DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
ESCUELA SUPERIOR DE INGENIERÍA
UNIVERSIDAD DE SEVILLA

Motion planning with dynamics and environment awareness for aerial robotic manipulation in inspection and maintenance

por

Álvaro Caballero Gómez

Graduado en Ingeniería Aeroespacial / Master Universitario en Ingeniería Aeronáutica

PROPUESTA DE TESIS DOCTORAL
PARA LA OBTENCIÓN DEL TÍTULO DE

DOCTOR POR LA UNIVERSIDAD DE SEVILLA
SEVILLA, 2022

Directores

Dr. Ing. Aníbal Ollero Baturone, Catedrático
Dr. Ing. Manuel Béjar Domínguez, Profesor Titular

UNIVERSIDAD DE SEVILLA

Memoria para optar al grado de Doctor por la Universidad de Sevilla

Autor: **Álvaro Caballero Gómez**
Título: **Motion planning with dynamics and environment awareness for aerial robotic manipulation in inspection and maintenance**
Departamento: **Departamento de Ingeniería de Sistemas y Automática**

V° B° Director:

Aníbal Ollero Baturone

V° B° Director:

Manuel Béjar Domínguez

El autor:

Álvaro Caballero Gómez

A mi familia,
A Cristina.

Agradecimientos

En primer lugar, me gustaría dar las gracias a Aníbal Ollero por todas las oportunidades que me ha ofrecido durante todos estos años en el GRVC. Él me abrió las puertas de este apasionante mundo. De igual forma, mi más sincero agradecimiento a Manuel Béjar, quien me enseñó a caminar con serenidad e inteligencia, y siempre apuntando hacia la perfección. Me quedo con sus consejos, todos ellos de gran valor.

A mis compañeros del GRVC, a los que están y a los que estuvieron. Si algo me habéis enseñado es que nadie es tan bueno como todos juntos. Gracias especialmente a, por orden alfabético, Alejandro Suárez, Fran Real, Honorio Romero, María Polvillo, Pedro Sánchez, Rafael Salmoral, Ramiro Martínez, Víctor Quesada y Víctor Vega. Los demás también estáis en mi corazón.

Por supuesto, unas palabras a lo más valioso que tengo en esta vida. Muchas gracias familia. Todo lo que he conseguido ha sido gracias a vosotros. Me habéis dado tanto sin esperar nada a cambio... Mucho más de lo que vosotros mismos creéis. Vuestro apoyo incondicional merece, al menos, una tesis.

Para acabar, me gustaría dejar escritas aquí unas palabras para mi yo del futuro. No son un agradecimiento, sino un consejo. Seguro que vendrán momentos duros, no olvides que hay que lucharlos. También vendrán triunfos, disfruta hasta los más simples y recuerda que sin honor ninguno de ellos merecerá la pena.

Muchas gracias a todos,

Álvaro Caballero Gómez
Sevilla, 1 de septiembre de 2022

Este trabajo ha sido financiado a través de las ayudas para contratos predoctorales para la formación de doctores (FPI 2018) financiadas por el Ministerio de Ciencia, Innovación y Universidades, por los proyectos nacionales ARM-EXTEND (DPI2017-89790-R) y HAERA (PID2020-119027RB-I00) financiados por el Ministerio de Economía y Competitividad y por el Ministerio de Ciencia e Innovación, respectivamente, y por los proyectos europeos H2020 AERIAL-CORE (H2020-2019-871479), HYFLIERS (H2020-ICT-25-2017-779411) y AEROARMS (H2020-ICT-2014-1-644271) financiados por la Comisión Europea.

This work has been supported by the 2018 FPI programme funded by the Spanish Ministerio de Ciencia, Innovación y Universidades, by the national projects ARM-EXTEND (DPI2017-89790-R) and HAERA (PID2020-119027RB-I00) funded by the Spanish Ministerio de Economía y Competitividad and by the Spanish Ministerio de Ciencia e Innovación, respectively, and by the European H2020 projects AERIAL-CORE (H2020-2019-871479), HYFLIERS (H2020-ICT-25-2017-779411) and AEROARMS (H2020-ICT-2014-1-644271) funded by the European Commission.

Resumen

Los Vehículos Aéreos No Tripulados (UAVs, por sus siglas en inglés) dotados de capacidades robóticas de manipulación, también conocidos como Manipuladores Robóticos Aéreos (ARMs, por sus siglas en inglés), han demostrado un futuro prometedor en su aplicación para actividades de Inspección y Mantenimiento (I&M). Sin embargo, sus capacidades asociadas aún necesitan ser ampliadas para alcanzar mayores niveles de autonomía, fiabilidad, precisión, seguridad y eficiencia, entre otras. Además de importantes mejoras en la seguridad humana, esto supondrá un ahorro de costes significativo, haciendo de los ARMs una solución efectiva para ser explotada en condiciones reales.

Motivada por la afirmación anterior, esta tesis ha encontrado en la planificación de movimiento un medio para dotar a los ARMs de funcionalidades mejoradas. Así, la investigación presentada se ha centrado en el diseño, desarrollo y validación de métodos de planificación de movimiento para manipulación robótica aérea en I&M. Más en detalle, se ha identificado la necesidad de planificación para tres temas principales, que son manipulación usando ARMs dotados de brazos robóticos, manipulación con robots de locomoción híbrida y manipulación multiARM. Para cada uno de ellos, se ha formulado un método de planificación de movimiento y posteriormente se han introducido varias extensiones para aumentar sus capacidades.

En cuanto a manipulación usando ARMs dotados de brazos robóticos, se ha formulado un planificador de movimiento especialmente orientado a este tipo de ARMs para fases tanto de navegación como de manipulación en entornos con alta densidad de obstáculos. Este planificador considera la operación conjunta de la plataforma aérea y del sistema de manipulación dentro del proceso de planificación.

Sobre los fundamentos del método anterior, se han propuesto tres extensiones. Debido a la compleja dinámica que existe en los ARMs, la primera extensión introduce Conciencia de la Dinámica (DA, por sus siglas en inglés) en la operación del planificador para la evitación robusta de obstáculos. Complementando esta extensión DA, un nuevo mecanismo de Adaptación de Velocidad (VA, por sus siglas en inglés) permite una mejor optimización del tiempo de ejecución de las trayectorias planificadas pero sin aumentar considerablemente la carga computacional. Alternativamente, considerando la aerodinámica de los ARMs, el enfoque DA también se ha ampliado con Conciencia de la Aerodinámica (ADA, por sus siglas en inglés) para hacer frente a fenómenos aerodinámicos que pueden hacer que los robots colisionen.

Cambiando a manipulación con robots de locomoción híbrida, el planificador de movimiento se dedica a la generación de planes eficientes a lo largo de secuencias de puntos de manipulación, involucrando este tipo de ARMs en entornos con alta densidad de obstáculos. Para ello, el método aprovecha las capacidades de vuelo y rodadura que ofrecen estos robots.

Finalmente, con el foco en manipulación multiARM, se ha presentado un planificador de movimiento para sistemas multiARM sujetos a capacidades de carga de pago limitadas y restricciones dinámicas. Este método da respuesta a misiones que requieren visitar de manera eficiente un conjunto de regiones objetivo donde se despliegan cargas a bordo de los robots.

Resultados numéricos, simulaciones realistas y experimentos de vuelo reales en interiores y exteriores han demostrado los beneficios de los métodos de planificación para calcular trayectorias que llevan a una amplia variedad de ARMs a realizar operaciones reales de I&M en diferentes escenarios. Estos escenarios de aplicación van desde la instalación de sensores en entornos industriales hasta la inspección de racks de tuberías en refinerías de petróleo y gas o plantas químicas, pasando por la instalación de salvapájaros en líneas eléctricas.

Abstract

Unmanned Aerial Vehicles (UAVs) endowed with robotic manipulation capabilities, also known as Aerial Robotic Manipulators (ARMs), have demonstrated a promising future in their application for Inspection and Maintenance (I&M) activities. However, their associated capabilities still need to be extended to reach higher levels of autonomy, reliability, accuracy, safety and efficiency, among others. In addition to important improvements in human safety, this will lead to significant cost savings, making ARMs an effective solution to be exploited in real conditions.

Motivated by the previous statement, this thesis has found in motion planning a means to endow ARMs with enhanced functionalities. Thus, the presented research has been focused on the design, development and validation of motion planning methods for aerial robotic manipulation in I&M. More in detail, the need of planning has been identified for three main topics, which are manipulation using ARMs endowed with robotic arms, manipulation with hybrid-locomotion robots and multiARM manipulation. For each of them, a motion planning method has been formulated and then, several extensions have been introduced to increase its capabilities.

Concerning manipulation using ARMs endowed with robotic arms, a motion planner specially oriented to this kind of ARMs has been formulated for both navigation and manipulation phases in cluttered environments. This planner considers the joint operation of the aerial platform and the manipulation system within the planning process.

Over the fundamentals of the previous method, three extensions have been proposed. Due to the complex dynamics existing in ARMs, the first extension introduces Dynamics Awareness (DA) in the planner operation for robust obstacle avoidance.

Complementing this DA extension, a new Velocity Adaptation (VA) mechanism allows a better optimisation of the execution time of the planned trajectories but without increasing the computational burden considerably. Alternatively, accounting for the ARM aerodynamics, the DA approach has also been extended with Aerodynamics Awareness (ADA) to face aerodynamic phenomena that may address robots to collisions.

Switching to manipulation with hybrid-locomotion robots, the motion planner is devoted to the generation of efficient plans along sequences of manipulation points, involving this kind of ARMs in cluttered environments. For that, the method takes advantage of the flying and rolling capabilities offered by these robots.

Finally, with the focus on multiARM manipulation, a motion planner for multiARM systems subject to limited payload capacities and dynamic constraints has been presented. This method gives response to missions that require visiting efficiently a set of target regions where loads on board the robots are deployed.

Numerical results, realistic simulations and real-world indoor and outdoor flight experiments have demonstrated the benefits of the planning methods to compute trajectories that lead a wide variety of ARMs to fulfil real I&M operations in different scenarios. These application scenarios range from the installation of sensors in industrial environments to the inspection of pipe arrays in oil and gas refineries or chemical plants and even the installation of bird flight diverters on electric power lines.

Contents

Agradecimientos **vii**

Resumen **ix**

Abstract **xi**

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions 3

1.2.1 Motion planning methods. Overall view 5

1.2.2 Scientific publications 8

1.3 Framework 9

1.4 Organisation of this thesis 10

2 State of the art **13**

2.1 Aerial robotic manipulation 13

2.2 Motion planning for aerial robotic manipulation 17

2.3 Extensions on motion planning: dynamics and aerodynamics awareness 18

2.4 Motion planning for hybrid robots 20

2.5 Multirobot motion planning: installation of bird diverters 21

2.6 Conclusions 23

3 Motion planning for aerial manipulators with robotic arms **25**

3.1 Motivation 25

3.2 Motion planner 26
 3.2.1 Fundamentals of planner operation 27
 3.2.2 In-flight planning capabilities 35
3.3 Simulation results 38
3.4 Experimental results 43
 3.4.1 Indoor motion planning 43
 3.4.2 Outdoor motion planning 47
 3.4.3 In-flight outdoor motion planning 50
 3.4.4 Industrial validation 54
3.5 Conclusions 62

4 Extensions on motion planning for aerial manipulators 65
4.1 Dynamics awareness 66
 4.1.1 Motivation 66
 4.1.2 Motion planner 67
 4.1.3 Simulation results 68
4.2 Velocity adaptation 80
 4.2.1 Motivation 80
 4.2.2 Motion planner 80
 4.2.3 Simulation results 83
4.3 Aerodynamics awareness 88
 4.3.1 Motivation 88
 4.3.2 Motion planner 89
 4.3.3 Simulation results 91
4.4 Conclusions 98

5 Hybrid motion planning for inspection with aerial-ground robots 101
5.1 Motivation 101
5.2 Motion planner 103
 5.2.1 Fundamentals of planner operation 103
 5.2.2 Dynamics Awareness 116
 5.2.3 Hybrid replanning 118

- 5.3 Simulation results . . . 119
  - 5.3.1 Fundamentals of planner operation . . . 121
  - 5.3.2 Dynamics Awareness . . . 124
  - 5.3.3 Hybrid replanning . . . 129
- 5.4 Experimental results . . . 131
  - 5.4.1 Fundamentals of planner operation . . . 132
- 5.5 Conclusions . . . 135

6 Motion planning for fleets of aerial manipulators 137

- 6.1 Motivation . . . 137
- 6.2 Motion planner . . . 139
  - 6.2.1 Fundamentals of planner operation . . . 140
  - 6.2.2 Energy awareness . . . 145
  - 6.2.3 Event-based replanning . . . 147
- 6.3 Simulation results . . . 147
  - 6.3.1 Fundamentals of planner operation . . . 149
  - 6.3.2 Energy awareness . . . 155
  - 6.3.3 Event-based replanning . . . 157
- 6.4 Experimental results . . . 159
  - 6.4.1 Fundamentals of planner operation . . . 160
- 6.5 Conclusions . . . 161

7 Conclusions and future work 165

- 7.1 Conclusions . . . 165
- 7.2 Future work . . . 168

A Aerial robotic manipulators 171

- A.1 Aerial robotic system for long-reach manipulation (version 1) . . . 171
  - A.1.1 System description . . . 171
  - A.1.2 Modelling and control . . . 172
- A.2 Aerial robotic system for long-reach manipulation (version 2) . . . 178
  - A.2.1 System description . . . 178

A.2.2 Modelling and control 182
A.3 Aerial robotic system for long-reach manipulation (version 3) 185
A.3.1 System description 185
A.3.2 Modelling and control 187
A.4 Hybrid robotic manipulator 190
A.4.1 System description 190
A.4.2 Modelling and control 192
A.5 Aerial robotic manipulator with linear actuator 194
A.5.1 System description 194

B Characterisation of aerodynamic effects 197
B.1 Test bench 197
B.2 Experimental procedure 198

C Fundamentals of signal temporal logic for motion planning 203
C.1 Preliminaries 203
C.2 Signal temporal logic 204
C.3 Robust signal temporal logic 204
C.4 Smooth approximation 205
C.5 STL motion planner 206

References 207

List of Figures

1.1 Organisation of this thesis. General scheme. 12

3.1 Collision checking for the multirotor based on the position of its centre of mass M^O. Round obstacle (left) and rectangular obstacle (right). . 31

3.2 Collision checking for the right upper link of the ARS-LRM_v1 system. Round obstacle (left) and rectangular obstacle (right). 31

3.3 Regions in which the right arm of the ARS-LRM_v1 system collides with an obstacle for fixed UAV and obstacle positions. Round obstacle (left) and rectangular obstacle (right). The shape of the collision regions varies with the relative position between the multirotor and the obstacle. 32

3.4 Effect of the weighting parameters $p_{1,2}$ in the cost function CF_E. . . . 34

3.5 Integration process of the point-cloud representation into the motion planner. 36

3.6 Application scenario given by a riveting task. 38

3.7 Riveting task (navigation phase). The ARS-LRM_v1 system navigates between the obstacles. The dotted line represents the simulated centre of mass M^O position of the multirotor. 40

3.8 Riveting task (navigation phase). Evolution of the planning-space variables: planned (dashed black) and executed (blue) trajectories. . . 41

3.9 Riveting task (manipulation phase). The ARS-LRM_v1 system places the first rivet and then switches between the ready-to-go configurations. 42

3.10 Riveting task (manipulation phase). Evolution of the planning-space variables: planned (dashed black) and executed (blue) trajectories. . . 42

3.11 Application scenario given by a pipe inspection. 44

3.12 Mock-up for pipe inspection. Reference points and operation limits (black lines). 45

3.13 Pipe inspection. Snapshots of the operation. 46

3.14 Pipe inspection. Evolution of the planning-space variables: experiment (left) and simulation (right). The dashed lines represent the planned trajectory. The time intervals between the vertical dotted lines correspond to the inspection operation itself. 46

3.15 Application scenario given by a long-bar transportation. 48

3.16 Long-bar transportation. Snapshots of the operation. 49

3.17 Long-bar transportation. Evolution of the planning-space variables: planned (dashed black) and executed (red) trajectories. 50

3.18 Application scenario given by a long-bar transportation used for the validation of in-flight planning capabilities. 51

3.19 Long-bar transportation with in-flight planning capabilities. Snapshots of the operation. 52

3.20 Long-bar transportation with in-flight planning capabilities. Trajectory planned in flight: 3D view including the representation of the scenario generated by the Multisensor Mapping module (top) and 2D views for the collision spaces associated with the ARS-LRM_v2 system (centre) and the transported bar (bottom). 53

3.21 Long-bar transportation with in-flight planning capabilities. Evolution of the planning-space variables: planned (dashed black) and executed (red) trajectories. 55

3.22 Mock-up sensor to be installed on the surface of the ball mill. 56

3.23 Application scenario given by a sensor installation on an industrial ball mill. 56

3.24 Sensor installation on an industrial ball mill. Snapshots of the operation. 58

3.25 Sensor installation on an industrial ball mill. Trajectory planned in flight: 3D view including the representation of the scenario generated by the Multisensor Mapping module (top) and 2D view for the collision space associated with the ARS-LRM_v2 system (bottom). 59

3.26 Sensor installation on an industrial ball mill. Evolution of the planning-space variables: planned (dashed black) and executed (red) trajectories. 60

3.27 Sensor installation on an industrial ball mill. Trajectory planned for the installation on the back surface of the mill. 61

3.28 Inspection by contact of different surfaces of interest. Application scenarios (left) and planned trajectories (right). 63

4.1 ARS-LRM_v1 system spreading the left arm from initial position 0 to final position f according to a simple plan. The ARM oscillation during the execution (shaded intermediate position i) produces a collision with the yellow obstacle. 66

4.2 Operation basis of the $COLLISION(x_{nearest}, x_{new}, map)$ function: blue dashed line for the MP-ARM algorithm and green solid line for the MP-ARM-DA algorithm. 67

4.3 Guiding obstacles enforcing a rectilinear movement perpendicular to the contact surface. 68

4.4 Application scenario given by a riveting task used for the validation of motion planning with Dynamics Awareness. 70

4.5 Application scenario given by a chimney repairing task. 71

4.6 Riveting task (navigation phase). Execution (dark blue) of the trajectory planned (light blue) with the MP-ARM algorithm. 73

4.7 Riveting task (manipulation phase, rivet placement 1). Execution of the trajectory planned with the MP-ARM algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configuration 0 to final configuration f through shaded intermediate configuration i. 73

4.8 Chimney repairing task (manipulation phase, repair). Execution of the trajectory planned with the MP-ARM algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configuration 0 to final configuration f through shaded intermediate configuration i. 74

4.9 Riveting task (navigation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$. . . 75

4.10 Riveting task (navigation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm. 75

4.11 Chimney repairing task (navigation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2}$. . . 76

4.12 Chimney repairing task (navigation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm. 76

4.13 Riveting task (manipulation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configurations 0 to final configurations f through shaded intermediate configurations i. 78

4.14 Riveting task (manipulation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm. 78

4.15 Chimney repairing task (manipulation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configurations 0 to final configurations f through shaded intermediate configurations i. 79

4.16 Chimney repairing task (manipulation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm. 79

4.17 Operation basis of the MP-ARM-DAVA algorithm. $u_{ref}^{max}, w_{ref}^{max}$ are the initial maximum velocities for the aerial platform and the manipulator joints, respectively, and u_{ref}, w_{ref} correspond to the resulting maximum allowable velocities that guarantee safe trajectories for the branch under study. 81

4.18 Flow diagram representing the basis operation of the MP-ARM-DAVA algorithm. 82

4.19 Chimney repairing task (navigation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$ and has a density of points inversely proportional to the UAV velocity. . . 84

4.20 Chimney repairing task (navigation phase). Evolution of the UAV velocity: execution of the trajectory planned with the MP-ARM-DA algorithm. 85

4.21 Chimney repairing task (navigation phase). Execution of the trajectory planned with the MP-ARM-DAVA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$ and has a density of points inversely proportional to the UAV velocity. 86

4.22 Chimney repairing task (navigation phase). Evolution of the planning-space variables: trajectories planned (light purple) and executed (dark purple) using the MP-ARM-DAVA algorithm. For comparison purposes, the execution of the trajectory planned with the MP-ARM-DA algorithm (dark green) is also represented. 87

4.23 Chimney repairing task (navigation phase). Evolution of the UAV velocity: execution of the trajectory planned with the MP-ARM-DAVA algorithm (dark purple). For comparison purposes, the execution of the trajectory planned with the MP-ARM-DA algorithm (dark green) is also represented. 87

4.24 UAV trying to track a straight planned trajectory between two obstacles. The vehicle crashes with the upper obstacle as a consequence of an undesirable ascent caused by the ground effect. 88

4.25 Operation basis of the MP-ARM-ADA algorithm (left) and closed-loop scheme required to perform the simulations that give support to it (right). The terms highlighted in red correspond to the variables employed to model the Aerodynamics Awareness. 90

4.26 Reinforced concrete bridge. Image extracted from Google Maps [1]. . 92

4.27 Application scenario given by a bridge inspection. 93

4.28 3D map of aerodynamic effects for the bridge scenario. 94

4.29 Bridge inspection. Execution (dark blue) of the trajectory planned (light blue) with the MP-ARM algorithm. The dashed and dotted lines represent, respectively, the movement of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$. 95

4.30 Bridge inspection. Evolution of the configuration variables: trajectories planned (light blue) and executed (dark blue) using the MP-ARM algorithm. 95

4.31 Bridge inspection. Execution (dark green) of the trajectory planned (light green) with the MP-ARM-DA algorithm. The dashed and dotted lines represent, respectively, the movement of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$. 96

4.32 Bridge inspection. Evolution of the configuration variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm. 96

4.33 Bridge inspection. Execution (dark orange) of the trajectory planned (light orange) with the MP-ARM-ADA algorithm. The dashed and dotted lines represent, respectively, the movement of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$. 97

4.34 Bridge inspection. Evolution of the configuration variables: trajectories planned (light orange) and executed (dark orange) using the MP-ARM-ADA algorithm. 98

5.1 Cluttered industrial environment with many pipe arrays. 102

5.2 Scheme with both aerial and pipe states during the hybrid expansion of the search tree associated with the MP-HR method. 104

5.3 Effect of the weighting parameter μ in the process of discretisation and sampling of the hybrid planning space: the complete set of n_{pipe} pipe and n_{air} aerial states corresponding to the discretisation (left), the selected subset with uniform sampling (centre) and the selected subset corresponding to the sampling with compensating effect that results in a more balanced distribution (right). 106

5.4 Illustrative examples of the criteria to select yaw angles for the different types of nodes in the hybrid planning space. 108

5.5 Basic behaviours corresponding to the optimisation of the proposed cost functions. Flying is prioritised over rolling when minimising the operation time T ($u_{fly} > u_{roll}$), while rolling is prioritised over flying when minimising the energy consumption E ($e_{roll} < e_{fly}$). 109

5.6 Feasibility checking in transitions between aerial nodes and pipe nodes. Transitions taking place out of the region of safe transition are discarded. 114

5.7 Collision checking based on the minimum distance between the HR and the pipe. The robot is approximated by its bounding sphere while the pipe is modelled as a cylinder. Margins of safety are also considered. . 116

5.8 Operation basis of the MP-HR-DA algorithm: feasibility analysis of a new air-air branch (left) and feasibility analysis of a new air-pipe branch (right). 117

5.9 Example of application scenarios for hybrid reactivity. 119

5.10 Example scenario illustrating the criteria followed to represent the different application scenarios. 120

5.11 Application scenario given by an inspection plan in moderately-cluttered area: topology definition (left) and the corresponding safety regions (right). 122

5.12 Inspection plan in moderately-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the operation time. 123

5.13 Overshooting in reference waypoints extracted from Fig. 5.12: landing on the inspection point 3 (left) and final approaching to the start point (right). 123

5.14 Inspection plan in moderately-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the energy consumption. 125

5.15 Application scenario given by an inspection plan in highly-cluttered area: topology definition (left) and the corresponding safety regions (right). 126

5.16 Inspection plan in highly-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the operation time. 127

5.17 Detailed views of Fig. 5.16: violations of safety margin (left) and region of safe transition (right). 127

5.18 Inspection plan in highly-cluttered area. Execution (dark green) of the trajectory planned (light green) with the MP-HR-DA algorithm when optimising the operation time. 128

5.19 Detailed views of Fig. 5.18: the MP-HR-DA method overcomes the violations of the safety margin (left) and the region of safe transition (right). 129

5.20 Application scenario given by an inspection plan in moderately-cluttered area with unmapped obstacle: topology definition (left) and the corresponding safety regions (right). The pipe highlighted in green will not be included in the map provided to the motion planner. 130

5.21 Inspection plan in moderately-cluttered area with unmapped obstacle. Hybrid-reactive capabilities for rolling manoeuvres: trajectory initially planned by the MP-HR algorithm without considering the green pipe (left) and execution (dark blue) of the trajectory replanned (light blue) with the MP-HR algorithm when the green pipe is detected by onboard sensors (right). 131

5.22 Application scenario given by an inspection plan in scarcely-cluttered area. 132

5.23 Inspection plan in scarcely-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the energy consumption. 133

5.24 Inspection plan in scarcely-cluttered area. Snapshots of the operation. 134

6.1 Example of bird flight diverters installed on an electric power line. . . 138

6.2 Graph-based representation of the bird diverter installation problem. Example of graph G built upon the assumption of two target regions (square vertices), two refilling stations (round solid vertices) and two ARMs. The depots are embedded into the round dashed vertex. . . . 143
6.3 Application scenario given by a bird diverter installation task. 150
6.4 Bird diverter installation task. ARM trajectories planned with the MP-MARM algorithm. 151
6.5 Bird diverter installation task. Evolution of the ARM variables: trajectories planned using the MP-MARM algorithm. The installation and refilling time windows are shaded using blue and green slots, respectively, and the dashed red lines highlight the admissible limits in the variables. 152
6.6 Bird diverter installation task. Snapshots of the simulation in Gazebo. The white circles highlight the ARMs while the red spheres symbolise the installed bird diverters. 153
6.7 Bird diverter installation task. Initial guess used to compute the ARM trajectories represented in Fig. 6.4. 154
6.8 Bird diverter installation task. Comparison between the initial guess (left) and the final solution computed by the MP-MARM planner and also represented in Fig. 6.4 (right). 155
6.9 Bird diverter installation task. ARM trajectories planned with the MP-MARM algorithm including energy awareness. 156
6.10 Bird diverter installation task. Comparison between the evolution of the forward velocities planned using the MP-MARM algorithm in its standard operation (left) and including energy awareness (right). The dashed black lines highlight the optimal forward velocity. 157
6.11 Bird diverter installation task. ARM trajectories after replanning the task in Fig. 6.4 by applying the MP-MARM algorithm when the ARM 2 suffers a failure in the refilling station $RS\ D$. The ARM 1 continues its ongoing operation while the back-up ARM 3 completes the remaining mission of the faulty ARM 2. 158

6.12 Bird diverter installation task. Execution of the ARM trajectories planned with the MP-MARM algorithm. 161
6.13 Bird diverter installation task. Evolution of the ARM variables: execution of the trajectories planned using the MP-MARM algorithm. The installation and refilling time windows are shaded using blue and green slots, respectively, and the dashed red lines highlight the admissible limits in the variables. 162
6.14 Bird diverter installation task. Snapshots of the real experiment. The power line has been represented using augmented reality. The white circles highlight the robots while the red spheres symbolise the installed bird diverters. 163

A.1 First version of a set of Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v1). 172
A.2 Geometry and mass distribution of the ARS-LRM_v1 system. 173
A.3 Configuration variables of the ARS-LRM_v1 model. In green, the variables selected in Chapters 3 and 4 for the planning space. 175
A.4 Forces and torques applied to the ARS-LRM_v1 system. 175
A.5 Block diagram of the ARS-LRM_v1 distributed control scheme. . . . 176
A.6 Block diagram of the ARS-LRM_v1 UAV controller. 177
A.7 Block diagram of the ARS-LRM_v1 dual-arm controller. 177
A.8 Second version of a set of Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v2). First (left) and final (right) prototypes. 179
A.9 Software architecture of the ARS-LRM_v2 system. 180
A.10 Configuration variables of the ARS-LRM_v2 model (blue and green), and forces and torques applied to it (yellow). In green, the configuration variables selected in Chapter 3 for the planning space. 183
A.11 Block diagram of the ARS-LRM_v2 distributed control scheme. . . . 184
A.12 Third version of a set of Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v3). Geometry and mass distribution. 186

A.13 Configuration variables of the ARS-LRM_v3 model (green), and forces and torques applied to it (purple). 188
A.14 Block diagram of the ARS-LRM_v3 distributed control scheme. . . . 189
A.15 Hybrid robotic manipulator (HRM) with aerial-rolling locomotion capabilities. 191
A.16 HRM configuration for the inspection of pipe arrays. 191
A.17 Block diagram of the HRM control scheme. 193
A.18 Aerial Robotic Manipulator with Linear Actuator (ARM-LA) for the installation on power lines of clip-type bird diverters (bottom corners). 195

B.1 Test bench used for the characterisation of the aerodynamic effects. . 198
B.2 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). 200
B.3 3D map corresponding to the aerodynamic effects associated with a rectangular-shaped obstacle. The colour scale represents the magnitude of the ratio τ 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. 202

Chapter 1

Introduction

> Si te lo dice tu corazón, no es
> necesario que te lo diga alguien más.
>
> Ana M. Gómez

1.1 Motivation

At the beginning of aerial robotics, the application of Unmanned Aerial Vehicles (UAVs) was essentially limited to operations where the robots had to gather some information from the environment, generally using onboard sensors like visual cameras. However, thanks to the fast growing in this field during the last years, this kind of robots is being used for much more complex tasks. Thus, it is not difficult to find current applications where the aerial robots have to interact physically with the environment. Potential tasks include instrument deployment, maintenance operation and contact inspection in industrial sites or civil constructions in which the access is very dangerous or costly. The motivation is to decrease risks and operational costs in these scenarios with the support of aerial robotic systems. Small size rotorcraft can indeed access to hard-to-reach places more easily than human operators, avoiding unnecessary risks for industrial workers and allowing Inspection and Maintenance (I&M) operations without shut-downs of the facilities (the mandatory safety policy in case of human operation) and without the use of scaffolding or cranes. Aligned with this kind of operations, the concept of *Aerial Robotic Manipulators (ARMs)* arose. In general terms, an ARM can be defined as a UAV endowed with robotic manipulation capabilities [2, 3, 4].

These new promising applications of aerial robotic systems for manipulation tasks bring also new challenges. First, it is necessary to develop new manipulation or inspection tools such as adapted arms or grippers that can be seamlessly integrated into the airframe to provide manipulation capabilities to UAVs. Furthermore, the existing algorithms to operate autonomously the UAV and the manipulator should be extended for the new system. In this respect, one of the most challenging issues is the development of new methods that consider both the UAV and the manipulator in an integrated way. From the point of view of the *dynamics*, there is a strong dynamical coupling between both subsystems that should not be neglected. In contrast to initial approaches where the manipulator was modelled as a disturbance for the aerial platform, the full-dynamics consideration can help to reach higher levels of safety and accuracy in the operation and, consequently, a better performance. Focusing on the efficiency, more system states can be accounted when the aerial manipulator is considered as a whole. Thus, the best movements minimising the total energy consumption can be computed to avoid a waste of battery. The cost to be paid is the higher computational requirements in the associated algorithms.

The environments where ARMs fly may also become another challenging issue. These robots frequently operate in industrial environments with a high density of obstacles likes pipes or other elements that might hinder the operation substantially. These cluttered scenarios force the ARMs to fly close to obstacles, which lead to an increment in the risk of collision. Also, these obstacles imply the presence of *aerodynamics* effects like ground effect that may destabilise the vehicle. However, the difficulties in these scenarios can be partially turned into advantages. In this sense, the high density of obstacles like pipes can be exploited as a means to move along the environment while saving energy. More particularly, ARMs with hybrid air and ground locomotion could fly towards an operation area located at height, and once there, move between points rolling on the pipes when it is possible, or flying otherwise.

In this manner, the lower energy consumption associated with ground locomotion can be complemented with the higher accessibility associated with flying locomotion, making the operation more efficient for these *Hybrid Robots (HRs)*.

Even in applications where ARMs offer some unique advantages, the use of single robots might be insufficient. For instance, the execution of simple but repetitive tasks like the deployment of many sensors to monitor the state of certain infrastructure might be inefficient for only one ARM. The time required to place all the devices together with the limited payload and battery capacity might lead to an excessive mission time. In contrast, *multiARM* approaches are demonstrating to be a promising solution that helps to save time. However, there are still open challenges to coordinate all the robots in the search of both safety and global efficiency. Indeed, when a team of heterogeneous ARMs has to visit a set of targets for load deployments in complex environments, the combination of efficiency requirements, like the search of optimal travel sequences subject to payload constraints, with safety requirements, like obstacle avoidance, minimum distance between robots or dynamic feasibility in the trajectories, still requires further analysis.

Giving response to all the previous aspects requires a wide variety of integrated actions in the fields of Robotics. From design or communications to perception, control and navigation. This thesis is focused on addressing the problem from a *motion planning* perspective. The main contributions are described in the following section.

1.2 Contributions

This thesis is devoted to developing motion planning methods specially oriented to ARMs in I&M tasks. These methods should face the open issues presented in previous section, with the purpose of endowing ARMs with better performances and higher levels of autonomy. In this context, the main contributions of this thesis can be enumerated as follows:

1. Development of motion planners specially oriented to ARMs for safe operation in cluttered environments. These motion planners have been adapted to the

particularities of ARMs, exploiting the full-system capabilities (aerial platform and manipulator) when required in scenarios with a high density of obstacles like industrial sites.

2. Optimisation mechanisms to increase efficiency in terms of operation time or energy consumption. The motion plans minimise the waste of mission time or battery. Accordingly, effective and fast trajectories with a reduced number of ARM movements are prioritised when possible for time optimisation. In contrast, the optimisation of the energy lies in the use of consumption models.

3. Enhancement of the planners with extended capabilities: in-flight planning, reactivity and replanning. The planning strategies are flexible enough to answer against changes or uncertainties in the environment. Depending on the situation, the plans can be computed in flight after building a map of the environment, react to unknown obstacles interfering the trajectories or be replanned once an event turns the plan outdated.

4. Awareness of the ARM dynamics in the motion planning. In order to mitigate the effect of differences between planned and executed trajectories, increasing the accuracy and safety for robust obstacle avoidance in the operation, the plans can be computed according to the dynamical behaviour of the ARMs. Models of the robot dynamics are used.

5. Consideration of aerodynamic phenomena in the motion planning. The inclusion into the planning process of aerodynamic effects close to surfaces in the environment, like ground or ceiling effects, helps to predict areas prone to disturbing the ARM flight. The motion planner can act in consequence, favouring the flight in zones that lead to a more robust ARM operation. These aerodynamic effects have been characterised previously.

6. Design of motion planners for ARMs with hybrid aerial-ground locomotion. When a robot has the capability to move either flying or rolling on the elements in the operation area, the motion planner is able to exploit the environment to

benefit the system operation. Its performance in terms of efficiency is maximised without neglecting the rest of aspects like the safety.

7. Evolution of motion planning techniques for ARM fleets. When certain application can be hardly addressed in practice using a single ARM but the coordinated use of a team of ARMs can be an effective solution, multiARM planning is formulated. The motion planner does not focus on computing isolated trajectories for each robot, which may be suboptimal. Instead of that, an integrated approach where all the robots are coordinated together is followed. The method accounts for potential heterogeneities in the ARMs as well as multirobot constraints.

8. Validation of the motion planners using existing ARMs in realistic applications. The resulting planning methods demonstrate that are suitable to operate real ARMs in real situations, breaking the frontier of theoretical approaches with difficult applicability in real-world conditions.

The contributions described above correspond to significant advances in the state of the art of motion planning for aerial manipulation in inspection and maintenance. The associated motion planning methods are summarised in the next subsection. Such results have given place to the scientific publications collected in Section 1.2.2.

1.2.1 Motion planning methods. Overall view

This thesis consists of a set of new motion planning methods that can be arranged around four blocks. Each of them is introduced below.

Firstly, a motion planning method specially oriented to ARMs endowed with robotic arms has been investigated for both navigation and manipulation tasks in cluttered environments. The planner is addressed considering jointly the aerial platform and the manipulator within the planner operation, which constitutes a remarkable difference to previous contributions where the motion planning problem was addressed in a decoupled way. This integrated strategy allows the consideration of a more complete set of system states that in turn will make it possible to achieve wider operating conditions. The objective is to guarantee safe and efficient trajectories for systems with limited manoeuvrability like the ARMs. Moreover, the planning algorithm offers the possibility to be integrated with a mapping module that builds a representation of the environment in flight. This integration, together with the effort to have a computationally-efficient method, allows in-flight motion planning capabilities. Finally, the approach has been intensively validated for different ARMs in simulation and real experiments, including autonomous outdoor flights in industrial sites.

Over the fundamentals of the previous method, several extensions have been proposed. First, the complex dynamics existing in ARMs makes necessary to introduce *Dynamics Awareness (DA)* in the planner operation for robust obstacle avoidance. Accordingly, the search process is based on the behaviour of the controlled ARM instead of using classic geometrical checks. Complementing the DA approach, a new *Velocity Adaptation (VA)* mechanism allows a better optimisation of the execution time of the planned trajectories but without extending the planning space to the complete set of motion variables (position and velocity). In this sense, only the position variables are considered in the search process, which minimises the computational burden. In order to maintain the velocity as an additional exploration degree, the new VA mechanism based on an bounded iterative process has been implemented. Alternatively, the DA approach is extended with *Aerodynamics Awareness (ADA)*. This new formulation implies that the search process will be based not only on the dynamics of the controlled system but also on its aerodynamic behaviour. A characterisation of the aerodynamic effects is considered within the trajectory generation process to discard states whose associated aerodynamic phenomena may provoke undesired collisions. In contrast, alternatives that lead to the most efficient trajectories within the area of safe operation are explored. The resulting motion planning strategies have been validated satisfactorily through simulation in different realistic scenarios.

In a third stage, the research focuses on exploiting hybrid locomotion in ARMs for autonomous inspection tasks in industrial environments like oil and gas refineries or chemical plants. For that, a novel motion planning method that takes advantage of flying and rolling on pipes capabilities offered by HRs has been developed to generate safe plans along complete sequences of inspection points. Moreover, the algorithm brings the possibility to consider the hybrid motion capabilities to optimise metrics in the generated trajectories like operation time or energy consumption. This is a remarkable feature since it allows better adaptation of the plans to specific missions. For instance, urgent inspections motivated by emergency situations would benefit from optimising the time operation, whereas routine inspections could better fit with energy optimisation in order to maximise the coverage. Additionally, an extension of the planning algorithm to guarantee safer trajectories based on Dynamics Awareness has also been derived. Finally, replanning capabilities of the planner endow HRs with hybrid reactivity; that is, the possibility to switch between pipe and air to avoid unexpected obstacles detected while rolling. Simulation and experimental results showcase the advantages of the proposed techniques.

Finally, with the focus on motion planning for multiARM systems, a joint collaboration with the Multi-Robot Systems Group (MRS) at the Czech Technical University in Prague (CTU) [5] extends the previous work [6] proposing a convenient motion planning method for the autonomous installation of bird diverters on power lines using a fleet of heterogeneous ARMs. However, the approach is flexible enough to be adapted to other applications like sensor deployments or load deliveries. The method considers the limited payload capacity of the ARMs and their dynamic constraints, as well as the existence of refilling stations where the robots can be supplied with more diverters once run out on board. The final results are optimal and feasible trajectories connecting the installation points and the refilling stations that follow efficient installation sequences while avoiding obstacles and keeping safety distances between the ARMs. Moreover, an adaptation of the general method is intended to improve the energy consumption. In this sense, the motion planner accounts for the ARM aerodynamics to compute trajectories whose velocities are prone to minimise the energy consumption of the robots without disregarding the rest of requirements. Finally, a replanning strategy is also proposed in case a particular ARM suffers a failure. A new trajectory allows a back-up ARM to continue with the remaining mission respecting the ongoing operation of the rest of the robots. Numerical results in MATLAB [7], realistic simulations in Gazebo [8] and real-world outdoor experiments reinforce the validity of the method.

Transversally to previous motion planning methods, each of them has been tested with particular ARMs that help to show its benefits. In this context, another complementary development is the derivation of dynamic models and control laws for such ARMs when they are required for Dynamics Awareness in the planning process.

1.2.2 Scientific publications

As a result of the research presented along this thesis, the following scientific dissertations were published:

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

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

- Caballero, A., Bejar, M., and Ollero, A. On the use of velocity adaptation to outperform the motion planning with dynamics awareness in aerial long-reach manipulators with two arms. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1125-1133). IEEE, 2018.

- Caballero, A., Suarez, A., Real, F., Vega, V. M., Bejar, M., Rodriguez-Castaño, A., and Ollero, A. First experimental results on motion planning for transportation in aerial long-reach manipulators with two arms. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 8471-8477). IEEE, 2018.

- Caballero, A., Bejar, M., Rodriguez-Castaño, A., and Ollero, A. Reactivity and dynamic obstacle avoidance. In *Aerial Robotic Manipulation* (pp. 333-348). Springer International Publishing, 2019.

- Suarez, A., Caballero, A., Garofano, A., Sanchez-Cuevas, P. J., Heredia, G., and Ollero, A. Aerial manipulator with rolling base for inspection of pipe arrays. *IEEE Access*, 8, 162516-162532. 2020.
- Caballero, A., Sanchez-Cuevas, P. J., Bejar, M., Heredia, G., Trujillo, M. A., and Ollero, A. An aerodynamic extension for motion planning with dynamics awareness in aerial long-reach manipulators. *International Journal of Aerospace Engineering*, 2020.
- Cacace, J., Orozco-Soto, S. M., Suarez, A., Caballero, A., Orsag, M., Bogdan, S., Vasiljevic, G., Ebeid, E., Acosta, J. A., and Ollero, A. Safe local aerial manipulation for the installation of devices on power lines: AERIAL-CORE first year results and designs. *Applied Sciences*, 11(13), 6220. 2021.

1.3 Framework

The research collected in this thesis has been developed in the GRVC Robotics Lab at the University of Seville in Spain [9], mainly within the framework of the European research and innovation projects AEROARMS [10], HYFLIERS [11] and AERIAL-CORE [12], all of them funded by the European Commission. Chronologically, AEROARMS proposed the development of the first aerial robotic system with multiple arms and advanced manipulation capabilities to be applied in industrial I&M. HYFLIERS aims to develop the first industrial integrated robot with hybrid air and ground mobility with a long-reach hyper-redundant manipulator capable of reaching sites in oil and gas refineries and chemical plants where no other robot can access. AERIAL-CORE has the main goal of developing an integrated aerial cognitive robotic system that will have unprecedented capabilities on the operational range and safety in the interaction with aerial co-workers and objects in the environment. The system will be validated in the I&M of large linear infrastructures and, particularly, in the application to electrical lines. More details about such projects can be found in their respective webpages. In these projects, there has been an active participation in the elaboration of deliverables as well as in the execution of live demonstrations in different countries, including experiments in real industrial sites. Some of them were disseminated in the Spanish media.

Also in the context of the project AERIAL-CORE, part of the research presented in this thesis was done during a three-month stay in the Multi-Robot Systems Group (MRS) at the Czech Technical University in Prague (CTU) [5], consolidating a joint collaboration with the institution.

Finally, it should be highlighted that part of the generated technology has been transferred to leading companies like NAVANTIA. In particular, the fundamentals of some advanced planning techniques developed in this thesis have been used as the basis for motion planners in real Unmanned Surface Vehicles (USV) operating on the sea. The details of this collaboration are confidential and out of the scope of this thesis.

1.4 Organisation of this thesis

This thesis is organised in seven chapters, being the introduction the first one concluding with this section. The rest of the chapters are:

- Chapter 2: State of the art focuses on reviewing the state of the art related to motion planning for aerial manipulation in inspection and maintenance.

- Chapter 3: Motion planning for aerial manipulators with robotic arms presents the motion planning method specially oriented to ARMs endowed with robotic arms. This chapter includes the fundamentals of the planner operation and its adaptation for in-flight planning capabilities. A wide set of validation tests is also added.

- Chapter 4: Extensions on motion planning for aerial manipulators introduces the advanced extensions of the motion planner in the previous chapter. These are the Dynamics Awareness for robust obstacle avoidance, the Velocity Adaptation mechanism for a better optimisation of the execution time of the planned trajectories, and the Aerodynamics Awareness (ADA) to discard states whose associated aerodynamic phenomena may provoke undesired collisions. The simulations that validate the extensions complement the chapter.

- Chapter 5: Hybrid motion planning for inspection with aerial-ground robots explains the motion planning method that takes advantage of hybrid locomotion in ARMs for autonomous inspection tasks. The extension based on Dynamics Awareness and the associated hybrid replanning have also been presented. Simulation and experimental results close the contribution.

- Chapter 6: Motion planning for fleets of aerial manipulators shows the motion planning method for the autonomous installation of bird diverters on power lines using a fleet of heterogeneous ARMs. The adaptation of the general method to improve the energy consumption and the replanning strategy in case a particular robot suffers a failure are also included. The chapter finishes with the planning validation through simulations and flight tests.

- Chapter 7: Conclusions and future work is devoted to the general conclusions of this thesis as well as the future work beyond the associated research.

Complementing the previous chapters, there are also three annexes:

- Annex A: Aerial robotic manipulators collects a description of the different ARMs that have been used to test the motion planning methods presented along this thesis. This annex includes dynamic models and control laws when they are required for Dynamics Awareness or simulation as well as software and hardware details for experimentation.

- Annex B: Characterisation of aerodynamic effects describes an experimental approach that allows characterising the aerodynamic phenomena that arise when a rotor operates close to a surface; these are, the ground effect, the ceiling effect and the wall effect. The resulting models are essential to formulate Aerodynamics Awareness in the planning process.

- Annex C: Fundamentals of signal temporal logic for motion planning summarises the fundamentals of STL (Signal Temporal Logic) for motion planning. The multiARM motion planner for the installation of bird diverters on power lines is built over them.

Fig. 1.1 depicts a general scheme that shows the interconnections among the different chapters and annexes that have been introduced before.

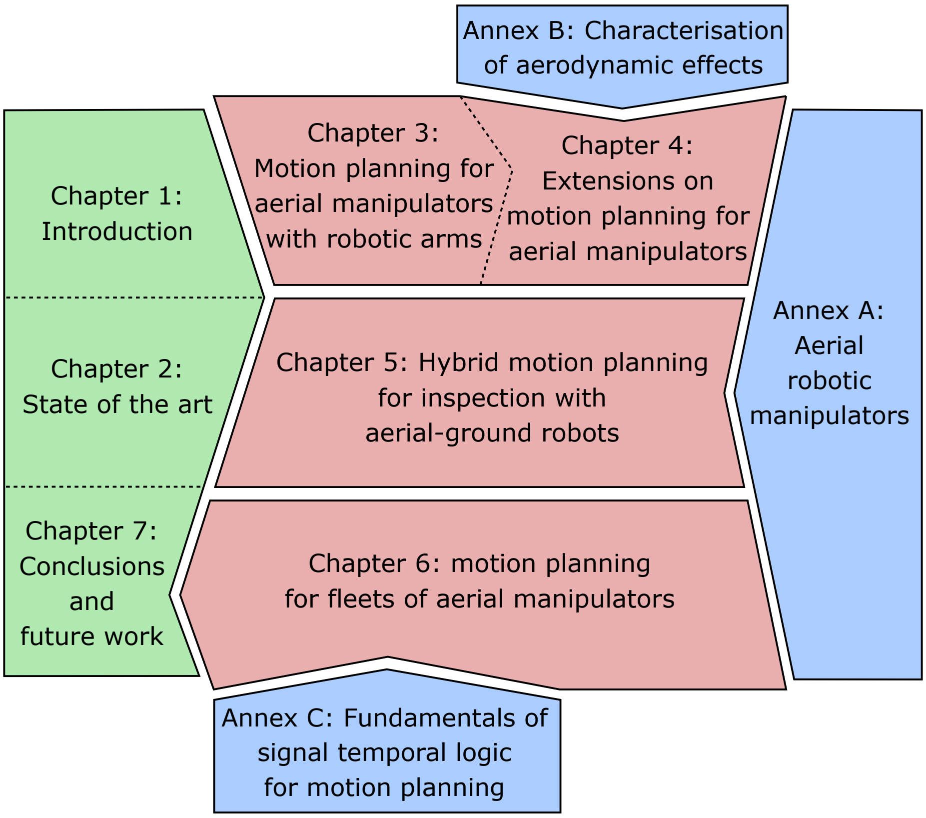

Figure 1.1: Organisation of this thesis. General scheme.

This thesis also provides a set of videos with simulations and experiments that help to illustrate the obtained results. This multimedia material, referenced along the document, is available at `https://hdvirtual.us.es/discovirt/index.php/s/8cnw6ePWGJCfsTD`. A digital copy of this present document in its latest version has also been included there.

Chapter 2

State of the art

> Créelo, créalo y vuela.
>
> Ana B. Caballero

This chapter presents a review of the state of the art related to motion planning for aerial robotic manipulation in inspection and maintenance. After analysing the trends in aerial robotic manipulation, this review puts the focus on the different topics that are addressed along this thesis in the framework of motion planning.

2.1 Aerial robotic manipulation

In the last ten years, there has been a growing interest in the application of robots for aerial manipulation in Inspection and Maintenance (I&M) tasks that has given place to a good number of publications. Highlighting some of the most relevant, work [13] presents a novel aerial robotic manipulator that provides physical-contact inspection on surfaces at any orientation making use of a specially-designed manipulator. The system, intended to perform non-destructive tests in the oil and gas industry, can keep its relative position with respect to the inspected surface without using GNSS (Global Navigation Satellite System) measurements and allows semi-autonomous operation. Among contributions without outdoor experiments but with promising results indoors, two articles can be pointed out. In the first one [14], the design, motion planning, and control of an aerial manipulator and its application to the inspection by contact of metal pipes are presented. The system is composed of a multidirectional-thrust aerial platform and a 2-DoF (Degrees of Freedom) lightweight arm which integrates an Eddy-current probe as end effector. The benefits of this approach are demonstrated with the scan of a pipe surface by sliding the Eddy current probe while keeping the contact. In the second one [15], another aerial vehicle integrating a lightweight delta manipulator and an extrusion system for aerial repair is introduced. This robot takes advantage of a kinematics-based approach and visual-inertial odometry-aided navigation for robust operation in hovering and windy conditions. Experimental results show that the aerial manipulator can be suitable for the sealing of cracks and holes on industrial oil pipes. More focused on civil applications, the design, modelling, and control of a multirotor for safe inspection of bridges with full contact are presented in [16]. For this application, the robot uses the aerodynamic ceiling effect to improve the accuracy in the operation. The work includes flight tests in a real bridge where the system completes autonomously a structural assessment of it making use of a robotic total station for both positioning and measurement acquisition. Alternatively, in other valuable contributions [17, 18], a multirotor with a robotic arm attached to its top part is developed also for inspection tasks on bridge ceilings.

Regarding aerial robotic manipulation in a wider sense but with potential applications in I&M, there are also many research works. [19] presents the design of several light-weight, low-complexity grippers that allow quadrotors to grasp and perch on branches or beams and pick up and transport payloads. The publication [20] focuses on the design of a feedback control strategy for aerial manipulators that allows both position and orientation tracking for the end effector. Furthermore, [21] proposes a small-sized multirotor endowed with a simple specially-designed manipulator for the installation and retrieval of sensors with an exchange of forces with the environment. After dealing with the design, modelling and control, the system is tested in multiple flight experiments, both indoors and outdoors, including interactions with different kinds of surfaces. In a bigger size, contribution [4] presents a helicopter integrating a 7-DoF industrial manipulator. The complex system, motivated by practical applications identified in the industry, requires the use of a control technique that deals with the coupling between the aerial platform and the robotic arm. Additionally, a vision system is added for localisation and tracking of the manipulated target. The robot performance is successfully evaluated in outdoor flights. Another interesting research topic is the usage of cables for aerial manipulation. Along with this line, [22] suggests the use of an aerial carrier with a cable-suspended manipulation system consisting of an active platform and a KUKA LWR 4 manipulator. This novel configuration, suitable for I&M tasks, allows a reduction in collision risks with the rotor blades because the aerial platform does not operate near the manipulation area.

