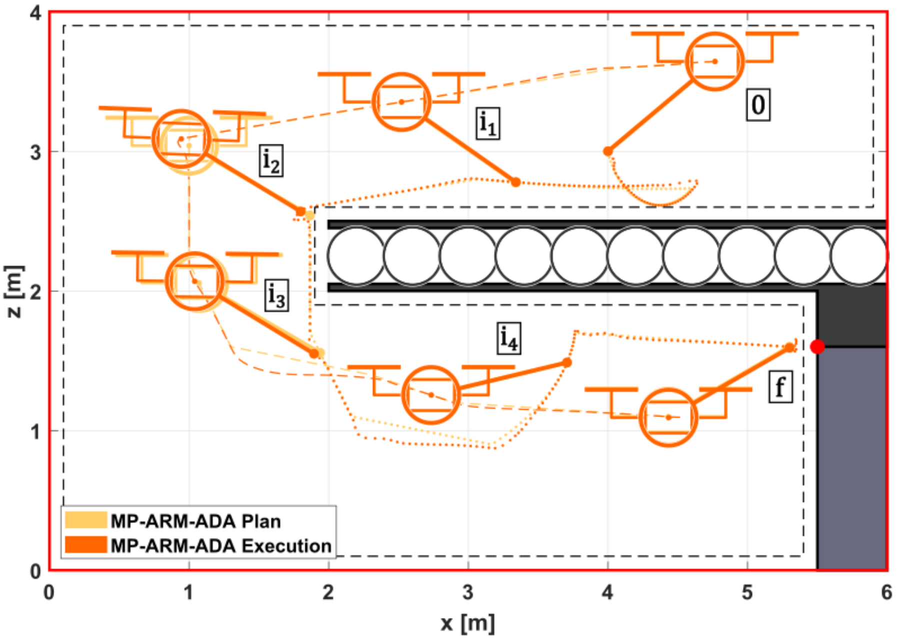


Figure 29. Snapshot diagram corresponding to the closed-loop execution (dark orange) of the trajectory planned with the MP-ARM-ADA algorithm (light orange).

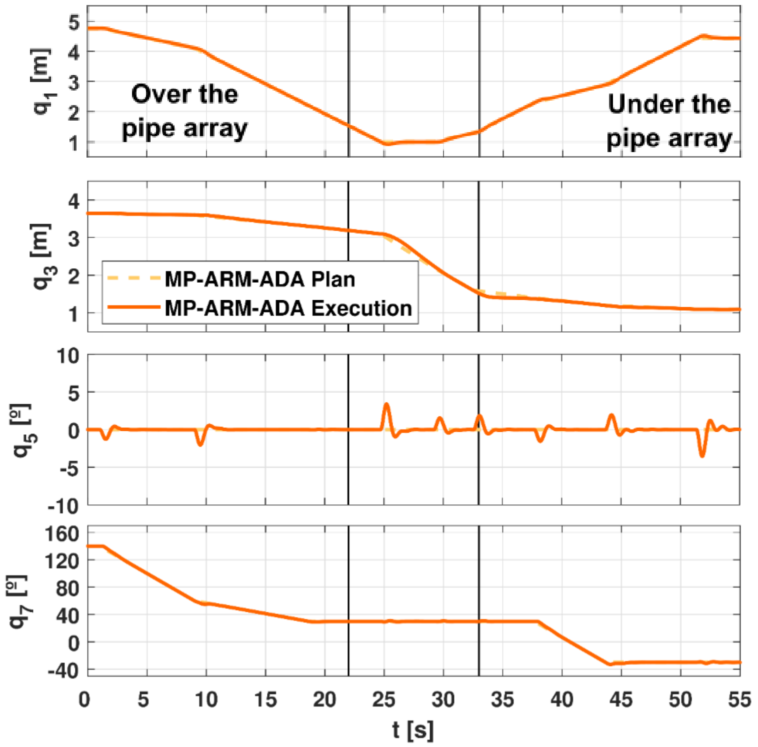


Figure 30. Evolution along time of the trajectory planned with the MP-ARM-ADA algorithm (dashed light orange) and executed by the controlled ARS-LRM system (solid dark orange).

In order to endorse the validity of previous conclusions about the MP-ARM-ADA algorithm, a wider set of simulations has been carried out. More particularly, this algorithm has been extensively executed in the inspection scenario to obtain a sample of 50 planned trajectories, all of these leading to collision-free executions. The same extensive approach has been followed with the MP-ARM and

MP-ARM-DA algorithms and only one instance of MP-ARM-DA led to a collision-free trajectory. This analysis reinforces the capability of the MP-ARM-ADA method to plan efficient trajectories that at the same time ensure safe conditions of operation.

Moreover, Figure 31 presents the analysis of two general metrics of the MP-ARM-ADA planned trajectories; these are, the length of both the UAV and the arm trajectories. The central red line is the median while the edges of the box are the 25^{th} and 75^{th} percentiles. The whiskers extend to the most extreme values obtained in the simulations. As can be seen for the length of the UAV trajectory, there exists a small variability around the median while the worst and best cases are relatively close to each other. At the same time, the angular displacement of the robotic arm is the same for all the simulations (the minimum distance between the initial and final poses) because the planner always computes the optimal trajectory for this subsystem. These results thus reveal a deterministic behaviour of the algorithm in the performance indices under consideration, which can also be considered a desirable feature in critical operation conditions like the proximity of the pipe array.

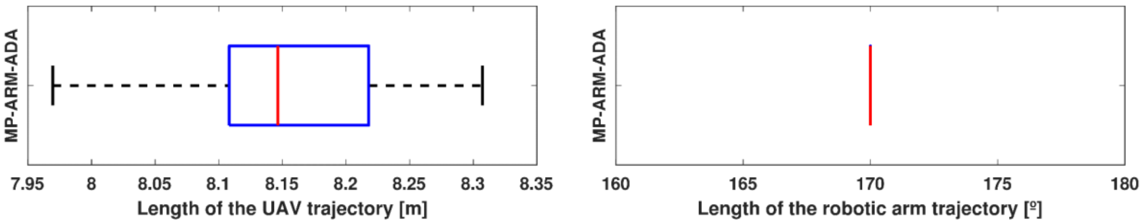


Figure 31. Statistical analysis for the set of 50 trajectories planned with the MP-ARM-ADA algorithm.

6. Mobile ground support platform (T4.3 part II)

This section brings information on the work done in task T4.3 “Mobile ground support platform” more related to the mobile ground support platform itself and its hardware and software components. The next subsection provides a summary of previously reported outcomes, followed by the description of the most recent ones.

6.1. Introduction

Earlier deliverables D4.1 and D4.2 described the developments in the design of the mobile support platform itself, including its flexible power supply subsystem. In particular, D4.1 and D4.2 presented

- The hardware and software related to the mobile support platform, for linking the activities of the inspection personnel and the hybrid robot.
- The power management for inspection, hybrid robot operation, including options of future safe wireless HR recharging.
- The pushable cart including the waterproof ground support subsystems.

6.2. Overview

The mobile support platform (MSP) consists of a pushable filmcart, Smartone Micro, which has a mounted high-capacity battery pack and enough space for a computer and other support hardware required to remote operate the HR. The regulators related to the computer, screen, and other utilities can be mounted in a single carry case, that can be placed on top of the cart, as shown previously in D4.2, section 5.

6.3. MSP battery system

Currently, the LiFePo4 8-cell 2.5 kWh battery system in the remote operating cart is operated and charged externally, using a separate 1 kWh battery charger. The battery case has a 1 kW power supply. The printed circuit board (PCB) schematics has been designed for a custom integrated 480W charging system for the case, described in D5.1, Section 4. It was not yet integrated into the system as the PCB layout would have required more revising to achieve intended specifications, but the MSP battery system is fully usable and controllable without the custom charging board as well, to power all the required equipment on the MSP. The Android app UI of the Overkill Solar battery management system (BMS), during charging, is shown in Figure 32 and Figure 33. Here it is connected to the tablet via UART-Bluetooth converter. Alternatively, the BMS can be monitored and controlled with a third-party Arduino library, when used with the NUCLEO-F446RE of the custom charging board. The same charger unit that is used to charge the MSP, could be also used to charge the UAV/HR batteries, in a safe location.