Among the different contributions focused on aerial manipulation, not many of them consider configurations with more than one arm. The need to employ several arms can be justified in special tasks such as transportation of long elements (to avoid swinging movements), application of torques or execution of different tasks simultaneously. Thus, papers [3, 23] propose a dual-arm aerial manipulator for valve turning. The robot integrates a control strategy that considers the coupling between the system and the valve, as well as a detection algorithm to assist the operation. The approach is validated through flight tests. For a more general applicability, works [2, 24] focus on the design and construction of a human-size and lightweight dual-arm manipulator with compliant joints and its integration with a multirotor platform. The custom design of the manipulator demonstrates to be suitable for physical interactions with the environment. Moreover, the aerial manipulator also integrates a visual servoing method for object grasping. The complete system is tested outdoors. In the same research line, paper [25] makes use again of a lightweight and compliant dual arm attached to a multirotor platform. However, in this case, the work is oriented to perform aerial manipulation with one arm in grabbing condition. This approach allows an increment in the positioning accuracy of the operation arm thanks to the sensors of the grabbing arm, which are used to estimate the position of the robot with respect to the grabbing point. The strategy is validated in an indoor testbed. Concerning theoretical contributions, [26] introduces a generic planar aerial manipulator with any number of arms attached at the centre of mass of a UAV. The authors prove that this kind of systems are differentially flat regardless the number of joints of each arm and their kinematic and dynamic parameters. This theory is validated by simulating object grasping and transportation tasks.

Due to their versatility, the design and development of new concepts of Hybrid Robots (HRs) is arousing much interest. Thus, there are a lot of research lines pointing to the synergistic use of UAVs and water [27, 28] or ground locomotion based on wheels [29], rolling cages [30, 31] and other crawling mechanisms [32, 33]. However, most of these HRs were presented to demonstrate their extended capabilities but without being applied to any particular mission. In contrast, [34] introduces a flying HR with wall-climbing mobility intended to inspect visually vertical surfaces like facades in urban spaces. In the context of industrial I&M, the European project HYFLIERS [11] proposes the development of the world's first industrial integrated robot with hybrid air and ground mobility endowed with a long-reach hyper-redundant manipulator capable of reaching sites where no other robot can access. Associated with this project, publication [35] presents a HR that makes use of a 6-DoF robotic arm for the inspection of long arrays of pipes in the oil and gas industry. For the navigation, the robot with flying locomotion is capable of landing and rolling along the arrays of pipes without wasting energy in the propellers and increasing the accuracy during the manipulation. With a similar purpose, [36] introduces a modular HR with interchangeable rolling bases for its locomotion in different pipe configurations. Moreover, the robot design is compatible with the operation in potentially-explosive atmospheres.

Turning to manipulation tasks on electric power lines, the interest in the application of ARMs to install bird flight diverters on them has been growing recently. Some commercial solutions like [37, 38] try to keep human workers away from the line, increasing substantially the safety in the operation and reducing costs. From a research perspective, some publications can also be found in the framework of the AERIAL-CORE project [12, 39]. In this sense, [40] and [41] propose a multirotor endowed with a linear actuator and a clamp mechanism to install clip-type bird diverters exerting high forces in flight (see Annex A.5). Not bird diverters but sensors with similar deployment, [42] presents a UAV that can place them autonomously on a power line. Focusing on the control, [43] presents a simulator that is suitable to develop and test control strategies for autonomous bird diverter installation tasks.

2.2 Motion planning for aerial robotic manipulation

With the focus on motion planning for ARMs, the number of relevant publications has grown substantially in the last few years. At the beginning, the existing contributions like [44] usually assumed a strong simplification by addressing the planning problem in a decoupled way; this is, adopting independent planners for the UAV and the manipulator that switched their operation according to the mission phase. This means that during the navigation phase, the arm configuration was supposed to be fixed and hence the UAV planner was in charge of planning the motion. In contrast, the manipulation phase was resolved by using the manipulator planners and supposing that the aerial platform was not moving. However, the evolution rapidly led to the joint consideration of both subsystems within the planning process. [45] proposed an optimal planning strategy for a quadrotor with two arms that minimises the interaction between the aerial platform and the arms. Later, [46] presented a motion planner to pick and place objects using ARMs. Assuming that the position of an object is given by a waypoint, the technique, based on bidirectional RRT* (optimal version of the Rapidly Exploring Random Tree RRT, [47]), computes optimal trajectories that accomplish waypoint constraints with only partial specifications. Furthermore, recent contributions also include planning methods for cooperative ARMs in tasks like transportation of heavy or large objects [48, 49]. On the usage of cables for aerial manipulation, [50] studies a motion planning approach for reliable 6-dimensional quasi-static manipulation with an aerial towed-cable system. The system consists of a platform attached to three flying robots through two cables each one. Not for aerial but spatial manipulation, paper [51] presents an online path planner for a spacecraft robot with a manipulator to capture moving objects. The planner allows the end-effector approximation to the grasping point thanks to the coordinated movement of the spacecraft base and the robotic arm. Moreover, the trajectory is modified in real time using updated states of the target. This desirable feature is possible due to the simplicity in the derivation of the planner without a complex numerical calculation. Moreover, the technique does not consider obstacles within the operation area, which also simplifies the problem. More contributions where the motion planning of the vehicle and the manipulator is addressed in an integrated way can be found for manipulators mounted on mobile ground bases. As an example, [52] deals with this problem on flexible mobile manipulators.

Paper [53] introduces the motion planning of a transformable multilinked flying robot and its application to pass through a small opening. Although the work does not involve manipulation, it is useful in order to highlight that, when planning trajectories in difficult scenarios for robots with a high number of DoF, as in the case of aerial manipulators, the computation time of the planned trajectory (around 10 hours for the proposed problem) may complicate the online use of the planner. This can be required for navigation purposes when there is not available any prior map of the environment that allows planning a trajectory offline. To reduce the computation time, planning methods based on motion primitives are mainly exploited [54, 55]. In this manner, article [54] provides an interesting approach oriented to online motion planning for unmanned aerial vehicles in cluttered environments. The method avoids the online search to reduce the computational complexity and relies on determining the next navigation step that maximises the likelihood to reach the goal through the use of a trajectory library. Further, the method also uses the position of the obstacles detected onboard by a laser scanner. As a conclusion, a path can be found in less than 1 ms. The approach is tested with a multirotor in a dense forest.

2.3 Extensions on motion planning: dynamics and aerodynamics awareness

Taking into account that ARMs usually operate in cluttered areas or physically interacting with the environment, the system dynamics may play an important role in the planning problem. Concerning motion planning where the robot dynamics can not be neglected, there are important contributions aiming at finding admissible, collision-free and accurate trajectories. [56] present an exhaustive review in this respect. In a relevant publication, [57] proposes kinodynamic motion planners that consider kinematic and dynamic constraints during trajectory generation. However, these planners require a planning space twice the dimension of the configuration space (configuration variables together with their associated velocities), which implies high computational costs. Alternatively, [58, 59] propose other methods that split the motion planning problem into two stages. In the first stage a basic planner searches for a path compatible with a bounding sphere that replaces the system. In the second stage the path is transformed into a trajectory compatible with the kinodynamic constraints. A significant drawback of these methods comes from the approximation required in first stage since it could complicate the existence of a collision-free path. [60, 61] propose some enhancements to improve the weaknesses of previous approaches. On the first contribution, a steering method to compute physically-realistic local trajectories of quadrotors is presented. This method, which is computationally-efficient, connects kinodynamic states using fourth-order splines. The second publication, built on the first, presents an accurate but computationally fast quasi-metric to determine the proximity of dynamic states of a quadrotor, and an incremental state-space sampling technique to avoid generating local trajectories that violate kinodynamic constraints. These contributions have been validated through simulation. In aerial manipulation, there are also motion planners that integrate some kind of dynamics awareness during the planning process for ARMs [62, 63]. In the case of [62], the technique consists in a sampling-based control-aware planner for task-constrained motions. As it is stated by the authors and demonstrated through simulation, the method lets to sample in the task space, predict the behaviour of the controlled ARM, and ensure a proper execution of the trajectory.

Another aspect that should not be neglected when planning the motion of an aerial system is the hazardous influence of the aerodynamic effects associated with the operation of rotors in cluttered environments. Indeed, 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, may be dangerous for standard multirotors provoking their destabilisation or even crashes. To the author's knowledge, there is not any precedent contribution in the literature that addresses these undesired effects of the aerodynamic phenomena in the planning process. Instead, there are some contributions that deal with the aerodynamic phenomena at the control level. In this way, [64] proposes an enhanced control scheme for quadrotors with the purpose of improving the landing under the presence of the ground effect. Moreover, [65] analyses the influence of the ground effect in several control approaches for multirotors. Additionally, [66] proposes a new UAV configuration that takes advantage of the ceiling effect. The latter allows precise contact inspection on bridges to be carried out.

2.4 Motion planning for hybrid robots

The use of HRs brings also new challenges for motion planning. When considering the joint exploitation of different locomotion capabilities, a common approach consists of planning the motion of Unmanned Ground Vehicles (UGVs) using a map provided by a UAV [67, 68]. In this case, the heterogeneous fleet of robots works together but the planning problem is usually concentrated on the UGVs after assuming that they are more exposed to potential obstacles. In contrast, there are also some papers focused on integrated planning methods for heterogeneous teams of UAVs/UGVs operating as a whole in applications such as planetary exploration [69] or aerial manipulation [70, 71, 72].

However, very few publications about motion planning for HRs can be found in the literature, with most of them concentrated on the last years. Among them, [73] highlights for the description of a unified framework for control, planning and autonomy of hybrid ground/air vehicles. In particular, a unified planner is presented for both rolling and flying by leveraging differential flatness mapping. Experimental results in unknown environments endorse the validity of the method. Outperforming the results in previous contribution, [74] has recently developed an alternative adaptive navigation framework to bring complete autonomy to HRs. The approach includes a hierarchical motion planner that generates safe and energy-efficient terrestrial-aerial trajectories even in unknown unstructured dense environments. Extensive tests demonstrate the benefits. Concerning HRs in I&M tasks, [39] outlines a graph-based planner for the efficient installation on power lines of devices like bird flight diverters or electrical spacers using HRs.

2.5 Multirobot motion planning: installation of bird diverters

The motion planning associated with bird diverter installation tasks on power lines has many requirements in terms of collision avoidance, efficiency in the total operation time, dynamic feasibility, limited payload capacity, multiARM operation and other high-level requirements. The achievement of all of them is not straightforward as it requires the common functionalities of different kinds of motion planners into a single one.

Point-to-point motion planners like sampling-based planners have demonstrated to be a good solution in the search of trajectories for challenging environments. In particular, the family of RRT algorithms [75] can compute collision-free trajectories in cluttered environments for high-dimensional robots in a reasonable time. Additionally, these algorithms can also optimise some particular metrics like the distance, the operation time or the energy consumption [47], or even consider the system dynamics in the planning process [57]. Alternatively, when fast trajectories are needed, planning methods based on motion primitives are mainly exploited as it was commented in Section 2.2 [54, 55]. However, all the previous algorithms assume that the start and goal points are known, therefore when the full mission demands visiting several points, the sequence must be provided as an input.

In contrast, Vehicle Routing Problems (VRPs) are usually the first choice in missions where the sequence computation is not trivial and may condition significantly the efficiency of the solution. These algorithms offer a wide range of variations that allow the consideration of important features like limited payload capacity [76] or heterogeneous multivehicle capabilities [77], among many others. Since VRPs are suitable for planning in scenarios that can be modelled as a graph, most of the state-of-the-art work in planning for power lines is addressed as variations of the VRP [39, 77, 78]. However, VRP algorithms also present some inconveniences. Firstly, they do not compute smooth trajectories, needed to avoid obstacles, but only sequences. Secondly, they can hardly consider the obstacles in the environment. Thus, optimal sequences computed without obstacles could be suboptimal with them. Thirdly, VRPs are NP-hard (Non-deterministic Polynomial-time hard) combinatorial problems whose complexity increases quickly with the number of variables. Consequently, they usually need to be solved using heuristic approaches that can not ensure the search of the optimal solution.

MultiUAV systems offer some unique advantages since they allow a significant reduction in the mission time by dividing the full operation between several vehicles. Furthermore, the robustness is increased because the mission can continue in case a particular robot suffers a failure. Concerning multiUAV motion planning, there are many strategies in the literature ranging from decomposition graph-based methods to bioinspired methods [79]. In the category of planning through optimal control, Model Predictive Control (MPC) has become popular. These methods tend to have a high computational burden but [80] presents a distributed MPC that is able to generate in real time point-to-point trajectories with a reduced flight time. Other publications like [81], based on neuronal networks, allow simultaneously target assignment and path planning. Although these kinds of techniques can compute safe trajectories for hundreds of UAVs efficiently [82], they may struggle to deal with high-level specifications like "one UAV must visit region A and B in the time interval $[t_1, t_2]$ while another UAV must reach region C and stay there for t_3 seconds, both fulfilling safety requirements".

In an attempt to extend the complexity of the specifications that can be managed in motion planning, several publications can be found in the literature. [83] proposes a relaxation in the complexity by distributing it into local specifications that are assigned to particular robots. In this method, the motion capabilities of each robot are represented using a transition system whose modelling may become hard or computationally expensive. In contrast, Control Barrier Functions (CBFs) [84] may be suitable in the efficient computation of robust trajectories. In spite of this, they do not fulfil soundness and completeness for the full syntax of Temporal Logics (TLs), restricting the application in complex scenarios. At the same time, [85] presents robust algorithms for multirobot coordination that encode high-level TL specifications into a Mixed-Integer Linear Programming (MILP) problem. However, the local planner used in this method relies on a sequential multiagent RRT algorithm that leads to suboptimal trajectories. With a practical application to the visual inspection of power-line towers using only a single UAV, [86] formulates an optimisation problem that considers flight time, image quality, and tower coverage. The problem is solved using particle swarm optimisation and simulated annealing, achieving a good balance over the three performance ratios.

2.6 Conclusions

The review of the state of the art has shown a clear trend during the last decade in the development of ARMs for I&M activities or with potential application to them. In order to address these activities, many ARM configurations have been proposed, ranging from aerial robots endowed with one ore more robotic manipulators to HRs with flying and rolling locomotion. Additionally, some of these ARMs have been specially designed for particular tasks where they can provide significant advantages. The inspection of pipe arrays in industrial environments and the installation of bird flight diverters on electric power lines are two examples.

Motivated by the previous evidence, a parallel evolution has also been experienced in motion planning for aerial robotic manipulation. Thus, initial contributions considered strong simplifications but they rapidly led to more elaborated approaches. In this sense, some of those motion planners started to include advanced functionalities such as cooperation between ARMs or kinodynamic planning. However, the research community still has a long way to go. Highlighting important needs, there is a lack of motion planners for the safe and efficient operation of ARMs in cluttered environments like industrial sites, the contributions about motion planning for HRs are scarce and multirobot planning for aerial manipulation still admits relevant improvements.

Chapter 3

Motion planning for aerial manipulators with robotic arms

> Siguiendo el camino de la perseverancia cualquier destino es alcanzable.
>
> Mario B. de la Rosa

3.1 Motivation

As it was pointed out in Section 1.1, the application of ARMs for manipulation tasks in I&M brings new challenges. These robots frequently operate in industrial environments with a high density of obstacles likes pipes or other elements that might hinder the operation substantially. Such cluttered scenarios force ARMs to fly close to obstacles, which lead to an increment in the risk of collision. Accordingly, new planning algorithms should be developed to operate autonomously ARMs endowed with robotic arms while considering both the UAV and the manipulator in an integrated way. Focusing on safety and efficiency, more system states can be accounted when the aerial manipulator is considered as a whole. Thus, there is a higher control of system configurations in strategies for collision avoidance and optimisation. As a result, safer trajectories that also minimise the total energy consumption can be computed to avoid obstacles and waste of battery. Moreover, these planners should be able to adapt the robot motion to different operation phases. In this sense, the motion planners should not focus only on the manipulation phase, where the manipulator is expected to operate mainly with the support of small movements on the aerial platform, but also on the navigation phase towards the manipulation area, where the main movements may correspond to the aerial platform.

The complex topology of industrial scenarios as well as the frequent modifications of their layouts in maintenance operations constrain the availability of accurate maps of the operation area. To solve this, ARM onboard sensors need to explore the operation area while flying in real I&M applications. The resulting information must be transformed into the standard map template required by the planning algorithms. Additionally, in the search of a fully-integrated approach where the robot can complete all its operation in a single flight, which reduces the total operation time, the motion planners have to operate in flight just after the exploration performed by the onboard sensors. For this reason, although high computational requirements are expected for all the functionalities above, the planning algorithms must be efficient computationally to minimise the response time of the robot during the mission execution.

3.2 Motion planner

With the purpose of endowing ARMs with better performances and higher levels of autonomy, giving response to all the open issues mentioned in previous section, a new motion planning method specially oriented to ARMs endowed with robotic arms has been developed. This section includes the fundamentals of the planner operation (Section 3.2.1) and its adaptation for in-flight planning capabilities (Section 3.2.2).

The presented planning approach is based on general design principles and hence can be applied to different kinds of aerial manipulators. One possible configuration, the first version of a set of new Aerial Robotic Systems for Long-Reach Manipulation, called ARS-LRM_v1 and presented in Annex A.1, has been used in this section to guide the presentation.

3.2.1 Fundamentals of planner operation

According to [56, 87], sampling-based planners like the family of RRT algorithms [75] have demonstrated high potential in the search of fast solutions for high-dimensional robots. Furthermore, some of these methods bring the possibility to generate motion plans that optimise certain cost functions, as for the case of RRT* variations [47]. This makes it possible to find an optimal solution in terms of a specific metric. Taking all these considerations into account, together with the limited manoeuvrability which is usual in ARMs, an RRT*-based algorithm that optimises the energy and the execution time of the planned motion has been developed.

An important aspect for the planner performance is the planning space considered when exploring the different possibilities of motion. As a general concept, ARMs are robots composed of two subsystems that work as a whole; these are, the aerial platform and the manipulator. Consequently, the proposed method adopts an integrated approach where the configuration variables of both subsystems are considered jointly within the planning space. This integrated strategy allows the consideration of a more complete set of system states. In this way, it is possible to achieve wider and safer operating conditions since equivalent configurations in terms of final effector positions can be differentiated according to the positions of both the multirotor and the manipulator intermediate links. For the particular case of the ARS-LRM_v1 system, the presented planner explores the configuration variables of the aerial platform and the dual arm, which correspond to the green variables in Fig. A.3. It should be mentioned that the pitch angle q_5 of the aerial platform is considered negligible for planning purposes. The complex dynamics of ARMs makes more suitable to operate following smooth movements, which are associated with small values of q_5.

The pseudocode of the planning algorithm that has been developed is shown in Algorithm 3.1. It mainly corresponds to the common structure of the RRT* algorithm [47] but most of the intermediate functionalities have been customised, giving rise to the MP-ARM (Motion Planner for Aerial Robotic Manipulators) algorithm. These particular developments have been dealt with in detail hereinafter.

Algorithm 3.1 Motion Planner for Aerial Robotic Manipulators (MP-ARM).

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

Discretisation of the planning space

Due to the high dimension of the planning space integrating both UAV and manipulator states that is considered by the MP-ARM planner, a continuous treatment of the variable ranges considered in the sampling operation would require an excessively high execution time to converge to a solution. The former suggests the adoption of discretisation patterns that bound the execution time for the planner. However, this discretisation must be addressed carefully since an excessively reduced set of data might endanger the algorithm convergence. Hence, there must be a trade-off between the computational gain and the convergence properties that should be determined for each application scenario.

Computation of the nearest node

The $NEAREST(Tree, x_{rand})$ function finds in the tree nodes the nearest node $x_{nearest}$ to the random state x_{rand} generated in the sampling-based exploration of the planning space. Since the nodes include state information for both multirotor and manipulator accordingly with the integrated operation basis of the planner, there will be two different measurements to calculate the nearest node; these are, the difference in position for the multirotor and the difference in angle for the manipulator joints. Thus, it appears the need to define a homogenising metric. The reference velocities u_{ref} for the UAV and w_{ref} for the joints have been introduced to transform the heterogeneous measurements into a common metric. This metric is given by the time magnitude required for each system component (t_{UAV}, t_{MAN}) to move between the configurations associated with the nodes under analysis. The equations corresponding to this normalisation approach are presented for the ARS-LRM_v1 system as follows:

$$\begin{aligned} t_{UAV} &= \frac{\sqrt{(\Delta q_1)^2 + (\Delta q_3)^2}}{u_{ref}} \\ t_{MAN} &= \frac{\max\left(\left|\Delta q_7^R\right|, \left|\Delta q_8^R\right|, \left|\Delta q_7^L\right|, \left|\Delta q_8^L\right|\right)}{w_{ref}} \\ \\ x_{nearest} &= \min_{x \in Tree} \left(\max\left(t_{UAV}|_x, t_{MAN}|_x\right)\right) \end{aligned} \tag{3.1}$$

where Δq_i denotes the increment in variable q_i when going from the tree node x to the sampled node x_{rand}; that is, $\Delta q_i = q_i^{rand} - q_i^x$.

Steering

The $STEER(x_{nearest}, x_{rand})$ function guarantees that new nodes x_{new} are close to those that already exist in the tree. For that purpose, the advancing segment from the nearest node $x_{nearest}$ to the random state x_{rand} is limited to maximum values for both the position of the aerial platform and the orientation of each link in the manipulator.

Collision checking

The $COLLISION(x_{nearest}, x_{new}, map)$ function checks if the branch that would link two nodes produces any collision with the obstacles included in the map. To this end, a representative set of intermediate configurations between the nodes is generated using interpolation. Then, each intermediate configuration is analysed to see if any part of the ARM collides with the obstacles defined in the scenario.

This operation deserves special attention since it plays an important role in the advanced functionality of the MP-ARM planner that allows differentiating equivalent configurations in terms of final effector positions according to the positions of both the multirotor and the manipulator intermediate links. The consideration of the different geometries of the system components, together with the joint exploration of the planning space, are crucial features in this respect. Concerning the former, the consideration of the system geometry with high accuracy when implementing methods for collision checking leads to better results but the computation time increases considerably. On the other hand, standard algorithms usually approximate the geometry of the vehicle by a simple bounding volume. This speeds up the process but implies a degradation in the performance of the algorithm. In view of this, an intermediate approach is the desirable option, where simplified models alleviate the computational burden of collision checking but maintain at the same time the capability to express the heterogeneity existing in the geometry of the different parts. To this end, the geometry of the aerial platform is approximated by its bounding box while the manipulator is modelled like rectilinear bars with negligible section. Thus, the collision checking operation fits better with the characteristics of the system than considering other general-purpose approximations like circle or rectangular bounds associated with the full ARM system. This granularity eases the planner convergence to feasible trajectories in cluttered environments and the associated increment in the computational load is not significant. Regarding the obstacles, all of them have been considered round or rectangular. In this way, it is possible to approximate complex-shaped obstacles with simple shapes that also reduce the complexity of the collision checking algorithm.

Another aspect that requires further consideration is the algorithm that analyses potential collisions between the approximated geometries of the robot and the obstacles. This is divided into two processes that optimise the computational efficiency for each ARM component. In the case of the multirotor, the approach is straightforward since it only requires checking whether the position of its centre of mass is within the limits of the rectangular region that produces collisions with the obstacle (see Fig. 3.1).

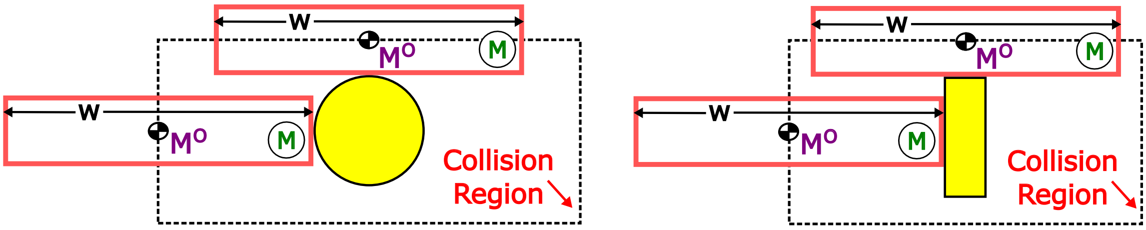

Figure 3.1: Collision checking for the multirotor based on the position of its centre of mass M^O. Round obstacle (left) and rectangular obstacle (right).

In contrast, the collision management for the manipulator admits several approaches. Although it would be possible to follow the standard procedure of generating intermediate configurations between the initial and final positions of the manipulator links, and then proceed to check collisions for a set of points sufficiently dense to represent each configuration, a better approach has been derived for the MP-ARM planner. It mainly consists in translating the collision condition into the angular space as shown in Fig. 3.2 for the ARS-LRM_v1 system. In this way, the obstacles are characterised in terms of the minimum and maximum link angle that produces a collision. Then, taking into account also the distance to the obstacle, it is possible to check the collision with a considerably reduction in the computational load. Fig. 3.3 shows two examples of this angular approach for the ARS-LRM_v1 system. The collision management consists in checking if a point representing certain configuration in the angular space falls inside the collision regions defined through the minimum-maximum angular characterisation of the obstacles. It should be noted that the shape of such regions varies with the relative position between the multirotor and the obstacle since the distance to the obstacle employed for the collision checking is varying accordingly.

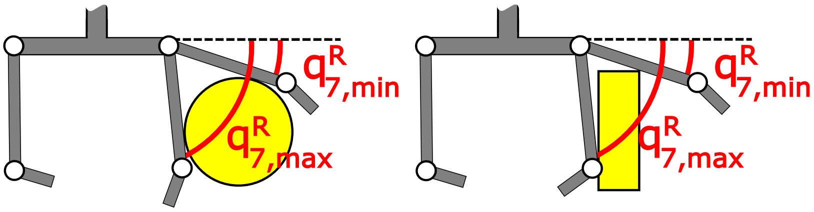

Figure 3.2: Collision checking for the right upper link of the ARS-LRM_v1 system. Round obstacle (left) and rectangular obstacle (right).

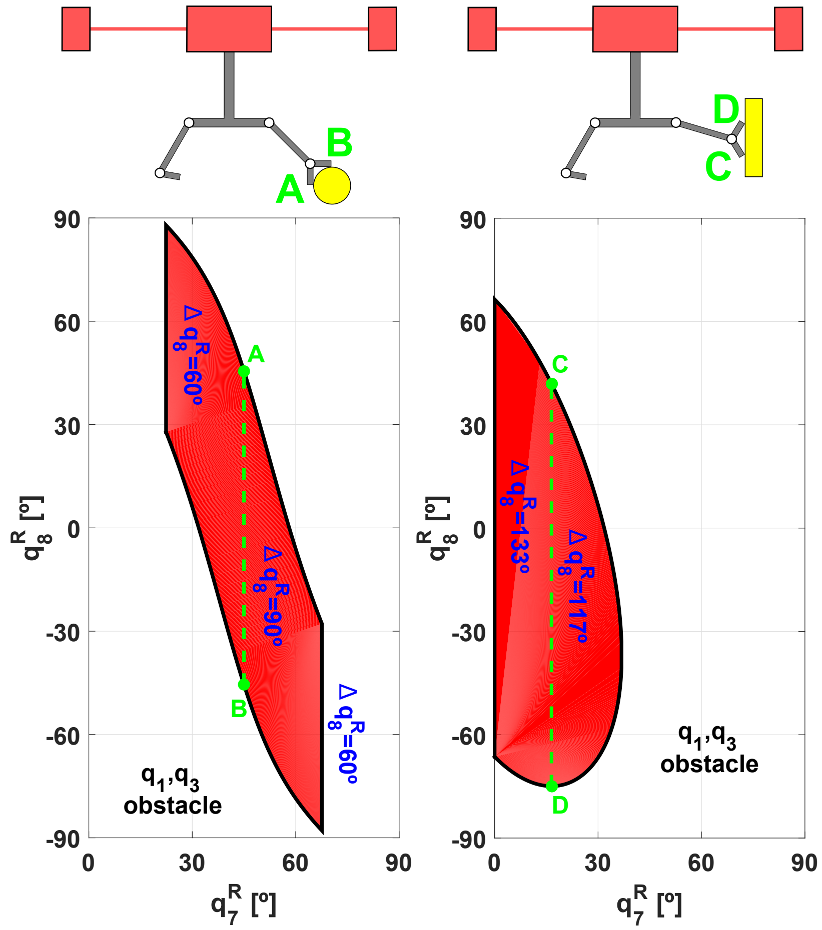

Figure 3.3: Regions in which the right arm of the ARS-LRM_v1 system collides with an obstacle for fixed UAV and obstacle positions. Round obstacle (left) and rectangular obstacle (right). The shape of the collision regions varies with the relative position between the multirotor and the obstacle.

Pre-computation of collisions

In order to further reduce the computation time devoted to collision management, which is the planner process demanding more computational burden, a pre-computation of collision conditions can be implemented. It should be noted that this approach is only valid for environments with obstacles whose locations are known prior to the ARM flight. Under this assumption, a representative set of ARM states in the application scenario is selected to implement the collision pre-computation. The results are stored in a binary matrix that will allow checking collisions in two simple steps. Firstly, the most similar element within the aforementioned set of representative states to the configuration under collision checking is searched for. Secondly, its corresponding value in the logic-values matrix is returned as the result for the collision checking operation. In the first step, a proper selection of the state granularity is essential to avoid false conclusions on the configuration under analysis. Taking into account that this pre-computation will be executed only once for each environment, a dense discretisation pattern will be used to achieve reliable results.

Computation of the set of near nodes

The $NEAR(Tree, x_{new})$ function finds the set of tree nodes x_{near} that satisfy simultaneously the following conditions with respect to their distances to the new candidate node x_{new}: the difference in multirotor position is less than threshold γ_{UAV} and the differences in link orientations are all less than threshold γ_{MAN}. This definition can be expressed mathematically for the ARS-LRM_v1 system as follows:

$$\begin{aligned}
\rho_{UAV} &= \sqrt{(\Delta q_1)^2 + (\Delta q_3)^2} \\
\rho_{MAN} &= \max\left(\left|\Delta q_7^R\right|, \left|\Delta q_8^R\right|, \left|\Delta q_7^L\right|, \left|\Delta q_8^L\right|\right)
\end{aligned}$$

$$x_{near} = x \in Tree \ / \ \left\{ \begin{array}{l} \rho_{UAV}|_x \leq \gamma_{UAV} \\ \rho_{MAN}|_x \leq \gamma_{MAN} \end{array} \right. \tag{3.2}$$

where Δq_i denotes the increment in variable q_i when going from the tree node x to the new candidate node x_{new}; that is, $\Delta q_i = q_i^{new} - q_i^x$.

Cost functions

In order to apply the RRT* optimisation sequence within the $ADD(x_{nearest}, x_{near}, x_{new})$ and $REWIRE(x_{near}, x_{new})$ functions, two different cost indices have been defined: the operation time of the complete ARM (CF_T), and the energy measurement given by the linear and angular displacements produced in the multirotor and the manipulator joints respectively (CF_E). For the ARS-LRM_v1 system, these cost indices can be formulated as follows:

$$\begin{aligned} CF_T &= \max\left(t_{UAV}, t_{MAN}\right) \\ CF_E &= p_1 \rho_{UAV} + p_2 \sigma_{MAN} \end{aligned} \tag{3.3}$$

where t_{UAV} and t_{MAN} were defined in equations (3.1), ρ_{UAV} was defined in equations (3.2), $\sigma_{MAN} = \left|\Delta q_7^R\right| + \left|\Delta q_8^R\right| + \left|\Delta q_7^L\right| + \left|\Delta q_8^L\right|$ with Δq_i denoting the increment in variable q_i between the nodes in which the cost function is being evaluated ($\Delta q_i = q_i^{to} - q_i^{from}$), and $p_{1,2}$ are two weighting parameters that allow the prioritisation of movements with minimum displacements in the multirotor or in the manipulator. Thus, $p_1 \gg p_2$ prioritises trajectories in which the UAV displacement is minimum while $p_2 \gg p_1$ prioritises the opposite. Fig. 3.4 illustrates the effect of these weighting parameters.

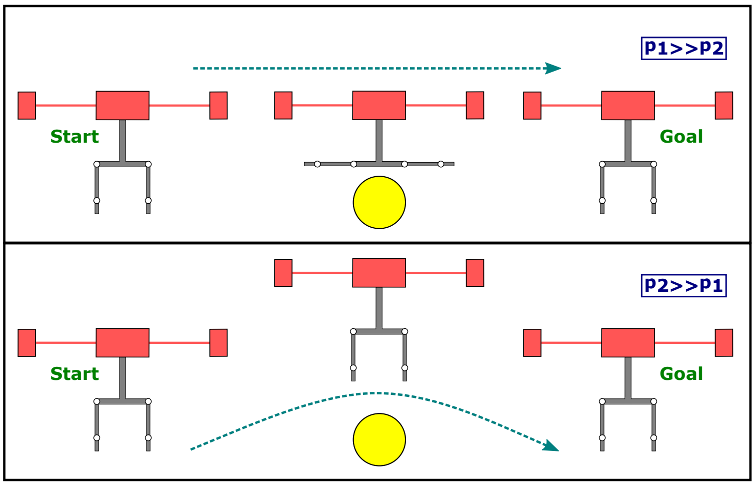

Figure 3.4: Effect of the weighting parameters $p_{1,2}$ in the cost function CF_E.

3.2.2 In-flight planning capabilities

The MP-ARM motion planner can guarantee efficient obstacle-free trajectories that are suitable for the autonomous operation of ARMs in cluttered environments. However, some additional requirements are needed to endow such method with in-flight planning capabilities for real industrial scenarios. According to Section 3.1, these requirements can be summarised as follows:

1. In-flight mapping in uncertain environments.
2. Computational efficiency in the trajectory generation process.

The design decisions that meet previous requirements are presented hereunder. They constitute an adaptation of the MP-ARM algorithm presented in Section 3.2.1. Concerning the first requirement, it is fulfilled in two stages. Firstly, the Multisensor Mapping module mentioned in Annex A.2.1 (Software integration) will provide accurate point-cloud-based information about the obstacles in the scenario. After that, a module of the motion planner will transform this information into a standardised map that supports the operation of the MP-ARM algorithm. With regard to the second requirement, both the algorithmic design and the implementation approach of the planner will aim for computational efficiency to enable in-flight planning. Further details about the implementation of these design guidelines are introduced below.

Integration of point-cloud representations into the planner

The first step of the motion planning process is the in-flight generation of a map of the scenario. This map will integrate the accurate point-cloud-based representation of the environment provided by the Multisensor Mapping module mentioned in Annex A.2.1 (Software integration). The resulting map will be a planar characterisation of the scenario since this will contribute to fulfil the requirement of computational efficiency. In complex manoeuvres, a decomposition into planar trajectory segments can be addressed. This planar approach will demonstrate experimentally later in Section 3.4 to be valid.

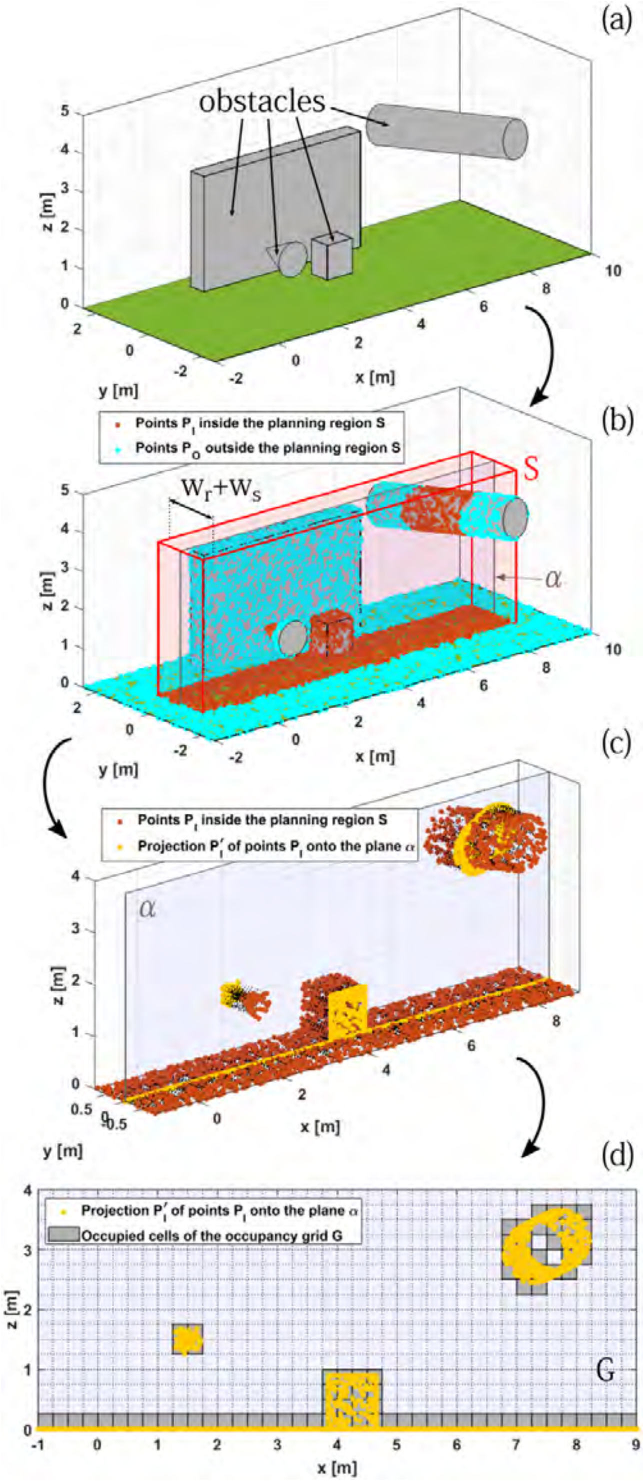

Figure 3.5: Integration process of the point-cloud representation into the motion planner.

The complete procedure to integrate in a map the point cloud delivered by the Multisensor Mapping architecture is illustrated in Fig. 3.5. The example scenario is presented in Fig. 3.5.a whereas the planning plane $\alpha \subset \mathbb{R}^2$ where the motion of the ARM centre of mass will be computed is defined in Fig. 3.5.b. The objective is to identify the subset of points $P_I' \in \alpha$ that might affect the system operation. To this end, the planning region $S \subset \mathbb{R}^3$ is defined with red lines. It is centred around the plane α and its width is equal to the robot width w_r plus a safety margin w_s. This region encloses the subset of points $P_I \in S$ (brown points in Fig. 3.5.b) that could provoke collisions when the ARM centre of mass is operating in α. The projection of this subset P_I onto the planning plane α gives place to the desired subset P_I' (yellow points in Fig. 3.5.c). Finally, the point-cloud information of subset P_I' is transformed in Fig. 3.5.d into a discrete occupancy grid $G \subset \alpha$ that allows reducing considerably the computational load associated with collision checking. The resolution of the discretisation should be a trade-off between computational efficiency and accuracy in the representation of the scenario. After this integration process, the resulting map is compatible with the collision checking approach presented in Section 3.2.1 (Collision checking) since the occupancy grid can be treated as a set of rectangular obstacles.

Computational efficiency

Since the motion planner is intended to work in flight, computational efficiency is a valuable feature to guarantee agile operation. This efficiency has been managed at two levels. The first level affects directly to the planning algorithm itself and is oriented to speed up both the search of the first solution and its convergence towards the optimal one. With this purpose, in addition to the approaches already presented in Section 3.2.1 like the use of discretisation patterns in the planning space or efficient collision checking, other implementations such as smoothing techniques and biased sampling that brings the exploration tree closer to the goal state have also been incorporated into the MP-ARM algorithm. The second level concerns the adoption of coding strategies that optimise the execution time. This level has also been supported through the usage of tools for automatic generation of efficient code.

3.3 Simulation results

In order to demonstrate the validity of the motion planning strategy presented in previous section, the fundamentals of the MP-ARM algorithm (Section 3.2.1) will be tested firstly using simulation in a realistic industrial scenario given by a riveting task.

Application scenario: riveting task

The schematic description of the scenario is shown in Fig. 3.6, where a planar characterisation is adopted for the sake of clarity. In this environment, coloured circles correspond to pipes existing in the industrial facility and surrounding circumferences denote the safety regions whose violation would be treated as a collision. As can be seen, the ARS-LRM_v1 system will be commanded to place two rivets with its right arm (target points marked in red) while the left arm provides visual feedback by pointing a visual camera integrated as end effector (see Fig. A.1). In this first proof of concept, the riveting operations will assume ideal conditions; this is, absence of interaction forces.

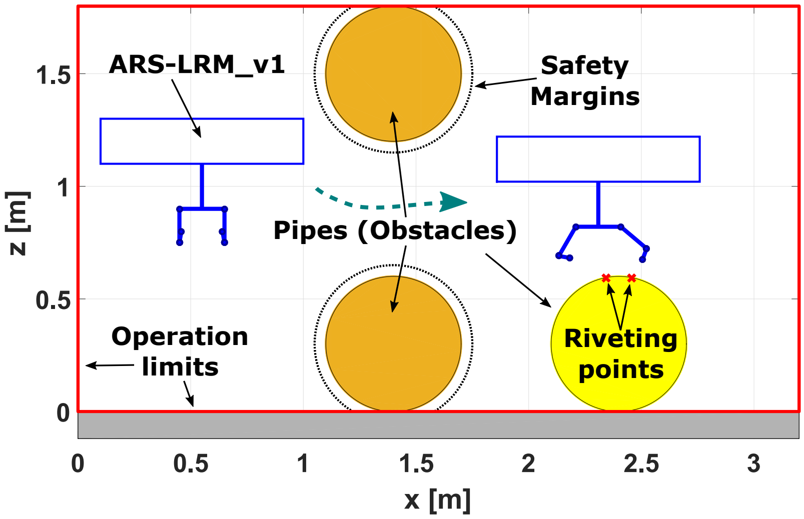

Figure 3.6: Application scenario given by a riveting task.

The achievement of the riveting objectives defined previously requires the execution of certain intermediate operations that include both navigation and manipulation manoeuvres:

1. Navigation phase: this phase corresponds to the robot displacement required to reach an observation position over the riveting area. After this, a short transition phase not requiring planner execution will enforce a ready-to-go configuration for the first riveting manoeuvre accomplished during the manipulation phase.

2. Manipulation phase: this phase covers the different manoeuvres involved in the manipulation task under consideration; this is, the riveting operation. In all the subphases described below it is assumed that the left arm will adapt its configuration to optimise the visual feedback provided by its integrated camera:

    (a) Rivet placement: approaching to the target point by the riveting effector integrated in the right arm.

    (b) Release: opposite manoeuvre to the rivet placement in which the riveting effector leaves the target point.

    (c) Switching: manoeuvre of the complete ARM to switch between the ready-to-go configurations for riveting points 1 (right) and 2 (left).

Analysis of the results

The MP-ARM algorithm has been used here to calculate a motion plan that commands the ARS-LRM_v1 system to complete the riveting task. The index selected for optimisation has been the cost function CF_E defined in Section 3.2.1 (Cost functions). Concerning the discretisation patterns of the planning space adopted for the navigation and manipulation phases, these are shown in Table 3.1. Furthermore, not only the planned trajectory (represented with dashed black lines in the figures) will be a matter of study but also the trajectory executed by the controlled ARS-LRM_v1 model (represented with blue lines in the figures) when receiving the former as control reference. The objective is the analysis of its closed-loop behaviour. For this, the model and controller presented in Annex A.1.2 have been used. This simulation

Table 3.1: Riveting task: discretisation of the planning-space.

Phase	Navigation	Manipulation
Variable	Discretisation pattern	
q_1	$[45\,,\,50\,,\,...\,,\,275]cm$	$[225\,,\,230\,,\,...\,,\,255]cm$
q_3	$[30\,,\,35\,,\,...\,,\,170]cm$	$[80\,,\,85\,,\,...\,,\,120]cm$
q_7^R	$[0\,,\,45\,,\,90]^o$	$[30\,,\,40\,,\,...\,,\,150]^o$
q_8^R	Fixed value of 170^o	$[-45\,,\,0\,,\,45]^o$
q_7^L	$[0\,,\,45\,,\,90]^o$	Fixed value of 60^o
q_8^L	Fixed value of 170^o	Fixed value of 110^o

work has been carried out in a MATLAB-Simulink framework [7] that represents the graphical evolution of the system variables.

The results corresponding to the navigation phase are presented in Fig. 3.7 and Fig. 3.8. In Fig. 3.7 the trajectory followed by the ARS-LRM_v1 system is illustrated by the dotted line representing the movement of the UAV centre of mass M^O. In Fig. 3.8, the evolution of the planning-space variables, for both the planned trajectory (dashed black line) and the closed-loop executed trajectory (blue line), has been shown. As can be observed, the planned trajectory succeeded in commanding efficiently the controlled ARM through the navigation phase without producing collisions with the obstacles existing in the scenario.

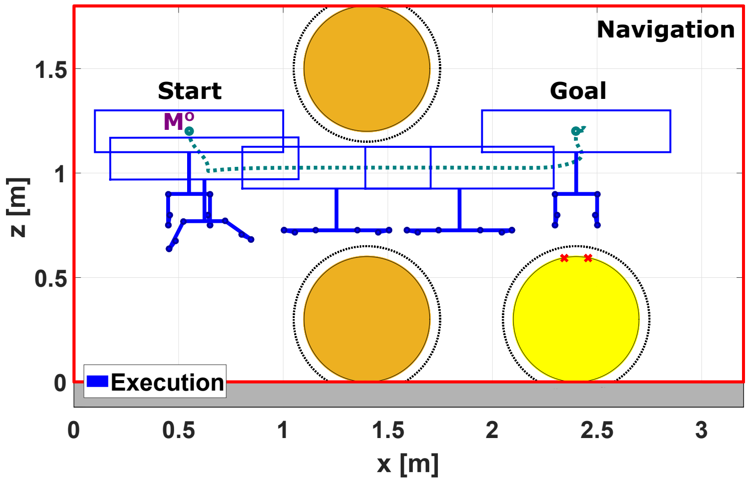

Figure 3.7: Riveting task (navigation phase). The ARS-LRM_v1 system navigates between the obstacles. The dotted line represents the simulated centre of mass M^O position of the multirotor.

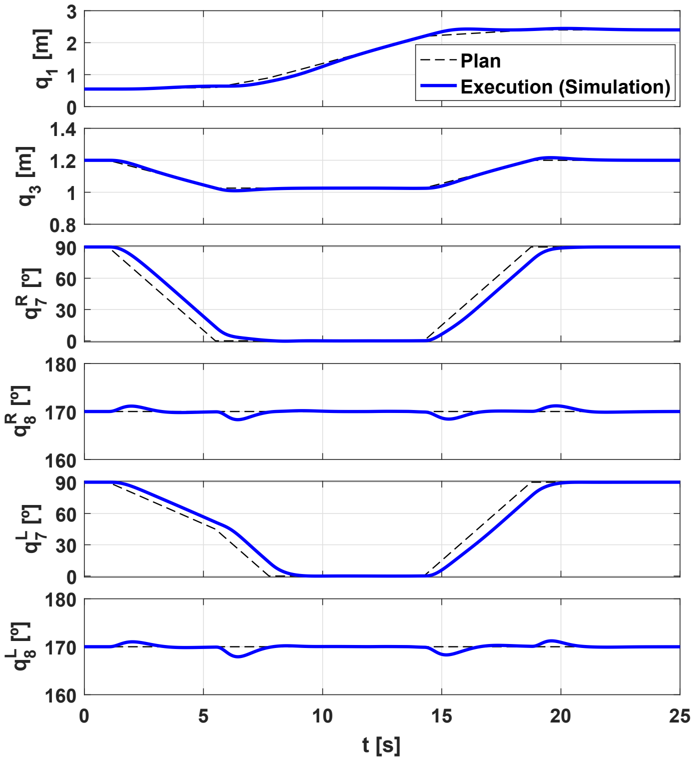

Figure 3.8: Riveting task (navigation phase). Evolution of the planning-space variables: planned (dashed black) and executed (blue) trajectories.

Regarding the manipulation phase, Fig. 3.9 and Fig. 3.10 illustrate the achieved results. As in the navigation phase, Fig. 3.9 shows a schematic representation of the manoeuvres associated with the manipulation phase. Similarly, in Fig. 3.10 the evolution of the planning-space variables, for both the planned trajectory (dashed black line) and the closed-loop executed trajectory (blue line), has been represented for this phase. Once again, the planned trajectory succeeded in commanding efficiently the controlled ARS-LRM_v1 system through the different manipulation manoeuvres involved in the riveting task. It is worth highlighting that Fig. 3.9 (switching) illustrates how the jointly consideration of the planning space for the multirotor and the dual arm allows the optimisation of the switching manoeuvre between the riveting points. More precisely, the MP-ARM planner takes advantage of the multirotor vertical displacement to carry out the switching manoeuvre of the riveting effector in a more efficient way.

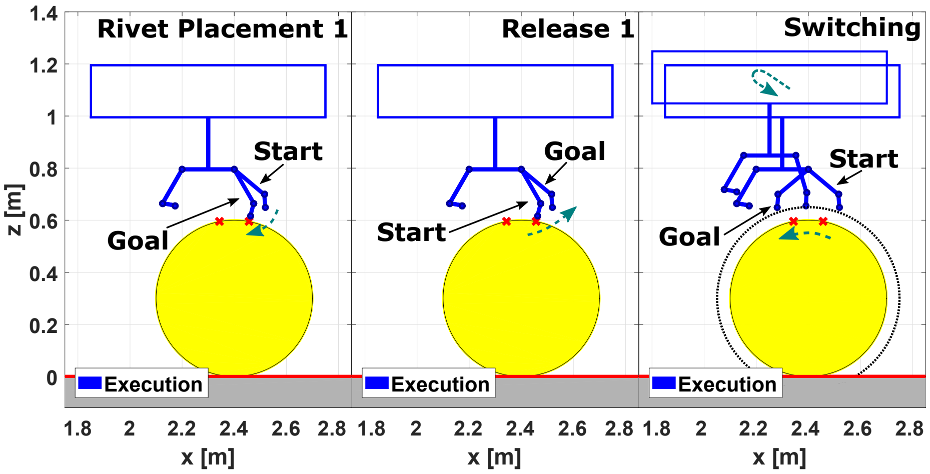

Figure 3.9: Riveting task (manipulation phase). The ARS-LRM_v1 system places the first rivet and then switches between the ready-to-go configurations.

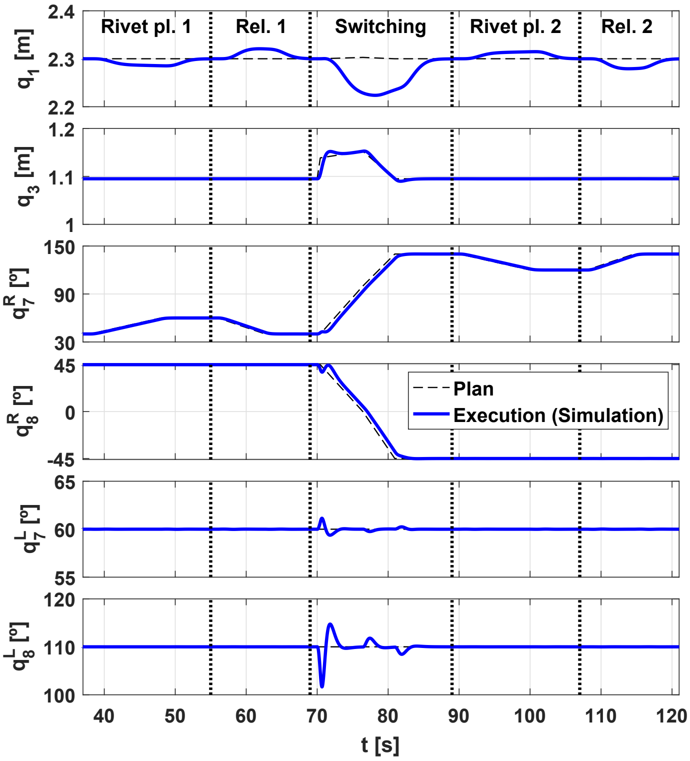

Figure 3.10: Riveting task (manipulation phase). Evolution of the planning-space variables: planned (dashed black) and executed (blue) trajectories.

3.4 Experimental results

Once the preliminary simulations have demonstrated that the fundamentals of the MP-ARM planner provides planned trajectories that are safe enough for the operation of ARMs in cluttered environments, the next step is the experimental validation. Starting from flight tests in a controlled indoor environment, the motion planner will be progressively tested in different outdoor scenarios until finishing with its validation in real industrial sites.

For the experiments performed in this section, the second version of the set of new Aerial Robotic Systems for Long-Reach Manipulation, called ARS-LRM_v2 and presented in Annex A.2, has been used. As it is explained in the annex, its safety features makes this ARM suitable for experimentation.

For the ARS-LRM_v2 system, the configuration variables that have been selected to define the planning space correspond to the green variables in Fig. A.10. However, as the long-bar extension is attached to the aerial platform using a passive revolute joint and the mass distribution of the manipulator is known, the variable q_0 can be expressed as a function of the variables $q_7^L, q_8^L, q_7^R, q_8^R$ by establishing an equilibrium condition in the revolute joint. As for the ARS-LRM_v1 system, the pitch angle q_5 of the aerial platform is considered negligible for planning purposes.

3.4.1 Indoor motion planning

The first experimental tests have been performed in a controlled indoor testbed located in the facilities of the Advanced Center for Aerospace Technologies (CATEC) in Spain [88]. The mitigation of disturbances like wind and the availability of a Vicon-based positioning system [89] with millimetre accuracy offer the safest conditions to start with experimentation. The fundamentals of the MP-ARM planner have been tested in a realistic industrial scenario given by a pipe inspection.

Application scenario: pipe inspection

The schematic description of this scenario is shown in Fig. 3.11 where a planar characterisation is adopted for the sake of clarity. In this environment, yellow circles correspond to pipes existing in the industrial site and surrounding circumferences denote the safety regions whose violations will be treated as collisions. As can be seen, the ARS-LRM_v2 system (first prototype) will be commanded to carry out a contactless inspection of the bottom part of the right pipe using two tools integrated as end effectors in both arms. After this inspection, the ARM will be commanded to return to the initial position. It is important to note that the manipulation at the bottom part of the pipe would not be possible with ARMs in standard configurations (manipulator directly attached to the aerial platform on its bottom part) since the proximity between the multirotor propellers and the pipe might endanger the safety of the system.

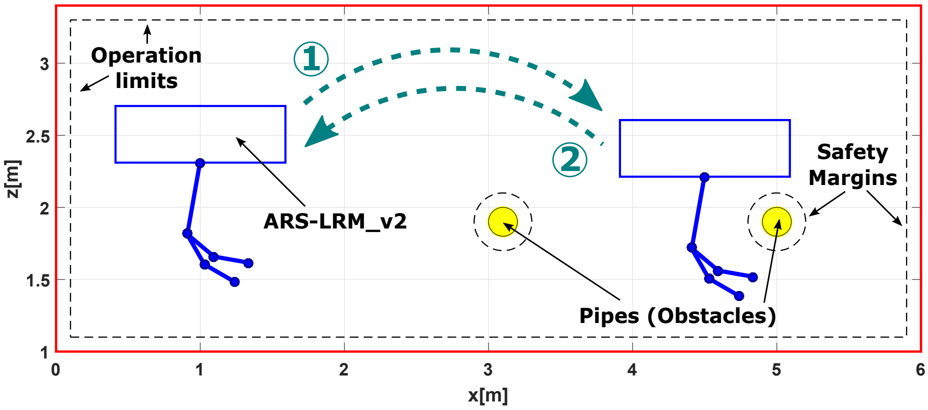

Figure 3.11: Application scenario given by a pipe inspection.

The pipe inspection scenario has been reproduced in a mock-up. Fig. 3.12 presents a visualisation of the indoor setup where the pipes structures are made of plastic. As can be observed, the figure also includes an extended visualisation that shows the initial and final ARM positions, as well as the operation limits superimposing black lines. For the upper limit, it is imposed for safe operation as the maximum height at which the Vicon cameras do not lose coverage.

Analysis of the results

The MP-ARM algorithm has been used here to calculate a motion plan that commands the ARS-LRM_v2 system to complete the pipe inspection. Thus, the motion planner

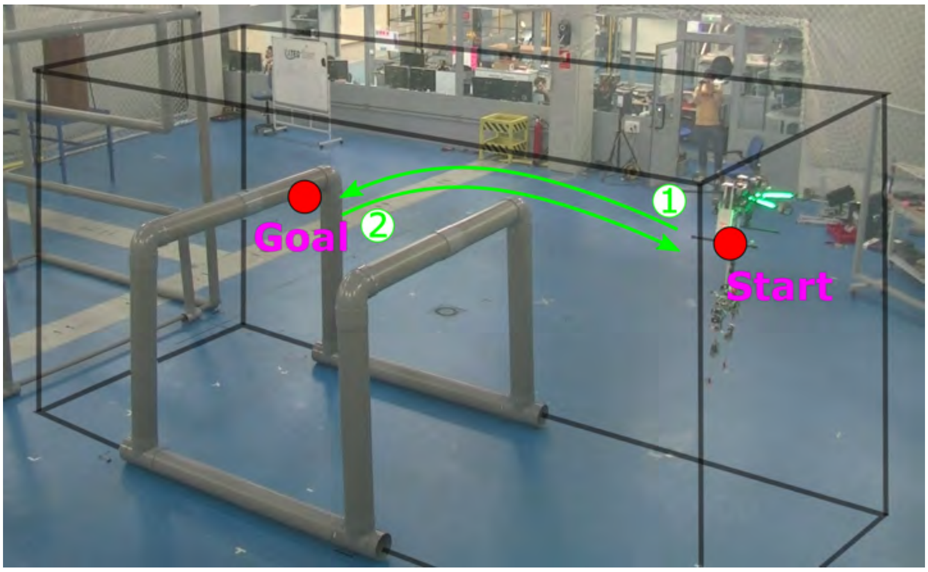

Figure 3.12: Mock-up for pipe inspection. Reference points and operation limits (black lines).

has been firstly executed for the scenario under study and then, the resultant plan has been provided to the ARM. As for the simulation results, the index selected for optimisation has been the cost function CF_E defined in Section 3.2.1 (Cost functions).

Fig. 3.13 collects several snapshots of the experiment. Additionally, the complete video *MP-ARM_pipe_inspection.mp4* can be downloaded from the provided multimedia content, as indicated in Section 1.4. According to the results, the ARS-LRM_v2 system is able to carry out the inspection task through the cluttered scenario, without any collision and following an efficient trajectory.

Furthermore, the operation at the bottom part of the pipe is performed safely due to the suitability of the ARS-LRM_v2 system for this kind of inspection. In order to better analyse the latter, Fig. 3.14 shows the evolution of the planning-space variables when the same planned trajectory is commanded to both the real ARM (left) and the simulation model presented in Annex A.2.2 (right). The similarity existing between both endorses the validity of the proposed approach.

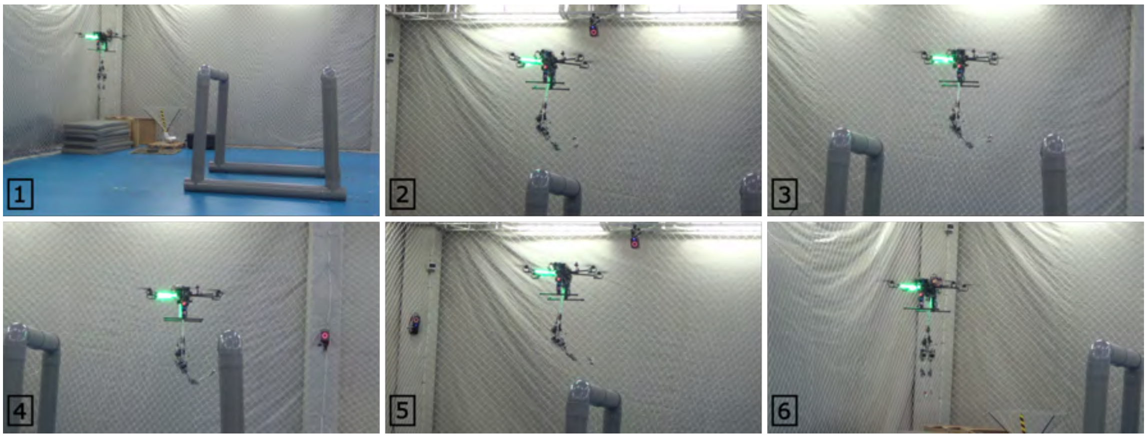

Figure 3.13: Pipe inspection. Snapshots of the operation.

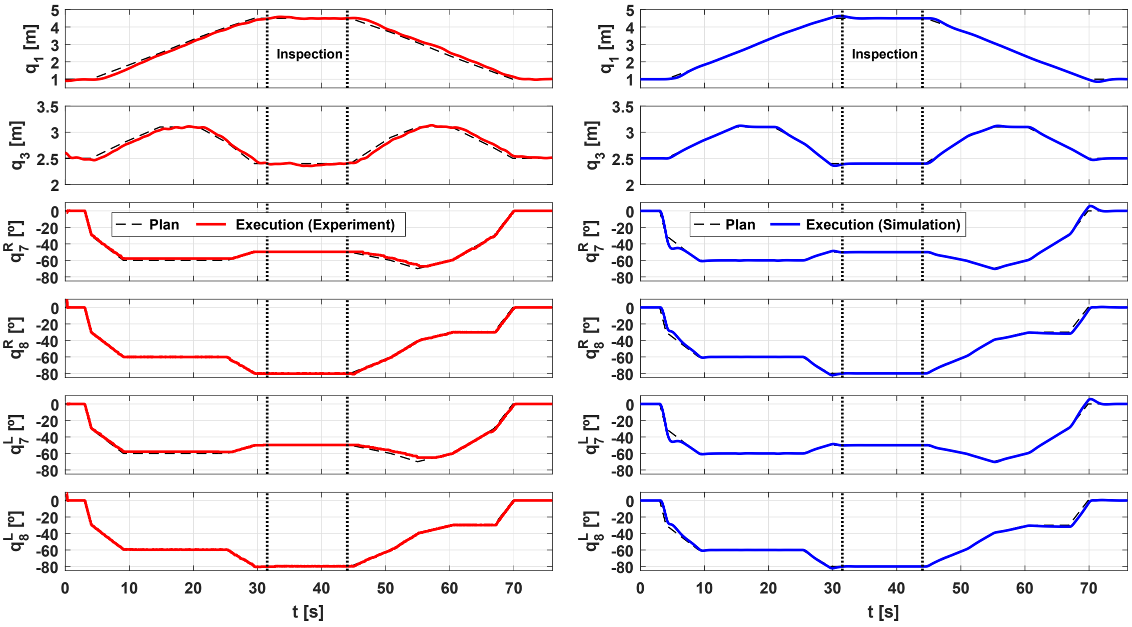

Figure 3.14: Pipe inspection. Evolution of the planning-space variables: experiment (left) and simulation (right). The dashed lines represent the planned trajectory. The time intervals between the vertical dotted lines correspond to the inspection operation itself.

3.4.2 Outdoor motion planning

After the satisfactory results obtained in the controlled indoor environment, the performance of the motion planner is ready to be assessed outdoors. Outdoor scenarios are closer to real industrial sites but the presence of wind disturbances or uncertainties introduced by GNSS measurements in the position estimation can hinder the safe operation. For the new evaluation, the fundamentals of the MP-ARM planner have been tested in a realistic industrial scenario given by the transportation of a long bar in an environment with a complex piping layout.

Application scenario: long-bar transportation

In this practical application, motion confined to a plane shall suffice to satisfactorily complete the proposed task. Hence, a planar characterisation of the scenario is adopted for simplification. The schematic description of the scenario is shown in Fig. 3.15 (above), where yellow elements correspond to several pipe structures (also presented on the side below) and the dashed surrounding areas denote the safety regions for both pipe structures and operational limits whose violations will be treated as collisions. As depicted in the figure, the ARS-LRM_v2 system (first prototype) will be commanded to transport a long bar along the longitudinal direction (the left side of Fig. A.8 illustrates the real system with this purpose). Since both the transported bar and the pipe structures exceed the planar characterisation presented in Fig. 3.15, it is worth mentioning that the potential collisions that could take place between both elements outside the ARS-LRM_v2 movement plane are represented with the following criteria in Fig. 3.15: green colour denotes the region where both the ARS-LRM_v2 system and the transported bar do not produce any collision, and red colour represents the area where the ARS-LRM_v2 system could navigate but in contrast the presence of the transported bar would provoke a collision. Accordingly with the former, the transported bar could only be passed through the lower (and wider) part of the large pipe structure in the right side.

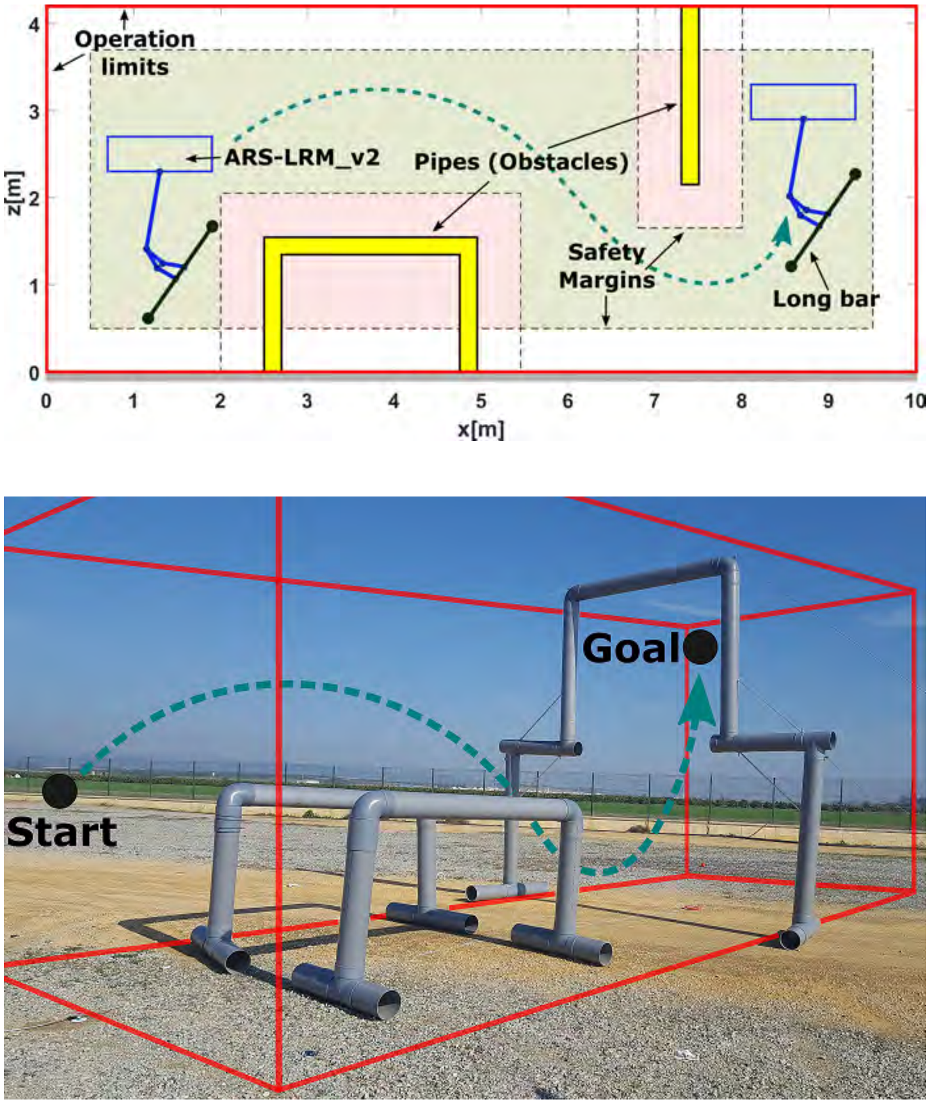

Figure 3.15: Application scenario given by a long-bar transportation.

Analysis of the results

Once more, the MP-ARM algorithm has been used to plan the trajectory that leads the ARS-LRM_v2 system to transport the long bar autonomously. As in previous results, the index selected for optimisation has been the cost function CF_E defined in Section 3.2.1 (Cost functions).

The complete video of the execution *MP-ARM_long-bar_transportation.mp4* can be found in the provided multimedia content, as indicated in Section 1.4, and a representative set of snapshots is shown in Fig. 3.16. According to these results, the ARS-LRM_v2 system is able to transport the long bar along the cluttered scenario without producing collisions with the pipe structures and following an efficient trajectory.

Figure 3.16: Long-bar transportation. Snapshots of the operation.

In order to better analyse the results, Fig. 3.17 shows the evolution of the planning-space variables when tracking the planned trajectory. More particularly, the figure shows how the motion planner generates an efficient trajectory with a reduced number of ARM movements thanks to the optimisation sequence of the planner as well as the joint consideration of both the multirotor and the long-reach manipulator within the planning operation.

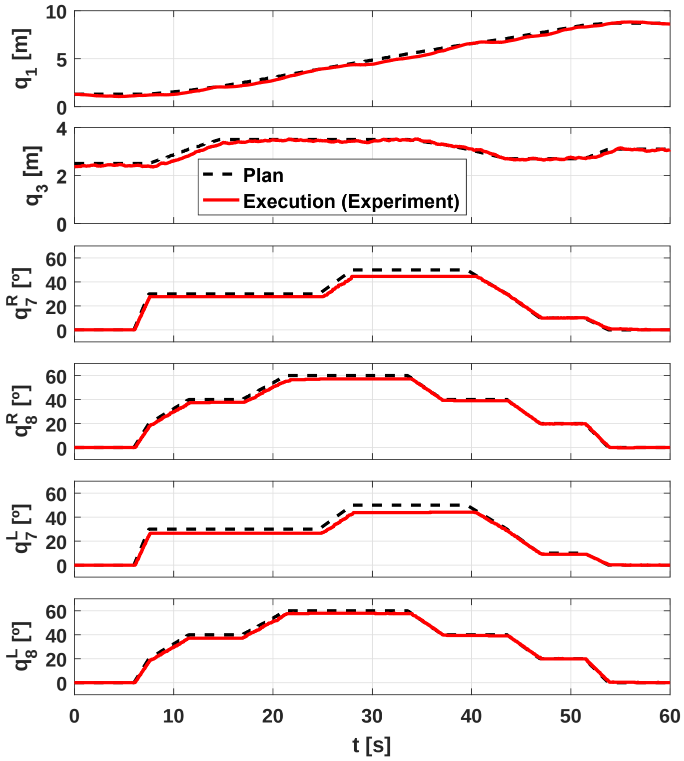

Figure 3.17: Long-bar transportation. Evolution of the planning-space variables: planned (dashed black) and executed (red) trajectories.

3.4.3 In-flight outdoor motion planning

With the fundamentals of the MP-ARM motion planner already validated in different environments, this section will put the focus on the in-flight planning capabilities presented in Section 3.2.2. For that, the MP-ARM planner including the in-flight capabilities has been tested in a similar scenario to the long-bar transportation presented before.

Application scenario: long-bar transportation

The application scenario is the same as the one presented in Section 3.4.2 (Application scenario: long-bar transportation) with slight variations in the piping layout. In particular, as can be seen in Fig. 3.18, one more pipe structure has been added in the left side to make the environment more cluttered. Also, in contrast to previous experiments, the final prototype of the ARS-LRM_v2 system has been used here since this is the ARM integrating all the software and hardware modules described in Annex A.2.1 (Software integration and Hardware integration) and required for in-flight planning.

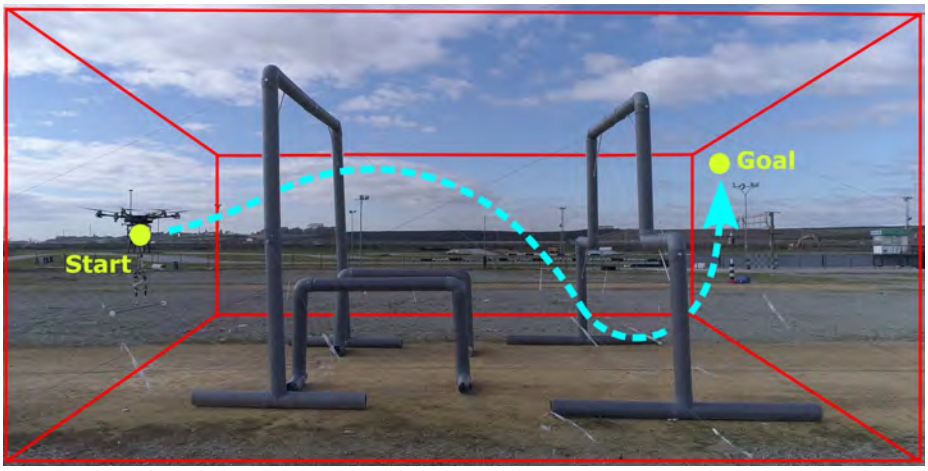

Figure 3.18: Application scenario given by a long-bar transportation used for the validation of in-flight planning capabilities.

Regarding the integrated procedure that is needed to complete the transportation task, this is composed of several steps. From the beginning of the operation, the Multisensor Mapping module mentioned in Annex A.2.1 (Software integration) should give an accurate representation of the environment with only a vertical movement of a few metres. The representation is kept updated online during the flight and, when requested, its latest version is provided to the MP-ARM motion planner. In particular, the planner computes an online plan using the updated representation after the take-off. Finally, the long bar is transported autonomously through the pipe structures following the optimised plan and making use of the long-reach manipulation capabilities of the ARS-LRM_v2 system.

Analysis of the results

Fig. 3.19 collects some snapshots of a real execution of the long-bar transportation. Additionally, the complete video *MP-ARM_in-flight_long-bar_transportation.mp4* can be downloaded from the provided multimedia content, as indicated in Section 1.4.

Figure 3.19: Long-bar transportation with in-flight planning capabilities. Snapshots of the operation.

Following the operation sequence, the Multisensor Mapping module is continuously building online an accurate representation of the scenario in a point-cloud format. Fig. 3.20 (top) shows the representation generated during the execution of the experiment. This has an accuracy of a few centimetres, does not include false-positive measurements, and is very detailed, even capturing very thin objects such as the cords used to fix the pipe structures. Consequently, since this representation is the main input for the motion planner, its very high sensitivity and detection capability are highly valuable to increase the navigation safety.

Using the generated representation as input, the MP-ARM method including the in-flight capabilities computes online the motion plan. As in previous results, the index

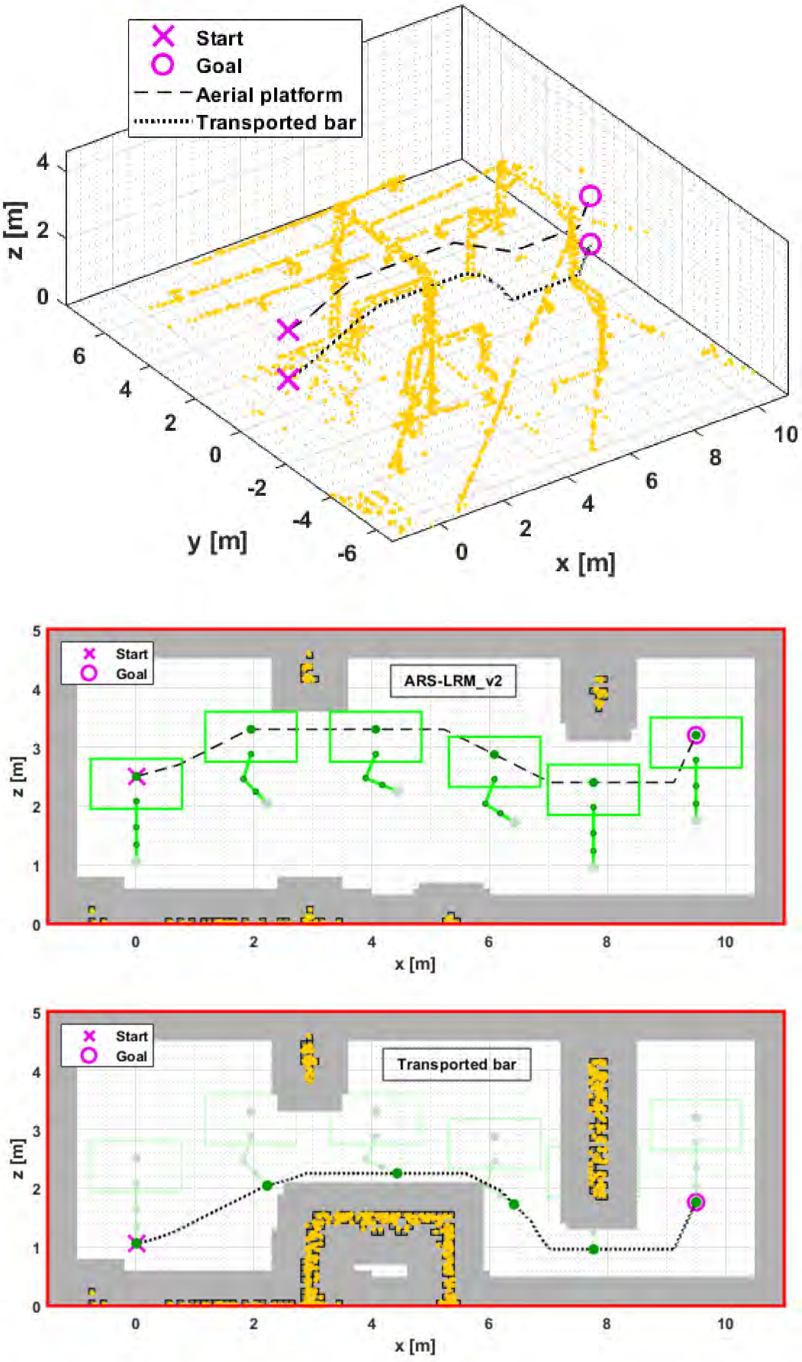

Figure 3.20: Long-bar transportation with in-flight planning capabilities. Trajectory planned in flight: 3D view including the representation of the scenario generated by the Multisensor Mapping module (top) and 2D views for the collision spaces associated with the ARS-LRM_v2 system (centre) and the transported bar (bottom).

selected for optimisation has been the cost function CF_E defined in Section 3.2.1 (Cost functions). For the presented experiment, Fig. 3.20 depicts the planned trajectory, whose computation time is around 30 seconds. The trajectory has been presented in two ways. Firstly, a 3D view helps to spatially visualise the plan by including the point-cloud-based representation, the trajectory of the aerial platform and the trajectory of the centre of mass of the transported bar. Secondly, a 2D view has been presented for a better analysis of the results. At the same time, this 2D view has been divided into two graphs, where one of them is focused on the ARS-LRM_v2 system and the other one is focused on the transported bar. The reason of this division is that, as the transported bar exceeds the dimension of the ARM, the collision spaces of both ARS-LRM_v2 system and transported bar are different. These collision spaces have been represented in the figures following the same format as in Fig. 3.5.d but also adding the safety regions around the obstacles. In view of the results, the motion planner generates efficient trajectories considering jointly the configuration variables of both the aerial platform and the long-reach manipulator and with no collisions with the cluttered environment. In particular, it can be seen as the ARS-LRM_v2 system passes through the left and right pipe structures by combining efficiently movements of both the aerial platform and the arms.

Once the planned trajectory is available, this is commanded as reference to the ARS-LRM_v2 system for its autonomous execution. For the presented experiment, Fig. 3.21 depicts the evolution of the planning-space variables. The graphs show a proper tracking with enough accuracy to fulfil the mission in such cluttered environment. Finally, it should be highlighted that the complete experiment has been carried out around 10 times with similar results and a success rate of 100%.

3.4.4 Industrial validation

The final step in the full validation process of the proposed motion planner is the evaluation of its performance for I&M tasks in real industrial sites. With this purpose, the MP-ARM planner including the in-flight capabilities has been tested for the installation of a sensor on the surface of an industrial ball mill located in the facilities of

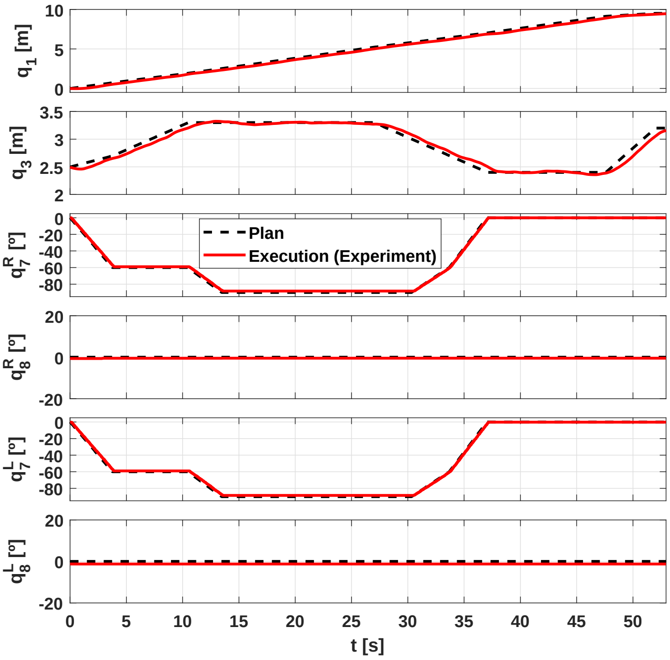

Figure 3.21: Long-bar transportation with in-flight planning capabilities. Evolution of the planning-space variables: planned (dashed black) and executed (red) trajectories.

a cement factory. Complementing this test, the results of other industrial experiments have also been presented briefly.

Application scenario: sensor installation on an industrial ball mill

The industrial validation has been conducted in a cement factory located in Spain. Here, the objective is the autonomous installation of the mock-up sensor shown in Fig. 3.22 on the surface of a ball mill in order to monitor its state. The description of the operation is shown in Fig. 3.23. As represented, the ARS-LRM_v2 system (final prototype) will be commanded to transport the sensor along the longitudinal direction, from the start location to the goal location, and then, to install it using magnets in the deployment point located on the upper part of the mill. Moreover, as in previous flight tests, safety regions have been considered for all the elements in the scenario as well as the operation limits denoted by the red lines.

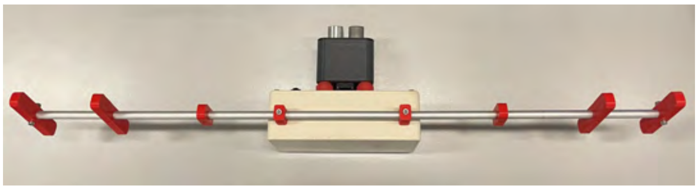

Figure 3.22: Mock-up sensor to be installed on the surface of the ball mill.

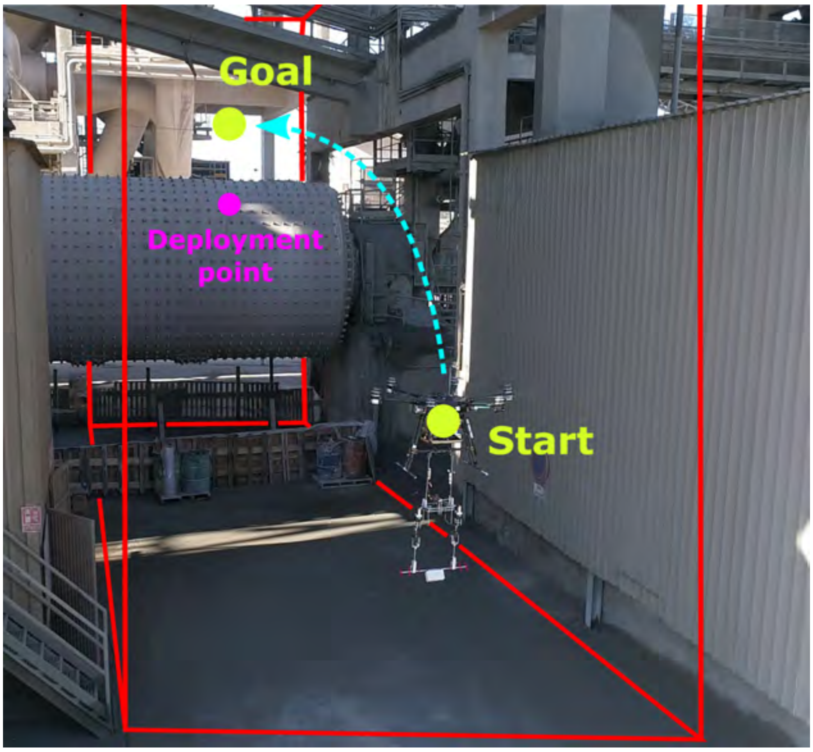

Figure 3.23: Application scenario given by a sensor installation on an industrial ball mill.

Traditionally, this kind of operation is carried out manually by a worker following a strict safety process. Firstly, the electrical transformer that feeds the ball mill must be uninstalled to avoid unexpected starts. Secondly, the free rotation of the mill must also be blocked using a brake system intended to it. Thirdly, the worker needs an specific training to work at heights. Moreover, the worker must perform the operation using a lifeline whose installation, usually permanent, must be certified. Finally, due to the time needed to prepare the operation itself, this should be planned in advance because these temporary shut-downs may affect the production capabilities of the factory. In contrast, the autonomous installation described here can be performed without risk for workers and much faster because most of the safety steps can be omitted. The latter contributes to save costs.

Similar to the flight tests in Section 3.4.3, the integrated procedure to perform the sensor installation is composed of several steps. Firstly, the Multisensor Mapping module mentioned in Annex A.2.1 (Software integration) should build and update online an accurate representation of the environment from the very beginning of the operation. Given the big dimensions of the ball mill and its surroundings, an initial vertical flight of several metres over the take-off point is executed to explore all the work area. Then, the representation is provided to the MP-ARM planner, which generates in flight a safe and efficient motion plan. Once the plan is available, it is used as reference to transport the sensor from the start configuration to the goal configuration required to place the sensor in the deployment point. Finally, the sensor is installed making use of the long-reach manipulation capabilities of the ARS-LRM_v2 system.

Analysis of the results

Fig. 3.24 collects some snapshots of the execution of the industrial validation. Additionally, the complete video *MP-ARM_industrial_sensor_installation.mp4* can be downloaded from the provided multimedia content, as indicated in Section 1.4.

Following the operation sequence, Fig. 3.25 (top) depicts the point-cloud representation generated online by the Multisensor Mapping module during the execution of the industrial validation. This representation stands out for the same features as

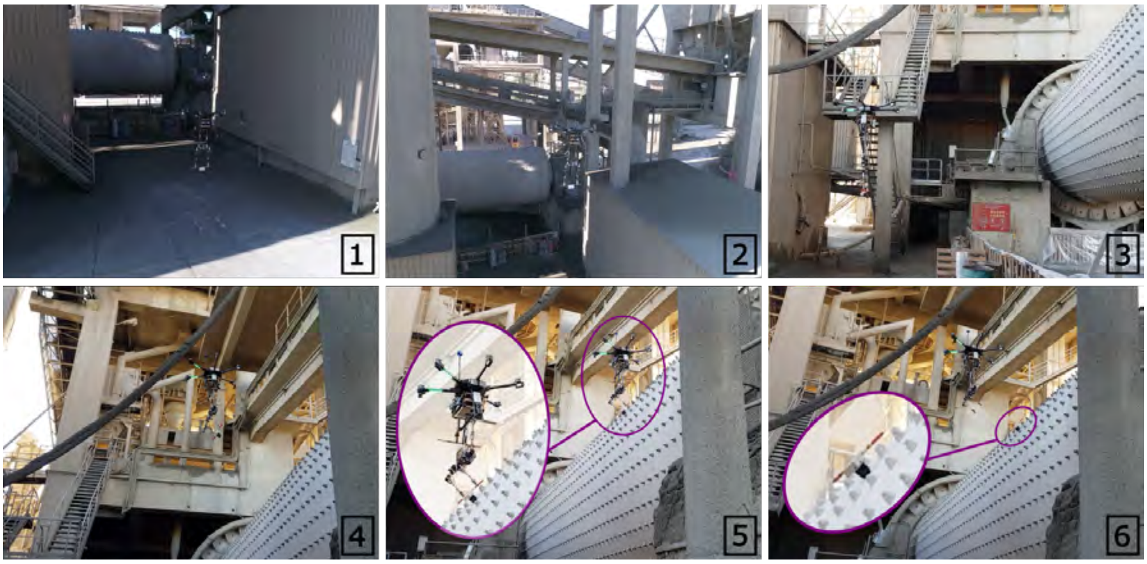

Figure 3.24: Sensor installation on an industrial ball mill. Snapshots of the operation.

in the experiments performed in Section 3.4.3; these are, the accuracy, the absence of false-positive measurements, and the high level of detail. The latter allows the detection of very thin objects present in the environment like the lifeline located horizontally over the mill. This is critical because the lifeline is very close to the goal state for the motion planner and the flying area during the deployment of the sensor.

Using the previous representation as input, the MP-ARM method including the in-flight capabilities computes online the motion plan depicted in Fig. 3.25, whose computation time is around 1 second. Once more, the index selected for optimisation has been the cost function CF_E defined in Section 3.2.1 (Cost functions). As for the analysis of the flight test in Section 3.4.3 (Analysis of the results), both 3D and 2D views of the plan have been represented following the same format. Again, the motion planner generates safe and efficient trajectories combining movements of the aerial platform and the long-reach manipulator. More in detail, the method plans a trajectory that smoothly connects the start and goal configurations avoiding any violation of the safety regions. Thus, during the flight of the aerial platform, the dual arm is adapted toward a partially-folded configuration that maximises its manoeuvrability during the subsequent placement of the sensor.

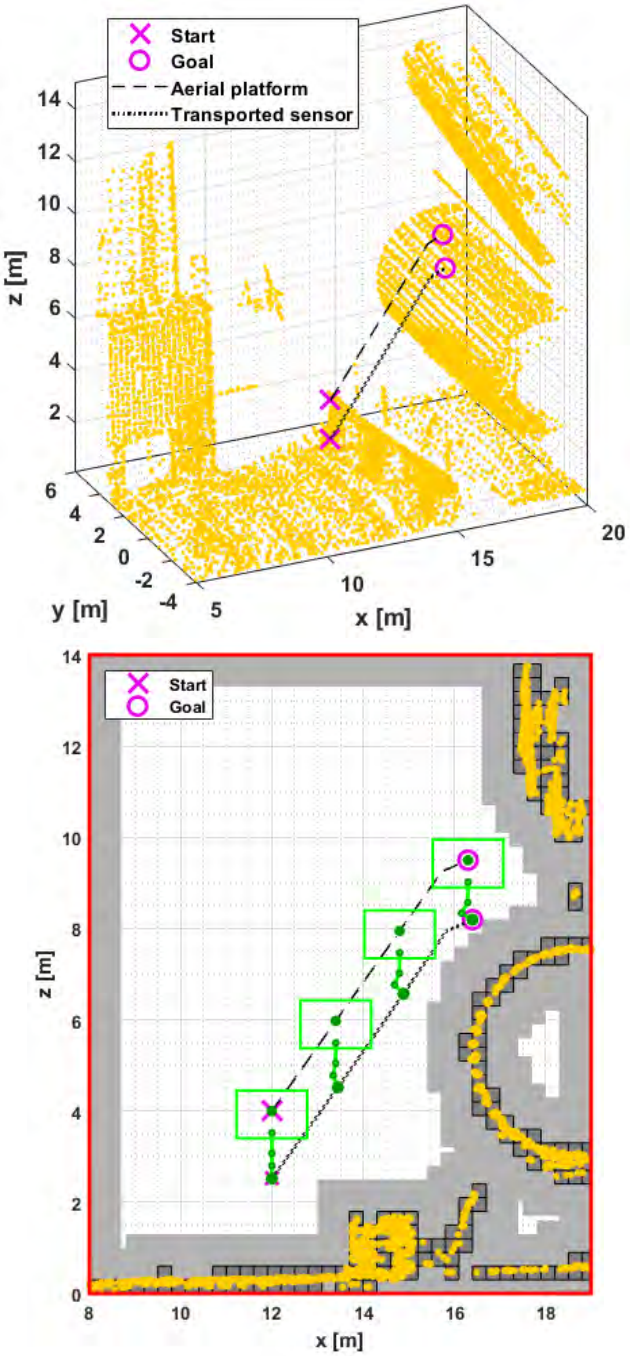

Figure 3.25: Sensor installation on an industrial ball mill. Trajectory planned in flight: 3D view including the representation of the scenario generated by the Multisensor Mapping module (top) and 2D view for the collision space associated with the ARS-LRM_v2 system (bottom).

After the planning of the trajectory, this is commanded to the ARS-LRM_v2 system for its autonomous execution. Fig. 3.26 represents the evolution along time of the configuration variables within the planning space. Once more, the graphs show an accurate tracking that allows reaching the goal configuration safely. Finally, the sensor is successfully placed in the deployment point. At this point, it should be emphasised the advantages of the long-reach manipulation since it allows an increment in the safety distance between the ARM propellers and the area where the manipulator is working. Thus, the sensor deployment could not be accomplished if the dual arm were attached directly to the bottom part of the aerial platform because the propellers might collide with the surface of the mill.

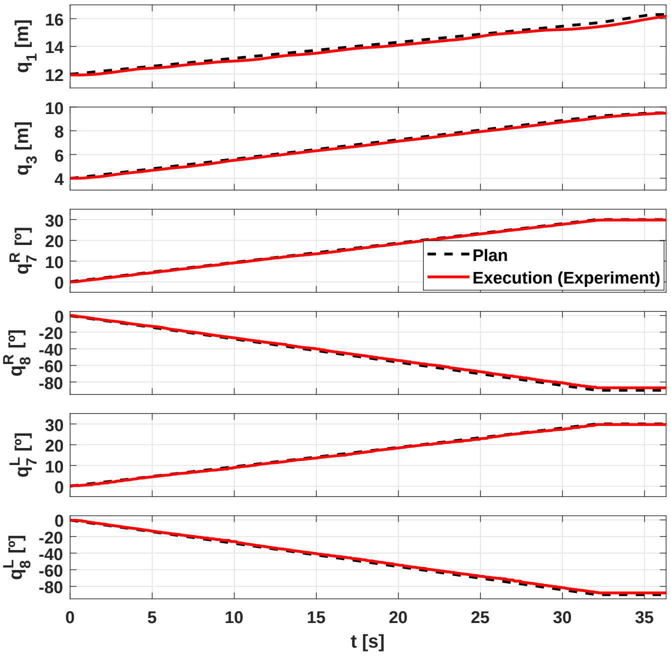

Figure 3.26: Sensor installation on an industrial ball mill. Evolution of the planning-space variables: planned (dashed black) and executed (red) trajectories.

To conclude the analysis, a second trajectory has been planned for the sensor installation on the the back surface of the ball mill. The purpose is to evaluate the motion planner in more challenging conditions, forcing the ARS-LRM_v2 system to fly through the gap between the ball mill and the structure over it in order to complete this alternative mission. However, the trajectory has not been commanded to the

ARM for its execution because the lifeline located horizontally over the mill blocks the accessibility to the gap. Removing the lifeline from the point-cloud representation, the MP-ARM algorithm computes in around 6 seconds the feasible motion plan presented in Fig. 3.27. In view of the result, the motion planner provides a safe and efficient trajectory considering the aerial platform and the manipulator jointly. The latter is essential to cross the gap between the ball mill and the structure over it without any collision with the environment.

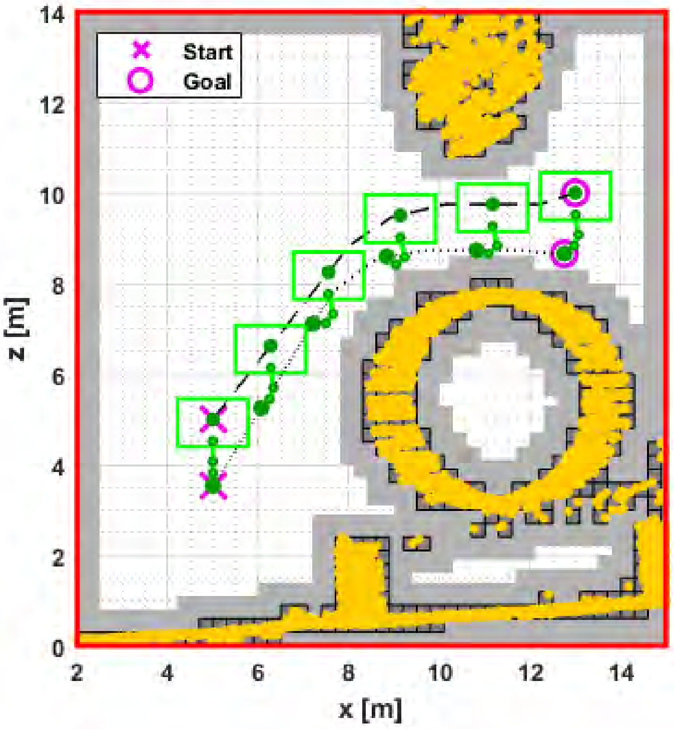

Figure 3.27: Sensor installation on an industrial ball mill. Trajectory planned for the installation on the back surface of the mill.

Other experiments

In a similar way to the approach presented above, the MP-ARM method including the in-flight capabilities has also been used for other I&M tasks in different industrial environments with equivalent results. These tasks include one of the final demonstrations of the AEROARMS project [10] in an Oil&Gas industry located in Germany.

Here, as the result of a joint collaboration with CATEC [88], the MP-ARM planner computed different trajectories for the ARM presented in [13]. The purpose of these trajectories was to allow the ARM to reach autonomously a location close to surfaces of interest to be inspected by contact using an eddy-current sensor installed as end effector. These surfaces include industrial tanks, horizontal pipes located at height and surplus gas separators. For the sake of brevity, only a set of images of these scenarios with the corresponding planned trajectories has been collected in Fig. 3.28. The same conclusions as in previous experiments can be applied here.

3.5 Conclusions

The MP-ARM motion planner presented in this chapter has demonstrated to be a suitable solution in the planning of safe and efficient trajectories for ARMs endowed with robotic arms in cluttered environments. This is possible thanks to the joint consideration of the aerial platform and the manipulator within the planner operation. In this way, more system states are accounted and wider operating conditions can be achieved since there is a higher control of system configurations in the strategies for collision checking and optimisation. Moreover, the presented planner is able to adapt the robot motion to different operation phases, including navigation and manipulation stages.

The MP-ARM approach has also been endowed with in-flight planning capabilities. This feature is essential to operate in uncertain environments like the industrial sites presented in Section 3.4.4 where there were not any available map prior to the operation. In this context, the method to integrate representations of the scenario provided by the Multisensor Mapping module has demonstrated to be suitable. Also, the search of computational efficiency has brought the possibility to plan trajectories whose computation time makes the in-flight computation feasible.

A wide set of experiments, involving ARMs with different configurations and realistic industrial applications, has endorsed the validity of the MP-ARM motion planner. For all these experiments, the planned trajectories have addressed the different ARMs to proper executions.

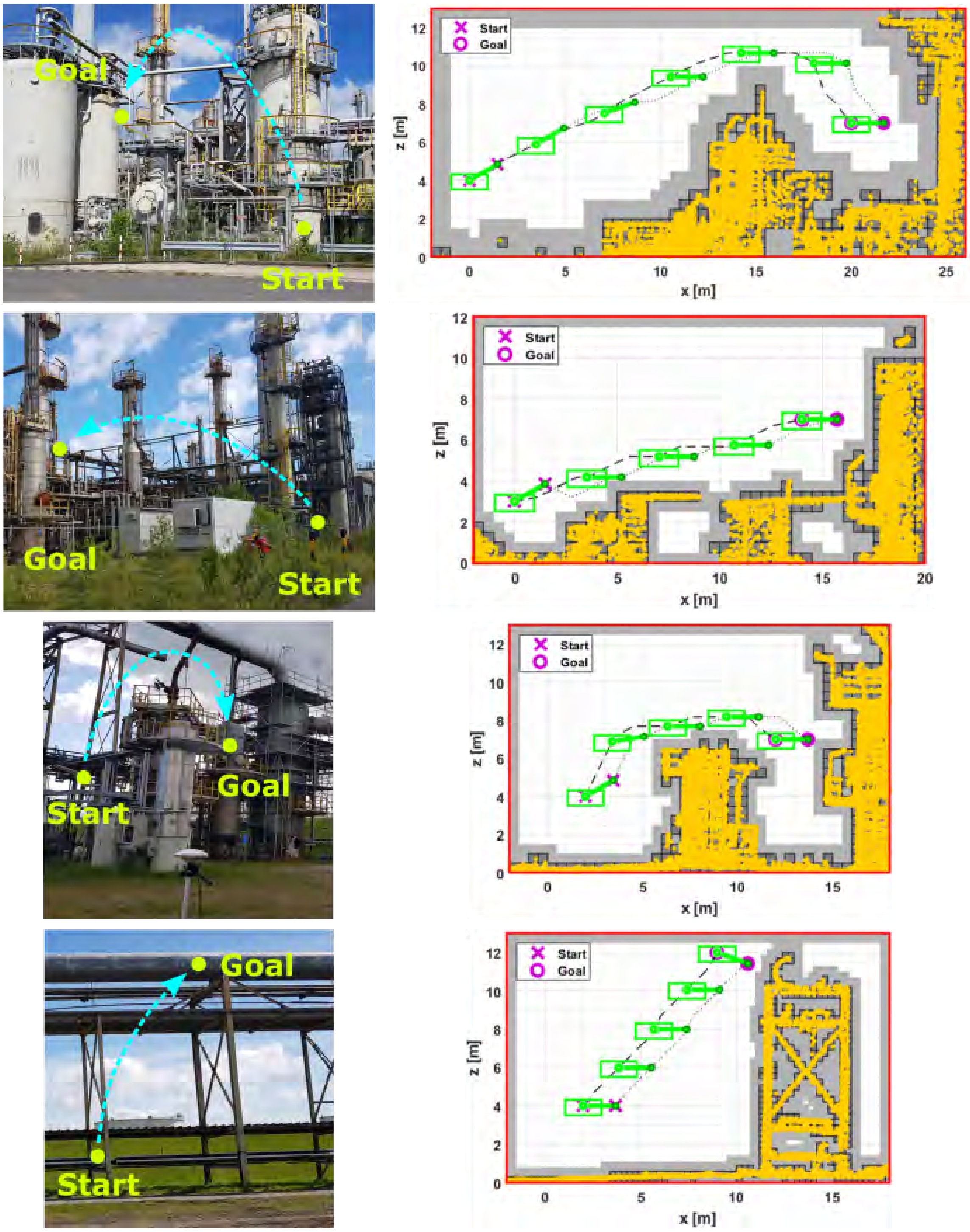

Figure 3.28: Inspection by contact of different surfaces of interest. Application scenarios (left) and planned trajectories (right).

Chapter 4

Extensions on motion planning for aerial manipulators

> No te conformes con lo que necesites,
> lucha por lo que mereces.
>
> ---
>
> Cristina M. Lira

The MP-ARM motion planner has demonstrated in Chapter 3 to be a suitable solution in the planning of safe and efficient trajectories for ARMs endowed with robotic arms in cluttered environments. However, other intensive tests of the planner have also shown that there are some particular circumstances that may compromise its good performance. These situations correspond to a demanding use of the planner and will be analysed in detail in this chapter. For each of them, an extension over the fundamentals of the MP-ARM planner will be proposed to enhance its execution. These are the Dynamics Awareness for robust obstacle avoidance, the Velocity Adaptation mechanism for a better optimisation of the execution time of the planned trajectories, and the Aerodynamics Awareness to discard ARM states whose associated aerodynamic phenomena may provoke undesired collisions. The simulations that validate these extensions complement the chapter.

4.1 Dynamics awareness

4.1.1 Motivation

The fundamentals of the MP-ARM approach presented in Section 3.2.1 guarantee planned trajectories that are efficient and collision-free. However, the strong dynamical coupling that usually exists in ARMs between the aerial platform and the manipulator requires further attention since it may provoke considerable differences between planned and executed trajectories. The collision risk introduced by these differences is especially critical for cluttered environments like industrial sites. For the particular case of the ARS-LRM_v1 system presented in Annex A.1, Fig. 4.1 illustrates a simple example of the undesired dynamic effects that must be avoided.

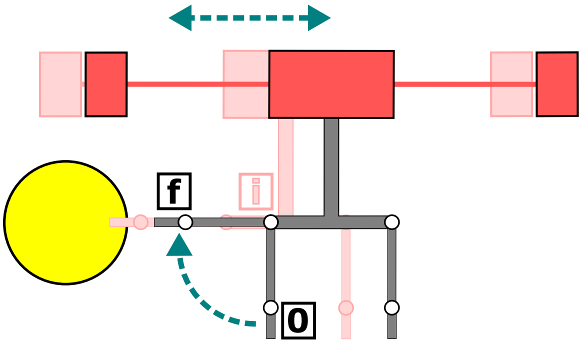

Figure 4.1: ARS-LRM_v1 system spreading the left arm from initial position 0 to final position f according to a simple plan. The ARM oscillation during the execution (shaded intermediate position i) produces a collision with the yellow obstacle.

In order to overcome the undesired influence of the ARM dynamical coupling, the inclusion of the system dynamics within the planning operation is required for robust obstacle avoidance. This Dynamics Awareness (DA) will allow addressing scenarios where the obstacle proximity narrows considerably the safe operation areas and consequently, the correlation between planned and finally executed trajectories is even more essential.