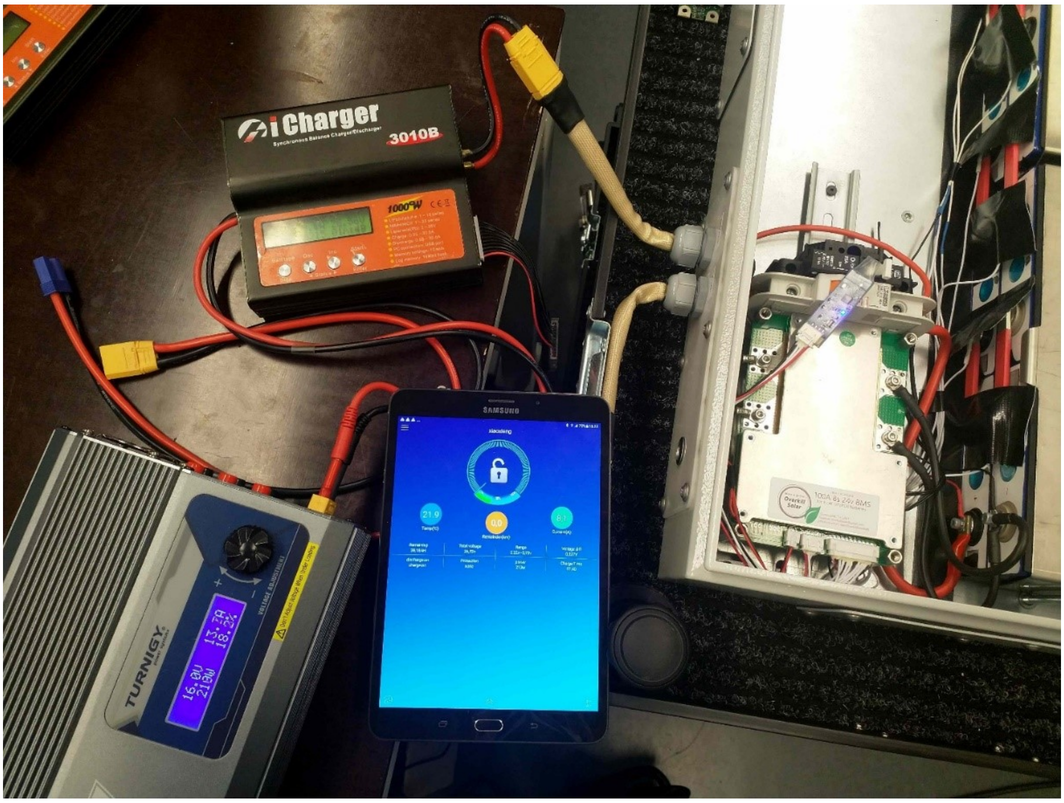


Figure 32. Battery management system (BMS) Android application graphical user interface (GUI) during charging testing.