4.1.2 Motion planner

With the purpose of solving the planning problem associated with the strong dynamical coupling existing in ARMs, the fundamentals of the MP-ARM approach presented in Section 3.2.1 are transformed into a more advanced planning method that incorporates Dynamics Awareness (MP-ARM-DA). The new algorithm guarantees robust obstacle avoidance by modifying the $COLLISION(x_{nearest}, x_{new}, map)$ function described in Section 3.2.1 (Collision checking). Instead of checking collisions in a set of geometrical intermediate configurations that are generated using lineal interpolation between the nodes under analysis (MP-ARM blue dashed line in Fig. 4.2), the checking will be carried out now through dynamical intermediate configurations that belong to the trajectory obtained after simulating the closed-loop dynamics of the controlled ARM (MP-ARM-DA green solid line in Fig. 4.2). Furthermore, the application of the previous technique for tree extension without collisions is based on a root-to-candidate validation, as can be seen in Fig. 4.2. Thus, not only the dynamical feasibility of the new possible branch reaching the candidate node is analysed, but also the complete trajectory from the tree root since such local feasibility will also depend on the absence of collisions in the full dynamical trajectory. The consequence of the application of this approach is the derivation of a search tree whose expansion is based on the behaviour of the controlled system and therefore the resultant planned trajectories are both compatible with dynamic constraints and free of obstacles.

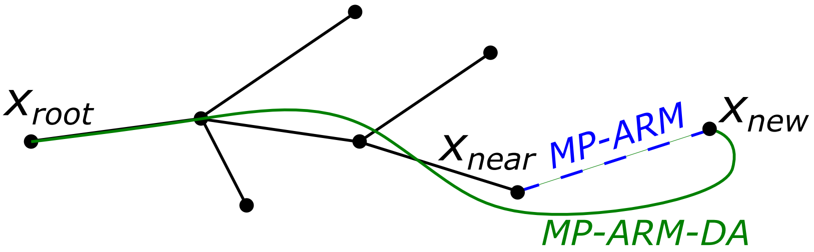

Figure 4.2: Operation basis of the $COLLISION(x_{nearest}, x_{new}, map)$ function: blue dashed line for the MP-ARM algorithm and green solid line for the MP-ARM-DA algorithm.

Associated with the paradigm of motion planning with Dynamics Awareness, the concept of guiding obstacles is also introduced. This is presented below.

Guiding obstacles to enforce manipulation patterns

In some applications, the definition of the manipulation task implies the enforcement of certain manipulation patterns like rectilinear movements of the end effector. This kind of requirements can be addressed in the planner by means of artificial guiding obstacles that enforce the resulting trajectory to match the desired manipulation pattern. Fig. 4.3 depicts an example of the usage of these guiding obstacles for the ARS-LRM_v1 system. In this scenario, the two guiding obstacles allow the right end effector to be approached to the yellow surface following a rectilinear movement that is perpendicular to the contact surface. In this way, other undesired trajectories that could make the end effector slides onto the surface are conveniently discarded.

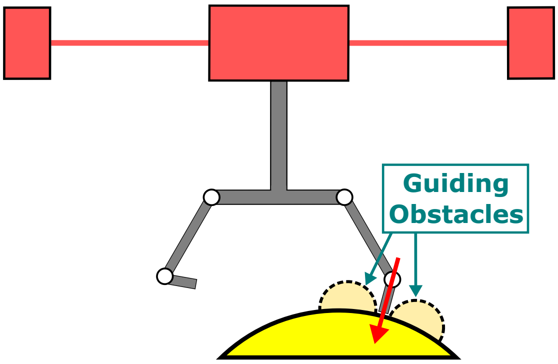

Figure 4.3: Guiding obstacles enforcing a rectilinear movement perpendicular to the contact surface.

It should be highlighted that the application of guiding obstacles only makes sense together with the use of the MP-ARM-DA planner, as it will be demonstrated later in Section 4.1.3. Due to the nature of these obstacles, ARMs are forced to operate very close to them and potential deviations associated with the robot dynamics to fulfil the motion plans would cause their violation.

4.1.3 Simulation results

As the MP-ARM planning approach, the new MP-ARM-DA motion planner is based on general design principles and can be applied to different kinds of aerial manipulators. To illustrate the simulation results, the ARS-LRM_v1 system presented in Annex A.1

has been used. For its dynamic simulations needed for motion planning with Dynamics Awareness, the model and controller derived in Annex A.1.2 have been adopted.

In order to demonstrate the validity of the MP-ARM-DA strategy, the algorithm will be tested in two realistic industrial scenarios; these are, a riveting task similar to the one included in Section 3.3 (Application scenario: riveting task), and a chimney repairing task. Both applications will require motion planning with Dynamics Awareness to perform robustly the desired operations in the associated cluttered environments.

Considering all the above information, the MP-ARM and the MP-ARM-DA motion planners have been executed for the proposed scenarios, selecting as index for optimisation the cost function CF_E defined in Section 3.2.1 (Cost functions). The resultant plans (represented in the following figures with light blue lines for the MP-ARM algorithm and with light green lines for the MP-ARM-DA algorithm) have also been provided to the controlled ARS-LRM_v1 system in order to analyse its closed-loop behaviour (represented in the following figures with dark blue lines for the MP-ARM algorithm and with dark green lines for the MP-ARM-DA algorithm) when following the planned trajectories. For that, the model and controller included in Annex A.1.2 have been used. The simulation work has been carried out in a MATLAB-Simulink framework [7] that provides the graphical evolution of the system variables. This graphical output will be utilised to illustrate the obtained results.

For the analysis of the results, the simulations carried out for both scenarios have been organised around two main lines. Firstly, the need to employ the MP-ARM-DA algorithm for robust obstacle avoidance in cluttered environments will be justified. To this end, the performance limitations of the MP-ARM algorithm will be pointed out. Secondly, a detailed analysis of the MP-ARM-DA performance in terms of the suitability of the executed trajectories will be presented.

Application scenario: riveting task

The application scenario is the same as the one presented in Section 3.3 (Application scenario: riveting task) with a small adaptation in the piping layout. In particular, one additional pipe has been added in the left side to make the environment more cluttered. The schematic description of the new scenario is shown in Fig. 4.4.

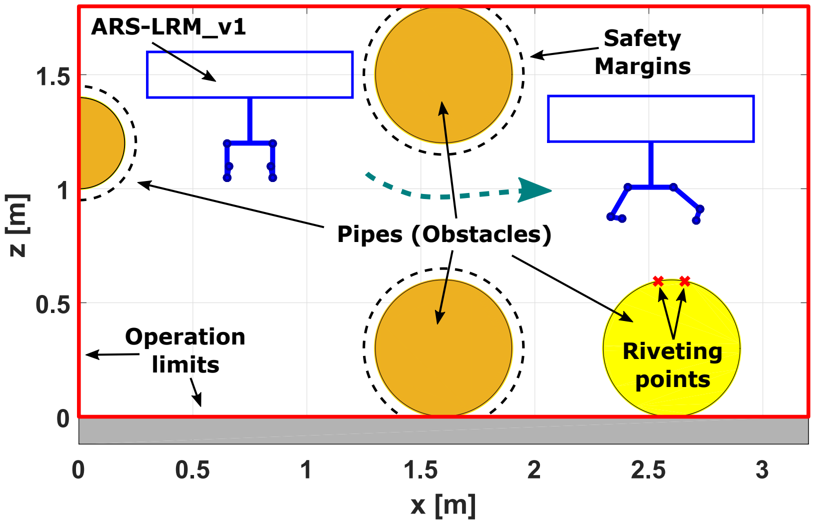

Figure 4.4: Application scenario given by a riveting task used for the validation of motion planning with Dynamics Awareness.

Again, the achievement of the riveting task requires the execution of the same intermediate operations described in Section 3.3 (Application scenario: riveting task). However, the movement of the right end effector during each riveting placement and its subsequent release should also follow now the perpendicular direction to the pipe in the target point. Thus, slides onto the manipulation surfaces can be avoided for a more robust operation. To cope with that, two guiding obstacles will be considered on either side of the riveting points during the manipulation phase.

Application scenario: chimney repairing task

This scenario is composed of two industrial chimneys represented in Fig. 4.5 by four dark grey rectangles. The two light grey rectangles corresponds to the free space inside the chimneys and the dashed surrounding areas denote safety regions whose violations will be treated as collisions. As depicted in the figure, the ARS-LRM_v1 system will be commanded to repair a crack inside the right chimney (target point marked in red) with a tool located in its right arm while the left arm provides visual feedback by pointing a camera integrated as end effector (see Fig. A.1). It is worth noting that, thanks to the long-reach configuration, the aerial platform can operate out of the chimney with certain separation distance while the repairing task is being performed. This contributes to reduce the undesired aerodynamic effects that can be present inside the chimney. For this scenario, the repairing operation will assume ideal conditions in the chimney surface; this is, absence of interaction forces.

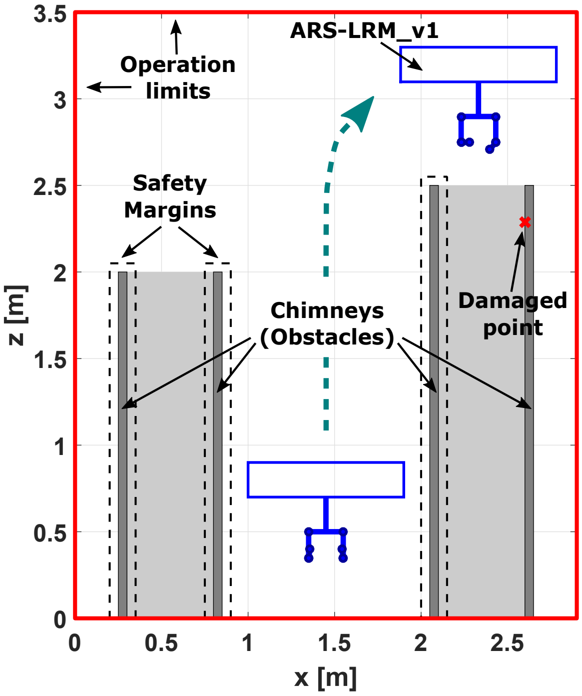

Figure 4.5: Application scenario given by a chimney repairing task.

Similarly to the riveting application, the achievement of the repairing objectives defined previously requires the execution of certain intermediate operations that include both navigation and manipulation manoeuvres:

1. Navigation phase: this phase corresponds to the ARM displacement required to reach an observation position over the right chimney. After this, a short transition phase not requiring planner execution will enforce a ready-to-go configuration for the repairing manoeuvre that will be accomplished during the manipulation phase.

2. Manipulation phase: this phase covers the different manoeuvres involved in the manipulation task under consideration; this is, the chimney repairing operation.

In the subphases described below it is assumed that the left arm will adapt its configuration to optimise the visual feedback provided by its integrated camera:

(a) Repair: approaching of the tool effector integrated in the right arm to the target point in the perpendicular direction to the chimney surface in order to avoid slides onto it. For that, two guiding obstacles will be considered on either side of the target point.

(b) Release: opposite manoeuvre to the repair in which the tool effector leaves the target point, again following the perpendicular direction to the chimney surface. The same guiding obstacles as in the previous subphase will be considered.

Analysis of the results: MP-ARM algorithm

The MP-ARM algorithm has certain performance limitations in particular circumstances since it does not consider the ARM dynamics. For instance, dynamic effects like oscillations are not accounted for during the collision-checking process, which may produce risky situations when commanding the planned trajectory to the real system. In order to illustrate the potential impact of this missing feature, two complete simulations with the MP-ARM algorithm, including both the planned and the closed-loop trajectories, have been considered for the two application scenarios under study.

According to these simulations, the MP-ARM algorithm plans efficient trajectories for navigation and manipulation phases in both scenarios. However, the corresponding closed-loop trajectories described by the controlled ARS-LRM_v1 system do not always satisfy the desired collision-free and slide-free properties. For the riveting task, Fig. 4.6 shows how the ARS-LRM_v1 system violates safety margins during the navigation phase whereas Fig. 4.7 reveals a slide onto the pipe surface during the manipulation phase; this is, the violation of a guiding obstacle. Similarly, for the chimney repairing task, Fig. 4.8 illustrates a collision of the right end effector with the chimney surface as well as the violation of the lower guiding obstacle.

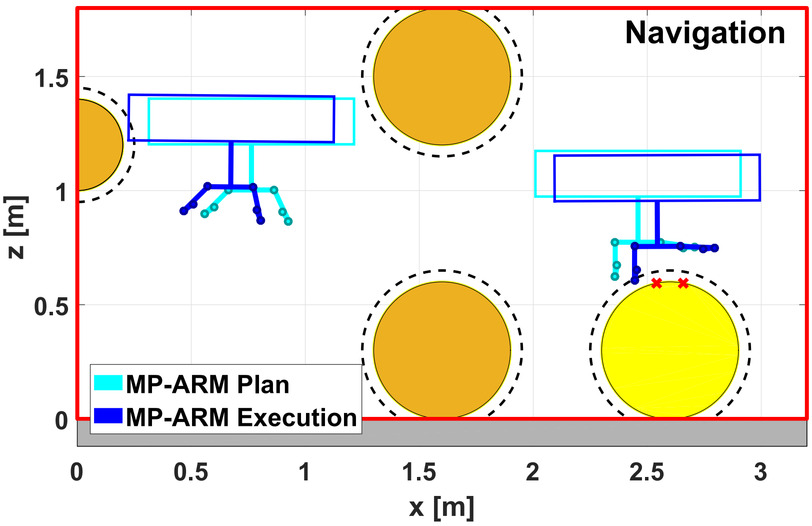

Figure 4.6: Riveting task (navigation phase). Execution (dark blue) of the trajectory planned (light blue) with the MP-ARM algorithm.

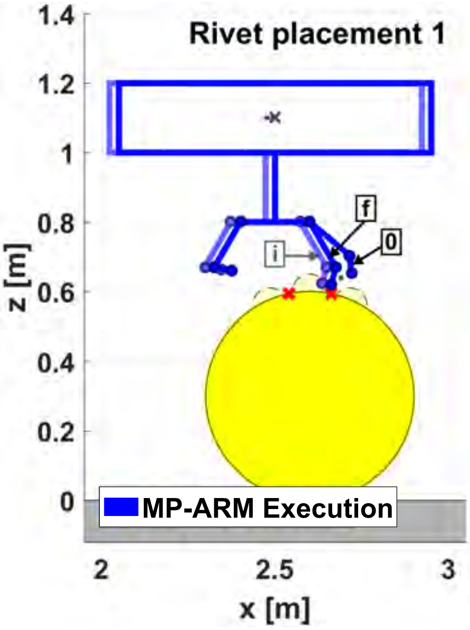

Figure 4.7: Riveting task (manipulation phase, rivet placement 1). Execution of the trajectory planned with the MP-ARM algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configuration 0 to final configuration f through shaded intermediate configuration i.

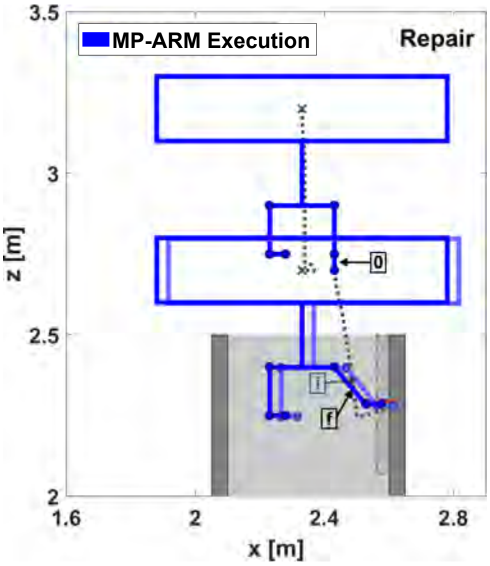

Figure 4.8: Chimney repairing task (manipulation phase, repair). Execution of the trajectory planned with the MP-ARM algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configuration 0 to final configuration f through shaded intermediate configuration i.

Analysis of the results: MP-ARM-DA algorithm

Previous results have motivated the need to consider the ARM dynamics within the motion planning problem when operating in certain cluttered environments. To solve the above, the MP-ARM-DA algorithm was proposed in Section 4.1.2. Now, the performance of such algorithm is analysed in the same simulation scenarios that were used previously with the MP-ARM algorithm. To this end, both the planned and the closed-loop trajectories have been studied again.

The results corresponding to the navigation phase of both scenarios are presented in Fig. 4.9 and Fig. 4.10 for the riveting task, whereas Fig. 4.11 and Fig. 4.12 correspond to the chimney repairing task. The trajectories followed by the ARS-LRM_v1 system are illustrated by the dotted lines representing the movements of the multirotor centre of mass from initial configurations 0 to final configurations f through intermediate configurations $i_{1,2,\ldots}$. As can be observed, the planned trajectories command safely and efficiently the controlled ARS-LRM_v1 system through the navigation phase of both scenarios.

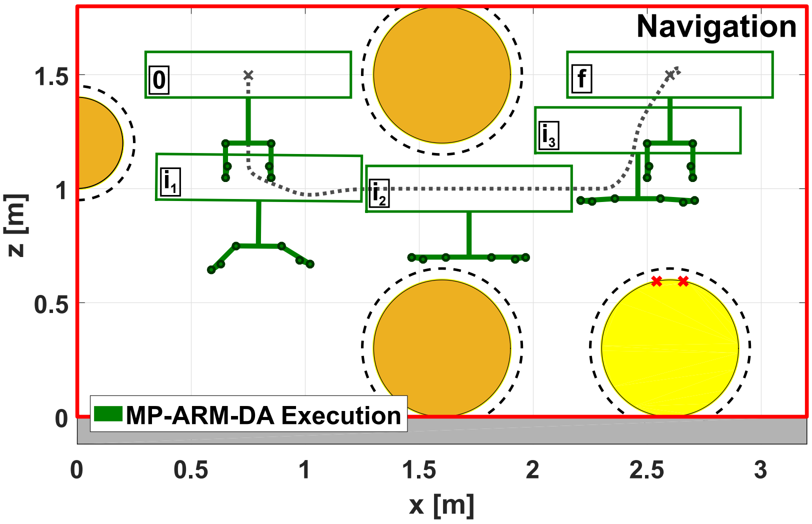

Figure 4.9: Riveting task (navigation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$.

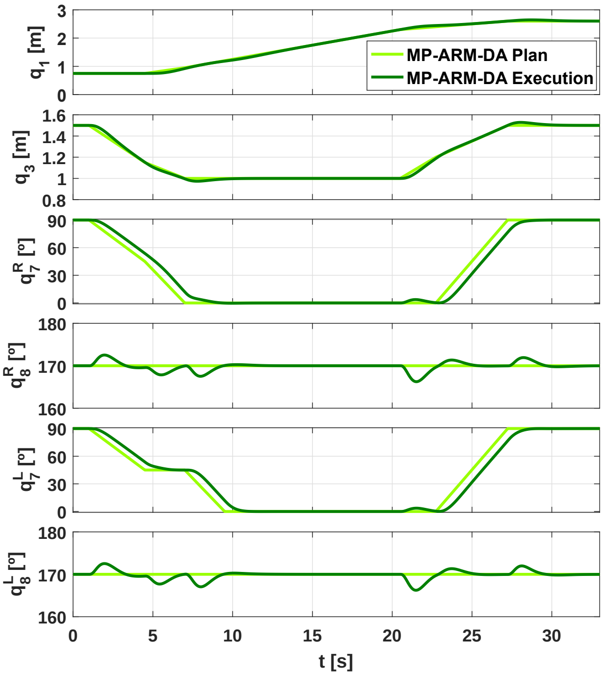

Figure 4.10: Riveting task (navigation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm.

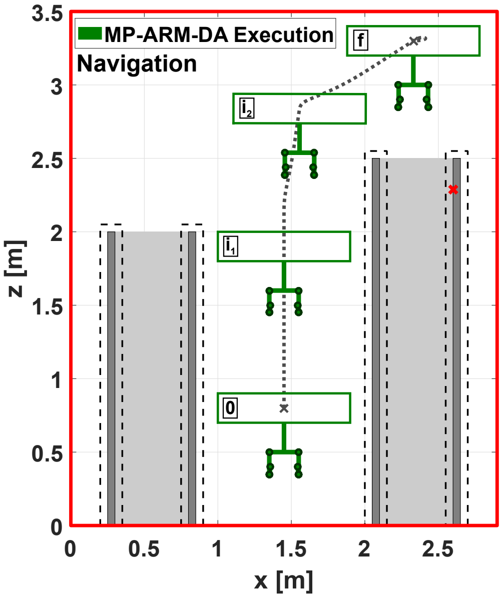

Figure 4.11: Chimney repairing task (navigation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2}$.

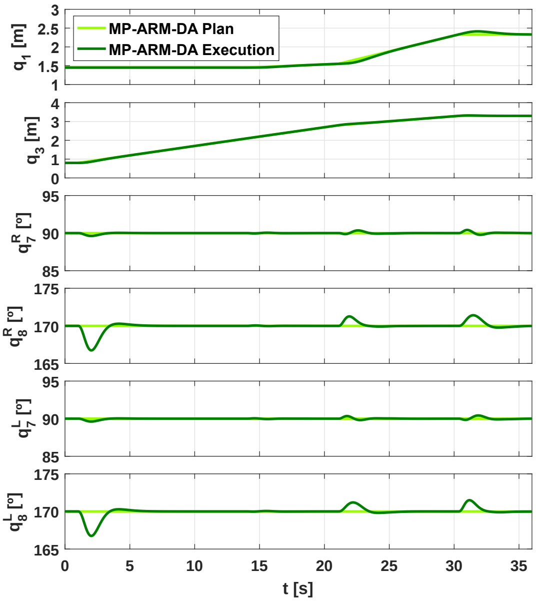

Figure 4.12: Chimney repairing task (navigation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm.

Regarding the manipulation phase, Fig. 4.13 and Fig. 4.14 present the achieved results for the riveting task. As in the navigation phase, Fig. 4.13 shows a schematic representation of the manoeuvres associated with the manipulation phase where the dotted lines represent the simulated movements of both the multirotor centre of mass and the right end effector from initial configurations 0 to final configurations f. Similarly, the evolution of the planning-space variables, for both the planned trajectory (light green line) and the closed-loop executed trajectory (dark green line), has been represented for this manipulation phase in Fig. 4.14. Again, the planned trajectory succeeds in commanding efficiently the controlled ARS-LRM_v1 system through the different manipulation manoeuvres involved in the riveting task. In contrast to the results using the MP-ARM algorithm, the guiding obstacles are not violated and hence, the right end effector executes the rivet placements without any slide onto the pipe surface. Moreover, Fig. 4.13 (right) illustrates how the joint consideration of the planning space for the multirotor and the dual arm improves the switching manoeuvre between the riveting points. More precisely, the motion planner takes advantage of the multirotor vertical displacement (see the shaded intermediate configuration i) to carry out the switching manoeuvre of the riveting effector in a more efficient way. A similar conclusion can be extracted from the rivet placements where the multirotor vertical displacements, together with the Dynamics Awareness, help to approach the riveting effector perpendicularly to the pipe surface. Turning now to the chimney repairing task, Fig. 4.15 and Fig. 4.16 show the results corresponding to the manipulation phase following the same format as for the riveting task. Once more, the planned trajectory commands the ARS-LRM_v1 system through efficient and collision-free trajectories. Also, the ARM approaches and moves away the right end effector to the target point in the chimney surface following perpendicular movements; this is, without the violation of any guiding obstacle.

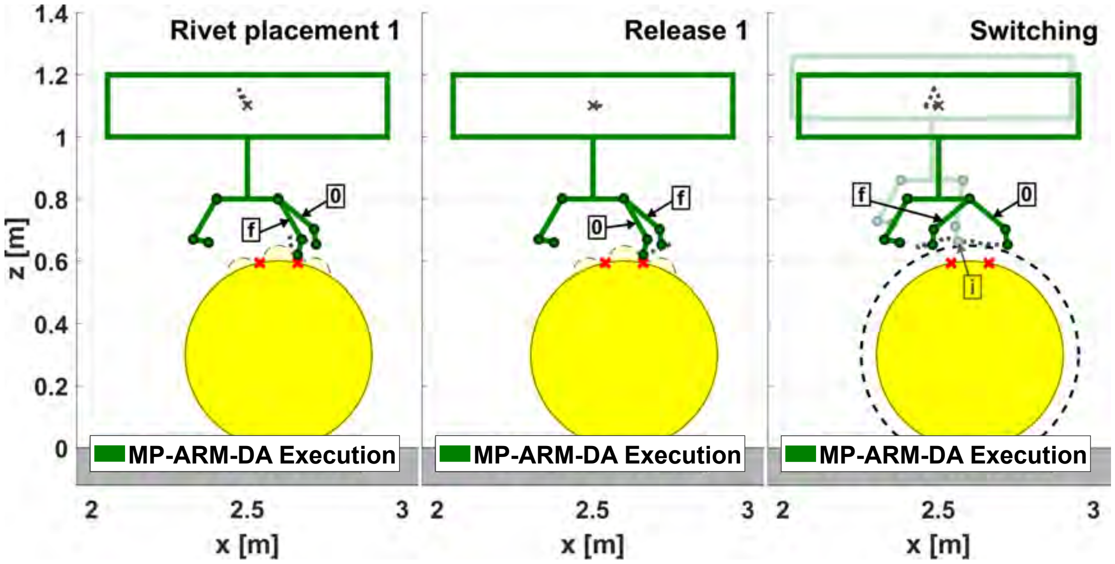

Figure 4.13: Riveting task (manipulation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configurations 0 to final configurations f through shaded intermediate configurations i.

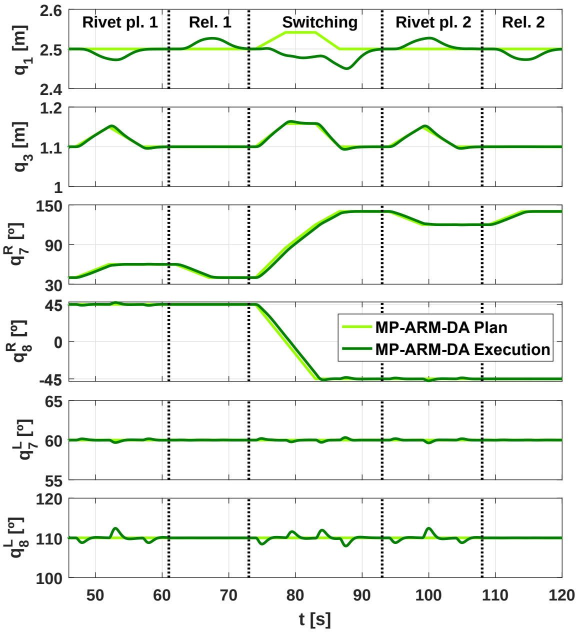

Figure 4.14: Riveting task (manipulation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm.

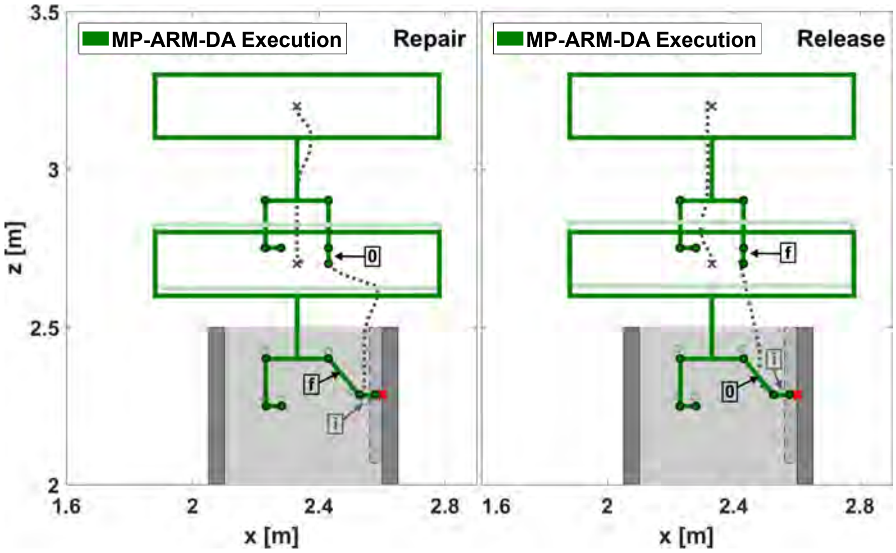

Figure 4.15: Chimney repairing task (manipulation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted lines represent the movements of both the multirotor centre of mass and the right end effector from initial configurations 0 to final configurations f through shaded intermediate configurations i.

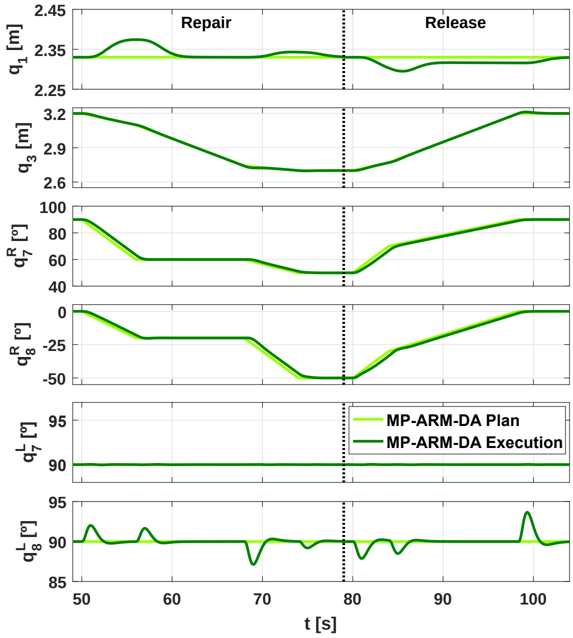

Figure 4.16: Chimney repairing task (manipulation phase). Evolution of the planning-space variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm.

4.2 Velocity adaptation

4.2.1 Motivation

The joint exploration within the planning space of the two ARM subsystems (the aerial platform and the manipulator) requires a trade-off between the desired completeness for the exploration and the associated computational cost. In this sense, the joint exploration for the complete set of motion variables (configuration variables together with their associated velocities) in ARMs implies a significant computational effort that can be alleviated if only the position variables are considered in the search tree. As it was stated in Section 2.3, when the robot dynamics is not neglected, some state-of-the-art motion planners require the exploration of the complete set of motion variables in the planning space. In contrast, previous developments in this thesis did not consider velocity variations to optimise the operation of the motion planners and hence they led to constant velocities in the resulting planned trajectories for both the aerial platform and the manipulator; these are, the reference velocities u_{ref} and w_{ref} introduced in Section 3.2.1 (Computation of the nearest node). The performance of this approach is clearly non-optimal for heterogeneously-cluttered environments. In these cases, the constant velocity profiles will end up equalling the lowest velocities required to safely avoid obstacles in the most cluttered areas. In order to maintain the velocity as an additional exploration degree, a mechanism for Velocity Adaptation (VA) should be implemented.

4.2.2 Motion planner

Due to the reasons explained before, this section presents the implementation of a new Velocity Adaptation mechanism that complements the Dynamics Awareness approach developed in Section 4.1.2. Thus, the MP-ARM-DA algorithm is transformed into the more advanced planning method MP-ARM-DAVA that incorporates Velocity Adaptation at branch level during the whole operation of the motion planner. This Velocity Adaptation is implemented by means of an iterative process that is computationally bounded. In other words, only a discrete set of velocities are considered for each branch.

In this way, the velocity optimisation reduces the expected execution time of the planned trajectory but without neither extending the planning space nor augmenting considerably the computational burden.

As it was introduced above, the operation principle of the MP-ARM-DAVA algorithm is given by an iterative analysis of the maximum allowed velocity for each new branch in the search tree. More precisely, the velocity profile is initialised to the maximum value according to the mechanical constraints of the ARM and it is then decreased gradually when analysing the dynamics of the branch under study until a feasible behaviour is found. The fundamentals of the approach are illustrated in Fig. 4.17, where $u_{ref}^{max}, w_{ref}^{max}$ are the initial maximum velocities due to mechanical constraints for the aerial platform and the manipulator joints, respectively, and u_{ref}, w_{ref} correspond to the resulting maximum allowable velocities that guarantee safe trajectories for the branch under study.

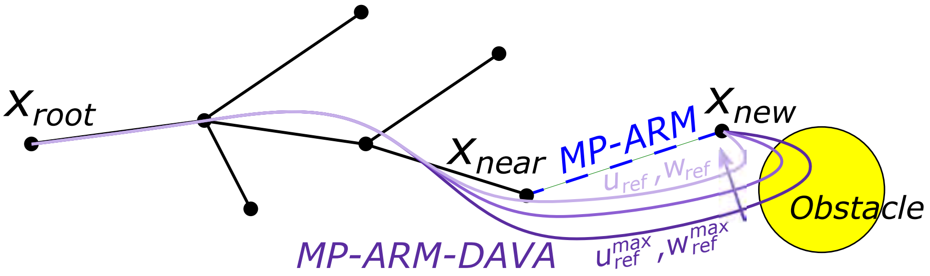

Figure 4.17: Operation basis of the MP-ARM-DAVA algorithm. $u_{ref}^{max}, w_{ref}^{max}$ are the initial maximum velocities for the aerial platform and the manipulator joints, respectively, and u_{ref}, w_{ref} correspond to the resulting maximum allowable velocities that guarantee safe trajectories for the branch under study.

Fig. 4.18 shows the detailed flow diagram corresponding to the Velocity Adaptation mechanism. The *Geometric collision* block discards unavoidable collisions because of the geometrical path of the branch under study. With this purpose, in both the $ADD(x_{nearest}, x_{near}, x_{new})$ and $REWIRE(x_{near}, x_{new})$ functions (see Algorithm 3.1), collisions are firstly checked by the $COLLISION(x_{nearest}, x_{new}, map)$ function in the way described in Section 3.2.1 (Collision checking). Then, if there are not geometric collisions, the *Dynamics collision* step analyses potential collisions motivated by the dynamical behaviour of the ARM (overshooting, ...) as described in Section 4.1.2.

In the first place, it will consider the maximum reference velocities $u_{ref}^{max}, w_{ref}^{max}$. If any collision is detected, the reference velocities u_{ref}, w_{ref} are reduced according to a scaling factor $\xi \in (0, 1)$ and the *Dynamics collision* process is triggered again until no collision appears. As it was advanced, the number of iterations is bounded with a limit of i_{max} iterations that should be properly tuned to achieve a trade-off between the optimality of the solutions and the generated computational load.

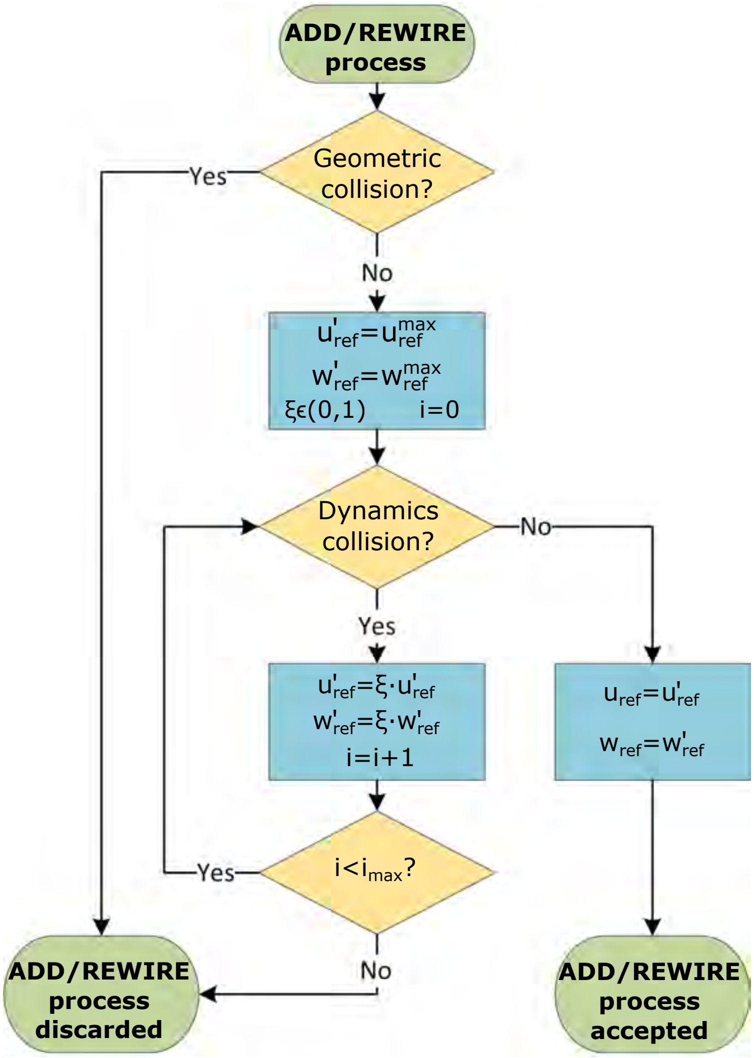

Figure 4.18: Flow diagram representing the basis operation of the MP-ARM-DAVA algorithm.

4.2.3 Simulation results

In an attempt to demonstrate the advantages of the MP-ARM-DAVA algorithm presented in previous section, the motion planning strategy has been tested in the chimney repairing task presented in Section 4.1.3 (Application scenario: chimney repairing task). During its navigation phase, the MP-ARM-DA algorithm was employed in Section 4.1.3 (Analysis of the results: MP-ARM-DA algorithm) with satisfactory performance in terms of energy efficiency and collision avoidance. However, these results can be improved as far as time efficiency is concerned. The latter makes this scenario suitable to address the manoeuvre using the MP-ARM-DAVA algorithm.

For the simulations, a parallel approach to the one described in Section 4.1.3 has been followed. This means that the ARS-LRM_v1 system has been adopted again to illustrate the results, and the same MATLAB-Simulink framework has been used. In the graphical representation of the results, the same format can be observed. While the green colour has been kept for the MP-ARM-DA results, the purple colour is selected now for the MP-ARM-DAVA algorithm (light purple for the resultant plans and dark purple for the ARM closed-loop behaviour when executing such plans). Also, the same dotted grey lines have been used to represent the complete simulated movements of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$. However, it should be emphasised now that these lines have a density of points inversely proportional to the UAV velocity.

The simulations have been organised around two main lines that endorse the need to employ the MP-ARM-DAVA algorithm to generate time-efficient trajectories in heterogeneously-cluttered environments. Firstly, the performance of the MP-ARM-DA algorithm has been evaluated, bringing to light the limitations of the method in some particular circumstances. Secondly, an analysis of the MP-ARM-DAVA performance in terms of time efficiency has been carried out.

Analysis of the results: MP-ARM-DA algorithm

For the sake of clarity, Fig. 4.19 recovers here the results presented in Section 4.1.3 (Analysis of the results: MP-ARM-DA algorithm) using the MP-ARM-DA method for the navigation phase of the chimney repairing task. As it was concluded there, this motion planner leads the ARS-LRM_v1 system to energy-efficient and collision-free trajectories thanks to both the optimisation approach and the Dynamics Awareness. More particularly, this figure shows how the planner generates an efficient trajectory with a reduced number of ARM movements while the feasibility of the executed trajectory in terms of collision avoidance is ensured.

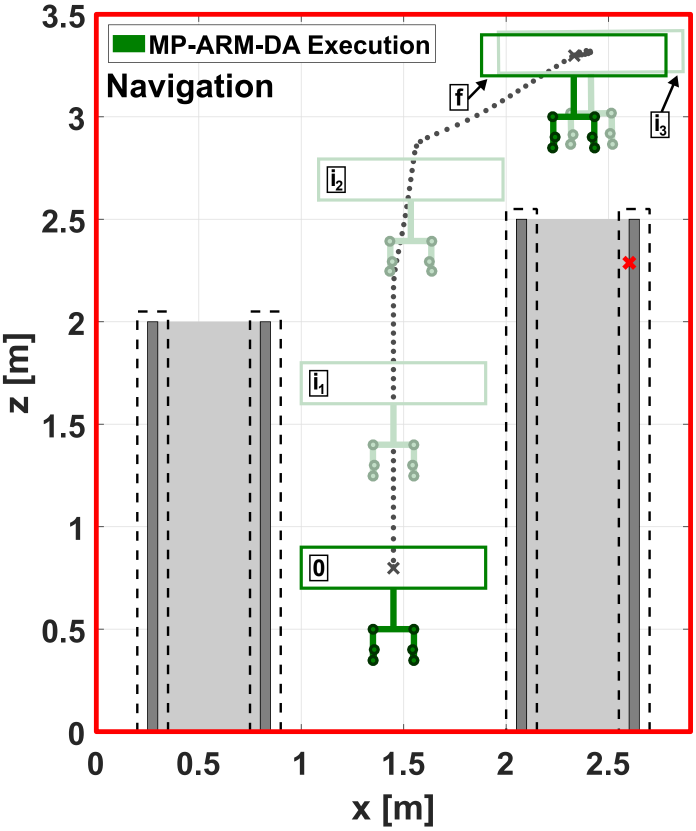

Figure 4.19: Chimney repairing task (navigation phase). Execution of the trajectory planned with the MP-ARM-DA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$ and has a density of points inversely proportional to the UAV velocity.

While previous results are satisfactory in terms of safety and energy efficiency, the time efficiency of the trajectory still admits some improvements. As it was introduced in Section 4.2.1, the application of the MP-ARM-DA method leads to constant velocity profiles in the resulting planned trajectories (see Fig. 4.20). This approach does not offer time-optimal results in heterogeneously-cluttered environments since those constant velocity profiles will end up equalling the lowest velocities required to safely avoid obstacles in the most cluttered areas (the upper right part in the chimney repairing task). Therefore, while the MP-ARM-DA algorithm is properly commanding a lower velocity in the upper right area to guarantee safe operation, the algorithm is not seizing the opportunity to impose higher velocities in the area between the two chimneys taking advantage of the absence of constraints in this region.

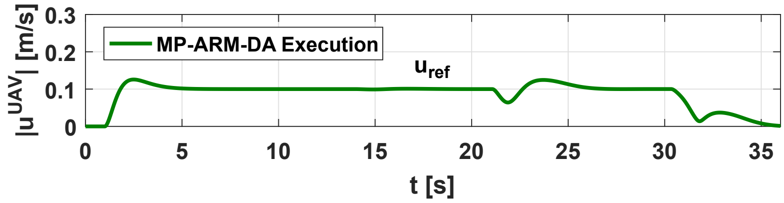

Figure 4.20: Chimney repairing task (navigation phase). Evolution of the UAV velocity: execution of the trajectory planned with the MP-ARM-DA algorithm.

Analysis of the results: MP-ARM-DAVA algorithm

Previous results have motivated the need to consider Velocity Adaptation within the motion planning problem when operating in heterogeneously-cluttered environments. In order to deal with this, the MP-ARM-DAVA algorithm has been applied.

Fig. 4.21 shows the simulation results in the same manner as Fig. 4.19 did for the MP-ARM-DA algorithm, while Fig. 4.22 depicts the evolution along time of the configuration variables within the planning space. In view of these results, all the advantages of the MP-ARM-DA algorithm have been maintained. Moreover, with the MP-ARM-DAVA algorithm it is possible now to reach the goal state in $12s$ less (a reduction of 33% in time). To better illustrate this reduction in the execution time, Fig. 4.23 shows the evolution of the UAV velocity. According to the figure, the ARS-LRM_v1 system navigates at maximum reference velocity $u_{ref}^{max} = 0.2m/s$ except when a reduction of the velocity is required, for instance, to avoid a collision with the right part of the red operation limit.

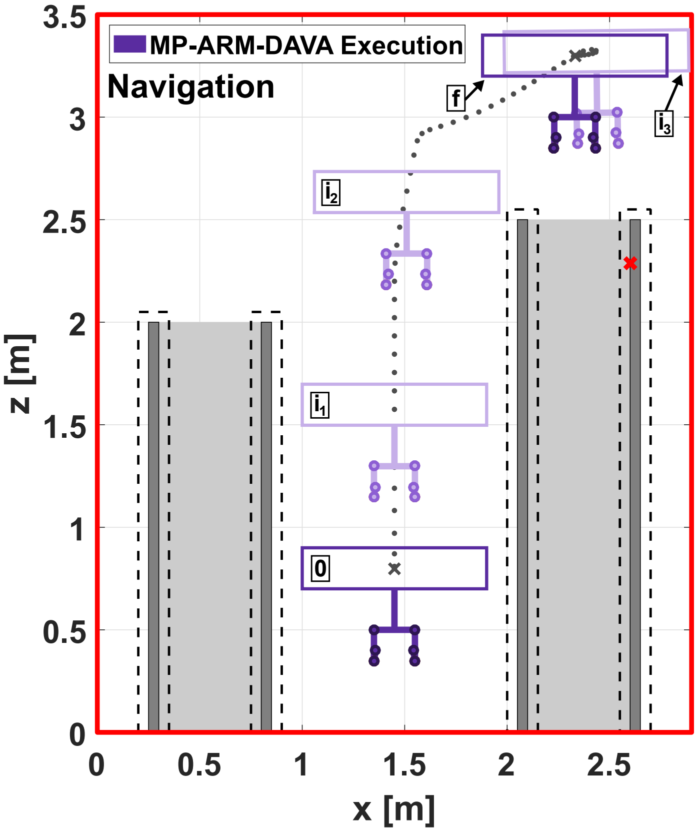

Figure 4.21: Chimney repairing task (navigation phase). Execution of the trajectory planned with the MP-ARM-DAVA algorithm. The dotted line represents the movement of the multirotor centre of mass from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$ and has a density of points inversely proportional to the UAV velocity.

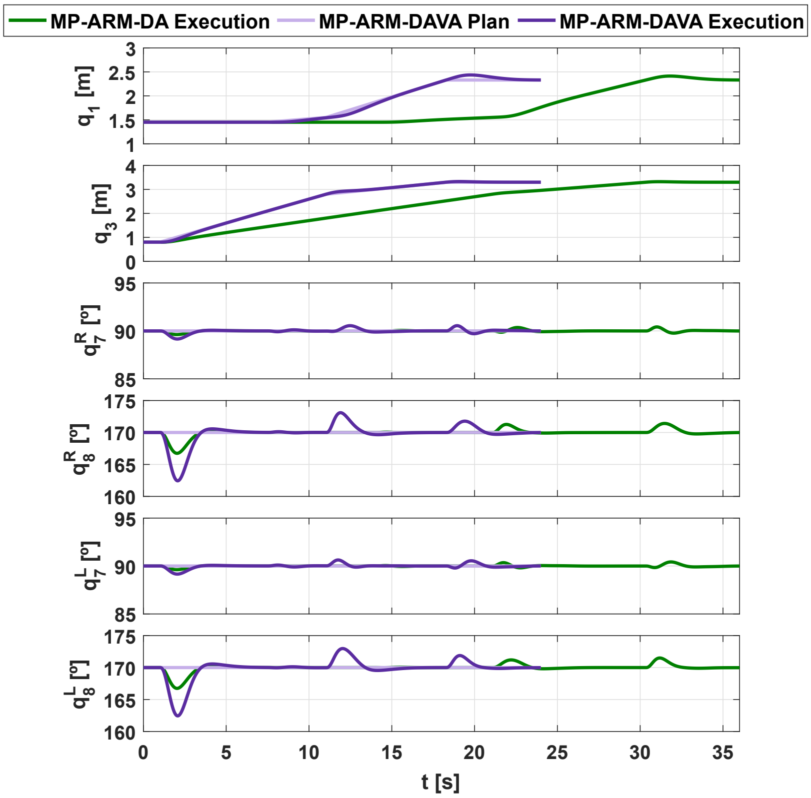

Figure 4.22: Chimney repairing task (navigation phase). Evolution of the planning-space variables: trajectories planned (light purple) and executed (dark purple) using the MP-ARM-DAVA algorithm. For comparison purposes, the execution of the trajectory planned with the MP-ARM-DA algorithm (dark green) is also represented.

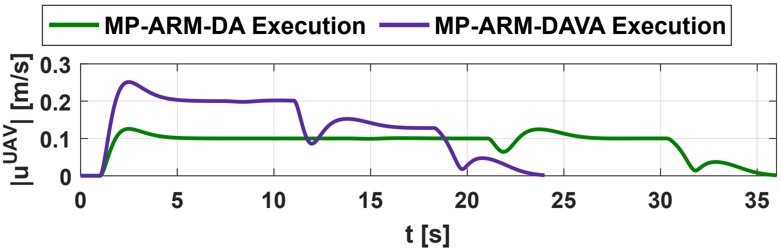

Figure 4.23: Chimney repairing task (navigation phase). Evolution of the UAV velocity: execution of the trajectory planned with the MP-ARM-DAVA algorithm (dark purple). For comparison purposes, the execution of the trajectory planned with the MP-ARM-DA algorithm (dark green) is also represented.

4.3 Aerodynamics awareness

4.3.1 Motivation

When ARMs moves autonomously in cluttered environments, the Dynamics Awareness becomes an essential feature in motion planning for safe operation. However, this mechanism might not be enough for robust obstacle avoidance in some particular scenarios. 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 such as the ground effect or the ceiling effect that can provoke destabilisations or even crashes. As an illustrative example of the potential risks of such aerodynamic effects, Fig. 4.24 shows the difficulties of flying under the aerodynamic phenomena associated with a cluttered scenario. As can be seen, the UAV tries to follow the straight trajectory in dashed light grey that has been generated using a standard motion planner to go through the two dark grey elements considered in the figure. However, due to the ground effect appearing when the system flies close to the lower dark grey surface, an undesirable ascent takes place and the vehicle collides with the upper dark grey element.

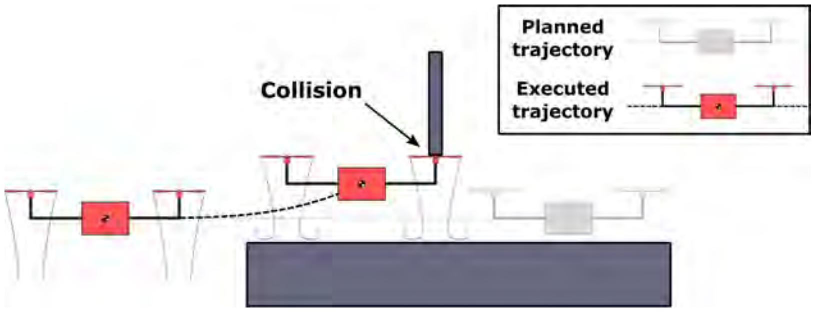

Figure 4.24: UAV trying to track a straight planned trajectory between two obstacles. The vehicle crashes with the upper obstacle as a consequence of an undesirable ascent caused by the ground effect.

In order to avoid these unsafe flight conditions, the compensation of aerodynamic effects should be considered in the design of the controller [65] or in the trajectory generation process of the motion planner, which is known here as Aerodynamics Awareness (ADA). This thesis will adopt the second approach in subsequent sections.

4.3.2 Motion planner

This section presents the implementation of a novel motion planner with Aerodynamics Awareness for robust operation close to surfaces involving aerodynamic effects. The resultant algorithm, referred to as MP-ARM-ADA, extends the capabilities of the Dynamics Awareness approach developed in Section 4.1.2.

As a means to guide the presentation of the MP-ARM-ADA approach, the third version of the set of new Aerial Robotic Systems for Long-Reach Manipulation, called ARS-LRM_v3 and presented in Annex A.3, has been used. Nevertheless, the applicability of the new algorithm extends beyond this specific ARM configuration.

Similarly 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 ARM. However, in this new approach, not only the system dynamics has been considered for the expansion of the search tree, but also its aerodynamics. Fig. 4.25 (left) schematises the operation basis of the MP-ARM-ADA algorithm. 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 ARM would not be able to compensate for the ceiling effect and might provoke a collision. Additionally, Fig. 4.25 (right) depicts the complete closed-loop scheme required to perform the simulations that give support to the Aerodynamics Awareness concept (see Annex A.3 to understand the nomenclature presented in this section, as well as the modelling and control associated with the blocks *Dynamics* and *Controller*). 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 (force $F_{3,OAE}$ and torque $T_{2,OAE}$) in terms of the corresponding PWM (Pulse Width Modulation) signals and whose output is the force and torque that actually govern the ARM movement as a consequence of aerodynamic effects ($F_{3,IAE}, T_{2,IAE}$).

In order to derive the block *Aerodynamics* introduced in Fig. 4.25 (right), the force and torque commanded by the controller ($F_{3,OAE}, T_{2,OAE}$) must be broken down into the contribution of each separate rotor. This will pave the way for the application of the equations presented in Annex B to characterise the aerodynamic effects of an

Figure 4.25: Operation basis of the MP-ARM-ADA algorithm (left) and closed-loop scheme required to perform the simulations that give support to it (right). The terms highlighted in red correspond to the variables employed to model the Aerodynamics Awareness.

individual rotor. The aforementioned decomposition can be implemented as follows:

$$\begin{aligned} F_{3,i} &= F_{3,i}^L + F_{3,i}^R \\ T_{2,i} &= d_1 \left(F_{3,i}^L - F_{3,i}^R \right) \qquad (i = IAE, OAE) \end{aligned} \tag{4.1}$$

where superscripts L, R make reference to left and right rotors in Fig. A.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:

$$\begin{aligned} q_1^L &= q_1 - d_1 cos\left(q_5\right) + d_3 sin\left(q_5\right) \qquad & q_1^R &= q_1 + d_1 cos\left(q_5\right) + d_3 sin\left(q_5\right) \\ q_3^L &= q_3 + d_1 sin\left(q_5\right) + d_3 cos\left(q_5\right) \qquad & q_3^R &= q_3 - d_1 sin\left(q_5\right) + d_3 cos\left(q_5\right) \end{aligned} \tag{4.2}$$

Using these positions as input in maps of aerodynamic effects computed according to Annex B, the aerodynamic ratio τ between the actual thrust governing the platform movement (IAE) and the thrust commanded by the controller (OAE) can be obtained. Thus, the lifting force of each rotor in presence of aerodynamic effects is given by:

$$F_{3,IAE}^j = \tau\left(q_1^j, q_3^j\right) F_{3,OAE}^j \qquad (j = L, R) \tag{4.3}$$

The combined use of equations (4.1) and (4.3) allows the final calculation of the total force and torque $F_{3,IAE}, T_{2,IAE}$ that are really exerted upon the aerial platform.

4.3.3 Simulation results

There are several applications that can benefit from motion planning with Aerodynamics Awareness. This section proposes the realistic application of bridge inspection as use case to evaluate the simulations corresponding to the developed MP-ARM-ADA motion planner.

In that context of bridge inspection, the usage of the ARS-LRM_v3 system presented in Annex A.3 is suggested to avoid human risks and minimise operational costs. The particular configuration of its manipulator makes this ARM a very convenient solution for such task. Additionally, since the ARS-LRM_v3 system will have to operate close to the bridge deck, aerodynamic phenomena such as the ceiling effect will play an important role. Consequently, the combination of bridge inspection and the ARS-LRM_v3 system will provide an excellent test bench to analyse the benefits associated with the addition of Aerodynamics Awareness within the trajectory generation process. For the ARS-LRM_v3 system, the configuration variables that have been selected to define the planning space correspond to the green variables in Fig. A.13 with the exception of the pitch angle q_5 of the aerial platform, which is considered negligible for planning purposes as in previous validations.

For the simulations, a parallel approach to the one described in Section 4.1.3 has been followed once more. This means that the same MATLAB-Simulink framework has been used and a similar format can be observed for the graphical representation of the results. While the blue and green colours have been kept for the MP-ARM and MP-ARM-DA results, respectively, the orange colour is selected now for the MP-ARM-ADA algorithm (light orange for the resultant plans and dark orange for the ARM closed-loop behaviour when executing such plans). Also, dashed and dotted lines have been used to represent, respectively, the complete movements of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$.

The validation analysis has been structured around two main phases. In a first step, the performance of the MP-ARM and MP-ARM-DA algorithms has been evaluated. Both have revealed their limitations in the inspection scenario under consideration because of their lack of Aerodynamics Awareness. In a second step, a complete analysis of the MP-ARM-ADA performance has also been addressed. The analysis of these results allows concluding that Aerodynamics Awareness is required to generate efficient and safe trajectories when ARMs operate close to elements that can modify significantly the rotor airflow.

Application scenario: bridge inspection

With the means currently available, the I&M task of crack detection in reinforced concrete bridges like the one shown in Fig. 4.26 requires visual inspection by qualified human operators. This inspection usually demands the usage of scaffolding or cranes in hard-to-reach and high-altitude locations, which endangers human lives. Consequently, the application of solutions based on aerial robotic manipulation is a desirable option.

Figure 4.26: Reinforced concrete bridge. Image extracted from Google Maps [1].

In order to illustrate the use case under consideration, the environment presented in Fig. 4.27 (left) has been adopted. In this scenario, the ARS-LRM_v3 system has to perform a visual inspection of the junction point between the deck and one pillar of the bridge (marked with a red point in the figure). To this end, a visual camera will be integrated as end effector in the long-reach manipulator of the ARM and the inspection plan will be the following. The robot will start the operation from the top part of the bridge and will navigate around the deck until reaching a position where the inspection point is in the field of view of the end effector. This operation is schematised in Fig. 4.27 (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.

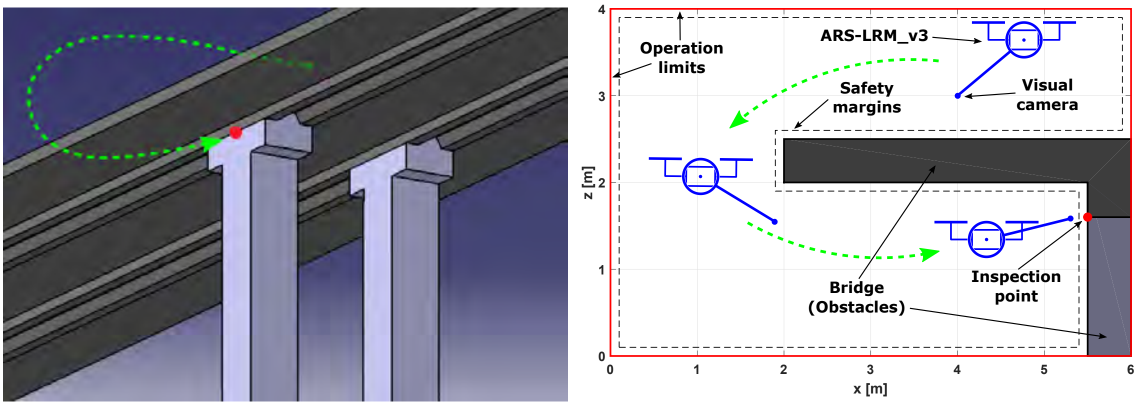

Figure 4.27: Application scenario given by a bridge inspection.

The aerodynamic challenges of this scenario arise because the most efficient trajectories for the inspection task under consideration demand that the ARS-LRM_v3 system flies close to the bridge surfaces. The latter makes this scenario suitable to establish the 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 Annex B has been followed to compute the 3D map of aerodynamic effects corresponding to the application scenario proposed here (see Fig. 4.28).

Analysis of the results: MP-ARM algorithm

Fig. 4.29 and Fig. 4.30 show the closed-loop simulation results when the controlled ARS-LRM_v3 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 ARM

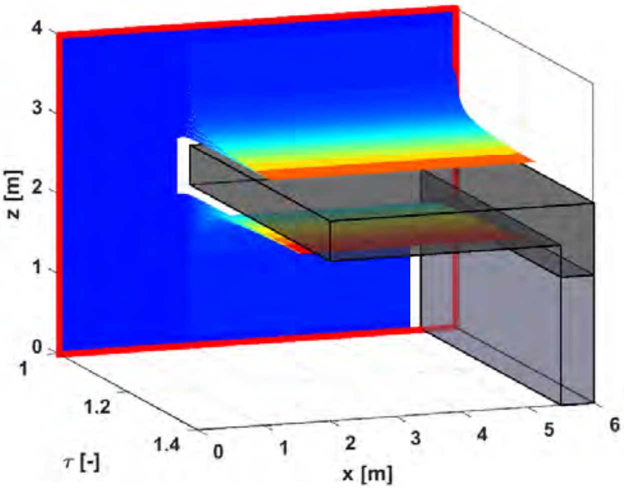

Figure 4.28: 3D map of aerodynamic effects for the bridge scenario.

system is commanded to navigate close to the bottom surface of the bridge deck (the shortest trajectory towards the final configuration f shown in Fig. 4.29), this implies flying in presence of the associated ceiling effect. The latter produces disturbances that the controller is not able to reject. Consequently, the robot becomes unstable until the rear rotors finally collide with the bridge.

Analysis of the results: MP-ARM-DA algorithm

Fig. 4.31 and Fig. 4.32 depict the simulation results when the controlled ARS-LRM_v3 system (dark green) tracks the trajectory generated by the MP-ARM-DA motion planner (light green). Although in this case the planner takes into account the system dynamics within the planning process, the disturbances associated with the ceiling effect lead again the ARM to collide with the bridge deck 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 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 scenario since the collision eventually takes place.

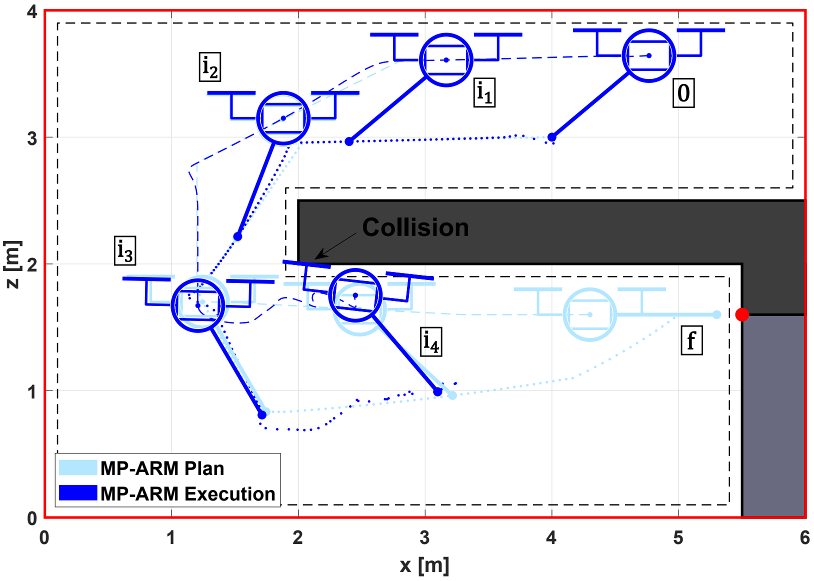

Figure 4.29: Bridge inspection. Execution (dark blue) of the trajectory planned (light blue) with the MP-ARM algorithm. The dashed and dotted lines represent, respectively, the movement of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$.

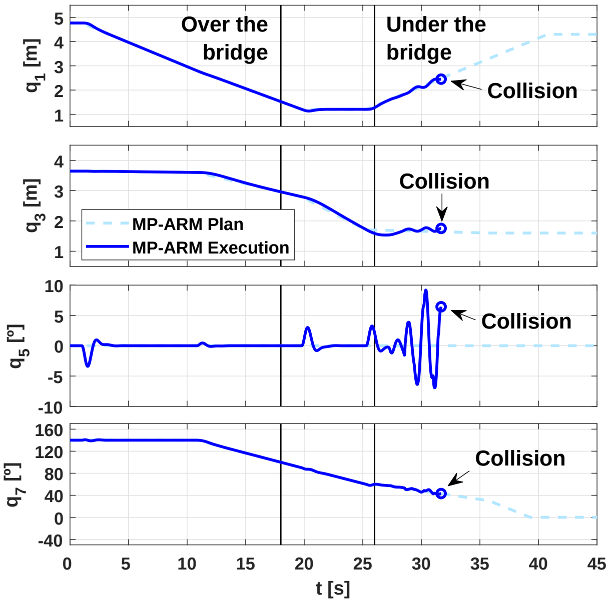

Figure 4.30: Bridge inspection. Evolution of the configuration variables: trajectories planned (light blue) and executed (dark blue) using the MP-ARM algorithm.

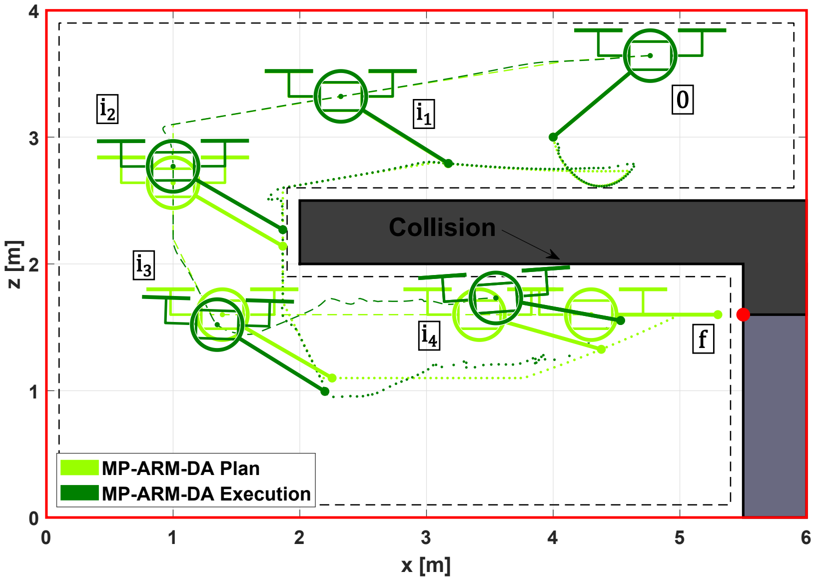

Figure 4.31: Bridge inspection. Execution (dark green) of the trajectory planned (light green) with the MP-ARM-DA algorithm. The dashed and dotted lines represent, respectively, the movement of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$.

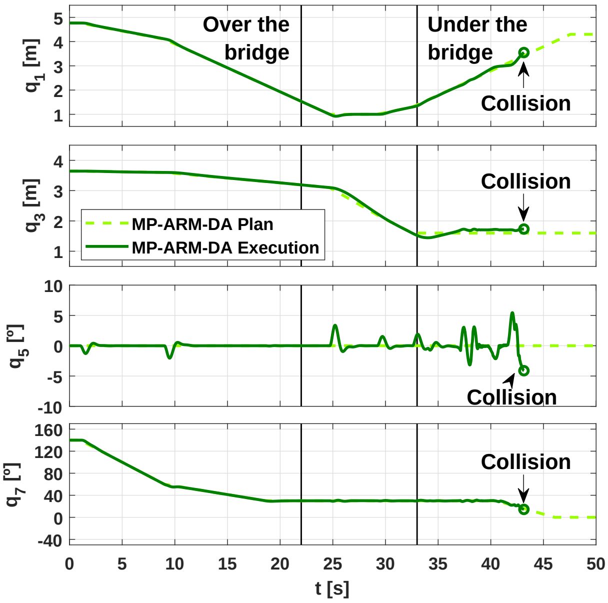

Figure 4.32: Bridge inspection. Evolution of the configuration variables: trajectories planned (light green) and executed (dark green) using the MP-ARM-DA algorithm.

Analysis of the results: MP-ARM-ADA algorithm

In order to conclude with the simulation sequence, Fig. 4.33 and Fig. 4.34 show the results when the trajectory generated by the MP-ARM-ADA motion planner (light orange) is commanded to the controlled ARS-LRM_v3 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 bridge. 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_v3 system. Thus, the aerial platform reaches the inspection point from positions that maintain a wider safety margin with respect to the elements causing the aerodynamic phenomena.

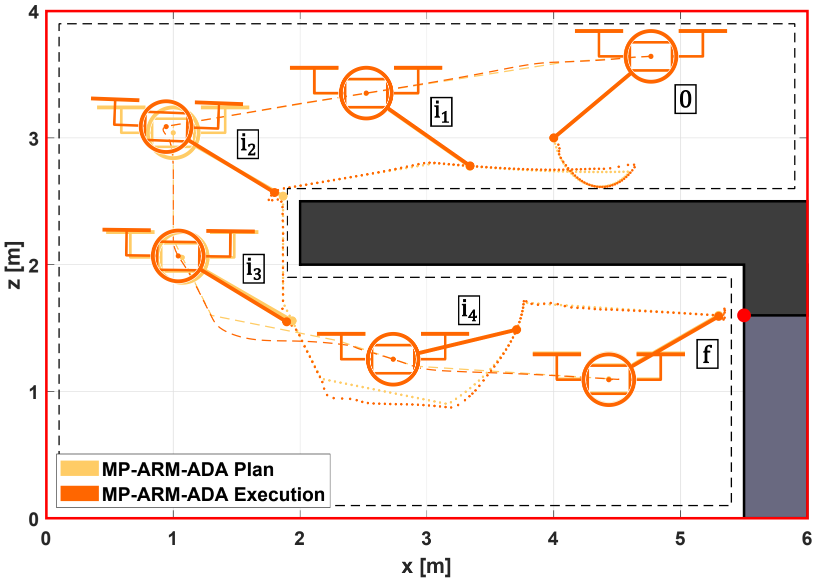

Figure 4.33: Bridge inspection. Execution (dark orange) of the trajectory planned (light orange) with the MP-ARM-ADA algorithm. The dashed and dotted lines represent, respectively, the movement of both the multirotor centre of mass and the end effector from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3,4}$.

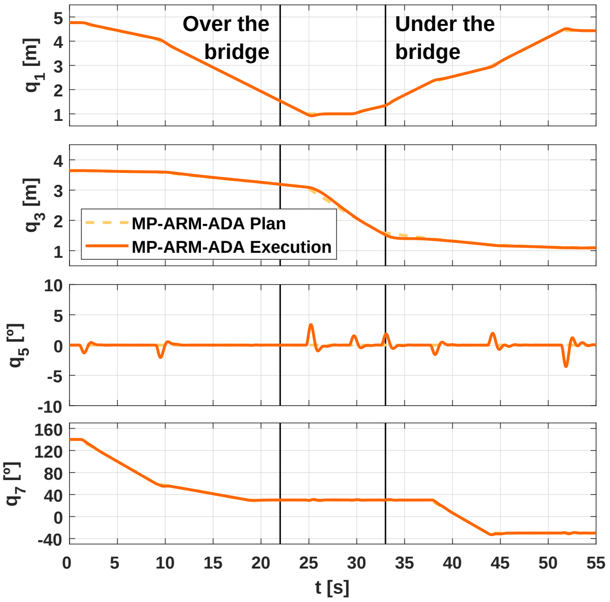

Figure 4.34: Bridge inspection. Evolution of the configuration variables: trajectories planned (light orange) and executed (dark orange) using the MP-ARM-ADA algorithm.

4.4 Conclusions

The MP-ARM motion planner demonstrated in Chapter 3 good results in the computation of safe and efficient trajectories for ARMs endowed with robotic arms in cluttered environments. Although this conclusion is valid for many industrial I&M tasks in which ARMs are involved, this chapter has shown that there are some particular circumstances that constrain the execution of its planned trajectories. To cope with that, three extensions have been proposed.

Firstly, the MP-ARM limitations due to the absence of Dynamics Awareness have been solved with the MP-ARM-DA motion planner. Simulation results allow concluding that the enhanced method offers energy efficiency and robust obstacle avoidance for both navigation in highly cluttered environments where the obstacle proximity narrows considerably the safe operation areas, and manipulation tasks that require high accuracy.

Secondly, a Velocity Adaptation mechanism has been added over the basis of the MP-ARM-DA algorithm. Thanks to this, the resulting MP-ARM-DAVA planner allows a reduction of the execution time for trajectories planned in heterogeneously-cluttered environments (reduction of 33% for the evaluated application) while keeping the computation time in similar values. The features of robust safety and energy efficiency are also preserved.

Finally, the MP-ARM-DA algorithm has also been improved by including Aerodynamics Awareness within the planning operation. According to simulation results, the novel MP-ARM-ADA approach becomes essential for the motion planning of safe and energy-efficient trajectories when ARMs have to operate close to large surfaces that can affect the rotor airflow. More in detail, the algorithm discards problematic states influenced by aerodynamic phenomena such as the ground effect or the ceiling effect and explores alternatives that lead to the most efficient trajectories within the area of safe operation.

Chapter 5

Hybrid motion planning for inspection with aerial-ground robots

> Raíces para ser fuertes y alas para poder volar.
>
> Marciano Caballero

5.1 Motivation

Aligned with statements in previous chapters, the high costs associated with I&M tasks in industrial scenarios, which are usually performed in hard-to-reach locations and in hazardous conditions for human operators, transform this kind of tasks in a suitable target for the exploitation of ARM capabilities. However, the real needs to perform I&M tasks effectively sometimes demand higher levels of efficiency and accuracy to conventional ARMs. Giving response to these requirements, the use of ARMs with hybrid air and ground locomotion plays an important role. In this way, a Hybrid Robot (HR) can easily fly towards the operation area, land on it, and address the I&M task in the target point without wasting energy in the propellers and taking advantage of the enhanced accuracy of fixed-base operations. After that, the HR can fly again to reach the next point or roll on the surface if the point is accessible by ground. This hybrid approach has been considered in the HYFLIERS project [11], where the work presented in this chapter is framed.

From the point of view of an operator, an HR offers some unique advantages but its teleoperation can become complex, especially in cluttered industrial environment like oil and gas refineries or chemical plants (see Fig. 5.1) where many pipe arrays and other obstacles can complicate the navigation. To cope with this, the HR needs to be equipped with a motion planning system that leads to a high level of autonomy. This system should take into account both the hybrid locomotion capabilities of the HR and the configuration of the environment in order to plan efficient trajectories that steer the robot while ensuring safety. Moreover, as it was demonstrated in Section 4.1, the robot dynamics should not be neglected for robust collision avoidance. Differences between planned and executed trajectories can produce collisions with obstacles, which is especially critical in cluttered scenarios. Finally, the system should also provide hybrid reactivity to face unexpected obstacles that may restrict locally the HR to advance when it is rolling according to the execution of a plan. Thus, the HR could switch between ground and air to avoid such obstacle. All these features would allow the operator to focus only on the I&M task itself, alleviating the workload.

Figure 5.1: Cluttered industrial environment with many pipe arrays.

5.2 Motion planner

Motivated by the reasoning explained in previous section, a new motion planning approach has been derived. The method is intended to exploit hybrid locomotion in ARMs for autonomous inspection tasks in industrial environments like oil and gas refineries or chemical plants, where there are many pipe arrays. In this sense, the final objective is the generation of motion plans that allow an HR to inspect a set of measurement points selected by an operator and located on several pipe arrays in a safe and efficient manner. For that purpose, the robot can navigate through the environment either flying or rolling on the pipes.

Although the planning strategies included in this chapter will consider pipes like the only potential means to move through the scenario by rolling, these strategies are based on general design principles and hence can be adapted to other surfaces for ground locomotion. For the same reason, they can also be applied to different HRs. In this section, the Hybrid Robotic Manipulator (HRM) described in Annex A.4 has been used to guide the presentation of the proposed method.

The rest of this section presents the fundamentals of the planner operation (Section 5.2.1), an extension of the method based on Dynamics Awareness to guarantee safer trajectories (Section 5.2.2) and the planner application for hybrid replanning in case unexpected obstacles are detected while rolling during the execution of a plan (Section 5.2.3).

5.2.1 Fundamentals of planner operation

In order to generate trajectories that steer an HR to perform its operation safely and efficiently, a novel motion planning methodology that takes advantage of the flying and rolling capabilities offered by the system has been developed.

Given a map of the environment and a complete sequence of inspection points located on several pipe arrays, the developed planning method computes the trajectory that allows autonomous hybrid navigation through the inspection targets to complete the mission. Afterwards, the resultant trajectory will be given to the HR as control reference.

Moreover, the algorithm brings the possibility to optimise metrics in the generated trajectories like operation time or energy consumption. This is a remarkable feature since it allows better adaptation of the plans to specific missions. For instance, urgent inspections motivated by emergency situations would benefit from optimising the time operation, whereas routine inspections could better fit with energy optimisation in order to maximise the coverage.

Inspired by the MP-ARM algorithm presented in Section 3.2.1, the developed motion planner, hereinafter referred to as MP-HR (Motion Planner for Hybrid Robots), is built over the basis of the RRT* algorithm [47]. However, its algorithmic modules have been completely adapted to efficiently integrate the hybrid nature of HRs into the motion planning technique. The main novelties are detailed below.

Hybrid planning space

The exploitation of the hybrid capabilities of HRs requires the joint consideration of aerial and pipe states within the planner operation. Fig. 5.2 illustrates this new hybrid paradigm during the expansion of the search tree associated with the MP-HR method. As can be seen, the pipe states are restricted to the upper part of the pipes conforming the arrays since this is the location from which the HR should operate, either rolling along the pipe arrays or inspecting them. Moreover, as it is justified in Annex A.4.1, these pipe states are centred on the arrays because this increases the stability and accuracy of the HR operation.

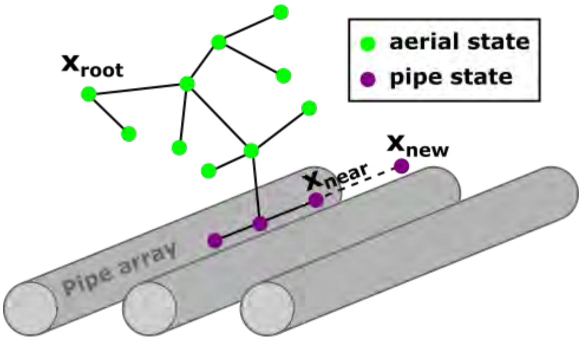

Figure 5.2: Scheme with both aerial and pipe states during the hybrid expansion of the search tree associated with the MP-HR method.

The variables considered within this hybrid planning space, whether in aerial states or in pipe states, include the robot position $\mathbf{p}$ (vector of coordinates x, y, z) in an Earth-fixed frame as well as the heading orientation (yaw angle ψ) with respect to the same frame. The exploration of these variables can be decoupled in order to minimise the computational load. The associated benefit is that the reactive capabilities presented in Section 5.2.3 will perform better with a lower processing time. With that purpose, the geometry of the HR is approximated by its bounding sphere. Under this assumption, collision checking will be irrespective of yaw angle and hence the planning space can be indeed divided into two different subspaces (position coordinates x, y, z and yaw angle ψ) that will be explored sequentially. The global approach followed to integrate these sequential explorations will guarantee in any case the consistence of the global exploration; that is, the fulfilment of the position and yaw orientation associated with each inspection point in the generated trajectory.

The specific characteristics of the exploration of the position subspace are introduced just below. After that, the following part is devoted to the particularities of the yaw exploration, including the procedure to bring into line the trajectory in the yaw subspace with that previously generated for the position subspace, at the same time that the yaw orientation associated with each inspection point is guaranteed.

Position subspace

In standard RRT*-based algorithms, the exploration of the planning space is based on uniform sampling. However, this approach is not suitable for the hybrid planning space considered here since the ratio between air and pipe volumes is very prone to be unbalanced. This situation could provoke that pipe volumes are not sufficiently explored and, as a result, that the hybrid capabilities of the HR are not fully exploited. Motivated by this, a smart process of discretisation and sampling has been proposed to tackle the problem.

The discretisation provides a two-fold benefit. Firstly, it eases the smart sampling of the hybrid planning space that will be described later to counteract the air-pipe unbalance. Secondly, it contributes to bound the computation time, as it was presented in Section 3.2.1 (Discretisation of the planning space) for the MP-ARM algorithm. This will be a valuable feature in the context of the application of the planning algorithm for reactive behaviours that is presented in Section 5.2.3. The discretisation has been performed uniformly with the same resolution δ for both the 3D aerial space and the 1D operation lines in the upper part of the pipes.

The smart sampling is the part of the process that allows adjusting the original ratio between air and pipe nodes that is typically very unbalanced. The corrective parameter μ will be in charge of this purpose. It will be introduced to steer the sampling process in such a way that the ratio between air and pipe nodes is transformed into the new ratio $\lambda \in (0, 1)$:

$$\lambda = \mu \frac{n_{pipe}}{n_{pipe} + n_{air}} \tag{5.1}$$

Fig. 5.3 illustrates previous concepts in an example scenario. As can be seen, when $\mu = 1$, the sampling of aerial states and pipe states is uniform (as in the standard approach) and the illustrative sample shown in the figure reveals the unbalanced condition anticipated before (only 1 out of 21 samples corresponds to pipes). In contrast, when the sampling is steered by a correcting parameter $\mu = \mu^*$, whose selection takes into account the density of pipe arrays, the resultant sample is more balanced (3 out of 21 samples correspond to pipes) and hence the hybrid capabilities of the system are better explored.

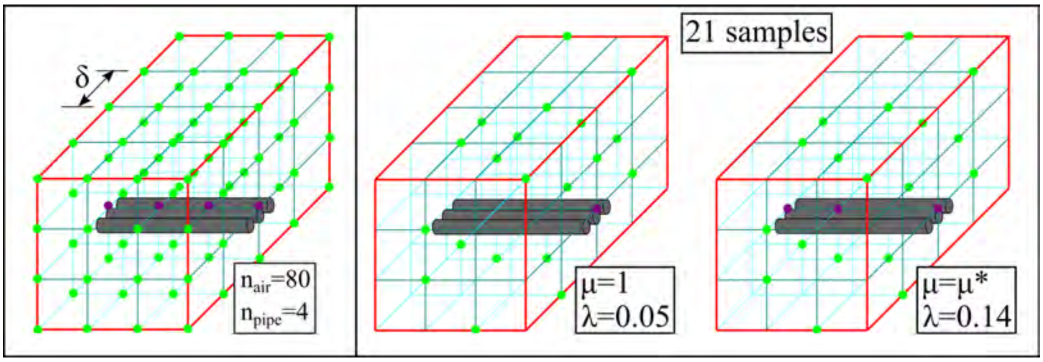

Figure 5.3: Effect of the weighting parameter μ in the process of discretisation and sampling of the hybrid planning space: the complete set of n_{pipe} pipe and n_{air} aerial states corresponding to the discretisation (left), the selected subset with uniform sampling (centre) and the selected subset corresponding to the sampling with compensating effect that results in a more balanced distribution (right).

Yaw subspace

The algorithm to derive the yaw trajectory will enforce two directives when selecting the yaw value for certain node. Firstly, the alignment with the displacement directions that can be inferred from the position trajectory generated for the HR. Secondly, the alignment with the orientation of the pipes where the HR has to land and operate. As will be seen later, these objectives might come into conflict with each other and the algorithm should provide mechanisms to solve these situations.

The resultant criteria to select the yaw values for each kind of node are enumerated below. They are also graphically illustrated in Fig. 5.4. As can be anticipated from this figure, the branches that join the yaw values selected for the different nodes will be built in such a way that smooth transitions at constant velocity are guaranteed.

P) In pipe nodes, the yaw angle is enforced to be aligned with the orientation of the pipe. Since two orientations fulfilling this requirement can be considered for each pipe, the final selection is based on the following considerations:

  P.1) In case that the pipe node is surrounded by two aerial nodes (landing, inspection on the landing point and finally, take-off), the selected orientation will be the one that minimises the aggregated angle variation produced along the two transitions air-pipe and pipe-air.

  P.2) In the remaining cases (another pipe node before or after), the selected orientation will be given by the direction of the rolling movement that allows the system to reach, or to arrive from, that other pipe node.

A) In aerial nodes, the final selection presents again two branches:

  A.1) When the aerial node is associated with landing transitions, the yaw angle is equal to that of the following pipe node. This allows a safer landing with constant yaw during the complete manoeuvre.

  A.2) In the remaining cases (whether in take-off transitions or in free flight), the yaw angle is aligned with the direction of the vector that connects the node under consideration with the next node in the position trajectory. This ensures that the robot is always flying in areas that have been covered by potential onboard perception sensors since they are typically forward-facing.

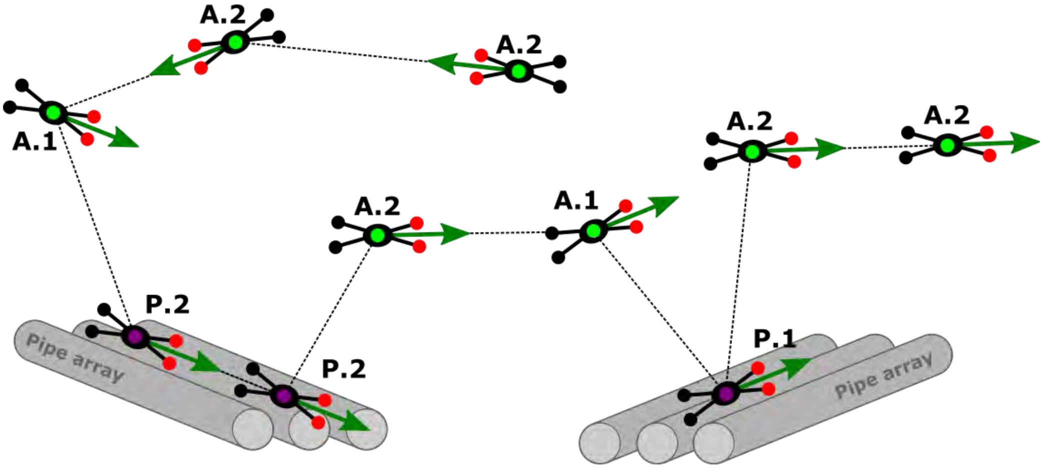

Figure 5.4: Illustrative examples of the criteria to select yaw angles for the different types of nodes in the hybrid planning space.

Optimisation capabilities in hybrid planning

For the optimisation of the HR operation at the planning level, two different cost functions that allow a better adaptation of the plans to specific missions have been defined: the operation time and the energy consumption. For instance, urgent inspections motivated by emergency situations would benefit from optimising the time operation, whereas routine inspections could better fit with energy optimisation in order to maximise the coverage.

These cost functions consider the hybrid capabilities of the robot through the associated velocities $u_{roll}, u_{fly}, u_{tran}$ and energy consumptions per unit of displacement $e_{roll}, e_{fly}, e_{tran}$ when the system is rolling, flying between aerial nodes or performing a hybrid transition (landing/take-off). Thus, the increment in cost Δc associated with a branch from node i to node j when minimising the operation time T or the energy consumption E can be defined, respectively, as:

$$\Delta c^{T}_{i,j} = \frac{\|\mathbf{p_j} - \mathbf{p_i}\|}{u_k} \qquad k = roll, fly, tran \tag{5.2}$$

$$\Delta c^{E}_{i,j} = \|\mathbf{p_j} - \mathbf{p_i}\|\, e_k \qquad k = roll, fly, tran \tag{5.3}$$

where k denotes the manoeuvre associated with the branch: rolling (roll), flying (fly) or hybrid transition (tran) either for landing or take-off. Since a normal operation complies with $u_{fly} > u_{roll}$ and $e_{roll} < e_{fly}$, the planner will prioritise flying branches when minimising operation time $\Delta c^{T}_{i,j}$ and rolling branches for energy consumption $\Delta c^{E}_{i,j}$. These basic behaviours are illustrated in Fig. 5.5. The derivation of an approximate estimation for the energy consumptions per unit of displacement that are associated with these basic behaviours; these are, e_{roll} and e_{fly}, is addressed below.

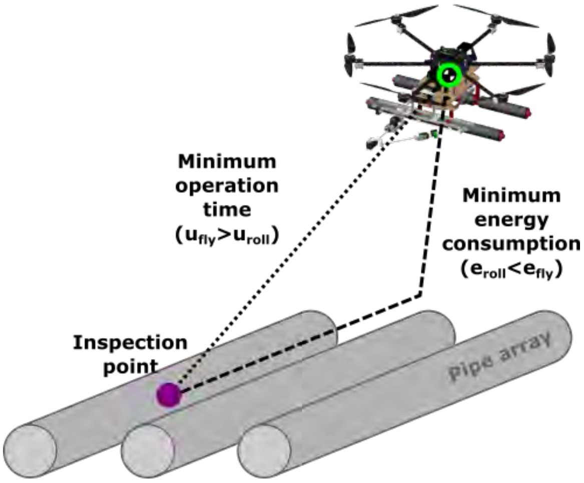

Figure 5.5: Basic behaviours corresponding to the optimisation of the proposed cost functions. Flying is prioritised over rolling when minimising the operation time T ($u_{fly} > u_{roll}$), while rolling is prioritised over flying when minimising the energy consumption E ($e_{roll} < e_{fly}$).

Additionally, a smoothing technique is applied at the end of the planning process. This technique follows a classic approach but incorporates some adaptations to operate in the hybrid planning space. In particular, only aerial nodes could be considered for this smoothing process. Pipe nodes are not processed because the 1D characterisation of the pipe arrays already ensures suboptimal trajectories on the pipes and, even more important, because the elimination of transition pipe nodes could negatively affect the optimality of the global trajectory.

Estimation of the energy consumption. General approach

This part is devoted to calculating an approximated estimation of the ratios of energy consumption $e_{roll}, e_{fly}, e_{tran}$ per unit of displacement that are included in equation (5.3).

In general, the electrical energy consumed by an HR while rolling, flying between aerial nodes or performing a hybrid transition (landing/take-off) $E_{roll}, E_{fly}, E_{tran}$ can be defined as:

$$E_{roll,fly,tran} = \int_0^{\Delta t} I(t)\mathrm{d}t \tag{5.4}$$

where $I(t)$ is the demanded current over time. Hereinafter, two different models have been derived to estimate the function $I(t)$ that allows solving equation (5.4): one for rolling phases and one for flying phases. Both of them are based on energetic principles. However, while the first focuses on the analysis of the mechanical power, the second makes use of the aerodynamic power.

Energy consumption in rolling phases

According to the law of conservation of energy, the electric power P_e generated by the robot batteries will be transformed to the mechanical power P_m that will be used to roll on the pipes, with certain power losses quantified through the efficiency parameter η_m. Consequently:

$$\begin{aligned} P_m &= \eta_m P_e = \eta_m V_{roll} I \\ I &= \frac{P_m}{\eta_m V_{roll}} \end{aligned} \tag{5.5}$$

where V_{roll} is the battery voltage and $P_e = V_{roll} I$ by definition. Additionally, considering that the mechanical power P_m will be used to counteract the friction force F_r

between the robot wheels and the pipes, this can be modelled as:

$$P_m = F_r u_{roll} = \mu_f m g u_{roll} \tag{5.6}$$

where μ_f is the coefficient of friction, m is the mass of the HR, g is the gravity acceleration, $P_m = F_r u_{roll}$ by definition, and $F_r = \mu_f m g$ is the used friction model. Substituting, equation (5.6) in equation (5.5):

$$I = \frac{\mu_f m g u_{roll}}{\eta_m V_{roll}} \tag{5.7}$$

In this point, all the parameter in $I(t)$ can be considered constant except V_{roll}. However, V_{roll} can be approximated, as a first approach, by its mean value between initial and final voltages, which are usually well defined. Integrating equation (5.4) after substituting equation (5.7):

$$E_{roll} = \frac{\mu_f m g u_{roll}}{\eta_m V_{roll}} \Delta t \tag{5.8}$$

Considering constant rolling velocity between nodes i and j:

$$u_{roll} = \frac{\|\mathbf{p_j} - \mathbf{p_i}\|}{\Delta t} \tag{5.9}$$

and substituting equation (5.9) in equation (5.8):

$$E_{roll} = \frac{\mu_f m g}{\eta_m V_{roll}} \|\mathbf{p_j} - \mathbf{p_i}\| \tag{5.10}$$

Finally, comparing equation (5.10) with equation (5.3):

$$e_{roll} = \frac{\mu_f m g}{\eta_m V_{roll}} \tag{5.11}$$

Energy consumption in flying phases

According to the law of conservation of energy, the electric power P_e generated by the robot batteries will be transformed to the aerodynamic power P_a that will be used to lift the robot, with certain power losses quantified through the efficiency parameter

η_a. Consequently:

$$\begin{aligned} P_a &= \eta_a P_e = \eta_a V_{fly} I \\ I &= \frac{P_a}{\eta_a V_{fly}} \end{aligned} \tag{5.12}$$

Additionally, according to the Momentum Theory of Fluid Dynamics [90], the aerodynamic power P_a of a multirotor with n_r rotors of radius r_r can be modelled as:

$$P_a = \frac{(mg)^{3/2}}{\sqrt{2\rho_a \pi n_r r_r^2}} \tag{5.13}$$

where ρ_a is the air density. This model is derived for aerial vehicles while hovering. However, its validity has also been demonstrated, as a first approach, for smooth flights [91]. In any case, this model supposes an overestimation of the aerodynamic power, which is slightly lower when the vehicle is moving. The latter allows an estimation on the side of safety. Substituting, equation (5.13) in equation (5.12):

$$I = \frac{(mg)^{3/2}}{\eta_a V_{fly} \sqrt{2\rho_a \pi n_r r_r^2}} \tag{5.14}$$

Integrating equation (5.4) after substituting equation (5.14) and assuming again a constant value of I following the same reasoning as in the derivation for rolling segments:

$$E_{fly} = \frac{(mg)^{3/2}}{\eta_a V_{fly} \sqrt{2\rho_a \pi n_r r_r^2}} \Delta t \tag{5.15}$$

Considering constant flight velocity between nodes i and j:

$$u_{fly} = \frac{\|\mathbf{p_j} - \mathbf{p_i}\|}{\Delta t} \tag{5.16}$$

and substituting equation (5.16) in equation (5.15):

$$E_{fly} = \frac{(mg)^{3/2}}{\eta_a V_{fly} u_{fly} \sqrt{2\rho_a \pi n_r r_r^2}} \|\mathbf{p_j} - \mathbf{p_i}\| \tag{5.17}$$

Finally, comparing equation (5.17) with equation (5.3):

$$e_{fly} = \frac{(mg)^{3/2}}{\eta_a V_{fly} u_{fly} \sqrt{2\rho_a \pi n_r r_r^2}} \tag{5.18}$$

The same approach also allows the estimation of e_{tran}:

$$e_{tran} = \frac{(mg)^{3/2}}{\eta_a V_{fly} u_{tran} \sqrt{2\rho_a \pi n_r r_r^2}} \tag{5.19}$$

Robustness in hybrid planning: air-pipe and pipe-pipe transitions

In standard derivations of RRT*-based planners, the admissibility of a new branch is analysed based on possible collisions of the robot with obstacles within the planning area. In contrast, the hybrid operational environment of HRs introduces additional aspects that can also affect whether or not it is possible to admit a new branch. These new aspects arise from the need to switch between air and pipes, or between pipe structures connected to each other. The integration of these new situations into the planning space requires specific conditions with regard to the corresponding transitions that guarantee safe operation at all times. Therefore, the tree expansion will now account not only for the collisions but also for these conditions that guarantee robustness in hybrid planning. In order to better encompass all these aspects, the admissibility of new branches will be denoted hereinafter by the new term feasibility. A more detailed description of the application of this new concept is presented below.

Feasibility of transitions between aerial and pipe states

Within the planning framework, take-off and landing manoeuvres correspond to branches that link aerial and pipe nodes. These manoeuvres will be considered safe when the reference trajectories associated with those branches fall within a volume that is compatible with the dynamic constraints of the HR. From here on, this volume will be referred to as region of safe transition.

More in detail, these safety regions will correspond to conical truncated volumes with aperture semi-angle β and radius r_0 at the intersection with the pipe node (see Fig. 5.6). The height references that determine the allowable upper and lower limits of this conical volume are given by a minimum separation h_{min} with respect to the pipe array (a smaller separation would be risky either to start the landing or to finish the take-off) and a maximum separation h_{max} (in order to bound the duration of the hybrid transition). It should be noted that the reference velocity imposed to these hybrid transitions will be smaller than the reference velocity corresponding to normal flight; that is, $u_{tran} < u_{fly}$. Mathematically, these feasibility conditions are given by:

$$\begin{gathered} h_{min} \leq z_{air} - z_{pipe} \leq h_{max} \\ \sqrt{\left(x_{air} - x_{pipe}\right)^2 + \left(y_{air} - y_{pipe}\right)^2} < r_0 + \tan\beta\left(z_{air} - z_{pipe}\right) \end{gathered} \tag{5.20}$$

where $(x, y, z)_{air,pipe}$ are the position coordinates of the aerial and pipe nodes that form the transition branch that is under analysis.

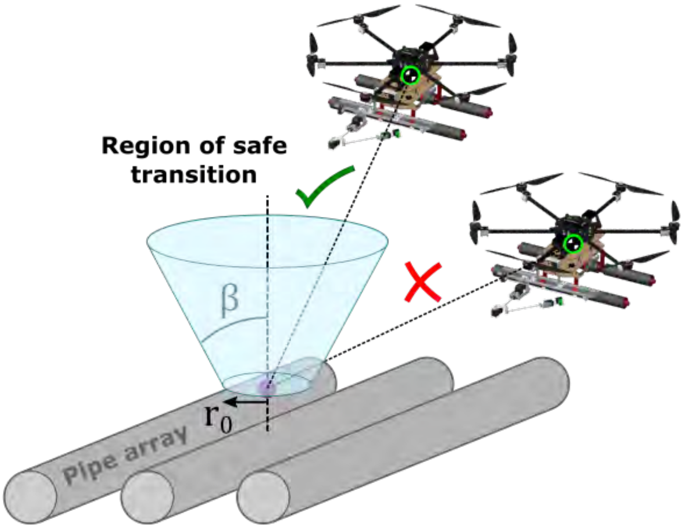

Figure 5.6: Feasibility checking in transitions between aerial nodes and pipe nodes. Transitions taking place out of the region of safe transition are discarded.

Feasibility of transitions between pipe states

During the expansion of the search tree, a potential new branch linking two pipe nodes can be classified into two categories: branches linking pipe nodes in the same pipe array or branches linking pipe nodes in different pipe arrays. Whereas rolling transitions can be assumed feasible in the first group, it is important to take into account whether both nodes belong to connected or unconnected pipe arrays in the second group. In the first case, the transition between arrays can be executed through the intersection region. However, in the second case, the potential new branch must be discarded because it is not possible to link two nodes in unconnected pipe arrays without an intermediate transition through aerial nodes.

All these situations can be characterised using a connection matrix $\mathbf{\Gamma} \in \mathcal{B}^{N\times N}$ where N is the number of pipe arrays in the environment. The elements $\gamma_{i,j}$ of this boolean matrix takes the logical values $\{0,1\}$ and represent the connection between pipe arrays i and j. Consequently, $\mathbf{\Gamma}$ is a symmetric matrix with ones in the main diagonal:

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & \gamma_{1,2} & \gamma_{1,3} & \cdots & \gamma_{1,N} \\ \gamma_{1,2} & 1 & \gamma_{2,3} & \cdots & \gamma_{2,N} \\ \gamma_{1,3} & \gamma_{2,3} & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \gamma_{N-1,N} \\ \gamma_{1,N} & \gamma_{2,N} & \cdots & \gamma_{N-1,N} & 1 \end{bmatrix} \qquad \gamma_{i,j} = \{0,1\} \tag{5.21}$$

This characterisation allows establishing a simple criterion to analyse feasibility in pipe-to-pipe transitions by simply checking the logical value of $\mathbf{\Gamma}$ for the pipe arrays associated with a specific branch.

Feasibility of transitions in respect of collisions

Finally, the simplified approach for a computationally-efficient treatment of collision checking has also been summarised in Fig. 5.7 for completeness in the definition of the feasibility concept. As can be seen, the geometry of the HR is approximated by its bounding sphere of radius r_{robot}, while the pipes are modelled as cylinders of radius r_{pipe}. Margins of safety MoS have also been considered. The strategy is based on simple geometrical considerations; that is, the minimum distance between the centre of mass of the robot and the segments corresponding to the axes of the pipes. In this context, the collision condition can be defined as:

$$d_{robot-pipe}^{min} < r_{robot} + r_{pipe} + MoS \tag{5.22}$$

This simplified approach allows speeding up the computation of collision checking and hence makes the MP-HR algorithm come closer to be a suitable strategy for cluttered environments, where this calculation tends to require high computational loads.

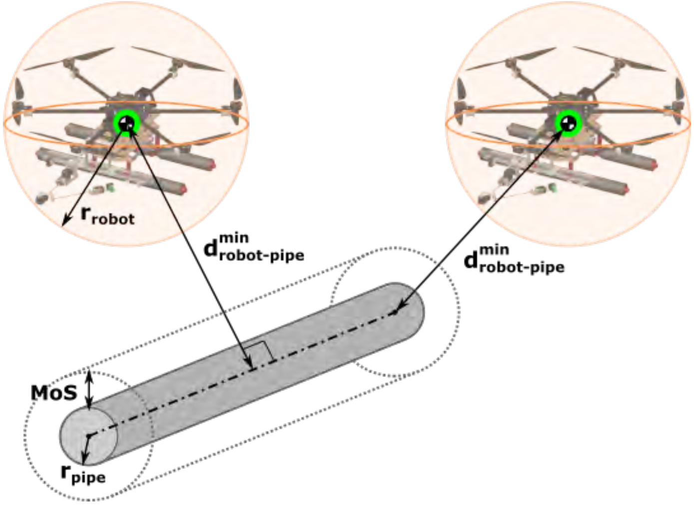

Figure 5.7: Collision checking based on the minimum distance between the HR and the pipe. The robot is approximated by its bounding sphere while the pipe is modelled as a cylinder. Margins of safety are also considered.

5.2.2 Dynamics Awareness

The MP-HR algorithm presented above guarantees planned trajectories that are kinematically feasible as well as efficient in terms of operation time or energy consumption. However, as it was demonstrated in Section 4.1, the complex dynamics that governs aerial systems requires further attention since it may provoke considerable differences between planned and executed trajectories. The collision risk introduced by these differences is especially critical for HRs while flying in highly-cluttered environments like some areas of the industrial sites considered in this chapter. In contrast, during rolling phases, the trajectory tracking is usually safe and accurate enough and the dynamic behaviour of HRs can be neglected.

In order to overcome this undesired influence of the dynamics while flying, the MP-HR algorithm previously presented is transformed into a more advanced algorithm that incorporates Dynamics Awareness (MP-HR-DA). This extension is inspired by previous contribution in Section 4.1.2. Following a similar approach, the expansion of the search tree for aerial branches in the new MP-HR-DA algorithm will be based on the behaviour of the controlled HR, which means that the feasibility of these branches is checked through closed-loop simulations of the controlled system (green solid lines in Fig. 5.8) instead of using geometrical interpolation between states (blue dashed lines in Fig. 5.8). Moreover, the scope for this dynamical analysis of the tree extension is again a root-to-candidate validation. Thus, not only the dynamical feasibility of the possible new branch reaching the candidate node (from node x_{near} to node x_{new} in Fig. 5.8) is analysed, but also the complete path from the tree-root node x_{root}.

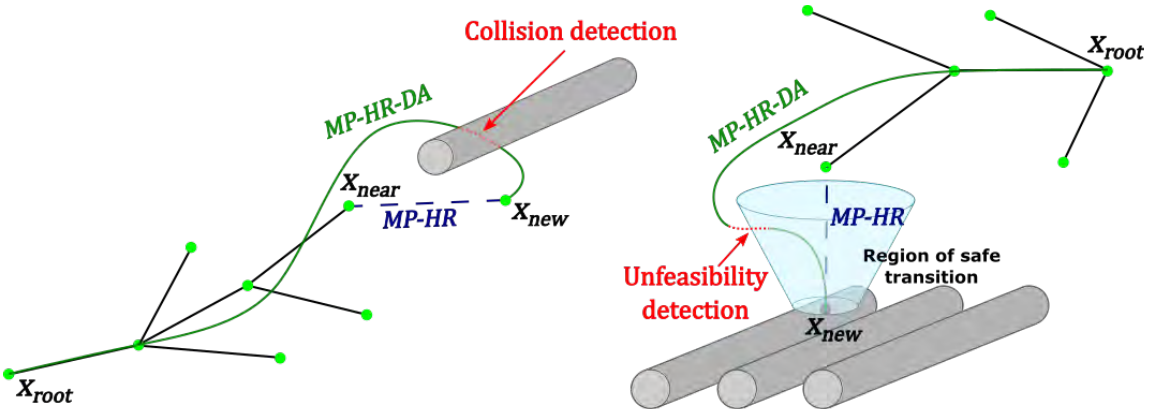

Figure 5.8: Operation basis of the MP-HR-DA algorithm: feasibility analysis of a new air-air branch (left) and feasibility analysis of a new air-pipe branch (right).

Since the proposed extension relies on closed-loop simulations of the controlled HR, the required dynamic model while flying and the associated controller have been adopted from Annex A.4.2 for the particular case of the HRM system.

This whole approach guarantees that the resultant planned trajectories are compatible with dynamic constraints while flying, hence scenarios where the obstacle proximity narrows considerably the safe operation areas can be addressed in safer conditions. This reinforces considerably the level of robustness and autonomy that HRs can offer.

5.2.3 Hybrid replanning

The industrial scenarios considered in this chapter are in certain cases highly-cluttered environments (for instance, an industrial refinery with many pipe arrays that intersect with each other as in Fig. 5.1) and it is very likely that the map provided to the planner does not include all the potential obstacles that the HR will come across during the inspection. In order to address this uncertainty, onboard perception sensors should be capable of detecting the unexpected elements during the plan execution. This detection in real time will allow modifying locally the global trajectory for collision avoidance. The motion planner plays an important role in this reaction since it will be in charge of the local replanning.

This kind of reactive behaviours has been widely studied in the literature for aerial systems under standard conditions; these are, in flight segments. However, this section explores hybrid reactivity that is specific to HRs. In other words, the unmapped obstacles will affect a rolling path on the pipes and hence the reactive avoidance will imply switching from pipe to air. After that, the system shall return to the pipe in order to keep following the original global inspection plan. This capability of hybrid reaction will reinforce the level of robustness and autonomy offered by HRs.