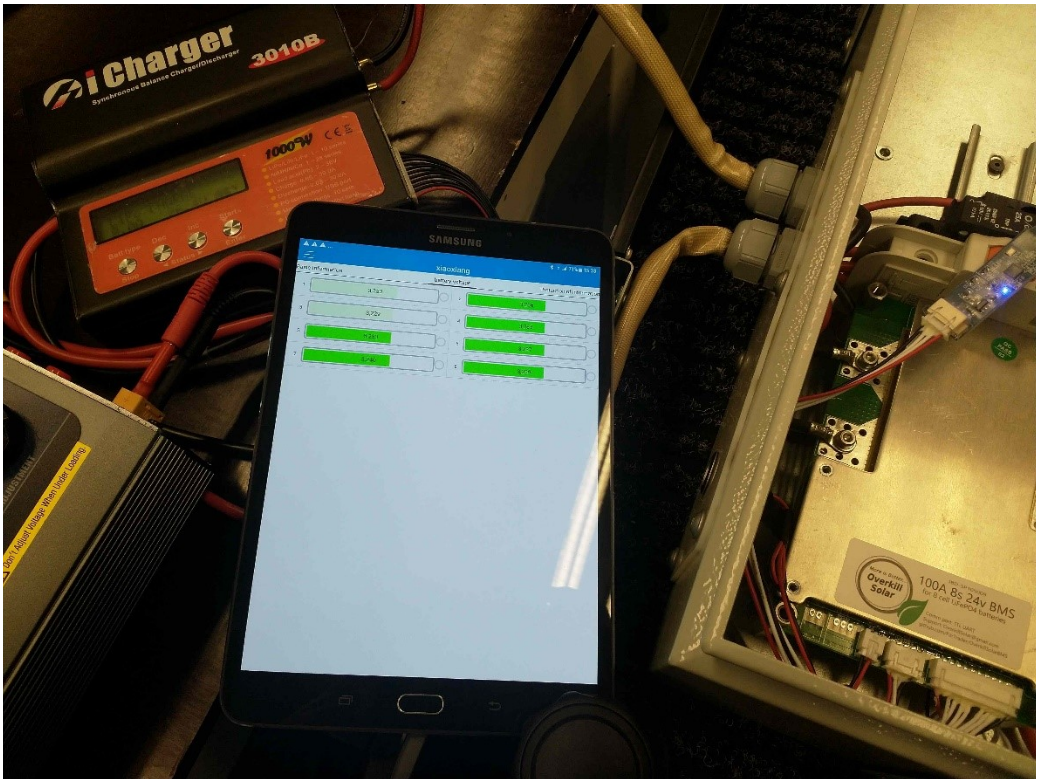

iCharger
Synchronous Balance Charger/Discharger
3010B
1000W
SAMSUNG
xiaoxiang
Overkill Solar
100A 8s 24v BMS
Don't Adjust Voltage When Under Loading

Figure 33. BMS GUI cell voltages monitoring.

6.4. MSP support computer

The MSP computer is a Zotac ZBOX EN072070, for which there is a carry case where the computer, touch screen, keyboard and mouse were housed as was presented previously in D4.2, Section 5. The carry case has passthroughs for input and output (IO) ports through its case. For connecting the HR and HR ground control station for real-time data transfer with the MSP, Wi-Fi router can be utilized based on the Asus ROG Rapture GT-AC5300, with 8 antennas and connectors for RJ-45 Ethernet. The Wi-Fi is enclosed in weather resistant box, on a pole, as shown in Figure 34. The pole also has a spot for RTK-GPS (Real Time Kinematic GPS) base station for high precision GPS positioning, but in practice RTK-GPS can not work near metal installations. As the remote operating cart is based on a film production cart, it supports attachment of 40 mm poles, so the Wi-Fi box can be mounted on it.