Fig. 5.9 represents two scenarios that benefit from hybrid reactivity. In both cases, the pipes circled in red would not have been included in the initial map provided to the planner. This is a realistic assumption since it is difficult to keep updated the map of the pipes after the frequent operations of repairing and substitution of the piping layout. As a result, the generated trajectory would command the HR to roll along the main arrays (blue arrows), which would produce a collision with the pipes circled in red. The hybrid reactivity presented in this section should allow the HR to switch from pipe to air to avoid the unmapped pipes and, after that, from air to pipe to continue with its navigation.

Figure 5.9: Example of application scenarios for hybrid reactivity.

In case that the HR has to enable the reactive response and to locally replan the global trajectory for collision avoidance, the same algorithmic basis of the MP-HR planner presented in Section 5.2.1 will be used. However, this new scenario requires a low computation time to ensure fast reactions. Although the reduction of the planning area associated with the local nature of the reactive behaviour contributes to satisfy this requirement, the approaches presented in Section 3.2.2 (Computational efficiency) have also been considered here to better meet the requirement.

5.3 Simulation results

In order to demonstrate the validity of the motion planning strategies presented in previous section, the algorithms will be tested though simulation in realistic industrial scenarios. All these scenarios require capabilities of hybrid motion to perform safely the desired operations in the associated cluttered environments. For that, the HRM system presented in Annex A.4 will be used.

The expected results will be inspection trajectories that allow the HRM system to navigate through the provided sequence of inspection points while avoiding the pipes and other restricted zones (operation limits, ...). Additional requirements such as metric optimisation or reactive behaviours will be introduced in the corresponding scenarios.

When illustrating graphically the scenarios, certain common criteria have been followed. Fig. 5.10 shows an example scenario to illustrate these criteria. Grey elements correspond to the existing pipe structures whereas the yellow volumes denote the safety regions associated with these pipe structures whose violations will be treated as collisions. In particular, these safety regions define the non-allowable regions for the displacement of the HR centre of mass. The purple points are reserved to highlight the inspection points while purple crosses represent the initial and final position of the HR. All these reference points are numbered according to the inspection sequence (0, 1, 2, ..., 0). Finally, the red lines stand for the operational limits of the HR for a particular inspection mission.

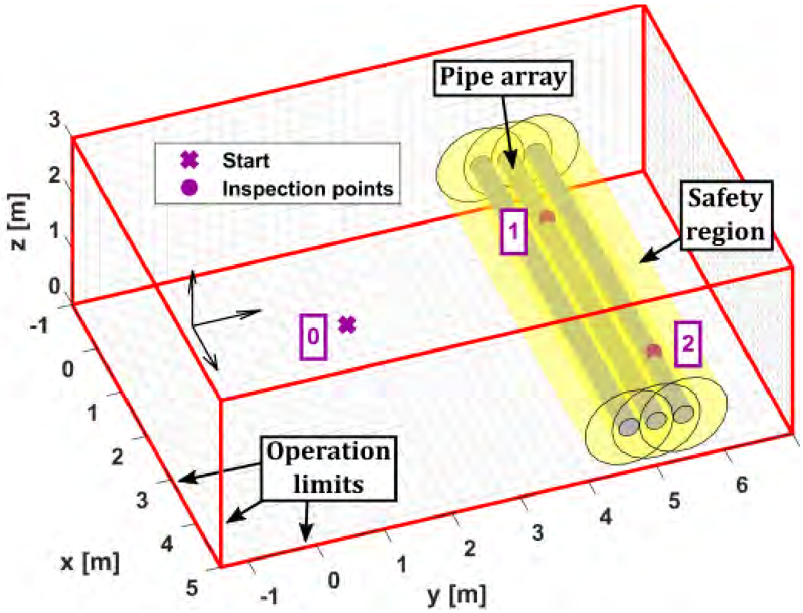

Figure 5.10: Example scenario illustrating the criteria followed to represent the different application scenarios.

In general terms, the validation tests have been carried out following the same sequence. Firstly, the algorithm under analysis has been executed to generate the motion plan. Then, the resultant plan has been provided to the closed-loop simulation of the controlled HRM system. For this, the model and controller presented in Annex A.4.2 have been used. The objective is therefore to analyse not only the planned trajectories, but also the closed-loop behaviour when the robot follows such trajectories.

The simulation work has been performed in a MATLAB-Simulink framework [7] that provides the graphical evolution of the system variables. This graphical output will be used throughout this section to illustrate the obtained results. Additionally, intuitive 3D snapshot diagrams have also been included. In these diagrams, the coloured lines represent, respectively, the complete planned (dashed light colours) and the simulated (solid dark colours) movements of the HR centre of mass. In contrast, the snapshots themselves only cover some intermediate configurations to intuitively illustrate the behaviour of the robot (air/pipe location, yaw angle, ...). These representative configurations are time-ordered in accordance with the inspection sequence along the reference points (0, 1, 2, ..., 0).

5.3.1 Fundamentals of planner operation

This section is focused on the validation of the MP-HR planner presented in Section 5.2.1 that serves as the basis for the more advanced developments derived in subsequent sections (Dynamics Awareness and hybrid replanning). The scenario proposed below has been selected for that purpose. Both cost functions defined in equations (5.2) and (5.3) have been used to optimise respectively the operation time and the energy consumption. The objective is to evaluate the performance of the generated trajectories under such metrics as well as their robustness against potential collisions with the pipes.

Application scenario: inspection plan in moderately-cluttered area

The application scenario considered for the validation of the MP-HR planner is depicted in Fig. 5.11. The inspection plan consists of 3 measurement points and the operation area can be classified as moderately cluttered since it comprises several pipe arrays and isolated pipes with different diameters and inclinations.

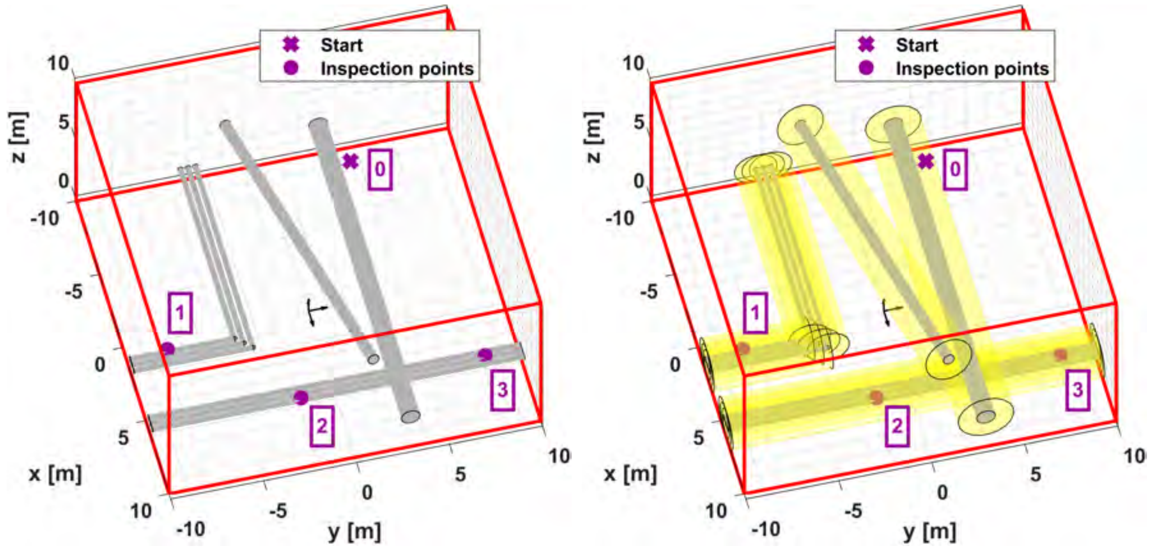

Figure 5.11: Application scenario given by an inspection plan in moderately-cluttered area: topology definition (left) and the corresponding safety regions (right).

Analysis of the results: efficiency in operation time

The validation simulations of the MP-HR algorithm when minimising the operation time have been included in Fig. 5.12. As can be seen, the HR tends to fly directly between inspection points. This behaviour is consistent with the higher velocity, and hence shorter time, associated with the flying mode in comparison with the rolling mode. Additionally, the restricted areas (safety margins of pipes, operation limits, ...) are not violated. Concerning the yaw angle, it is properly aligned with the flight path that guarantees optimal time execution. This alignment also allows better detection of unmapped obstacles since the common setups of onboard sensors usually maximise the perception capabilities on the front side of the system.

It should also be noted that the robot exhibits some overshooting around the reference waypoints (see Fig. 5.13). This observation is related to the control structure that has been adopted for the simulation work. That structure has not been tuned arbitrarily but tries to emulate the qualitative behaviour that can be observed in the experiments presented later in Section 5.4 with the real HRM system.

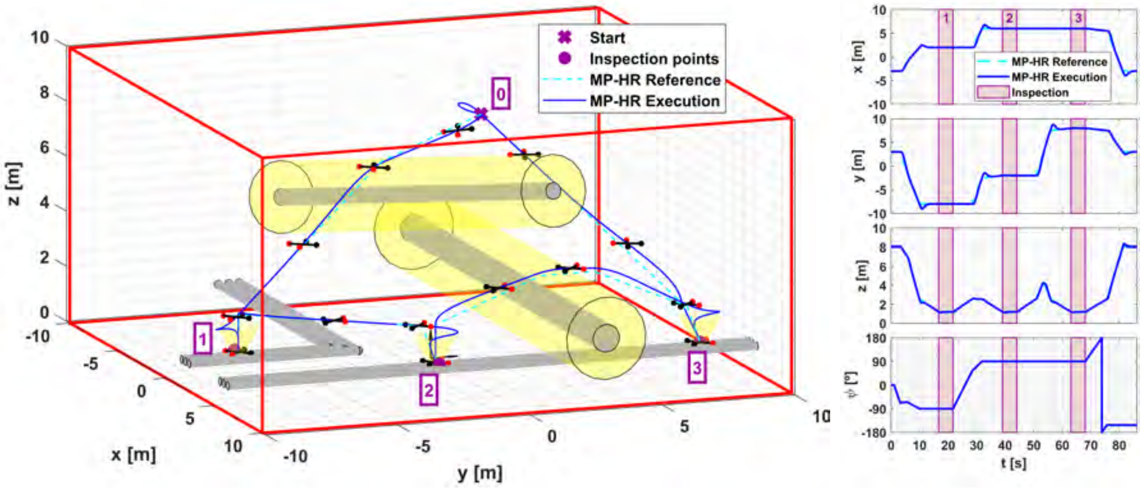

Figure 5.12: Inspection plan in moderately-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the operation time.

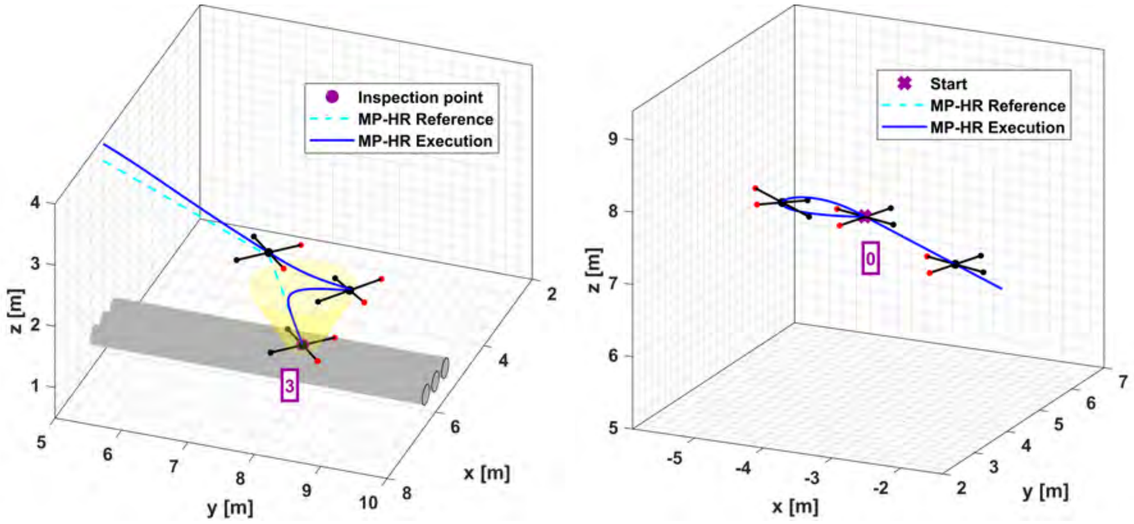

Figure 5.13: Overshooting in reference waypoints extracted from Fig. 5.12: landing on the inspection point 3 (left) and final approaching to the start point (right).

Following this analysis, it can be concluded that the MP-HR algorithm is capable of generating both safe and time-efficient trajectories for the HR. However, the existence of the overshooting previously mentioned is something that requires further follow-up. Subsequent scenarios will illustrate that this degraded performance of the controller could lead to collisions and other behaviours not allowed. The MP-HR-DA motion planner will be the solution to overcome these undesired behaviours of the closed-loop dynamics, as it will be presented later in Section 5.3.2.

Analysis of the results: efficiency in energy consumption

The validation simulations of the MP-HR algorithm when minimising the energy consumption have been included in Fig. 5.14. In contrast to previous case, the HR tends now to roll on the pipes as much as possible. This new behaviour is consistent with the smaller energy consumption associated with the rolling mode in comparison with the flying mode. Additionally, the restricted areas (safety margins of pipes, operation limits, ...) are properly avoided. Finally, the alignment of yaw angle in flight segments is fulfilled in the same manner as before. Concerning the pipe segments, these results endorse the proper selection of yaw references that are compatible with rolling displacements along the pipes.

Following this analysis, it can be concluded that the MP-HR algorithm is also capable of generating both safe and energy-efficient trajectories for the HR, with the same remarks concerning the overshooting that were discussed in previous simulations.

5.3.2 Dynamics Awareness

This section validates the MP-HR-DA extension presented in Section 5.2.2 that considers the dynamic constraints of the HR when flying. In order to better illustrate the need of this feature, a new highly-cluttered scenario is proposed. Moreover, the selected metric for planner optimisation is the operation time. The usage of this metric, which yields faster trajectories by maximising the flying segments, together with the new highly-cluttered scenario, narrow the safe operation areas of the generated flying trajectories with respect to the obstacles. The simulations developed under such

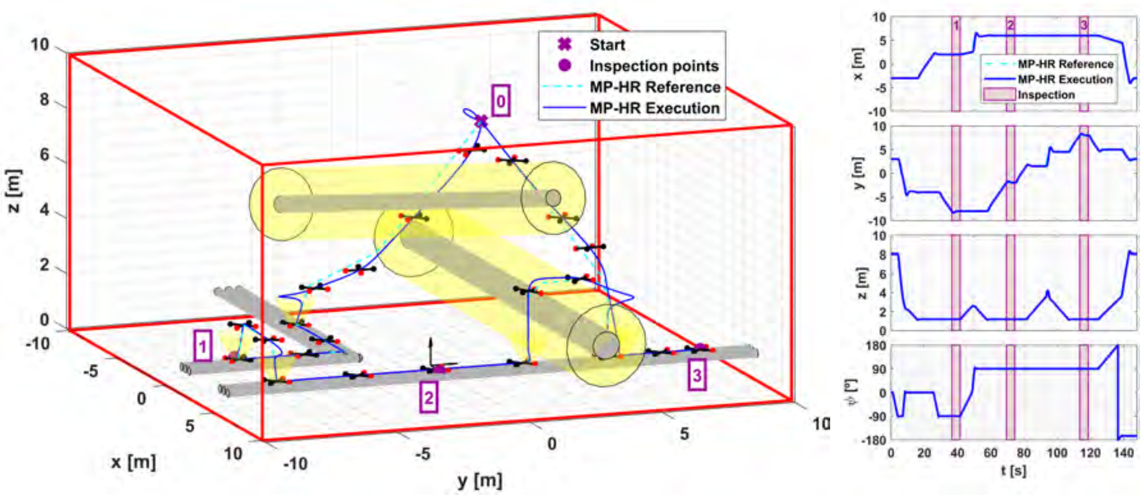

Figure 5.14: Inspection plan in moderately-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the energy consumption.

critical situation reveal the importance of taking into account the expected closed-loop behaviour of the system in planner operation.

It should be clarified that the MP-HR-DA algorithm is equally suitable for flying phases when minimising the energy consumption. The choice of the time-consumption metric for this section is simply made to maximise the number of situations that will benefit from the algorithm extension.

The validation simulations have been structured around two phases. On a first step, the performance of the MP-HR algorithm has been evaluated in the new scenario. On a second step, the extended MP-HR-DA algorithm is applied to the same scenario in order to illustrate the benefits associated with this enhanced version of the motion planner.

Application scenario: inspection plan in highly-cluttered area

The application scenario is presented in Fig. 5.15. The new inspection plan consists of 4 measurement points and the operation area can be classified as highly cluttered since it comprises more pipe structures than the previous scenario. Their relative topology is also more complex since they are located at different heights and there are also some connections between them. Additionally, a tank (red vertical cylinder) has been included in the central area, which will also limit considerably the manoeuvrability of the HR when flying.

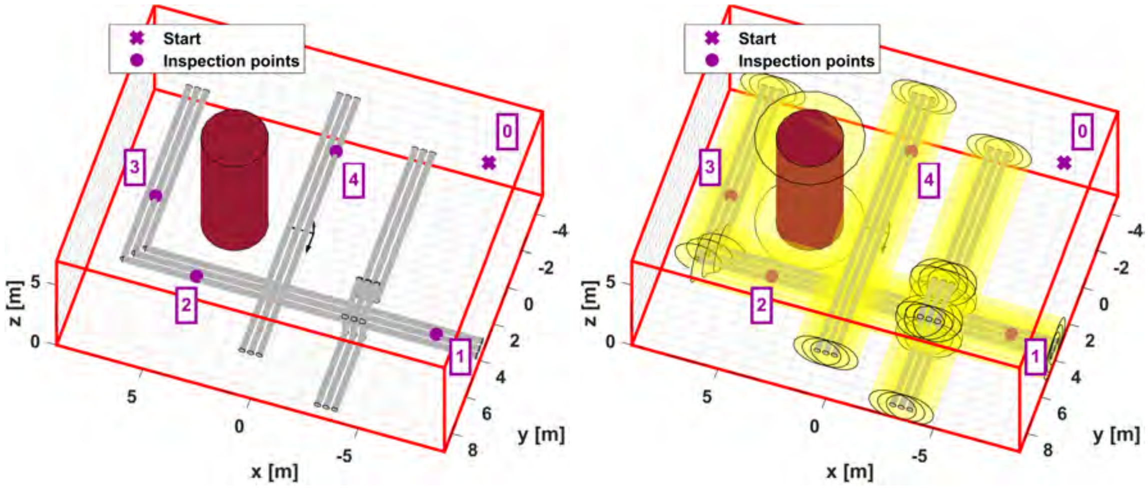

Figure 5.15: Application scenario given by an inspection plan in highly-cluttered area: topology definition (left) and the corresponding safety regions (right).

Analysis of the results: MP-HR algorithm (efficiency in operation time)

The trajectory generated by the MP-HR algorithm for the new highly-cluttered scenario has been included in Fig. 5.16. According to this result, the algorithm plans both safe and time-efficient trajectories for the proposed inspection sequence. However, the corresponding closed-loop trajectory described by the controlled HR does not satisfy the safety conditions. Firstly, it violates the safety margin of the pipe array located above, which is already considered a collision to prevent closer approximations (see the detailed view in the left side of Fig. 5.17). Secondly, the condition required for safe landing on the same array is not fulfilled either (see the detailed view in the right side of Fig. 5.17). These undesired behaviours are consistent with the fact that the MP-HR algorithm does not consider the HR dynamics and therefore dynamical effects like overshooting are not accounted for during the feasibility-checking phase of the motion planner.

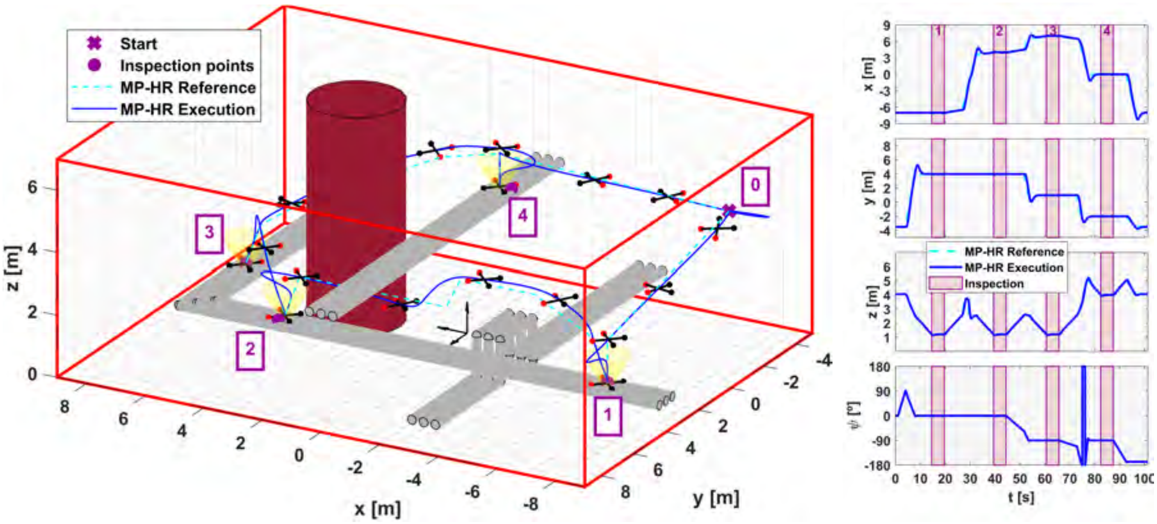

Figure 5.16: Inspection plan in highly-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the operation time.

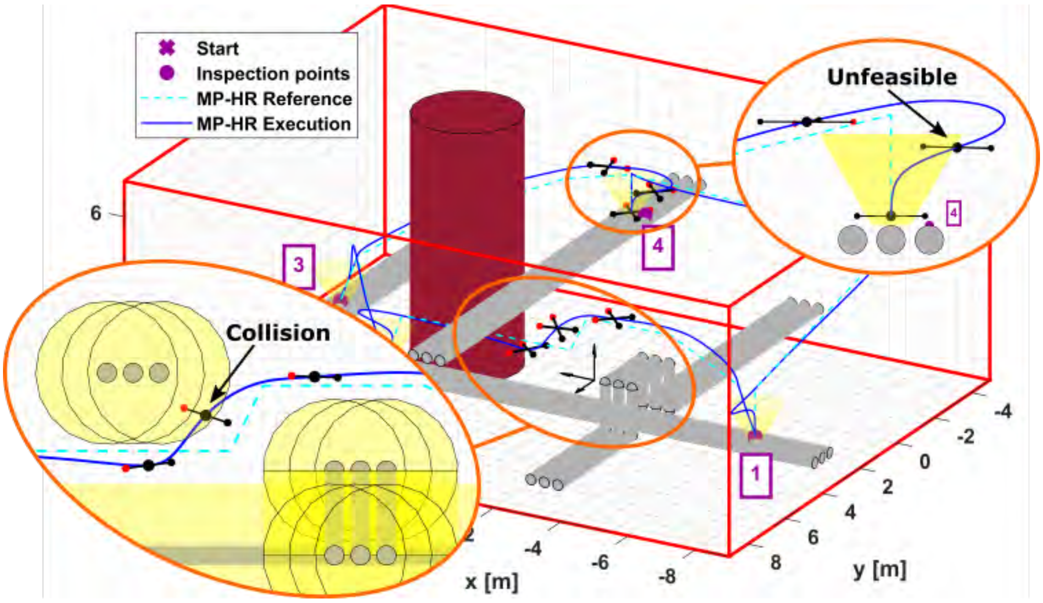

Figure 5.17: Detailed views of Fig. 5.16: violations of safety margin (left) and region of safe transition (right).

Analysis of the results: MP-HR-DA algorithm (efficiency in operation time)

Previous results illustrated the need to consider the dynamics of the robot when applying the MP-HR algorithm in highly-cluttered environments. The simulation results corresponding to the MP-HR-DA method that meets this requirement are presented in Fig. 5.18. As can be observed, Dynamics Awareness allows the algorithm to solve the problematic situations described previously: neither collisions with the safety margins for obstacles (it should be remembered that the safety regions define non-allowable regions for the displacement of the HR centre of mass) nor violations of the safety conditions for landing manoeuvres appear now (see Fig. 5.19). The analysis of these results allows concluding that Dynamics Awareness is required to generate both time-efficient and safe trajectories when the HR operates in highly-cluttered environments.

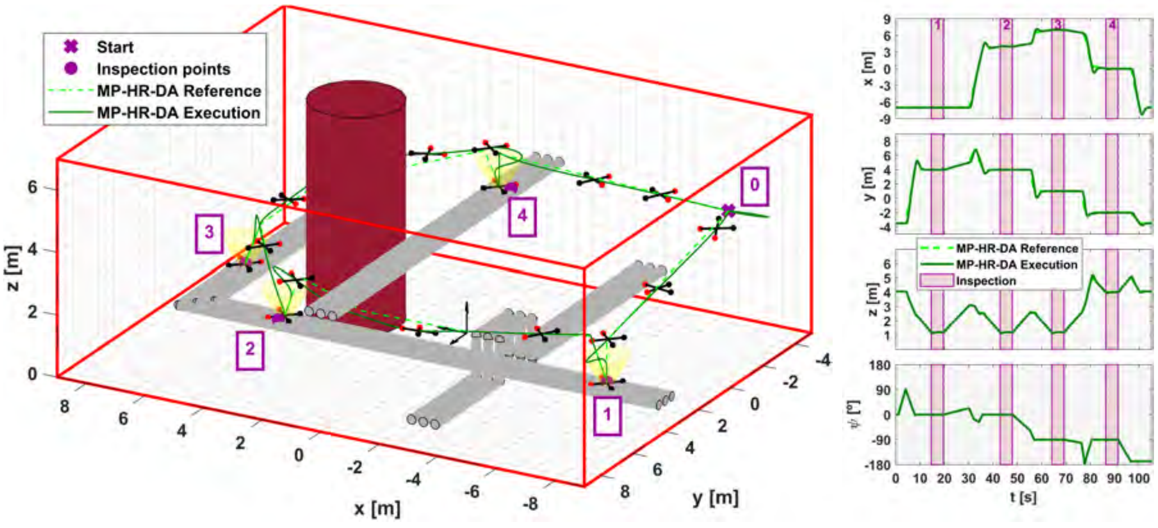

Figure 5.18: Inspection plan in highly-cluttered area. Execution (dark green) of the trajectory planned (light green) with the MP-HR-DA algorithm when optimising the operation time.

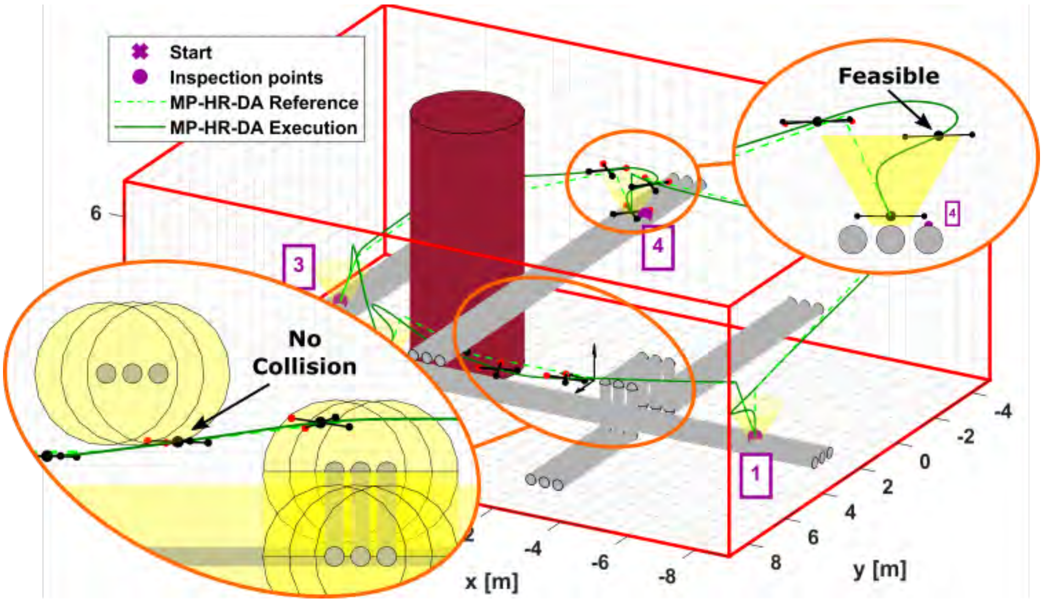

Figure 5.19: Detailed views of Fig. 5.18: the MP-HR-DA method overcomes the violations of the safety margin (left) and the region of safe transition (right).

5.3.3 Hybrid replanning

This section illustrates the hybrid-reactive capabilities of the MP-HR planner that have been presented in Section 5.2.3. With this motivation, a modified version of the scenario given in Section 5.3.1 (Application scenario: inspection plan in moderately-cluttered area) is presented below. Since the need of hybrid reactivity only arises while rolling on pipes, the selected metric for planner optimisation in this scenario will be the energy consumption, which is more prone to that kind of manoeuvres. In any case, this reactive application of the MP-HR algorithm is equally suitable for rolling phases when minimising the operation time.

Application scenario: inspection plan in moderately-cluttered area with unmapped obstacle

As it was advanced before, a modified version of the scenario given in Section 5.3.1 (Application scenario: inspection plan in moderately-cluttered area) has been proposed here. In this case, a pipe of the moderately-cluttered area will not be included in the map provided to the motion planner. Specifically, the horizontal pipe that passes crosswise through the scenario from (-10m, 3m, 2.5m) to (10m, 3m, 2.5m) has been removed (see Fig. 5.20). This is a realistic assumption since it is difficult to keep updated the map of the pipes after the frequent operations of repairing and substitution of the piping layout. As a consequence, onboard sensors must reveal the presence of this unmapped obstacle to trigger the online replanning that will allow the reactive response of the HR.

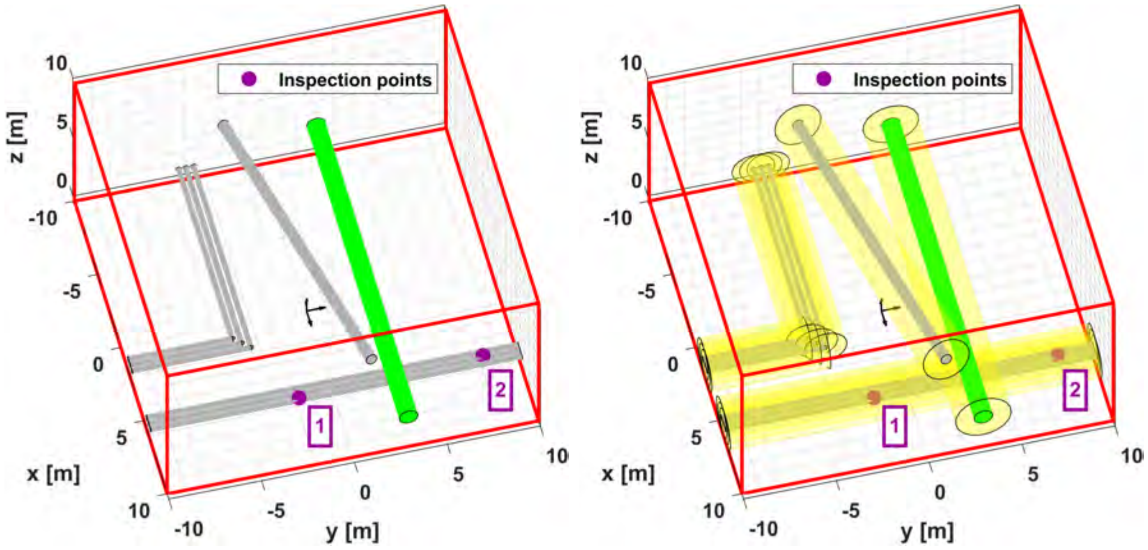

Figure 5.20: Application scenario given by an inspection plan in moderately-cluttered area with unmapped obstacle: topology definition (left) and the corresponding safety regions (right). The pipe highlighted in green will not be included in the map provided to the motion planner.

Analysis of the results

The validation simulations of the hybrid reactivity have been presented in Fig. 5.21. The left side of the figure shows the trajectory computed by the MP-HR planner when the unmapped pipe is not considered. On the right side, the replanned trajectory generated thanks to the reactive application of the MP-HR algorithm during the inspection is depicted. This trajectory, whose computation time is less than 1 second, allows the reactive avoidance of the pipe detected by onboard sensors. Also in the right side of the figure, the initial and final positions (green circles) as well as the operation area (red subset of the operational limits) of the replanning have been highlighted. From a more general view, it can also be concluded that the HR maintains its tendency to roll on the pipes as much as possible. Additionally, the restricted areas (safety margins of pipes, operation limits, ...) apart from the unmapped pipe are still properly avoided. Concerning the yaw angle, the reactive response robustly maintains the alignment that is required for rolling displacements along the pipes. Therefore, the generation of both safe and energy-efficient trajectories for replanning purposes is also validated when facing unexpected obstacles during rolling manoeuvres.

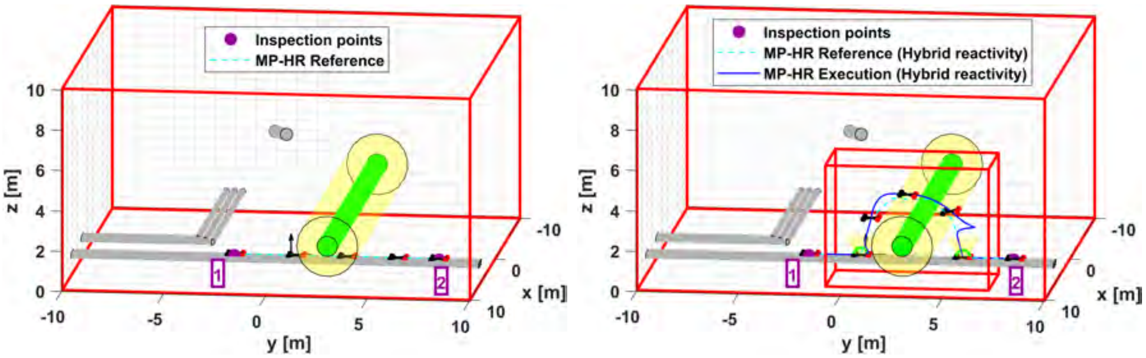

Figure 5.21: Inspection plan in moderately-cluttered area with unmapped obstacle. Hybrid-reactive capabilities for rolling manoeuvres: trajectory initially planned by the MP-HR algorithm without considering the green pipe (left) and execution (dark blue) of the trajectory replanned (light blue) with the MP-HR algorithm when the green pipe is detected by onboard sensors (right).

5.4 Experimental results

Once the different strategies to plan hybrid motions have been validated through simulation, the next step is the experimental validation in outdoor scenarios. For that, the real HRM prototype presented in Annex A.4 (see the right side of Fig. A.15) has been used for the autonomous execution of planned trajectories.

5.4.1 Fundamentals of planner operation

First experiments have been carried out with the MP-HR algorithm in a scarcely-cluttered area consisting of several mock-ups made of plastic that closely resemble real pipe arrays. These conditions are considered safe enough to guarantee the absence of collisions even in the case that any deviation from the expected operation is produced.

Since one of the main objectives of these experiments is the validation of the MP-HR capabilities to leverage the hybrid motion of HRs, the selected optimisation metric has been the energy consumption. This choice will make that the motion planner tends to maximise the rolling manoeuvres and hence, the hybrid operation of the robot.

Application scenario: inspection plan in scarcely-cluttered area

The proposed outdoor scenario is shown in Fig. 5.22. It consists of two arrays of three plastic pipes (2 m length, 20 cm diameter, 10 cm separation) supported by a frame structure built with aluminium bars. The inspection plan comprises the 2 measurement points that are highlighted in the figure. The laser tracking system that can also be seen in the figure (robotic total station Leica MS50 [92]) is used to measure accurately the motion of the HRM system.

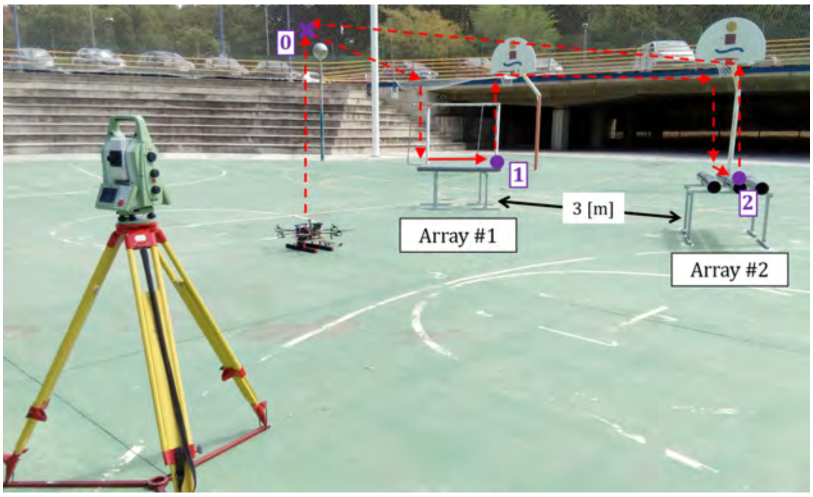

Figure 5.22: Application scenario given by an inspection plan in scarcely-cluttered area.

Analysis of the results

Fig. 5.23 represents the trajectory planned by the MP-HR algorithm (light blue) and the real trajectory executed autonomously by the HRM system (dark blue). As can be observed, the motion planner computes a trajectory that properly guides the HR through the selected inspection points. Furthermore, the safe and energy-efficient behaviour that could be expected from the simulation work, is also met with these experimental results. Indeed, the trajectory is efficient in terms of energy consumption since the 3 subtrajectories connecting the reference points prioritise rolling segments along the pipe arrays and incorporate flight segments only when they are unavoidable. Moreover, the trajectory also guarantees a safe execution of the inspection since the safety conditions concerning obstacle avoidance as well as landing and take-off manoeuvres are properly fulfilled.

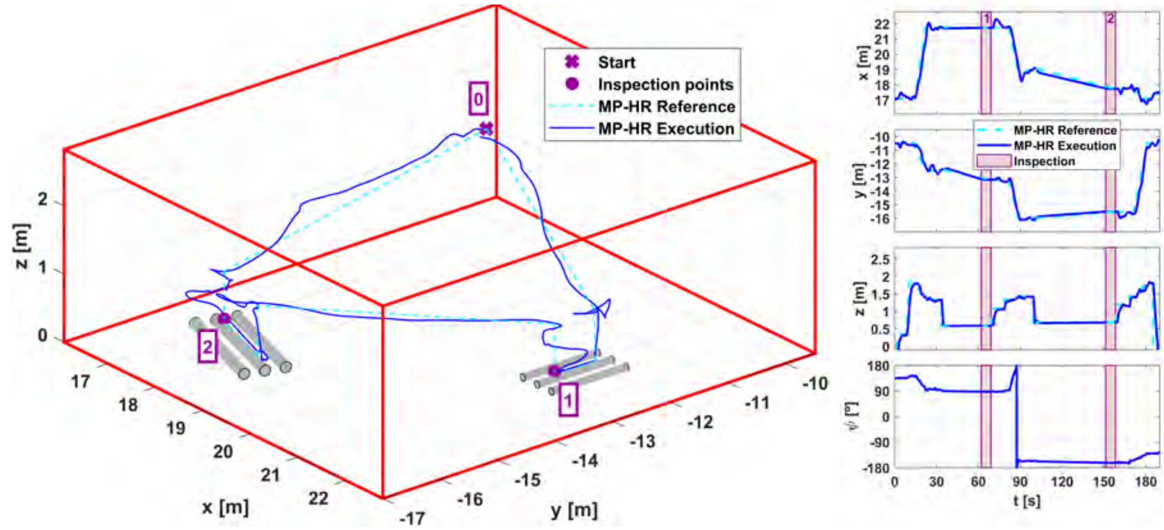

Figure 5.23: Inspection plan in scarcely-cluttered area. Execution (dark blue) of the trajectory planned (light blue) with the MP-HR algorithm when optimising the energy consumption.

The execution of the experiment can also be followed in the complete video *MP-HR_inspection_plan.mp4* that can be downloaded from the provided multimedia content, as indicated in Section 1.4. Additionally, Fig. 5.24 collects some snapshots of the real execution.

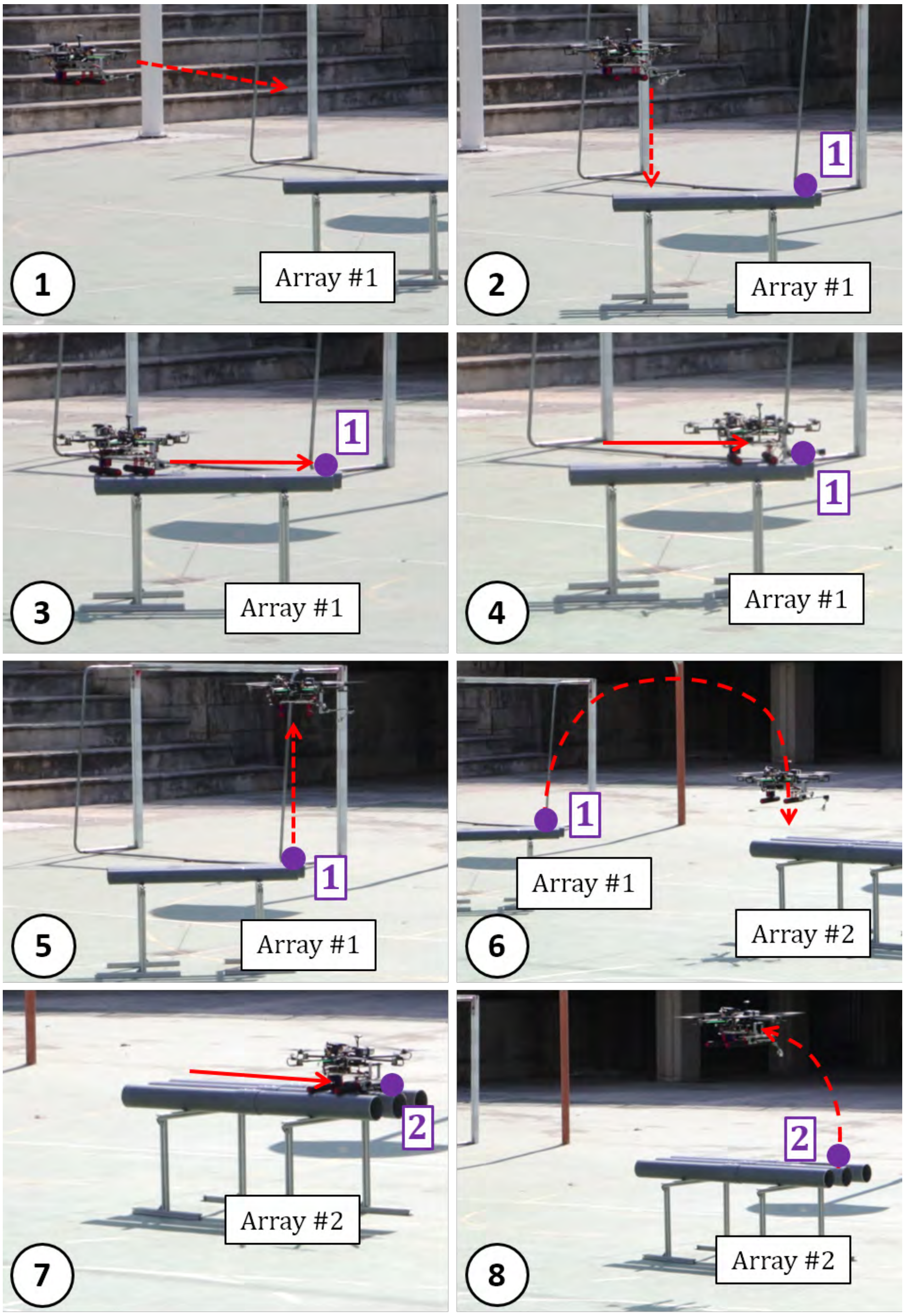

Figure 5.24: Inspection plan in scarcely-cluttered area. Snapshots of the operation.

5.5 Conclusions

The presented MP-HR motion planner has demonstrated to be a useful method for the effective exploitation of hybrid locomotion in HRs. This is possible thanks to the joint consideration of aerial and pipe states within the planner operation, the proper management of transitions between these states and the inclusion of optimisation metrics. Concerning the latter, the selection of either the operation time or the energy consumption as cost function allows different behaviours in the resultant planned trajectories. Thus, when the time is minimised, the HR tends to fly between inspection points, landing on them to perform the inspection while taking advantage of the enhanced accuracy of fixed-base operations and without wasting energy in the propellers. In contrast, the optimisation of the energy brings to trajectories that prioritise rolling segments as much as possible, flying only when it is unavoidable to accomplish the mission. In both cases, the hybrid capabilities of the robot are leveraged conveniently. From a general point of view, the computed trajectories lead to efficient actions, without collisions with obstacles in moderately-cluttered environments and suitable adaptations of the yaw angle to the mission.

Moreover, Dynamics Awareness has demonstrated to be also needed when the scenarios become more cluttered. The MP-HR limitations associated with HR overshooting in highly-cluttered environments are overcome with the MP-HR-DA extension. The consideration of the robot dynamics while flying in the planning process provides both robust collision avoidance and safer transitions between aerial and pipe states; these are, take-off and landing manoeuvres.

Finally, the application of the MP-HR algorithm for hybrid reactivity has also revealed a good performance. In particular, the method is able to locally replan safe and efficient hybrid trajectories when unexpected obstacles are detected by onboard sensors while the HR is rolling during the mission execution. Additionally, this issue can be solved fast, even when the computation time is not critical since the HR is landed and, consequently, its energy consumption is minimal.

Chapter 6

Motion planning for fleets of aerial manipulators

> Que se haga infinito todo aquello
> que te haga feliz.
>
> Ana Almodóvar

6.1 Motivation

Electric power lines comprise one of the most important infrastructures in any country. With thousands of kilometres, these lines supply electricity to millions of people and allow essential activities for the industry, playing an important role in the global economy. In order to ensure reliability in the electric supply, the electric companies invest huge amounts of money in the construction of the infrastructure as well as in good I&M strategies to avoid potential outages and their associated costs [93].

Between the different activities that are commonly carried out on power lines, the installation of bird flight diverters plays an important role. These devices (see Fig. 6.1), installed every 5-15 metres along the cables of a power line, are intended to increase their visibility and consequently, to mitigate the risk of bird collisions with them, estimated in up to 1.05 billion collisions per year in countries like the

United States [94]. As a result, the bird casualties are reduced [95] and with them, the potential outages that could cause. However, some difficulties arise in the installation of these diverters. Firstly, the installation is not concentrated on a particular location but long distances must be covered during the operation. Secondly, the power lines are frequently in hard-to-reach areas like mountains and the cables at high height. Thirdly, the contact with the cables during the installation can conduct to electric discharges since it is carried out on energised lines to avoid cut-offs in the service. To deal with all these drawbacks, the installation of bird diverters is traditionally performed manually by experienced workers from hanging platforms, aerial-lift equipment or manned helicopters [96], making the operation slow, costly and very hazardous.

Figure 6.1: Example of bird flight diverters installed on an electric power line.

As it was analysed in Section 2.1, there has been a growing interest in the application of ARMs for the installation of bird diverters on power lines. However, their limited battery and payload capacities and the use of single-robot solutions make the installation still slow. In contrast, a multiARM approach can speed up the overall installation process at the expense of increasing the operation complexity, which also forces to involve more ARM operators. The use of motion planning can mitigate the effect of all these issues since it can provide trajectories that allow several ARMs to accomplish the full mission autonomously. Thus, the planner can consider the vehicle constraints, eventually heterogeneous, the obstacles in the environment, the installation points and the location of refilling stations to ensure safety and time efficiency in the operation. In that context, the energy consumption of the ARMs should not be disregarded to maximise the number of diverters that can be installed without changing batteries. Additionally, the motion planner should also be able to replan the ARM trajectories in case a particular robot suffers a failure. In this sense, a back-up ARM could substitute the faulty one without the need to abort the remaining part of the mission. All the previous features would ease the operation with multiARM teams, making their field deployment more practical.

6.2 Motion planner

Giving response to the planning problem introduced in previous section, this section proposes a motion planning method for the autonomous installation of bird flight diverters on power lines using a fleet of ARMs. Such installation is performed on the upper cables between two consecutive towers of a power line. To fulfil the mission, a team of ARMs like the Aerial Robotic Manipulator endowed with a Linear Actuator (ARM-LA) presented in Annex A.5 should be coordinated to fly from their initial positions to the target regions that allow the installations. These regions are uniformly distributed along the cables and represent safe areas where the robots start and finish the installation sequence described in [41]. Moreover, considering the limited payload capacity of the ARMs; this is, the number of diverters that they can carry, a set of refilling stations are scattered around the power line. The visit to these stations puts the ARMs back into operation, restoring their payloads without mission interruptions.

It is worth mentioning that the planning approaches derived in this chapter are flexible enough to be adapted to other applications, being sensor deployments or load deliveries some examples. Similarly, the ARM-LA system has been selected to illustrate the bird diverter installation task but the derived approaches are also suitable for other kinds of ARMs.

This section follows with the presentation of the fundamentals of the planner operation (Section 6.2.1), an adaptation of the general method intended to reduce the energy consumption (Section 6.2.2) and a replanning strategy in case a particular ARM suffers a failure (Section 6.2.3).

6.2.1 Fundamentals of planner operation

This section describes a planning method that coordinates a team of ARMs with same capabilities and heterogeneous payload capacities to perform the bird diverter installation task introduced previously. To cope with that, Signal Temporal Logic (STL) [97] has demonstrated to be a suitable mathematical tool that can be used to encode complex temporal specifications into a single metric. Then, an optimisation problem can be solved to maximise the value of such metric. The results are feasible trajectories that are provided to the ARMs as tracking references. Accordingly, the motion planner, referred to as MP-MARM (Motion Planner for MultiARM systems) from here on, has been built over the fundamentals of STL for motion planning that are summarised in Annex C. The same nomenclature has been adopted.

Starting from the mission specifications, encoded in an STL formula φ, the optimisation problem in Annex C.5 can be set up for the smooth robustness $\tilde{\rho}_{\varphi}$ to compute feasible trajectories. It should be noted that, due to the need of ARM coordination for the installation task, there is a strong coupling between the ARM trajectories that forces to plan all of them in a single optimisation problem. Furthermore, a significant number of target regions and refilling stations might complicate the search of the optimal solution. Thus, the optimisation turns into a nonlinear non-convex min-max problem, solved using a NonLinear Programming (NLP) approach. As it is well known in the literature, these approaches suffer from the dependence on the initial guess, getting stuck on local optima if this is not suited.

The encoding of the mission specifications for the bird diverter installation task, as well as a methodology to compute an initial guess that eases the convergence of the optimisation problem towards the optimal solution are detailed hereinafter.