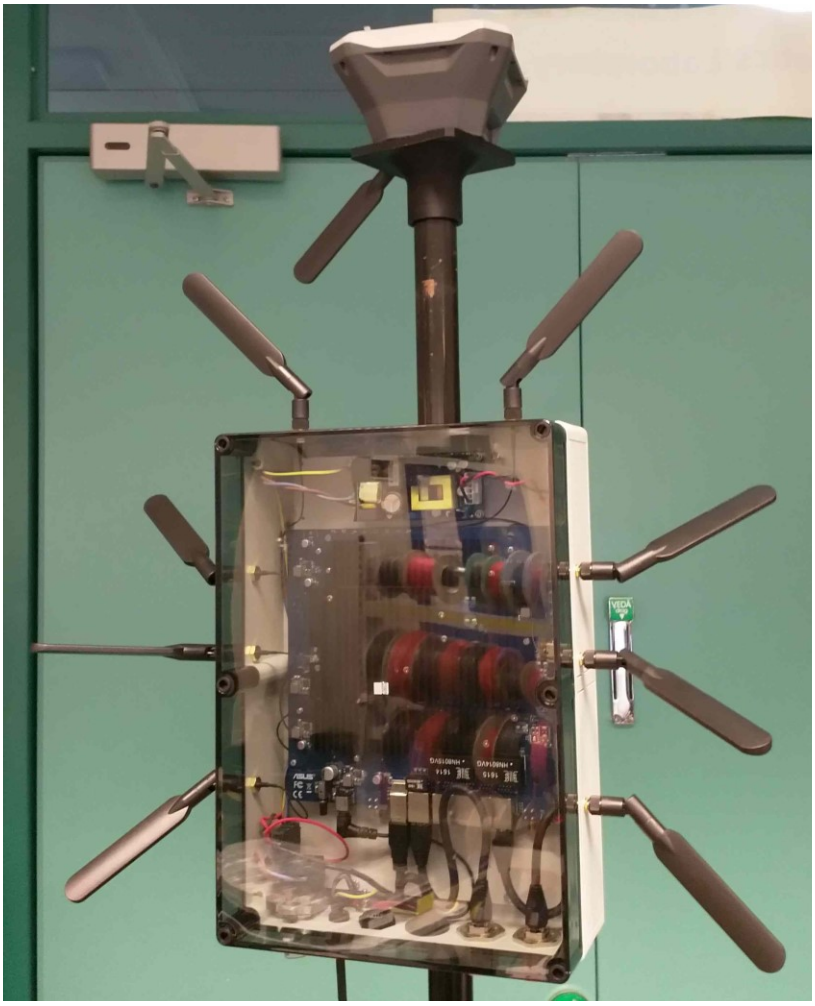
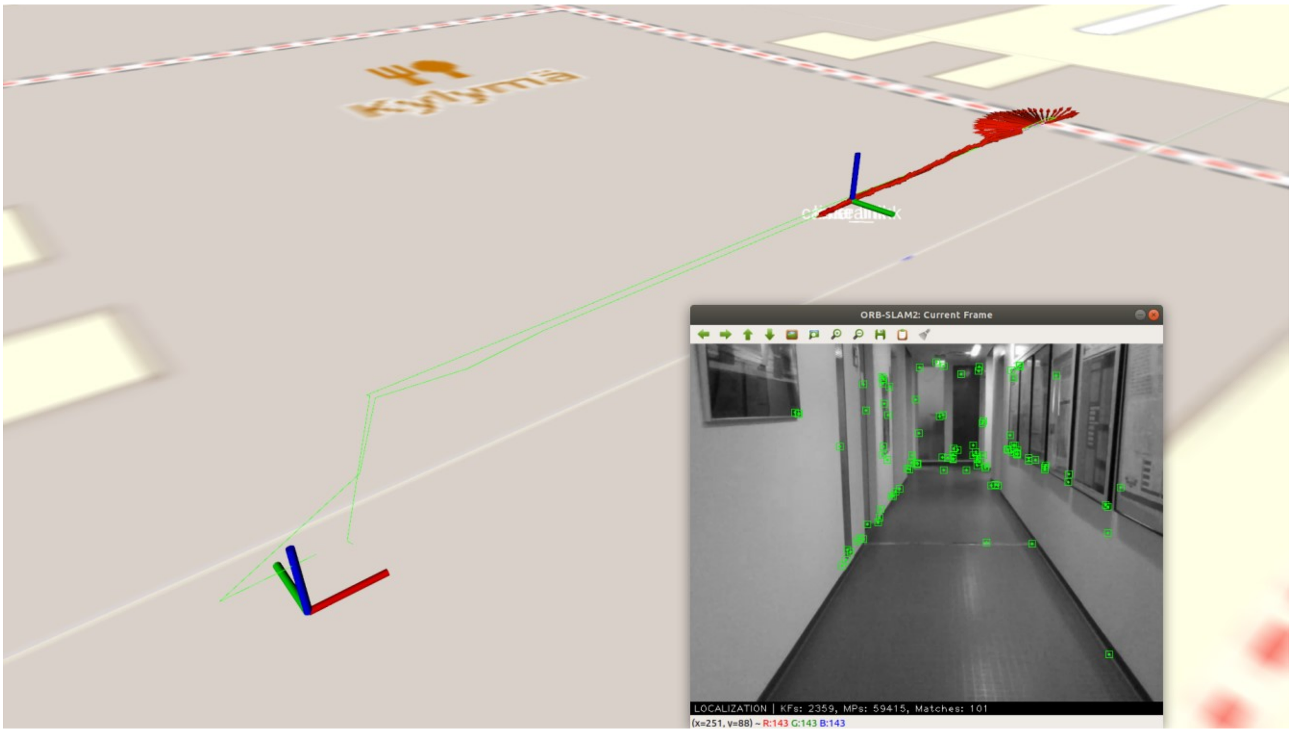


Figure 34. Weather resistant box for MSP Wi-Fi.

The remote visualization of the robot measurements and navigation paths can be done on the MSP computer, through ROS. In the visualization below, demonstrating the visualization when used with UOULU's navigation system, the robots planned path (green) and the executed path (red) can be seen, in Figure 35. The ROS node structure from D5.1, Section 4, is utilized with the added automatic map layout loading from the OpenStreetMap database (www.openstreetmap.org). The MSP computer can also perform localization using ORB-SLAM2, which the robot control algorithm can use to evaluate its position in ROS relative ENU frame (East-North-up), even without GPS. ORB-SLAM2 navigation maps of the work environments can be generated with a UAV/HR for the areas where the measurements are about to be performed. The fusion of UOULU ORB-SLAM2 navigation map and generated high detail 3D map of the related areas is left as a future development, as it is the migration of navigation and SLAM systems to ROS2 due to ROS being phased out.



```
LOCALIZATION | KFs: 2359, MPs: 59415, Matches: 101
(x=251, y=88) ~ R:143 G:143 B:143
```

Figure 35. RVIZ visualization of a given robot route plan and executed path in ROS, on the MSP computer.

6.5. Wireless charging sketches

As an optional feature, implementation of wireless charging was studied and schematics were sketched, but not yet implemented, being an optional feature. Wireless charging has been briefly discussed in D4.1, Section 4. Preliminary simulations were performed to estimate the dimensions and wire sizing of the helical toroidal coil that would be required for implementing a 500 W wireless charger. The simulation environment is shown in Figure 36. Based on the estimated coil parameters and requirements, also an initial sketch for the wireless transmitter side was made, shown in Figure 37. Implementation is left as future development.

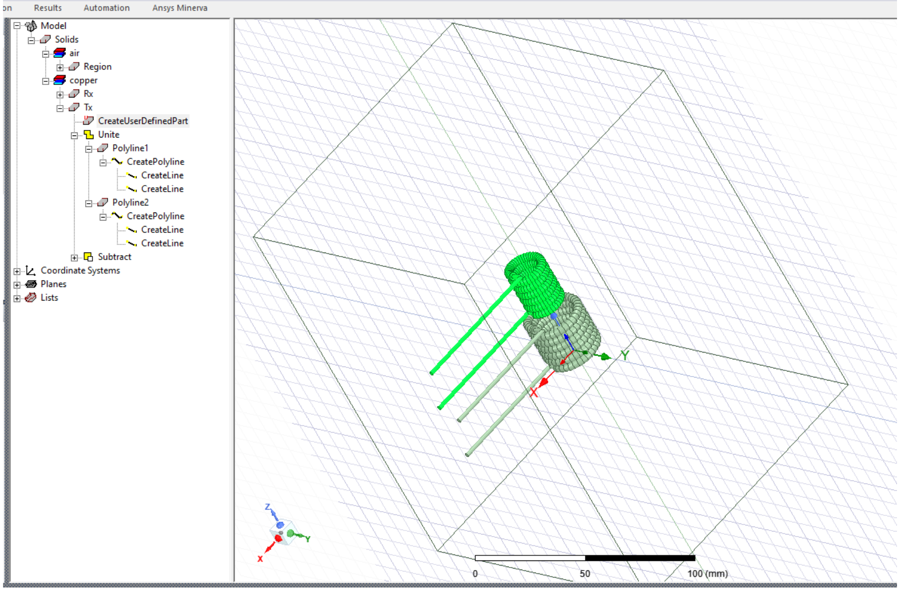


Figure 36. Ansys Maxwell simulator sketch of the toroidal coil setup.