Encoding of mission specifications

From a motion planning perspective, the bird diverter installation task consists of a set of different requirements. Some of them are associated with the safety and must be fulfilled during all the operation time t_N. These requirements include staying within the workspace (φ_{ws}), while avoiding collisions with the obstacles there (φ_{obs}), and keeping safety distances between ARMs (φ_{dis}). Other requirements linked to the mission itself must be satisfied following a sequence or after certain events. In this sense, the team of ARMs must visit all the target regions once and stay there for the installation time t_{ins} (φ_{tr}). Additionally, due to the limited payload capacity, the robots need to visit and stay in the refilling stations for the time t_{rs} to be supplied with more bird diverters once run out on board (φ_{cap}). Finally, each ARM must fly to its nearest refilling station after completing the installation task (φ_{rs}). All these requirements can be encoded in the STL formula:

$$\varphi = \Box_{t_N} \varphi_{\text{ws}} \wedge \varphi_{\text{obs}} \wedge \varphi_{\text{dis}} \bigwedge \Diamond_{t_N} \left(\varphi_{\text{tr}} \wedge \varphi_{\text{cap}}\right) \mathcal{U} \varphi_{\text{rs}} \tag{6.1}$$

where $\wedge, \Box, \Diamond, \mathcal{U}$ are the operators And, Always, Sometime and Until, respectively, and:

$$\varphi_{\text{ws}} = {}^{d}p_k^{(j)} \in (\underline{p}_{\text{ws}}^{(j)}, \bar{p}_{\text{ws}}^{(j)}) \tag{6.2a}$$

$$\varphi_{\text{obs}} = \forall_{\text{obstacle}} \; {}^{d}p_k^{(j)} \notin (\underline{p}_{\text{obs}}^{(j)}, \bar{p}_{\text{obs}}^{(j)}) \tag{6.2b}$$

$$\varphi_{\text{dis}} = \forall_{\text{ARM n} \neq \text{d}} \sqrt{\sum_{j} \left({}^{d}p_k^{(j)} - {}^{n}p_k^{(j)}\right)^2} > \Gamma \tag{6.2c}$$

$$\varphi_{\text{tr}} = \forall_{\substack{\text{target}\\ \text{region}}} \Box_{t_{\text{ins}}} \; {}^{d}p_k^{(j)} \in (\underline{p}_{\text{tr}}^{(j)}, \bar{p}_{\text{tr}}^{(j)}) \tag{6.2d}$$

$$\varphi_{\text{cap}} = \Box_{t_{\text{rs}}} (q_d == 0) \; {}^{d}p_k^{(j)} \in (\underline{p}_{\text{rs}}^{(j)}, \bar{p}_{\text{rs}}^{(j)}) \tag{6.2e}$$

$$\varphi_{\text{rs}} = {}^{d}p_k^{(j)} \in (\underline{p}_{\text{rs}}^{(j)}, \bar{p}_{\text{rs}}^{(j)}) \tag{6.2f}$$

Equation (6.2a) imposes the position of the ARM d at time instant t_k along the j-axis ($j = \{1, 2, 3\}$) of the inertial reference frame; this is ${}^{d}p_k^{(j)}$, to stay within the boundaries of the workspace area, with $\underline{p}_{\text{ws}}^{(j)}$ and $\bar{p}_{\text{ws}}^{(j)}$ denoting its minimum and maximum values along the j-axis of the reference frame. Similarly, equations (6.2b), (6.2d), (6.2e) and (6.2f) set the obstacle avoidance, bird diverter installation, payload capacity and mission completion specifications, respectively, where q_d represents the number of bird diverters on board the ARM d and $\underline{p}_{\text{obs}}^{(j)}$, $\underline{p}_{\text{tr}}^{(j)}$, $\underline{p}_{\text{rs}}^{(j)}$, $\bar{p}_{\text{obs}}^{(j)}$, $\bar{p}_{\text{tr}}^{(j)}$ and $\bar{p}_{\text{rs}}^{(j)}$ are the vertex positions along the j-axis of the reference frame of bounding boxes delimiting the obstacles, the target regions and the refilling stations. Finally, equation (6.2c) is the safety distance requirement of the ARM d with respect to the rest of ARMs n ($n \neq d$), with the square-root term representing the distance $\gamma_{d,n}$ between the pair of ARMs $\{d, n\}$ and Γ being its minimum admissible value. It should be noted that the notation in the previous equations has been particularised for the single ARM d to make them more readable. However, the formulation can be extended to the complete team of ARMs with an appropriate set of operators and indices.

Initial guess

In order to ease the convergence of the optimisation problem towards the optimal solution, this section presents a method to compute an initial guess $\mathbf{x}'$ sufficiently close to the optimum. The method is built over a high-level abstraction of the bird diverter installation problem, neglecting obstacles and other safety requirements. More in detail, a simplified graph-based representation of the problem is addressed firstly. Then, a Mixed-Integer Linear Programming (MILP) encoding works upon the graph to find the optimal assignment and sequence of target regions for each ARM. Finally, dynamically-feasible ARM trajectories are retrieved according to such assignment and sequence.

The graph that models the simplified problem (see Fig. 6.2) is an undirected weighted multigraph given by the tuple $G = (\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{D}, \mathcal{Q})$, where $\mathcal{V}$ is the set of graph vertices, $\mathcal{E}$ is the set of edges connecting the vertices, $\mathcal{W}$ is the set of weights corresponding to such edges, $\mathcal{D}$ is the set of δ ARMs used to cover the graph and $\mathcal{Q}$ is the set of maximum payload capacities associated with each ARM d in $\mathcal{D}$; these are, the maximum number of bird diverters Q_d that they can load. At the same time, $\mathcal{V} = \{\mathcal{O}, \mathcal{R}, \mathcal{T}\}$, with $\mathcal{O}$ embedding into a single vertex (vertex 0) the set of depots where each robot is located at the beginning of the mission (time t_0), $\mathcal{R}$ the set of refilling stations, and $\mathcal{T}$ the set of target regions. Concerning the graph connectivity, all the vertices in $\mathcal{T}$ are connected to each other and to every vertex in $\{\mathcal{O}, \mathcal{R}\}$ by δ edges each pair. Since ARMs with homogeneous constraints in terms of maximum velocities and accelerations are assumed, the weights of these connections can be approximated by the Euclidean distance, which means that the weight from vertex $i \in \mathcal{V}$ to vertex $j \in \mathcal{V}$, with $i \neq j$, using the ARM $d \in \mathcal{D}$ fulfils $\{w_{ij|d}, w_{ji|d} \in \mathcal{W} : w_{ij|d} = w_{ji|d}\}$. It should be noted that the weight $w_{ij|r}$ is not necessarily equal to $w_{ij|p}$, with $\{r, p\} \in \mathcal{D}$, because a single vertex in $\{\mathcal{O}, \mathcal{R}\}$ may encode different physical locations for each ARM.

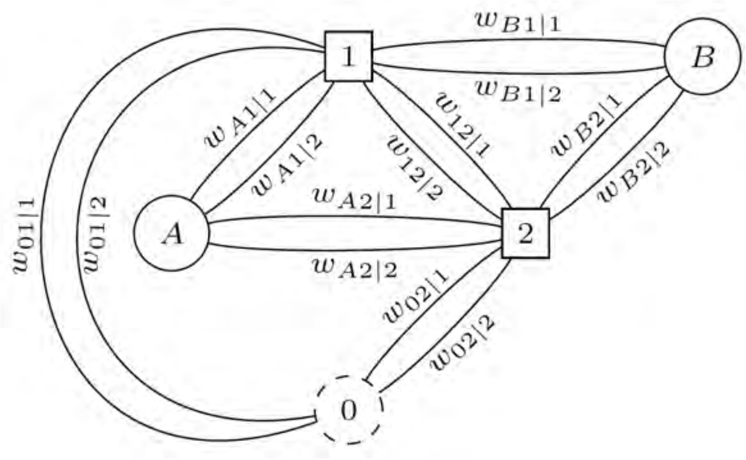

Figure 6.2: Graph-based representation of the bird diverter installation problem. Example of graph G built upon the assumption of two target regions (square vertices), two refilling stations (round solid vertices) and two ARMs. The depots are embedded into the round dashed vertex.

The MILP encoding to find the optimal assignment and sequence of target regions over the graph G aims at minimising the mission time. For that, the total distance covered by the multiARM team should not only be minimised but also distributed evenly between the robots as the mission finishes with the last ARM completing its sequence. Under this optimisation objective, the use of all the available ARMs is considered.

Associated with the edges in $\mathcal{E}$, $\mathcal{Z}$ can be defined as a set of integer variables $z_{ij|d}$, where i and j $(i \neq j)$ represent vertices in $\mathcal{V}$, and d corresponds to ARMs in $\mathcal{D}$, so that $\{z_{ij|d}, z_{ji|d} \in \mathcal{Z} : z_{ij|d} = z_{ji|d}\}$. These variables $z_{ij|d}$ encode the number of times that their associated edges are selected in the MILP solution. Thus, $z_{ij|d} \in \{0, 1\}$ if $\{i, j\} \in \{\mathcal{O}, \mathcal{T}\}$, and $z_{ij|d} \in \{0, 1, 2\}$ if $i \in \mathcal{R}$ and $j \in \mathcal{T}$. In this sense, $z_{ij|d} = 2$ corresponds to a return trip between a refilling station i and a target region j. Additionally, variables σ_{rp}, with $\{r, p\} \in \mathcal{D}$ and $r \neq p$, quantify the difference between the total distances covered by ARMs r and p. These variables are defined to be real and positive in the interval $[0, \sigma_{max}]$, being $\sigma_{max} \geq 0$ their maximum acceptable value.

With all the previous definitions, the MILP problem can be formulated as follows:

$$\underset{z_{ij|d},\sigma_{rp},y_{j|d}}{\text{minimise}} \sum_{\{i,j\}\in\mathcal{V},\, i\neq j,\, d\in\mathcal{D}} w_{ij|d}\, z_{ij|d} + \sum_{\{r,p\}\in\mathcal{D},\, r\neq p} \sigma_{rp} \tag{6.3a}$$

$$\text{s.t.} \sum_{i\in\mathcal{V},\, i\neq j,\, d\in\mathcal{D}} z_{ij|d} = 2,\ \forall j \in \mathcal{T} \tag{6.3b}$$

$$\sum_{i\in\mathcal{V},\, i\neq j} z_{ij|d} = 2y_{j|d},\ \forall j \in \mathcal{T},\ \forall d \in \mathcal{D} \tag{6.3c}$$

$$\sum_{i\in\mathcal{T}} z_{0i|d} = 1,\ \forall d \in \mathcal{D} \tag{6.3d}$$

$$\sum_{i\in\mathcal{S},\, j\notin\mathcal{S},\, d\in\mathcal{D}} z_{ij|d} \geq 2h\left(\mathcal{S}\right),\ \forall \mathcal{S} \subset \mathcal{T} \tag{6.3e}$$

$$\sum_{\{i,j\}\in\mathcal{V},\, i\neq j} \left(w_{ij|r}\, z_{ij|r} - w_{ij|p}\, z_{ij|p}\right) - \sigma_{rp} \leq 0 \tag{6.3f}$$

$$\sum_{\{i,j\}\in\mathcal{V},\, i\neq j} \left(w_{ij|p}\, z_{ij|p} - w_{ij|r}\, z_{ij|r}\right) - \sigma_{rp} \leq 0 \tag{6.3g}$$

$$0 \leq \sigma_{rp} \leq \sigma_{\max},\ \forall \{r,p\} \in \mathcal{D} \text{ with } r \neq p$$

where equation (6.3a) is the objective function, consisting of two terms that encode the total distance covered by the multiARM team and a penalty for uneven distance distribution between robots. Constraints (6.3b) and (6.3c) force that each target region is visited exactly once, being $y_{j|d} \in \{0, 1\}$ auxiliary integer variables guaranteeing that, if ARM $d \in \mathcal{D}$ reaches the target region $j \in \mathcal{T}$, the same ARM must leave it. Constraints (6.3d) ensure that each ARM starts the mission from its depot. Constraints (6.3e) prevent the formation of tours exceeding the payload capacities of the ARMs or with no connection between the depot vertex and a refilling station (it should be remembered that each ARM must fly to a refilling station after completing the installation task). For that, the function $h\left(\mathcal{S}\right)$ should impose the minimum number of times that the robots must enter and leave every subset $\mathcal{S} \subset \mathcal{T}$. However, since the number of this kind of constraints can be excessive, they can be omitted initially and added dynamically as they are broken [98]. In addition, to speed up the convergence, the following capacity constraints can also be added to the formulation:

$$\frac{1 + \sum_{i \in \mathcal{R}, j \in \mathcal{T}} z_{ij|d}}{2} \geq \left\lceil \frac{1 + \sum_{\{i,j\} \in \mathcal{T}, i \neq j} z_{ij|d}}{Q_d} \right\rceil, \ \forall d \in \mathcal{D} \tag{6.4}$$

which put a lower bound on the number of tours for each ARM. Finally, Constraints (6.3f) and (6.3g) balance the distance covered by each ARM as much as possible.

Once the MILP problem is solved, the selected edges in the graph G are known (they fulfil $z_{ij|d} \neq 0$) and with them, the optimal assignment and sequence of regions (depots, target regions and refilling stations) can be deduced. Then, the motion primitives presented in [99] are used to retrieve dynamically-feasible ARM trajectories between those regions. These motion primitives are computed fixing rest-to-rest motion between regions; this is, zero velocity and acceleration in such regions, and imposing the minimum feasible time that allows the fulfilment of the desired maximum values of velocity $v_{max}^{(j)}$ and acceleration $a_{max}^{(j)}$ along the j-axis ($j = \{1, 2, 3\}$) of the inertial reference frame. Installation times t_{ins} and refilling times t_{rs} with the ARMs stopped in the corresponding regions are also considered between the trajectories connecting the regions. Finally, the initial guess $\mathbf{x}'$ consists of the combination of previous trajectories and waiting manoeuvres.

6.2.2 Energy awareness

The fundamentals of the MP-MARM planning approach presented in Section 6.2.1 can be extended to consider a reduction in the energy consumption of the team of ARMs without disregarding the mission requirements. This extension accounts for the ARM aerodynamics to benefit trajectories whose velocities are prone to restrain the electric power demanded by the motors.

According to the fundamentals of Helicopter Aerodynamics [90], the electric power required by a rotary-wing vehicle decreases with increasing low forward velocities. In contrast, this power increases dramatically at high velocities due to parasitic losses.

The combination of both effects leads to the existence of an optimal value for the forward velocity that maximises the time that the vehicle can fly without changing batteries or, in other words, minimises its energy consumption for a specific flight time. Based on [90], an estimation of this optimal forward velocity v_∞^* adapted to multirotors is given by:

$$v_\infty^* = \sqrt{\frac{mg}{\varrho A}} \left(\frac{\kappa A}{3 n_r f} \right)^{1/4} \tag{6.5}$$

where m is the mass of the robot, n_r is its number of rotors, A is the rotor disk area, f is the equivalent flat plate area of the fuselage, κ is an induced power correction factor, ϱ is the air density and $g = 9.81 m/s^2$ is the gravity acceleration.

The previous fact allows a reformulation of the optimisation problem in Annex C.5 by including an energy term:

$$\begin{aligned} \underset{p_k^{(j)}, v_k^{(j)}, a_k^{(j)}}{\text{maximise}} \quad & \tilde{\rho}_\varphi(p_k^{(j)}, v_k^{(j)}, t_k) - \frac{\eta}{N} \sum_k \left(1 - \frac{v_{\infty,k}}{v_\infty^*} \right)^2 \\ \text{s.t.} \quad & |v_k^{(j)}| \leq v_{max}^{(j)}, |a_k^{(j)}| \leq a_{max}^{(j)} \\ & \mathbf{S_k^{(j)}}, \forall k = \{0, 1, \ldots, N-1\} \end{aligned} \tag{6.6}$$

where η is a parameter representing the weight of the energy term in the objective function and $v_{\infty,k}$ is the forward velocity at time instant t_k. This velocity is related to $v_k^{(j)}$ by the following equation:

$$v_{\infty,k} = \sqrt{\left(v_k^{(1)} \right)^2 + \left(v_k^{(2)} \right)^2} \tag{6.7}$$

It may happen that v_∞^* is higher than the maximum forward velocity $v_{\infty,max}$ obtained after imposing the desired maximum values of velocity $v_{max}^{(1)}$ and $v_{max}^{(2)}$ in equation (6.7). In that case, the velocity $v_{\infty,max}$ should replace v_∞^* in the optimisation problem (6.6) as the feasible forward velocity closest to v_∞^*.

The new formulation favours the computation of ARM trajectories that, when it is possible, lead the robots to fly at velocities close to the one minimising the energy consumption. The cost to be paid is a decrease in the robustness ρ_φ, but always complying with the mission requirements.

6.2.3 Event-based replanning

The MP-MARM motion planner provides feasible trajectories for the multiARM team that allow performing the bird diverter installation task successfully. However, a disruption in the service of a particular ARM might happen during the mission execution, putting it at risk. Alternatively, a back-up ARM could continue with the interrupted service of the faulty ARM, respecting the operation of the remaining robots. In an attempt to deal with this contingency, the application of the MP-MARM approach for event-based replanning is introduced.

Specifically, when the disruption is detected at time $t_{dis} \in [t_0, t_N]$, the replanning is triggered while the rest of the functional ARMs continue their operation. Then, the optimal planning process presented in Section 6.2.1 is addressed. However, for replanning purposes, there are several differences in the application of the MP-MARM planner. Firstly, only the trajectory of the back-up ARM needs to be computed since the trajectories of the rest of robots can still be assumed appropriate. Consequently, the replanner uses as input the part of such trajectories in the time interval $[t_{dis}+\Delta t_{rep}, t_N]$ being Δt_{rep} the maximum time expected to compute the replanning. This time fixes the start of the operation for the back-up ARM. Secondly, only the target regions assigned to the disrupted robot that have not been visited yet are considered in the replanning. The previous sequence should not be used because the new ARM may start from a position and with a payload that make this sequence suboptimal or even unfeasible. Finally, it is worth mentioning that the replanning time is considerably lower than the planning time for the entire mission since the replanning computation is carried out only for one ARM and a smaller set of target regions.

6.3 Simulation results

To demonstrate the validity and the effectiveness of the planning approaches presented in Section 6.2, the motion planning for the installation of bird flight diverters on power lines using a team of ARMs has been tested firstly through simulation.

For the installation task, a team of ARMs composed by three ARM-LA systems (see Annex A.5) has been considered, acting one of them as back-up robot. Table 6.1 reports the value of their parameters used to compute the planned trajectories. It should be noted that the installation time t_{ins} and the refilling time t_{rs} have been set using short symbolic times in these simulations to avoid analysing trajectories with excessively long waits. Turning to the ARM heading angles, which are not considered within the optimisation process of the motion planner, these are adjusted to have the ARMs aligned either with the displacement directions or with the cables during the diverter installations. Thus, a similar approach to the one presented in Section 5.2.1 (Hybrid planning space) has been followed, where the cables act here like the pipes in such section.

Table 6.1: Parameters of the multiARM team used for simulation.

Parameter	Value	Units
Payload capacity ARM 1 (Q_1)	2	–
Payload capacity ARM 2 (Q_2)	3	–
Payload capacity back-up ARM 3 (Q_3)	2	–
Maximum velocity ($v_{max}^{(j)}$)	3.1	m/s
Maximum acceleration ($a_{max}^{(j)}$)	2	m/s^2
Optimal forward velocity (v_{∞}^{*})	4.1	m/s
Minimum mutual safety distance (Γ)	3	m
Installation time (t_{ins})	5	s
Refilling time (t_{rs})	2	s

The MP-MARM motion planner has been coded using MATLAB [7], with the MILP problem to compute the initial guess formulated using the CVX framework [100] and the optimisation process using the CasADi tool [101] and Ipopt as solver [102]. Additionally, a MATLAB representation framework provides the graphical evolution of the variables associated with the multiARM system and thanks to this, intuitive 3D diagrams can also be depicted. These diagrams superimpose the planned trajectories over the environment, which includes the power line and all the regions of interest; these are, the target regions, the refilling stations and the initial positions. The previous graphical outputs will be used throughout this section to illustrate the obtained results.

In general terms, the validation tests have put the focus on the planned trajectories. In this sense, the motion planner has been executed to generate motion plans and then, the MATLAB representation framework is used to analyse them. At this stage, the dynamic simulation of the controlled team of ARMs is not considered. Once the planned trajectories have been validated, simulations in the Gazebo robotics simulator [8] can be performed. The purpose is to verify the feasibility in the execution of the trajectories under realistic conditions, exploiting the benefits of software-in-the-loop simulations [103].

6.3.1 Fundamentals of planner operation

The first part of the simulations is focused on the MP-MARM method in its standard operation. For that, the fundamentals of the motion planner presented in Section 6.2.1 have been tested for the installation of bird flight diverters in a segment of power line. The scenario is presented with more details below. These simulations aim to demonstrate:

- The compliance of the planned trajectories with the mission requirements and the feasibility in the execution of such trajectories.
- The effectiveness of initialising the MP-MARM planner with the initial guess computed according to Section 6.2.1 (Initial guess) and why this initial guess can not be used directly as a solution for the installation task.

Application scenario: bird diverter installation task

The application scenario that has been used for all the validation tests along this chapter is presented in Fig. 6.3. Here, the team of ARMs should perform the installation task described at the beginning of Section 6.2. For this particular scenario, seven target regions (represented in blue and labelled with the code $TR\ i$, being i the identifier number of the region) uniformly distributed over the upper cables have been considered. Also, four refilling stations (represented in green and labelled with the code $RS\ j$, being j the identifier letter of the station) located close to the corners of the workspace ($[50m \times 20m \times 15m]$ with the segment of power line centred on it) are available to restore the ARM payloads. It should be noted that each refilling station consists of two regions, one per functional ARM, to avoid potential conflicts in case that two ARMs need to visit the same refilling station at the same time. Finally, the initial positions of the ARMs are represented in pink and labelled with the code $IP\ k$, being k the identifier number of the associated ARM. Both the two towers and the six cables are treated as obstacles to be avoided.

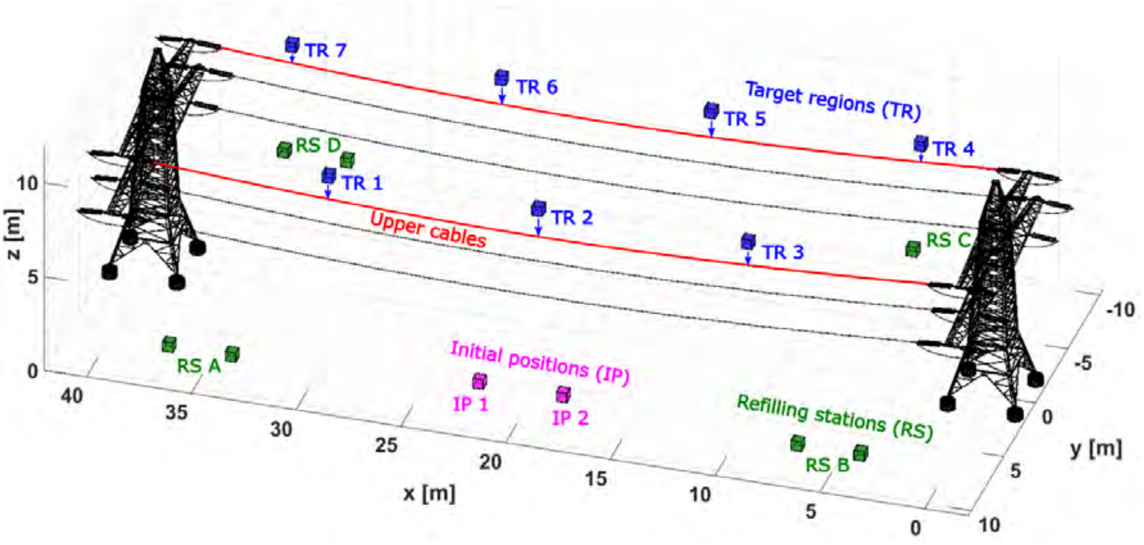

Figure 6.3: Application scenario given by a bird diverter installation task.

Analysis of the results: MP-MARM algorithm

The ARM trajectories planned with the MP-MARM algorithm have been showcased in Fig. 6.4 while the evolution along time of the main variables involved in the planning process has been depicted in Fig. 6.5. According to these figures, the multiARM system satisfies all the mission requirements. Particularly, from the point of view of the safety, there is not any collision with the environment and the ARMs stay in the workspace while they keep a mutual safety distance $\gamma_{1,2}$ that is always greater than the minimum admissible value Γ. Moreover, the desired maximum values of velocity and acceleration $v_{max}^{(j)}, a_{max}^{(j)}$ along the j-axis ($j = \{1, 2, 3\}$) of the inertial reference frame are never exceeded. Concerning the installation task itself, all the target regions have been visited, remaining in them during the installation time t_{ins} (see the shaded blue slots in Fig. 6.5). Also, these target regions have been properly assigned by the planner in order to minimise and balance the operation time per ARM. Similarly, the robots follow efficient and feasible sequences connecting the initial positions, the target regions and the refilling stations. Thus, the ARMs never visit a target region once run out of diverters. Instead of that, a direct flight to a refilling station allows restoring the number of bird diverters necessary to continue with the mission, which is never greater than the maximum payload capacities Q_1, Q_2. For these manoeuvres, the ARMs remain in the refilling stations during the refilling time t_{rs} (see the shaded green slots in Fig. 6.5). To conclude the mission, each ARM finishes in its nearest refilling station.

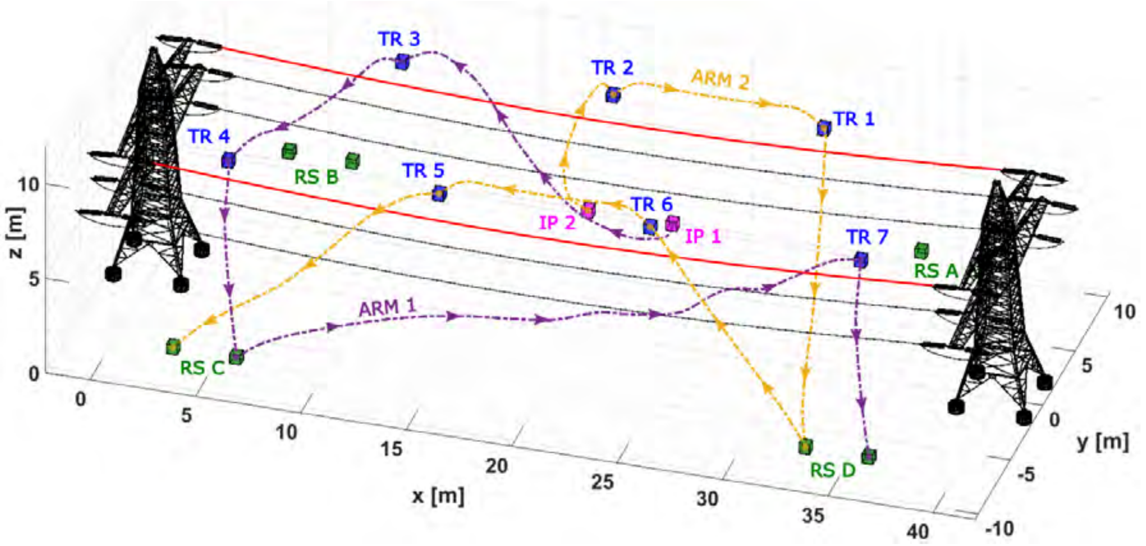

Figure 6.4: Bird diverter installation task. ARM trajectories planned with the MP-MARM algorithm.

Once the compliance of the planned trajectories with the mission requirements has been validated, a simulation in the Gazebo robotics simulator has been performed to verify the feasibility in the autonomous execution of such trajectories. Fig. 6.6 collects some snapshots of the simulation. Additionally, the complete video *MP-MARM_Gazebo_bird_diverter_installation.mp4* can be downloaded from the provided

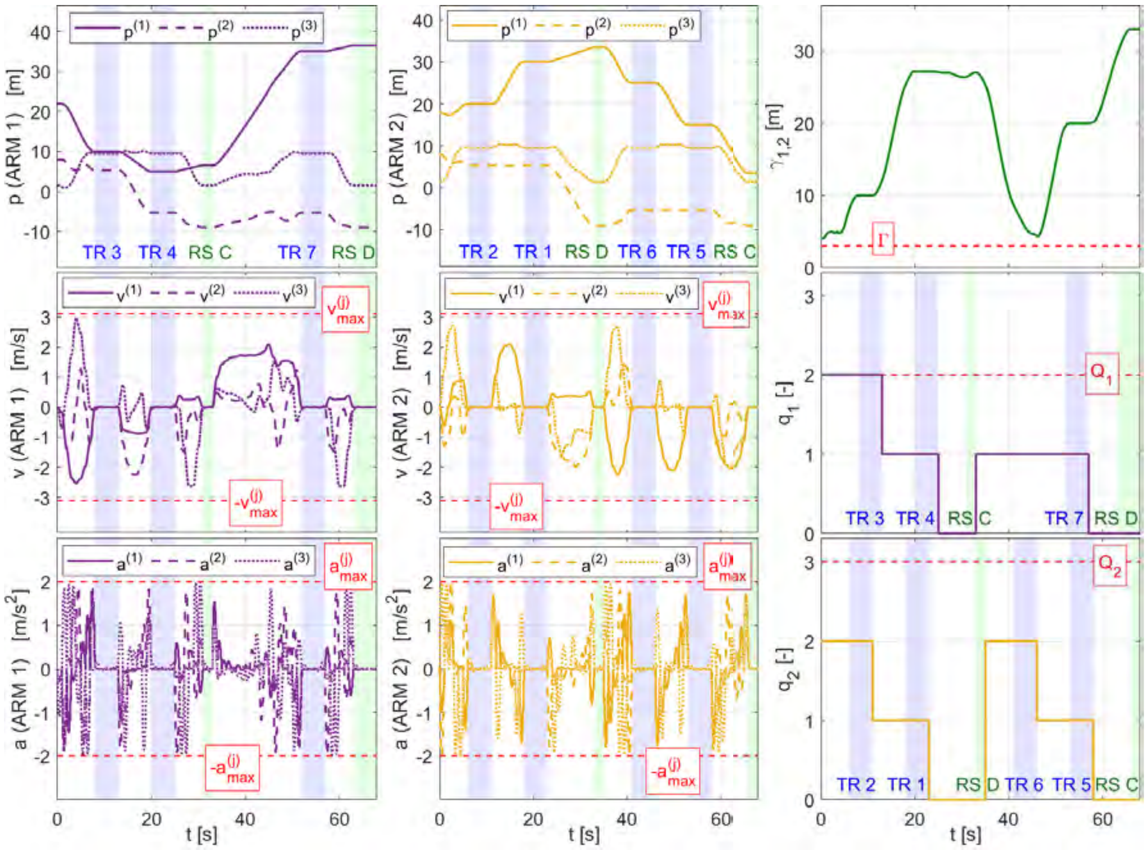

Figure 6.5: Bird diverter installation task. Evolution of the ARM variables: trajectories planned using the MP-MARM algorithm. The installation and refilling time windows are shaded using blue and green slots, respectively, and the dashed red lines highlight the admissible limits in the variables.

multimedia content, as indicated in Section 1.4. The analysis of these results shows that the planned trajectories lead the team of ARMs to perform the installation task successfully.

Analysis of the results: comparison with the initial guess

The proper results obtained above using the MP-MARM motion planner are partially due to the suitability of an initial guess computed according to Section 6.2.1 (Initial guess). In fact, preliminary tests showed that the optimisation process, motivated by its NP-hard structure, hardly converges to a feasible solution with moderated numbers

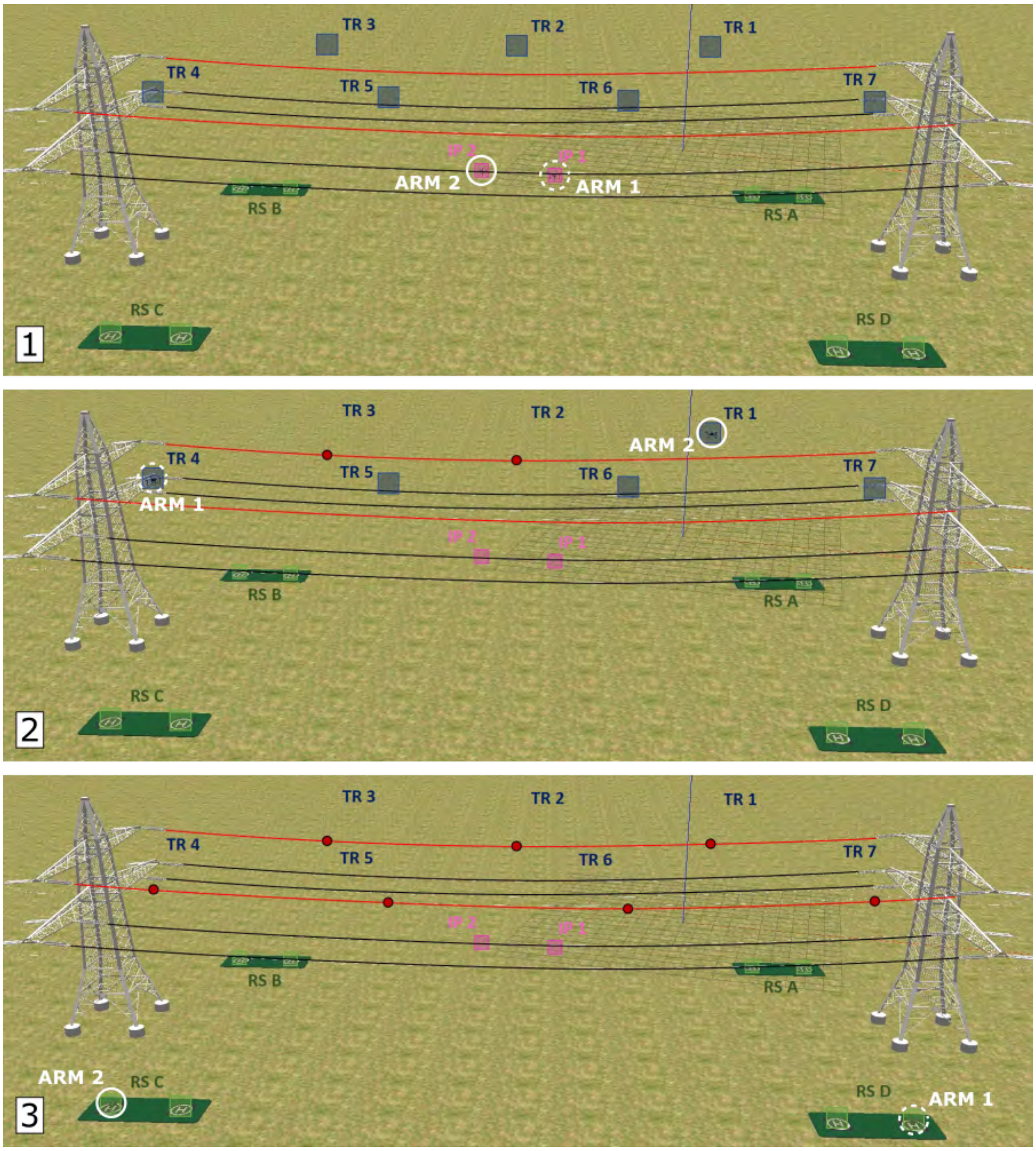

Figure 6.6: Bird diverter installation task. Snapshots of the simulation in Gazebo. The white circles highlight the ARMs while the red spheres symbolise the installed bird diverters.

of target regions and refilling stations if the initial guess is not suited. Moreover, as an MILP formulation lies behind the computation of the initial guess, its computation time is almost insignificant with respect to the time required to solve the full optimisation process using NLP approaches. In this sense, the computation of the initial guess takes around 4% of the total computation time for the application scenario under analysis. Consequently, the initial guess, represented in Fig. 6.7, provides a fast solution which is sufficiently close to the optimal one. After that, the MP-MARM planner continues working to refine the initial guess until the maximum robustness is reached.

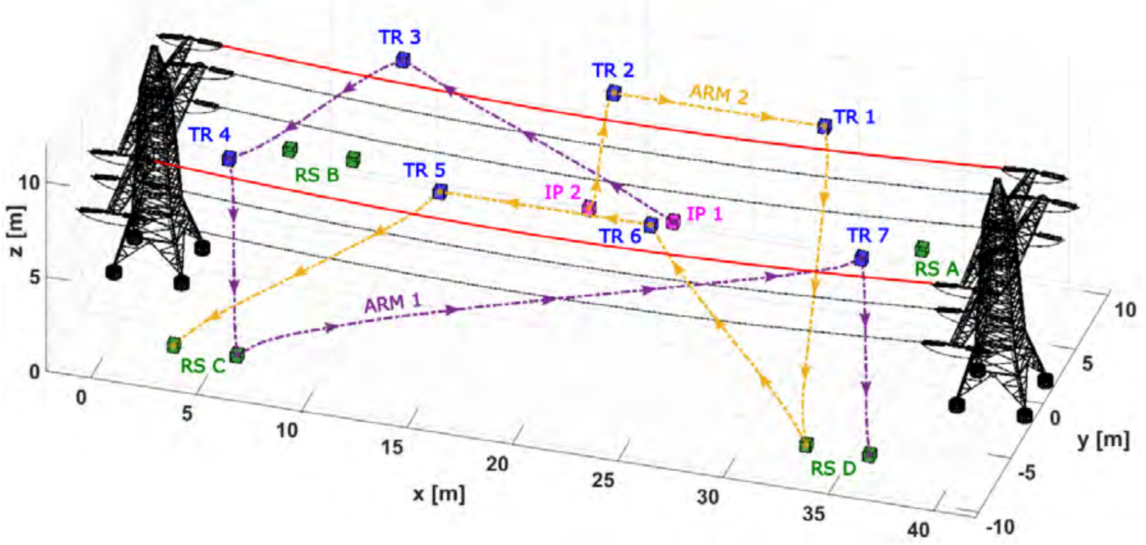

Figure 6.7: Bird diverter installation task. Initial guess used to compute the ARM trajectories represented in Fig. 6.4.

However, the initial guess can not be used directly as a solution for the bird diverter installation task since it mainly focuses on computing efficient and feasible sequences of target regions and refilling stations subject to payload constraints. In contrast, the initial guess does not consider any safety requirement like obstacle avoidance or minimum safety distance between ARMs. Hence, ARMs executing the initial guess could collide with the power towers or the cables. Regarding the minimum mutual distance, while this constraint can be achieved in the optimisation process of the MP-MARM planner by regulating the ARM positions, velocities and accelerations without affecting the optimal sequence of regions, this does not apply to the approach that computes the initial guess. To illustrate all the above, Fig. 6.8 depicts a comparison between the initial guess and the final solution computed by the MP-MARM planner (also represented in Fig. 6.4) from a point of view aligned with the x-axis of the scenario. As can be seen, it is evident how the ARM trajectories are adapted to reach higher levels of robustness. In particular, the MP-MARM planner needs to adjust the initial guess to achieve both obstacle avoidance with the cables and mutual distance requirements near the initial points while keeping the rest of mission specifications. Furthermore, the enhanced trajectories try to maximise their distances with the cables even in segments that do not produce any collision.

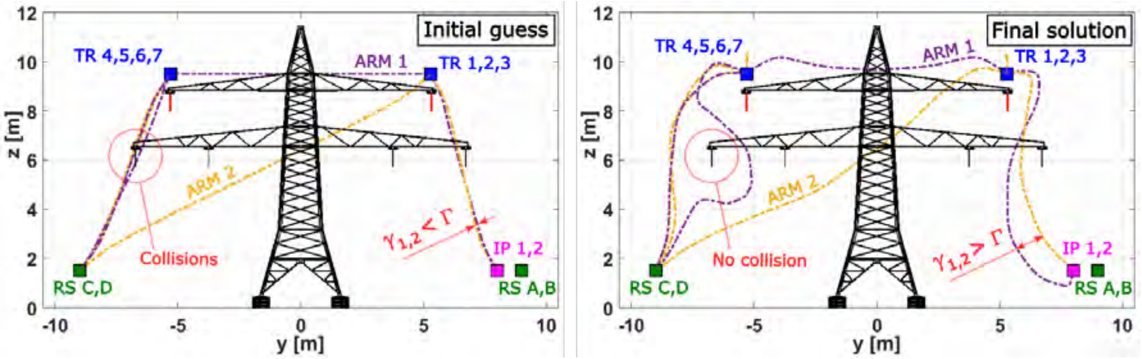

Figure 6.8: Bird diverter installation task. Comparison between the initial guess (left) and the final solution computed by the MP-MARM planner and also represented in Fig. 6.4 (right).

6.3.2 Energy awareness

This section aims to demonstrate the benefits of the MP-MARM motion planner endowed with the energy-aware feature formulated in Section 6.2.2. This extension accounts for the ARM aerodynamics to plan trajectories whose velocities help to reduce the energy consumption of the robots during the bird diverter installation task. For comparison purposes with respect to previous results, the same application scenario described in Section 6.3.1 (Application scenario: bird diverter installation task) has been considered.

Analysis of the results

The new trajectories for the multiARM system planned with the MP-MARM algorithm including energy awareness have been represented in Fig. 6.9. For these trajectories, all the benefits analysed in Section 6.3.1 (Analysis of the results: MP-MARM algorithm) when the plan is computed with the MP-MARM planner in its standard operation (see Fig. 6.4) are kept. In contrast, the main differences lie in the length and shape of them. Since the mission time is the same in both cases, the new longer trajectories should help to increase the forward velocity, making it closer to its optimal value; this is, v_∞^*. At this point, it should be remembered that this optimal forward velocity v_∞^* is the one that minimises the energy consumption as it was explained in Section 6.2.2.

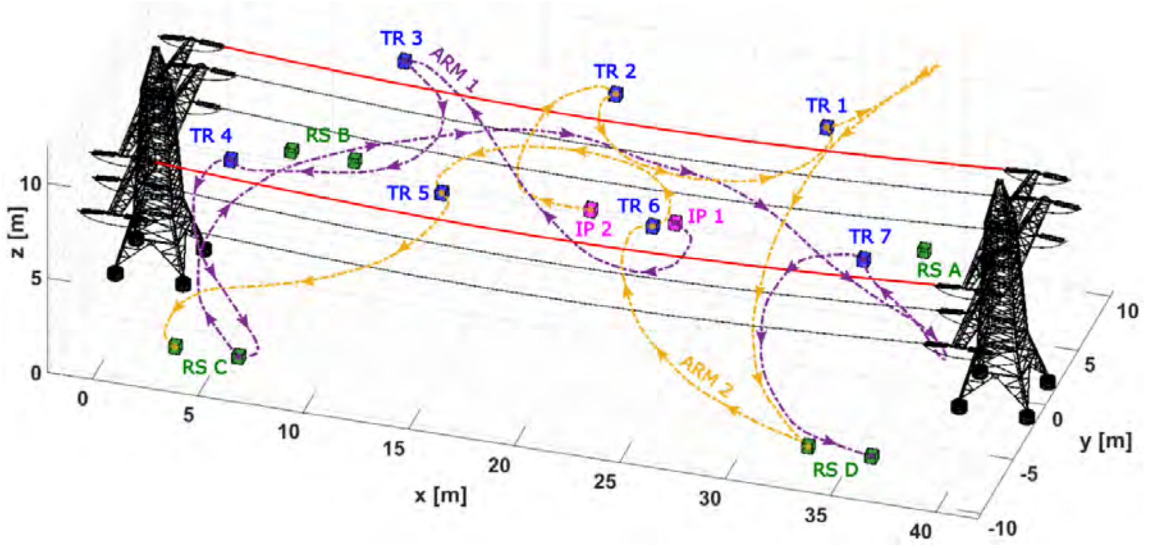

Figure 6.9: Bird diverter installation task. ARM trajectories planned with the MP-MARM algorithm including energy awareness.

For a better analysis of the previous evidence, Fig. 6.10 shows a comparison between the evolution of the forward velocities when the associated trajectories are planned using the MP-MARM algorithm in its standard operation and including energy awareness. In this figure, zero-velocity slots correspond to the unavoidable visits of the ARMs to the target regions and refilling stations. As it was expected, the forward velocities considering energy awareness reach values closer to v_∞^*, which allow inferring a reduction in the energy consumption. However, this improvement comes at the expense of the robustness ρ_φ. Thus, its resulting value experiences a reduction of 30% with respect to the robustness obtained with the MP-MARM algorithm in its standard operation, which means smaller safety distances of the ARMs with the obstacles and between them. Nevertheless, the robustness remains positive and consequently, all the mission specifications are still fulfilled (see Annex C.3).

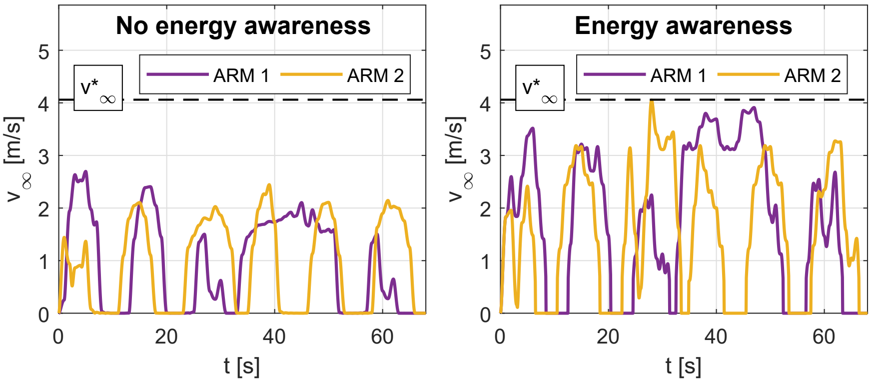

Figure 6.10: Bird diverter installation task. Comparison between the evolution of the forward velocities planned using the MP-MARM algorithm in its standard operation (left) and including energy awareness (right). The dashed black lines highlight the optimal forward velocity.

6.3.3 Event-based replanning

To conclude with the simulation results, this section is intended to demonstrate the applicability of the MP-MARM motion planner to replan missions in the event of ARM failures as it was expounded in Section 6.2.3. For this purpose, the same application scenario described in Section 6.3.1 (Application scenario: bird diverter installation task) has been adopted once more.

Analysis of the results

Using the results in Section 6.3.1 (Analysis of the results: MP-MARM algorithm) as starting point (see Fig. 6.4), a disruption in the service of the ARM 2 is simulated when it arrives at the refilling station *RS D* to load more bird diverters. Triggered by this event, the MP-MARM algorithm is applied to replan the bird diverter installation task by computing a feasible trajectory for the back-up ARM 3, which is initially at disposal in the refilling station *RS A*. This trajectory will allow the back-up ARM 3 to continue without interruptions with the remaining mission of the faulty ARM 2 while respecting the ongoing operation of the functional ARM 1.

The resulting trajectories after replanning are depicted in Fig. 6.11. According to the results, the installation task can be completed successfully thanks to the support of the back-up ARM 3 and the replanning capabilities. In particular, the analysis carried out in Section 6.3.1 (Analysis of the results: MP-MARM algorithm) when the plan progressed without any unforeseen circumstance is also valid here. Hence, all the mission specifications are still satisfied. Moreover, the disruption has not introduced a significant delay (only two seconds) in the attainment of the mission. As a conclusion, the good performance of the MP-MARM algorithm to ensure a proper mission continuity against contingencies in the service of a particular ARM has been demonstrated.

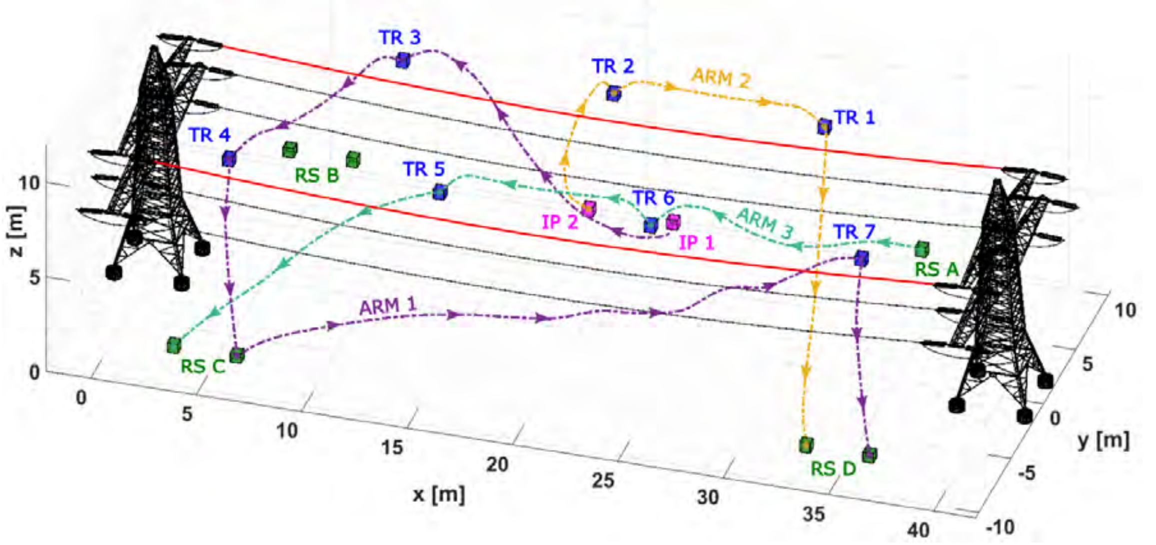

Figure 6.11: Bird diverter installation task. ARM trajectories after replanning the task in Fig. 6.4 by applying the MP-MARM algorithm when the ARM 2 suffers a failure in the refilling station *RS D*. The ARM 1 continues its ongoing operation while the back-up ARM 3 completes the remaining mission of the faulty ARM 2.

6.4 Experimental results

Simulation results have evinced the suitability of the motion planning strategies presented in Section 6.2 to plan ARM trajectories for the autonomous installation of bird flight diverters on power lines. Now, to complete the validation procedure, the viability of the usage of such trajectories in real-world outdoor environments should be demonstrated. In consequence, field experiments have been performed in this section.

In the search of high levels of safety, the use of the ARM-LA system presented in Annex A.5 has been replaced by customised DJI F450 quadrotors [104], whose reliability has been widely validated. Since the diverter installations themselves will not be carried out, the use of these robots shall suffice to satisfactorily complete the first experiments. Moreover, the desired maximum values of velocity and acceleration $v_{max}^{(j)}, a_{max}^{(j)}$ along the j-axis ($j = \{1, 2, 3\}$) of the inertial reference frame have been relaxed with respect to the simulations in previous section to favour smooth movements. Table 6.2 reports the value of the parameters that have been modified for experimentation when comparing with simulation (see Table 6.1). In this table, it should be noted how the refilling time t_{rs} has been increased because real landing and take-off manoeuvres have been conducted in the refilling stations.

Table 6.2: Parameters of the multiARM team used for experimentation.

Parameter	Value	Units
Maximum velocity ($v_{max}^{(j)}$)	1.2	m/s
Maximum acceleration ($a_{max}^{(j)}$)	1	m/s^2
Refilling time (t_{rs})	12	s

Aligned also with the safety principle, the flight experiments have been performed in a large outdoor area without obstacles located in the facilities of the Multi-Robot Systems Group (MRS) at the Czech Technical University in Prague (CTU) [5]. Thus, the crucial service of a real power line is not compromised in the case that any deviation from the expected operation is produced. Nevertheless, the application scenario has been integrated in the flight area using augmented reality in order to ease the analysis of the obtained results.

Concerning the execution of the experiments, the MP-MARM motion planner has been exploited to generate feasible motion plans and then, the output trajectories have been sent to the robots for their autonomous tracking. Finally, the MATLAB representation framework mentioned in Section 6.3 has been used to analyse the results.

6.4.1 Fundamentals of planner operation

Since the main objective of the flight experiments is to demonstrate the viability of the usage of the planned trajectories in real-world outdoor environments, the experiments shown in this chapter have been concentrated on the execution of trajectories computed using the MP-MARM planner in its standard operation. Nevertheless, similar behaviours have been found in the execution of trajectories that have been computed by the motion planner either including energy awareness or applied for event-based replanning.

For comparison purposes with respect to the simulation results, the same application scenario described in Section 6.3.1 (Application scenario: bird diverter installation task) has been considered again. In fact, the experiments have been addressed as an extension to the real world of the Gazebo simulation presented in Section 6.3.1 (Analysis of the results: MP-MARM algorithm).

Analysis of the results

Fig. 6.12 depicts the real autonomous execution of the trajectories planned with the MP-MARM algorithm in its standard operation for the bird diverter installation task. At the same time, Fig. 6.13 shows the associated evolution along time of the main variables involved in the mission execution. As can be analysed, these experimental results are quite similar to the simulation results presented in Section 6.3.1 (Analysis of the results: MP-MARM algorithm). In fact, the same conclusions extracted there in terms of safety, efficiency and feasibility can also be applied here. Consequently, the planned trajectories lead the team of robots to complete the installation task successfully, which is the direct effect of fulfilling all the mission specifications.

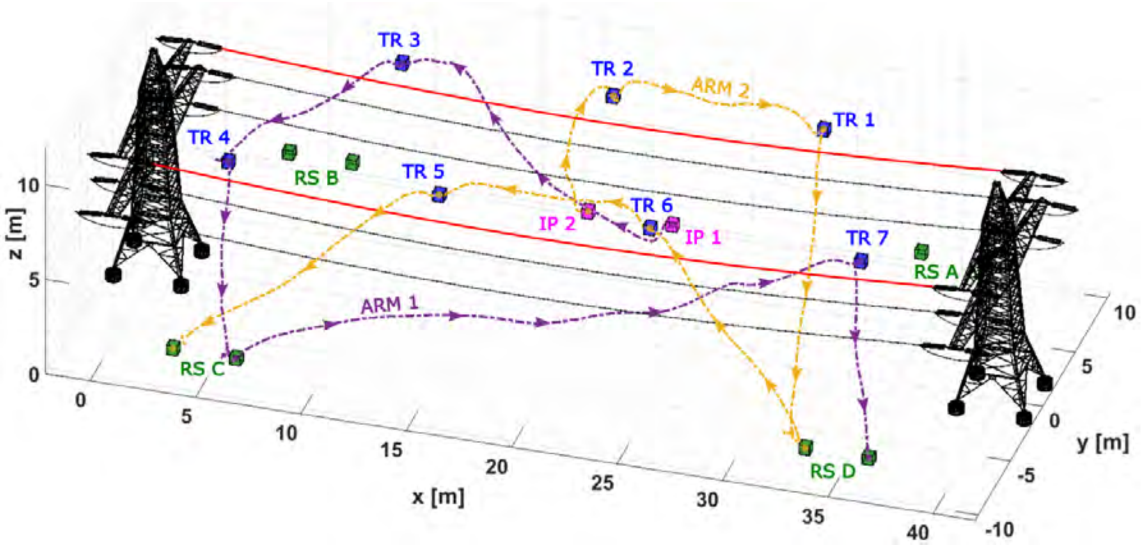

Figure 6.12: Bird diverter installation task. Execution of the ARM trajectories planned with the MP-MARM algorithm.