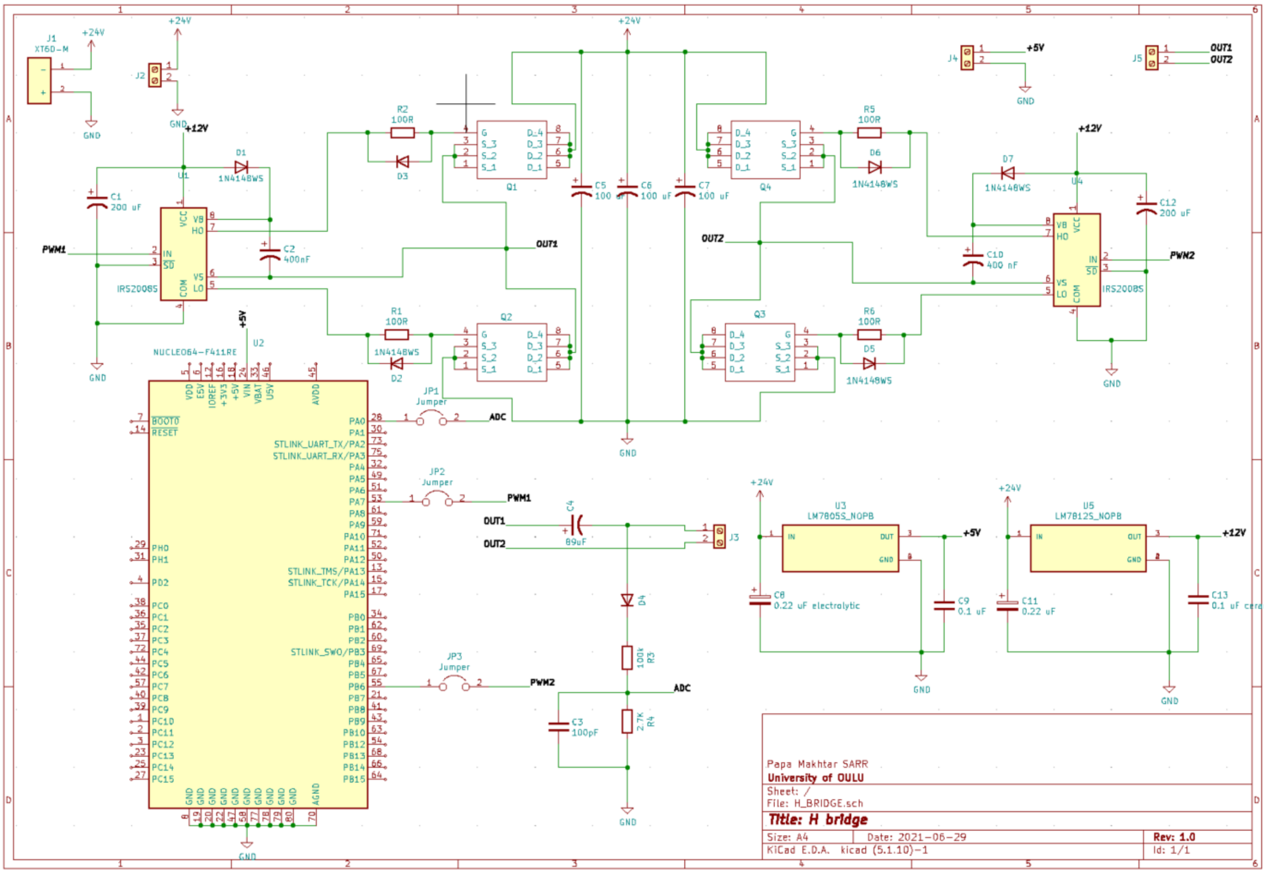


Figure 37. Sketch of the wireless charger transmitter side schematic.

7. Conclusions

This last deliverable of WP4 provided and extension of the outcomes presented in the previous work package deliverables.

It started in Section 2 with a discussion of the coverage of the overall objective of WP4 across the three deliverables. It continued presenting in Section 3 the final architecture of the ground station and its subsystems. Section 4 reported about the developments on the inspection data management, from the data acquisition from the ultrasound sensor on the hybrid robot's satellite/arm to the ground segment and their further processing into the relational database and the visualisation for the inspection engineer. Section 5 presented the outcomes on environment representation and path planning and navigation support for a hybrid robot with robotic arm and awareness of aerodynamic effects of obstacles found in oil and gas plants. Finally, Section 6 added considerations on the mobile support platform itself, including energy management.

References

D2.2 Heredia G, ed. (2022) Prototypes of the hybrid robots. H2020-ICT-25-2016-2017 779411 HYFLIERS project deliverable D2.2 (CO). Sep.

D4.1 Celentano U, ed., Kauppinen M, Pitkänen V, Celentano U, Heredia G, Alarcón F (2019) Measurement data management service and path planning algorithm. H2020-ICT-25-2016-2017 779411 HYFLIERS project deliverable D4.1 (CO). Jun (Revised Feb 2020).

D4.2 Celentano U, ed (2020), Kauppinen M, Celentano U, Caballero A, Bejar M, Heredia G, Ollero A, Trujillo MÁ, Viguria A (2020) Navigation support system and operator interface. H2020-ICT-25-2016-2017 779411 HYFLIERS project deliverable D4.2 (CO). Jun (Revised Dec 2020).

D5.1 Sanchez-Cuevas PJ, ed., Sanchez-Cuevas PJ, Neva D, Schmid R, Lippiello V, Trujillo MÁ, Heredia G, Viguria A, Ollero A, Celentano U, Kauppinen M, Röning J, Miika Sikala (2022) Hybrid robot hardware integration. H2020-ICT-25-2016-2017 779411 HYFLIERS project deliverable D5.1 (CO). Sep.

D5.2 P. J. Sanchez-Cuevas PJ, ed. Sanchez-Cuevas PJ, Benjumea D, Schmid R, Lippiello V, Trujillo MÁ, Heredia G, Viguria A, Ollero A, Celentano U, Kauppinen M, Sikala M, Mäenpää T, Röning J (2022) Hybrid robot software integration. H2020-ICT-25-2016-2017 779411 HYFLIERS project deliverable D5.2 (CO). Sep.

CabEtal2017 Caballero, A., Bejar, M., Rodriguez-Castaño, A., Ollero, A. (2017). Motion planning for long reach manipulation in aerial robotic systems with two arms. In: 2017 European Conference on Mobile Robots (ECMR), pp. 1–7.

CabEtal2018 Caballero, A., Bejar, M., Rodriguez-Castaño, A., Ollero, A. (2018). Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms. International Journal of Advanced Robotic Systems 15(3).

CabEtal2020 Caballero, A., Sanchez-Cuevas, P. J., Bejar, M., Heredia, G., Trujillo, M. A., Ollero, A. (2020). An aerodynamic extension for motion planning with dynamics awareness in aerial long-reach manipulators. International Journal of Aerospace Engineering.

CabEtal2022 Caballero, A., Bejar, M., Heredia, G., Ollero, A. Hybrid motion planning with Dynamics Awareness for aerial-ground robots in industrial inspection and maintenance. Robotics and Autonomous Systems. Submitted.

CheEtal1955 Cheeseman, I., Bennett, W. (1955). The effect of ground on a helicopter rotor in forward flight.

HorEtal2013 A. Hornung,. K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees" in Autonomous Robots, 2013; DOI: 10.1007/s10514-012-9321-0. Software available at https://octomap.github.io.

KarEtal2011 Karaman, S., Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research 30(7), 846–894.

KonEtal2007 Kondak, K., Bernard, M., Meyer, N., Hommel, G. (2007). Autonomously flying VTOL-robots: Modeling and control. In: Robotics and Automation (ICRA), 2007 IEEE International Conference on, pp. 736–741. IEEE.

MotGen Motiongenesis Kane 5.x. URL http://www.motiongenesis.com/.

SanEtal2017 Sanchez-Cuevas, P.J., Heredia, G., Ollero, A. (2017). Characterization of the aerodynamic ground effect and its influence in multirotor control. International Journal of Aerospace Engineering.

Sik2022 Sikala, Mika. “Data management system for wireless sensor ground station.” Thesis. Computer Science and Engineering, University of Oulu, Finland. 2022.

SuaEtal2020 Suarez, A., Caballero, A., Garofano, A., Sanchez-Cuevas, P. J., Heredia, G., Ollero, A. (2020). Aerial manipulator with rolling base for inspection of pipe arrays. IEEE Access, 8, 162516-162532.