The real execution of the experiment can also be followed in the complete video *MP-MARM_real_bird_diverter_installation.mp4* that can be downloaded from the provided multimedia content, as indicated in Section 1.4. Additionally, Fig. 6.14 collects some snapshots of the same execution.

6.5 Conclusions

The presented MP-MARM motion planner has demonstrated its high performance in the computation of trajectories that enable the safe and efficient installation of bird flight diverters on power lines using a coordinated team of ARMs. For that, different temporal specifications have been encoded into a single metric and then, an optimisation process has maximised its value. Moreover, the method is flexible enough to add extra specifications that might appear, making it also appropriate for other applications like sensor deployments or load deliveries. The results have been trajectories that comply with all the mission requirements and the ARM limitations; these are, maximum velocity, acceleration and payload capacities. However, the good results have been partially possible thanks to the formulation to rapidly generate an

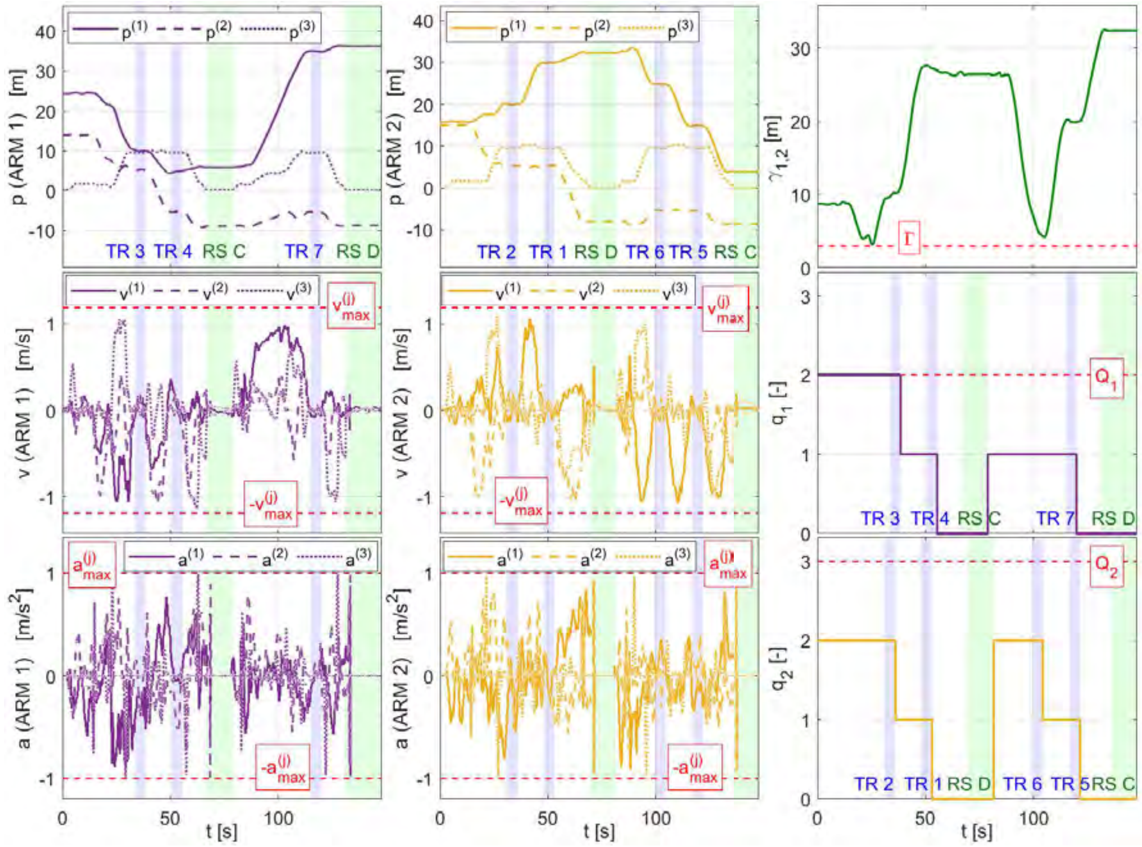

Figure 6.13: Bird diverter installation task. Evolution of the ARM variables: execution of the trajectories planned using the MP-MARM algorithm. The installation and refilling time windows are shaded using blue and green slots, respectively, and the dashed red lines highlight the admissible limits in the variables.

initial guess that eases the convergence of the aforementioned optimisation process. The combination of such optimisation with a suitable initial guess has allowed extending the scope of the proposed planning approach. Thus, global objectives, like the search of optimal assignments and sequences of target regions, and local constraints, like safety distances, have been put together. The resulting planned trajectories have demonstrated their feasibility to be used in real-world outdoor environments.

Additionally, the MP-MARM motion planner endowed with energy awareness has evidenced its benefits to reduce the energy consumption demanded by the team of ARMs while still fulfilling the mission requirements. In this sense, the planning of

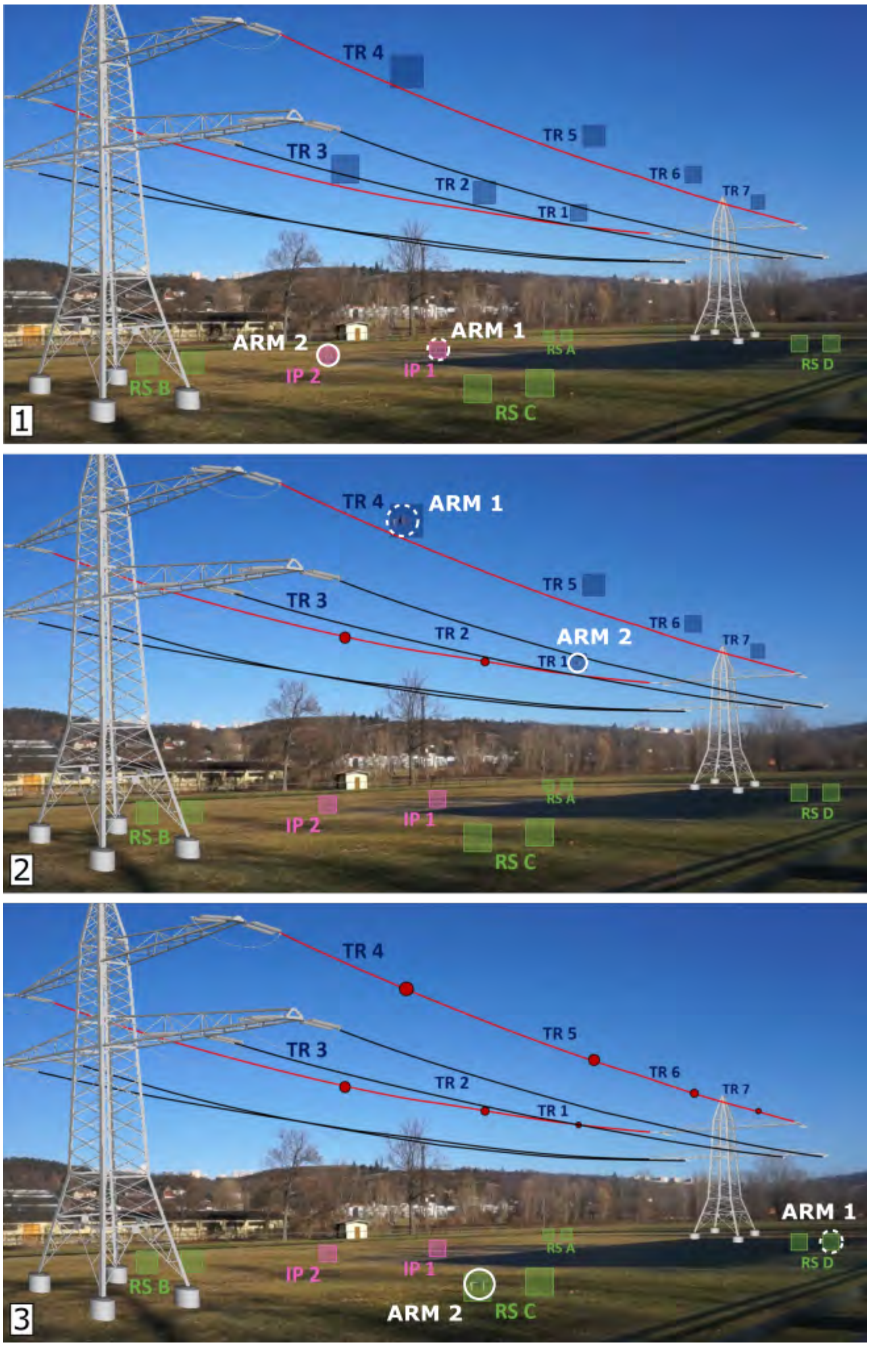

Figure 6.14: Bird diverter installation task. Snapshots of the real experiment. The power line has been represented using augmented reality. The white circles highlight the robots while the red spheres symbolise the installed bird diverters.

ARM trajectories with forward velocities close to the one minimising the consumption has been prioritised when it is possible. The cost to be paid has been a decrease in the safety distances of the ARMs with the obstacles and between them.

Finally, the application of the MP-MARM algorithm as event-based replanner has become useful to face the interruption in the service of a particular ARM when this is detected during the mission execution. To cope with that, the trajectory of a back-up ARM has been planned to address the remaining operation of the faulty robot while respecting the execution of the rest of functional ARMs. In this manner, the ongoing installation task can be completed successfully without introducing significant delays.

Chapter 7

Conclusions and future work

> Tan solo imagina que existe, piensa en cómo poder resolverlo y ten la confianza de que podrás hacerlo.
>
> Marcial Caballero

7.1 Conclusions

Aerial robotic manipulators have demonstrated a promising future in their application for inspection and maintenance activities by providing unique advantages with respect to conventional approaches. However, their associated capabilities still need to be extended to reach higher levels of autonomy, reliability, accuracy, safety and efficiency, among others. In addition to important improvements in human safety, this will lead to significant cost savings, making ARMs an effective solution to be exploited in real conditions.

Motivated by the previous statement, this thesis has found in motion planning a means to endow ARMs with enhanced functionalities. Consequently, the presented research has been focused on the design, development and validation of motion planning methods for aerial robotic manipulation in inspection and maintenance. In that context, the contributions have been concentrated not only on the exploitation of basic ARM features but also on the search of more advanced capabilities, such as Dynamics Awareness, Velocity Adaptation, replanning or planning oriented to reduce the energy consumption. Moreover, in some of the contributions, the cognition of the environment has played an important role as in the cases of in-flight planning or Aerodynamics Awareness. The results have been planned trajectories that bring ARMs closer to their practical deployment in real applications.

More in detail, the need of motion planning has been identified for three main topics related to aerial robotic manipulation, which are: manipulation using ARMs endowed with robotic arms, manipulation with HRs (Hybrid Robots) and multiARM manipulation. For each of them, a motion planning method has been derived and then, several extensions have been introduced to increase its capabilities. Concerning the fundamentals of such methods, all of them have been built over the basis of state-of-the-art approaches whose performances have been widely validated. However, they have been completely reformulated to give response to the particularities of the planning problems under consideration. The resulting motion planners can be summarised as follows:

- MP-ARM: motion planner specially oriented to ARMs endowed with robotic arms and operating in cluttered environments, considering the joint operation of the aerial platform and the manipulator. This method also counts on the extensions enumerated below:
    - In-flight planning thanks to the efficient integration with a mapping module that provides a representation of the environment in flight.
    - MP-ARM-DA formulation where the ARM dynamics is introduced in the planner operation for robust obstacle avoidance.
    - MP-ARM-VA approach for a better optimisation of the execution time of the trajectories without increasing the computational burden considerably.
    - MP-ARM-ADA extension accounting for the ARM aerodynamics to face the aerodynamic phenomena that may produce collisions.

- MP-HR: motion planner focused on taking advantage of the flying and rolling hybrid locomotion offered by HRs to generate efficient plans in inspection missions. Complementing this method, two extensions have been proposed:
  - MP-HR-DA approach where the HR dynamics while flying is integrated in the planning process to guarantee safer trajectories.
  - Hybrid replanning endowing the HR with the capability to switch between ground and air to avoid unexpected obstacles detected while rolling.
- MP-MARM: motion planner for multiARM systems subject to limited payload capacities and dynamic constraints in manipulation missions that require visiting efficiently a set of target regions where loads on board the robots are deployed. This method has been accompanied by two extensions:
  - Energy awareness reducing the ARM consumptions by planning trajectories with velocities prone to restrain the power demanded by the motors.
  - Event-based replanning to complete an ongoing mission without interruption when a particular ARM suffers a failure during its execution.

From the point of view of the application of the presented motion planning methods, this thesis has showcased several I&M scenarios where they have brought clear benefits. These application scenarios can be itemised in the following list:

- Maintenance of industrial elements, like pipes or chimneys, including repairs.
- Transportation of loads in cluttered industrial environments.
- Installation of sensors to monitor the state of industrial machinery.
- Inspection of surfaces of interest, like pipes or storage tanks, in industrial sites.
- Inspection of reinforced concrete bridges for crack detection.
- Inspection of pipe arrays in oil and gas refineries or chemical plants.
- Installation of bird flight diverters on electric power lines.

In any case, the developed planning methods have been based on general design principles, hence they can be adapted to other applications involving ARMs without entailing a significant effort.

To sum up, the different motion planning methods derived along this thesis have been essential to plan trajectories that have led a wide variety of ARMs to fulfil real I&M operations in demanding conditions with high performances. This evidence has been demonstrated through detailed analyses of numerical results, realistic simulations and real-world indoor and outdoor flight experiments.

7.2 Future work

The presented motion planning framework has established the basis of a research line that still admits further contributions. In this sense, several directions have been identified as a road map for future work. They are introduced hereinafter.

Advanced models of energy consumption

Most of the motion planning algorithms formulated in this thesis make use of consumption models to optimise the energy required by the ARMs to complete their missions. Although these models have helped in the computation of efficient trajectories, the integration of more advanced models can contribute to outperform the current results. Thus, models based on the ARM aerodynamics [105] allow a more accurate prediction of the energy consumption taking into account aspects like the velocity and, eventually, the wind effect in the operation area. Additionally, the consideration in these models of aerodynamic phenomena like the ground and ceiling effects can help planning approaches with Aerodynamics Awareness to consider such effects not only as disturbances leading to potential collisions but also as beneficial effects that can alleviate the energy consumption [66].

Parallel computing

During the programming of sampling-based motion planners like the MP-ARM and MP-HR algorithms, software profiling revealed that the most time-consuming process is the collision checking, which is in accordance with the literature [56]. Simultaneously, as it was explained in Section 3.2.1 (Collision checking), this is a process where many simple but repetitive independent checks are performed sequentially. In this context, the application of parallel computing using Graphics Processing Units (GPUs) has demonstrated its strength to alleviate the computation time required for collision checking [106]. In fact, the concept of parallel computing can also be applied to the expansion of the search tree in order to speed up the planning of a feasible solution [107, 108]. Inspired by these results, the usage of parallel computing can also be explored for the motion planning methods presented in this thesis, including not only the sampling-based planners but also the MP-MARM algorithm based on Signal Temporal Logic.

Force awareness

Many I&M applications require physical interaction of the ARM with the environment, which can be translated into reaction forces and torques exerted upon the robot. These forces and torques are influenced by the ARM configuration with respect to the contact point and, depending on the case, they can destabilise the ARM and even produce crashes. To cope with that, an extension of the MP-ARM motion planner, where the most favourable ARM configurations [109] for the contact operation are prioritised during the planning process, is a desirable option. These configurations must always keep the reaction forces and torques bounded within admissible limits while satisfying the rest of constraints like obstacle avoidance. Application scenarios such as sensor installations or repairing tasks would benefit from this extension.

Accurate models of the ARM dynamics

Motion planners with Dynamics Awareness have demonstrated clear advantages with respect to their basis algorithms. Their operation relies on the models of ARM dynamics that they integrate and consequently, their results are strongly influenced by the fitting of such models with the real dynamics of the robots. Thus, the more accurate the dynamic models are, the better results are expected during the execution of the planned trajectories. With this motivation, certain research effort deserves to be dedicated to the study of system identification techniques, specially oriented to ARMs or with application to them [110], and the integration of these techniques with the presented motion planners. In this line, the use of software in the loop can also help to have a good correlation between simulated and real closed-loop behaviours of the controlled ARMs by using exactly the same control software as in real flights.

Extended set of aerodynamic effects

Similarly to the relation between motion planning with Dynamics Awareness and models of ARM dynamics, the operation of the MP-ARM-ADA planner depends on maps of aerodynamic effects computed according to Annex B. Although the characterisation of aerodynamic effects used to build these maps includes the most common phenomena (ground, ceiling and wall effects), there is a wide range of additional effects whose consideration can extend the applicability of the MP-ARM-ADA motion planner. The influence in the ARM aerodynamics of inclined surfaces [111], pipes [112] or even more complex elements, such as tunnels or confined spaces, are some examples of aerodynamic effects that can also be added to the maps.

Appendix A

Aerial robotic manipulators

This annex collects a description of the different ARMs that have been used to test the motion planning methods presented along this thesis. Moreover, it also includes dynamic models and control laws when they are required for Dynamics Awareness or simulation as well as software and hardware details for experimentation.

A.1 Aerial robotic system for long-reach manipulation (version 1)

A.1.1 System description

Some of the ARMs presented in this thesis explore configurations that guarantee long-reach manipulation in hard-to-reach places where the target is far from the operation area of the UAV. In order to meet these requirements this section proposes the first version of a set of new Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v1). More precisely, the system consists of a multirotor with a long-bar extension that incorporates a lightweight dual arm in the tip (see Fig. A.1). Thus, the long-bar extension increases considerably the safety distance between rotors and manipulated objects while the dual arm offers extended manipulation capabilities with respect to other single-arm configurations existing in the literature.

Figure A.1: First version of a set of Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v1).

A planar characterisation of the system will serve to establish a first proof of concept for the ARS-LRM_v1 setup. This simplified approach eases the modelling and control derivations while maintaining the operation basis of the system. Following this assumption, the multirotor is characterised by a mass m_M, a principal moment of inertia I_{22}^M and dimensions $2d \times w$ (see Fig. A.2). Regarding the long bar, its length is given by l_P and it is assumed to be aligned with the UAV centre of mass M^O at a distance d. The cross-piece in the tip is defined by a length of $2l_C$. The total mass of the long bar and the cross-piece is m_P and will be treated as a punctual mass located where the long bar and the cross-piece intersect for simplicity purposes. Concerning the dual-arm manipulator, each separate arm L, R is composed of two links $L-U, L-D, R-U, R-D$, corresponding the lower ones to the end effectors. The two arms are characterised respectively by the lengths of their links -l_1 for upper links and l_2 for lower links- and their masses -m_1 and m_2-, where again the masses will be treated as punctual masses located at the distal end of each link in order to derive more manageable expressions. The values of all the aforementioned parameters are shown in Table A.1.

A.1.2 Modelling and control

Modelling

According to [113], the dynamics of a multirotor under 20kg is mostly determined by its mechanical model. This thesis embraces the same assumption and consequently, the behaviour of the different ARMs presented in this annex will be described by

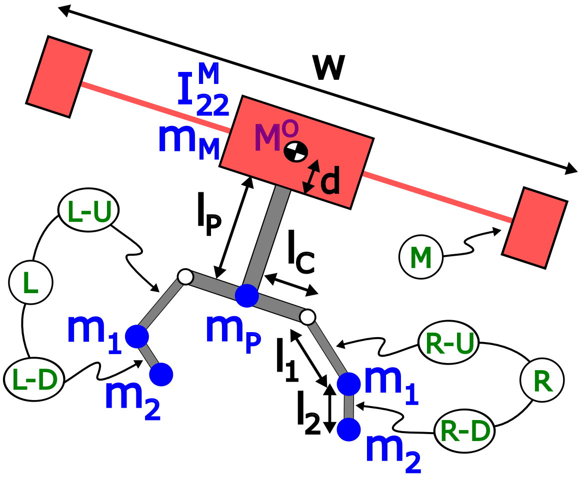

Figure A.2: Geometry and mass distribution of the ARS-LRM_v1 system.

Table A.1: ARS-LRM_v1 parameters.

	Parameter	Value	Units
Mass and Inertia	m_M	6.5	kg
-	I_{22}^M	0.0933	$kg\,m^2$
-	m_P	0.15	kg
-	m_1	0.06	kg
-	m_2	0.03	kg
Geometry	d	0.1	m
-	w	0.9	m
-	l_P	0.2	m
-	l_C	0.1	m
-	l_1	0.15	m
-	l_2	0.05	m

means of the mechanical models of the complete multibody systems. To this end, specific methodologies for multibody systems will be applied below.

Several approaches can be found in the literature to derive equations of motion for mechanical systems. However, Kane's method [114] has proved in [115] to hold some unique advantages over other traditional approaches when addressing multibody robotic systems like the ARS-LRM_v1. One of the most remarkable advantages is that the adoption of generalised speeds enables the derivation of a compact model in first-order differential equations that are uncoupled in the generalised speed derivatives. Other important features are the easy computerisation as well as 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 M^O in the inertial reference frame N, the multirotor pitch angle q_5 and the joint angles for both left L and right R arms q_7^L, q_8^L, q_7^R and q_8^R (see Fig. A.3). Generalised speeds u_i are defined as:

$$\begin{aligned} {}^{N}\mathbf{v}^{M^O} &= u_1\mathbf{n_1} + u_3\mathbf{n_3} & {}^{M}\boldsymbol{\omega}^{R-U} &= u_7^R\mathbf{n_2} \\ {}^{N}\boldsymbol{\omega}^{M} &= u_5\mathbf{n_2} & {}^{R-U}\boldsymbol{\omega}^{R-D} &= u_8^R\mathbf{n_2} \\ & & {}^{M}\boldsymbol{\omega}^{L-U} &= -u_7^L\mathbf{n_2} \\ & & {}^{L-U}\boldsymbol{\omega}^{L-D} &= -u_8^L\mathbf{n_2} \end{aligned} \tag{A.1}$$

where ${}^{N}\mathbf{v}^{M^O}$ is the velocity of the UAV centre of mass M^O with respect to the inertial reference frame N and ${}^{i}\boldsymbol{\omega}^{j}$ is the angular velocity of the element j with respect to the element i (see Fig. A.2 to identify the different elements). Previous equations lead to the following kinematic differential equations:

$$\begin{aligned} \dot{q}_i &= u_i \quad (i = 1, 3, 5) \\ \dot{q}_j^k &= u_j^k \quad (j = 7, 8\ ;\ k = R, L) \end{aligned} \tag{A.2}$$

Regarding forces and torques exerted on the ARS-LRM_v1 system (see Fig. A.4), the rotors generate a resultant lifting force $F_3\mathbf{a_3}$ applied at the multirotor centre of

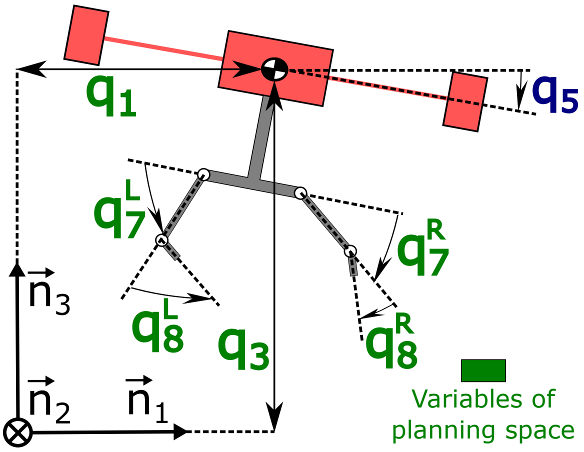

Figure A.3: Configuration variables of the ARS-LRM_v1 model. In green, the variables selected in Chapters 3 and 4 for the planning space.

mass M^O as well as a torque $T_2\mathbf{a_2}$ applied to rigid body M. At the same time, control actions governing the manipulator are given by the torques applied to the arm joints $T_7^R\mathbf{a_2}$, $T_8^R\mathbf{a_2}$, $-T_7^L\mathbf{a_2}$ and $-T_8^L\mathbf{a_2}$.

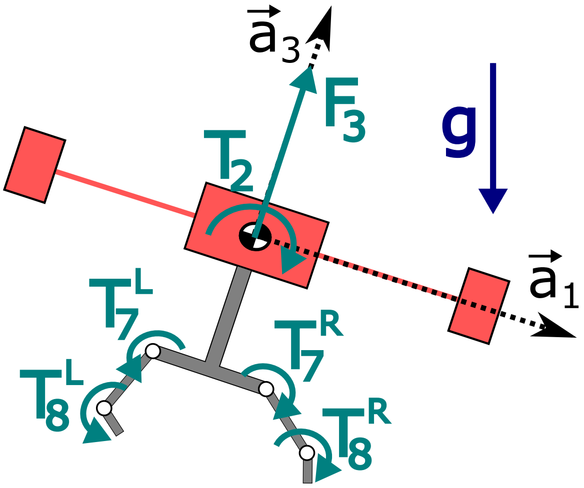

Figure A.4: Forces and torques applied to the ARS-LRM_v1 system.

Application of Kane's method through MotionGenesis software [116] leads to the following dynamic differential equations for translation and rotation, where **A**, **B**, **C** and **D** are dense matrices depending on the configuration variables q_5, q_7^R, q_8^R, q_7^L, q_8^L

and the system parameters defined in Table A.1, and $g = 9.81m/s^2$ is the gravity acceleration.

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_3 \\ \dot{u}_5 \\ \dot{u}_7^R \\ \dot{u}_8^R \\ \dot{u}_7^L \\ \dot{u}_8^L \end{bmatrix} = \mathbf{A} \begin{bmatrix} F_3 \\ T_2 \\ T_7^R \\ T_8^R \\ T_7^L \\ T_8^L \end{bmatrix} + \mathbf{B} \begin{bmatrix} (u_5)^2 \\ \left(u_7^R\right)^2 \\ \left(u_8^R\right)^2 \\ \left(u_7^L\right)^2 \\ \left(u_8^L\right)^2 \end{bmatrix} + \mathbf{C} \begin{bmatrix} u_5 u_7^R \\ u_5 u_8^R \\ u_5 u_7^L \\ u_5 u_8^L \\ u_7^R u_8^R \\ u_7^L u_8^L \end{bmatrix} + \mathbf{D} g \tag{A.3}$$

Control

After modelling the ARS-LRM_v1 system, a distributed control scheme (see Fig. A.5) has been derived to provide the system with the capability to execute navigation and manipulation manoeuvres. The objective is the completion of the simulation environment that will allow studying new planning strategies for the ARS-LRM_v1 platform. A standard control structure that makes use of nonlinear control strategies based on model inversion shall suffice to complete the testbed. This choice is considered representative of the average performance offered by the state-of-the-art controllers for this kind of systems.

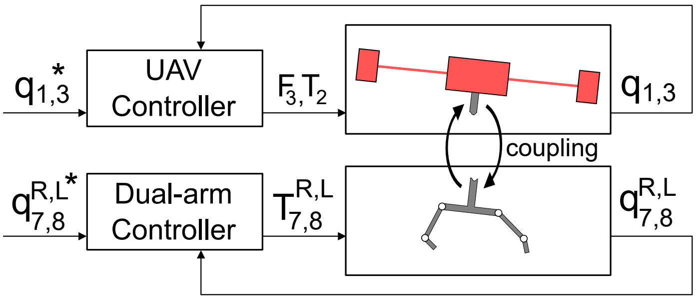

Figure A.5: Block diagram of the ARS-LRM_v1 distributed control scheme.

Regarding the multirotor, the control scheme is inspired by [113] and consists in linearising the system through model inversion and applying PID (Proportional-Integral-Derivative) control laws to the resultant dynamics. In the model inversion, the coupling action of the long-reach manipulator 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 multirotor lifting force vector, in order to generate the translational accelerations required to reduce position errors with respect to the control references $q^{\star}_{1,3}$. A general overview of the control scheme is shown in Fig. A.6, where D^{-1}_{13}, K^{-1}_{5} and D^{-1}_{5} blocks represent, respectively, the inversions of the translational dynamics, rotational kinematics and rotational dynamics.

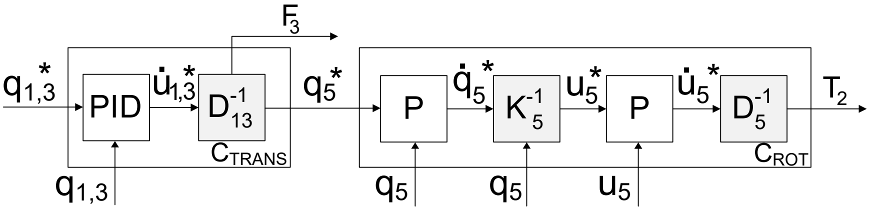

Figure A.6: Block diagram of the ARS-LRM_v1 UAV controller.

The control strategy selected for each arm is again based on linearisation through model inversion and PD (Proportional-Derivative) control, which yields a nonlinear control law capable of commanding the link positions of both arms. As in the previous case, the coupling action is not 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 Fig. A.7 where D^{-1}_{78} represents the block in charge of inverting the arm dynamics, and $q^{R,L\star}_{7,8}$ are the control references.

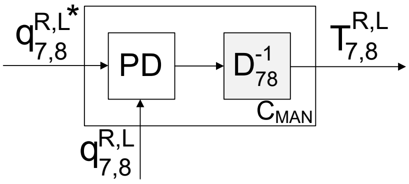

Figure A.7: Block diagram of the ARS-LRM_v1 dual-arm controller.

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.

A.2 Aerial robotic system for long-reach manipulation (version 2)

A.2.1 System description

This section proposes the second version of the set of new Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v2), which is produced by the GRVC Robotics Lab [9] in the framework of the AEROARMS project [10] as a variation of the ARS-LRM_v1 aerial manipulator (see Fig. A.8). For the new configuration, the long-bar extension has been integrated with a hexarotor using a passive revolute joint. In this way, high torques induced to the base of the aerial platform can be bounded when the dual arm contacts an external element. At the same time, flight experiments reveal that the air flow produced by the multirotor propellers reduces considerably the bar oscillations around the passive joint. Additionally, the manipulator, rotated 90 degrees with respect to the ARS-LRM_v1 configuration, is an enhanced version of the anthropomorphic and lightweight dual arm developed in [2]. Each arm provides four DoF in a human-like kinematic configuration. However, only two DoF (shoulder pitch and elbow pitch) have been considered in this thesis to simplify the analysis of the obtained results. The complete mechanical specifications of the ARS-LRM_v2 system are summarised in Table A.2.

Thanks to the passive revolute joint, the aforementioned mitigation of the dynamic coupling between the aerial platform and the long-reach manipulator makes the operation with this articulated version safer. Consequently, the ARS-LRM_v2 system is used for the validation experiments of the motion planning presented in Chapter 3. For this reason, software and hardware details are included below.

Figure A.8: Second version of a set of Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v2). First (left) and final (right) prototypes.

Table A.2: ARS-LRM_v2 parameters (final prototype).

Subsystem	Parameter	Value	Units
Aerial platform	mass	14.0	kg
-	dimensions	$\varnothing 1.7 \times 1.0$	m
Long-bar extension	mass	0.3	kg
-	length	0.45	m
Dual arm	mass	4.0	kg
-	cross-piece length	0.35	m
-	upper-arm length	0.25	m
-	forearm length	0.25	m

Software integration

Apart from the motion planner, presented in this thesis, the rest of software modules required for experimentation using the ARS-LRM_v2 system were developed previously by the GRVC Robotics Lab [9]. Firstly, a Multisensor Mapping module, developed in the context of the AEROARMS project [10], is able to generate online a significantly accurate map (median error smaller than 5 cm) that can be used for trajectory planning. The module integrates measurements from a 3D LIDAR (LIght Detection And Ranging) sensor, a stereo camera and UWB (UltraWide-Band) nodes. Secondly, given a reference trajectory for the complete ARM; this is, the aerial platform and the manipulator, a Trajectory Tracker module can command both ARS-LRM_v2 subsystems to track it properly.

Once the required modules are consolidated, the main challenge for the software integration is the communication and harmonisation of all them. For that, they have been integrated within the ROS (Robot Operating System) framework [117] in the way presented in Fig. A.9. Such integration was mainly addressed by different members of the GRVC Robotics Lab.

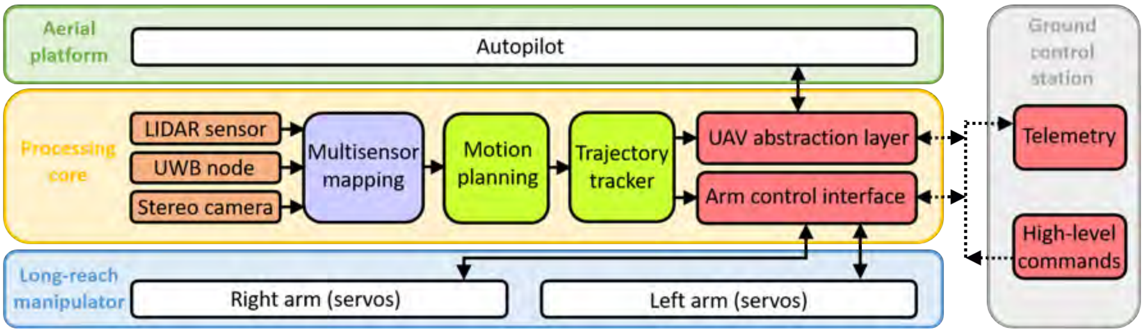

Figure A.9: Software architecture of the ARS-LRM_v2 system.

The Multisensor Mapping module was already implemented as a set of ROS nodes. A ROS service has been enabled to provide online-generated maps by demand. These maps are sent in real time to the Motion Planning module in a point-cloud-based format. The Motion Planning module was initially implemented in MATLAB [7] and the generated trajectories were saved in files. To improve the integration of this functionality, a C++ library generated using MATLAB Coder has been embedded into a ROS package. Such package offers a service that, given a target in configuration space and the map provided by the Multisensor Mapping module, generates a trajectory for the ARS-LRM_v2 system when required. The Trajectory Tracker already offered a ROS service that, every time a trajectory is generated by the Motion Planning module, this is provided to be tracked once accepted by an operator. Thus, this tracker commands the autopilot of the aerial platform through the UAV Abstraction Layer (UAL) [118] and the arm servo actuators through the Arm Control Interface (ACI). Finally, everything is controlled from a ROS-based Ground Control Station (GCS) software that allows sending high-level commands to the ARS-LRM_v2 system while receiving telemetry to monitor the system state.

Hardware integration

There are three main hardware modules that are needed for the operation of the ARS-LRM_v2 system. Apart from the GCS computer, working on land, the other two hardware modules are placed on board the ARM and they are the autopilot of the aerial platform, acting as Low-Level Controller (LLC), and the processing core to deal with the rest of software functionalities (see Fig. A.9). At the same time, the processing core consists of two different devices. Firstly, the Primary High-Level Computer (PHLC) is in charge of the Motion Planning module, the Trajectory Tracker module, the LLC management and the dual-arm control. Secondly, the Secondary High-Level Computer (SHLC) is fully dedicated to the Multisensor Mapping module. All the computers, either on board the ARM or on land, are on the same local network to favour direct communication between them. The ground-air communication is solved through an aerial bridge.

Regarding the sensors supporting the computers described above, these are a laser altimeter and an RTK GPS (Real-Time-Kinematic Global Positioning System), both connected to the autopilot, and all the sensors required for Multisensor Mapping, which are a 3D LIDAR sensor, a stereo camera and a UWB receiver node, all of them connected to the SHLC computer.

The models of all the hardware devices on board the ARS-LRM_v2 system are listed in Table A.3.

Table A.3: Hardware devices on board the ARS-LRM_v2 system.

Device	Model
LLC	Pixhawk 2.1
PHLC	Intel Nuc i7
SHLC	Nvidia Jetson TX2 Development Kit
Aerial bridge	Ubiquiti Rocket M5
Laser altimeter	Lightware SF11/C
RTK GPS	ProfiCNC Here+ (Ublox M8P)
3D LIDAR sensor	Velodyne FDL-32
Stereo camera	Stereolabs ZED
UWB node	InCircuit DW1000 Tag

A.2.2 Modelling and control

The use of the ARS-LRM_v2 system requires the development of new modelling and control derivations in order to carry out validation simulations before the experimental work.

Modelling

As for the ARS-LRM_v1 system, a planar characterisation of the ARS-LRM_v2 setup will be enough for a first proof of concept. 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 multirotor pitch angle q_5, the passive revolute joint angle q_0 and the joint angles for both left L and right R arms q_7^L, q_8^L, q_7^R and q_8^R (see Fig. A.10). Generalised speeds u_i are defined as:

$$\begin{aligned} {}^N\mathbf{v}^{A^O} &= u_1\mathbf{n_1} + u_3\mathbf{n_3} \\ {}^N\boldsymbol{\omega}^A &= u_5\mathbf{n_2} \\ {}^A\boldsymbol{\omega}^B &= u_0\mathbf{n_2} \end{aligned} \qquad \begin{aligned} {}^B\boldsymbol{\omega}^{U-R} &= u_7^R\mathbf{n_2} \\ {}^{U-R}\boldsymbol{\omega}^{F-R} &= u_8^R\mathbf{n_2} \\ {}^B\boldsymbol{\omega}^{U-L} &= u_7^L\mathbf{n_2} \\ {}^{U-L}\boldsymbol{\omega}^{F-L} &= u_8^L\mathbf{n_2} \end{aligned} \tag{A.4}$$

where ${}^N\mathbf{v}^{A^O}$ is the velocity of the UAV centre of mass A^O with respect to the inertial reference frame N and ${}^i\boldsymbol{\omega}^j$ is the angular velocity of the element j with respect to the element i (see Fig. A.10 to identify the different elements). Previous equations lead to the following kinematic differential equations:

$$\begin{aligned} \dot{q}_i &= u_i \quad (i = 1, 3, 5, 0) \\ \dot{q}_j^k &= u_j^k \quad (j = 7, 8\ ;\ k = R, L) \end{aligned} \tag{A.5}$$

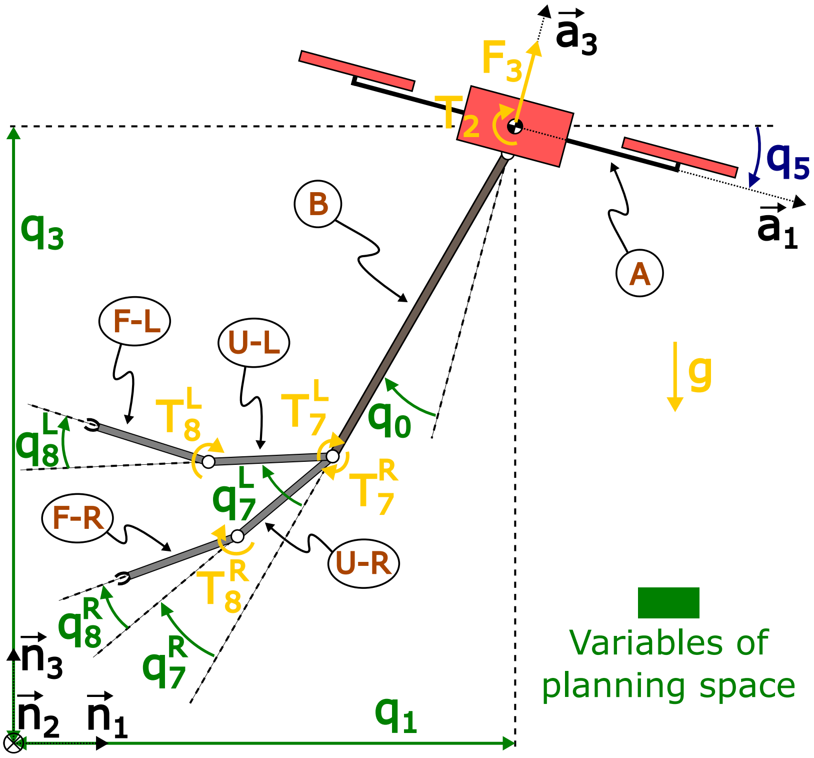

Figure A.10: Configuration variables of the ARS-LRM_v2 model (blue and green), and forces and torques applied to it (yellow). In green, the configuration variables selected in Chapter 3 for the planning space.

Regarding forces and torques exerted on the ARS-LRM_v2 system (see Fig. A.10), the rotors generate a resultant lifting force $F_3\mathbf{a_3}$ applied at the multirotor centre of mass A^O as well as a torque $T_2\mathbf{a_2}$ applied to rigid body A. At the same time, control actions governing the manipulator are given by the torques applied to the arm joints $T_7^R\mathbf{a_2}$, $T_8^R\mathbf{a_2}$, $T_7^L\mathbf{a_2}$ and $T_8^L\mathbf{a_2}$.

Application of Kane's method through MotionGenesis software [116] leads to the following dynamic differential equations for translation and rotation, where $g = 9.81m/s^2$ is the gravity acceleration and **A**, **B**, **C** and **D** are dense matrices depending on the configuration variables q_5, q_0, q_7^R, q_8^R, q_7^L, q_8^L and the parameters defining geometry and mass distribution of the ARS-LRM_v2 system.

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_3 \\ \dot{u}_5 \\ \dot{u}_0 \\ \dot{u}_7^R \\ \dot{u}_8^R \\ \dot{u}_7^L \\ \dot{u}_8^L \end{bmatrix} = \mathbf{A} \begin{bmatrix} F_3 \\ T_2 \\ T_7^R \\ T_8^R \\ T_7^L \\ T_8^L \end{bmatrix} + \mathbf{B} \begin{bmatrix} (u_5)^2 \\ (u_0)^2 \\ \left(u_7^R\right)^2 \\ \left(u_8^R\right)^2 \\ \left(u_7^L\right)^2 \\ \left(u_8^L\right)^2 \end{bmatrix} + \mathbf{C} \begin{bmatrix} u_5 u_0 \\ u_5 u_7^R \\ u_5 u_8^R \\ u_5 u_7^L \\ u_5 u_8^L \\ u_0 u_7^R \\ u_0 u_8^R \\ u_0 u_7^L \\ u_0 u_8^L \\ u_7^R u_8^R \\ u_7^L u_8^L \end{bmatrix} + \mathbf{D} g \tag{A.6}$$

Control

Concerning the control architecture, the same distributed control scheme proposed in Section A.1.2 has been adopted to provide the simulated ARS-LRM_v2 system with the capability to track trajectories generated by motion planning. Fig. A.11 depicts the associated block diagram.

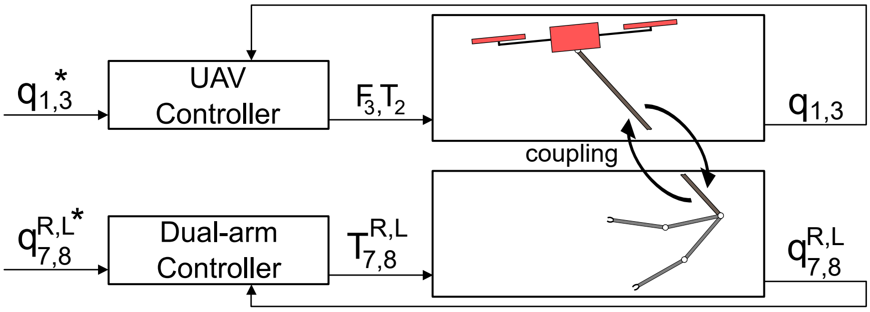

Figure A.11: Block diagram of the ARS-LRM_v2 distributed control scheme.

A.3 Aerial robotic system for long-reach manipulation (version 3)

A.3.1 System description

To complete the set of new Aerial Robotic Systems for Long-Reach Manipulation, a third version (ARS-LRM_v3), inspired by [13], is presented in this section. As can be seen in Fig. A.12, the proposed ARM consists of a multirotor and a robotic long-reach arm whose special integration with the aerial platform enables 360° rotation within the vertical plane that coincides with the centre of mass of the multirotor. This capability for multidirectional movement within the vertical plane widens significantly the operation workspace of the robotic arm, which in turn extends the range of positions from which the aerial platform can perform its task. The latter allows a better selection of the safest operation conditions for the ARM. Additionally, the long-reach feature increments considerably the safety distance between rotors and operation areas. This last characteristic is even more relevant when such areas include elements that can affect the rotor airflow, as is the case for the motion planning problem studied in Section 4.3. Both features, multidirectional and long-reach capabilities, constitute an important improvement with respect to other state-of-the-art aerial manipulators where the base of the arms is usually fixed at the bottom part of the airframe. Moreover, by simply interchanging the tool located as end effector, the ARS-LRM_v3 system is endowed with the capability to carry out a wide variety of manipulation tasks.

As for the ARS-LRM_v1 and ARS-LRM_v2 systems, a planar characterisation of the ARS-LRM_v3 setup will be enough for a first proof of concept. Following this assumption, the aerial platform A is characterised by a mass m_a, a principal moment of inertia I^a_{22} and dimensions $w \times h$ (see Fig. A.12). Concerning the location of the rotors, a quadrotor configuration with propellers of radius R is 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

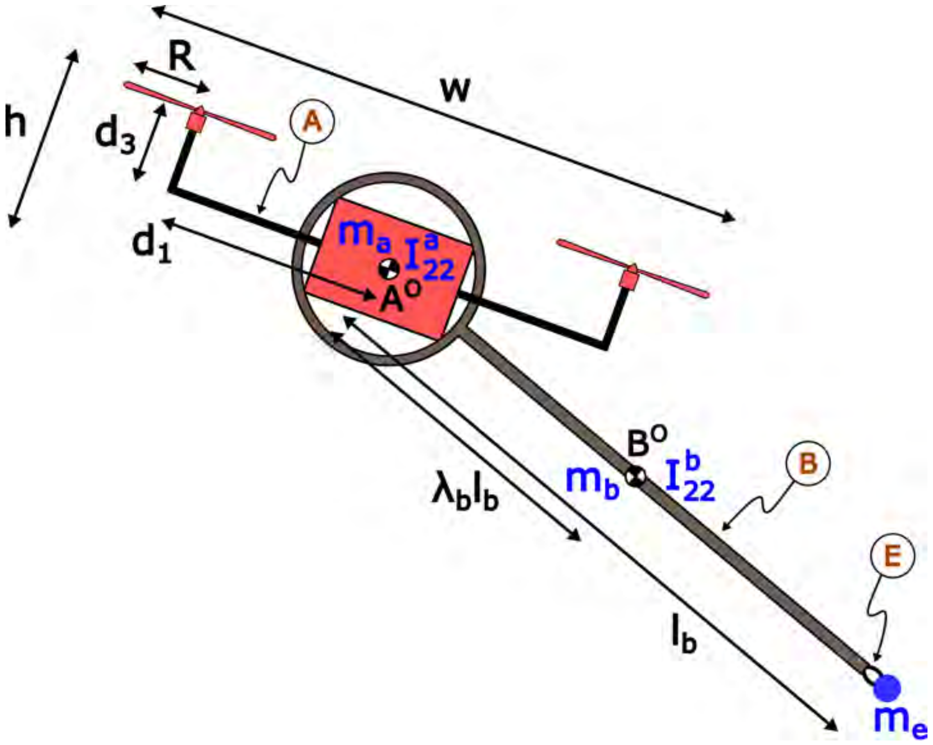

Figure A.12: Third version of a set of Aerial Robotic Systems for Long-Reach Manipulation (ARS-LRM_v3). Geometry and mass distribution.

long-reach manipulator has been considered as a rigid body of length l_b, mass m_b and principal moment of inertia I^b_{22}. 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 A.4.

Table A.4: ARS-LRM_v3 parameters.

	Parameter	Value	Units
Mass and Inertia	m_a	5	kg
-	I^a_{22}	0.093	$kg\,m^2$
-	m_b	0.3	kg
-	λ_b	0.5	$-$
-	I^b_{22}	0.025	$kg\,m^2$
-	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

A.3.2 Modelling and control

Modelling

The configuration variables selected as generalised coordinates for the ARS-LRM_v3 system 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 multirotor pitch angle q_5 and the joint angle of the robotic arm q_7 (see Fig. A.13). Generalised speeds u_i ($i = 1,3,5,7$) are defined as:

$$^N\boldsymbol{v}^{A^O} = u_1\mathbf{n_1} + u_3\mathbf{n_3} \quad ^N\boldsymbol{\omega}^A = u_5\mathbf{n_2} \quad ^A\boldsymbol{\omega}^B = u_7\mathbf{n_2} \tag{A.7}$$

where $^N\mathbf{v}^{A^O}$ 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:

$$\begin{bmatrix} \dot{q}_1 & \dot{q}_3 & \dot{q}_5 & \dot{q}_7 \end{bmatrix}^T = \begin{bmatrix} u_1 & u_3 & u_5 & u_7 \end{bmatrix}^T \tag{A.8}$$

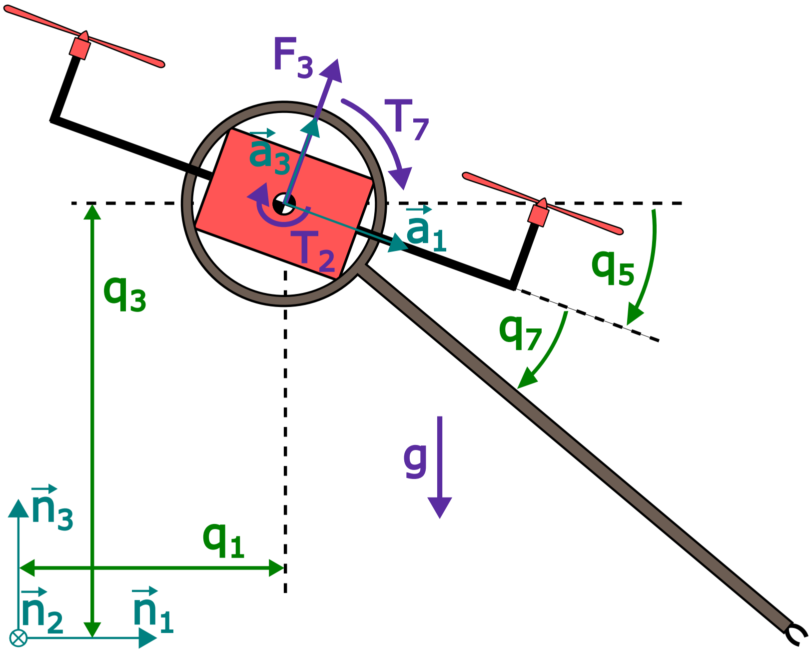

Figure A.13: Configuration variables of the ARS-LRM_v3 model (green), and forces and torques applied to it (purple).

Regarding forces and torques exerted on the ARS-LRM_v3 system (see again Fig. A.13), the rotors generate a resultant lifting force $F_3\mathbf{a_3}$ applied at the multirotor centre of mass A^O as well as a torque $T_2\mathbf{a_2}$ applied to the aerial platform A. Additionally, the control action governing the manipulator is given by the torque applied to the arm joint $T_7\mathbf{a_2}$.

Application of Kane's method through MotionGenesis software [116] leads to the following dynamic differential equations for translation and rotation:

$$\mathbf{A} \cdot \begin{bmatrix} \dot{u}_1 & \dot{u}_3 & \dot{u}_5 & \dot{u}_7 \end{bmatrix}^T = \mathbf{B} \tag{A.9}$$

where matrix **A** is:

$$\mathbf{A} = \begin{bmatrix} m_a + m_b + m_e & 0 & -l_b \xi_{m,1} s_{5,7} & -l_b \xi_{m,1} s_{5,7} \\ 0 & m_a + m_b + m_e & -l_b \xi_{m,1} c_{5,7} & -l_b \xi_{m,1} c_{5,7} \\ -l_b \xi_{m,1} s_{5,7} & -l_b \xi_{m,1} c_{5,7} & I_{22}^a + I_{22}^b + l_b^2 \xi_{m,2} & I_{22}^b + l_b^2 \xi_{m,2} \\ -l_b \xi_{m,1} s_{5,7} & -l_b \xi_{m,1} c_{5,7} & I_{22}^b + l_b^2 \xi_{m,2} & I_{22}^b + l_b^2 \xi_{m,2} \end{bmatrix} \tag{A.10}$$

matrix **B** is:

$$\mathbf{B} = \begin{bmatrix} F_3 s_5 + l_b \xi_{m,1} c_{5,7} (u_5 + u_7)^2 \\ F_3 c_5 - (m_a + m_b + m_e) g - l_b \xi_{m,1} s_{5,7} (u_5 + u_7)^2 \\ T_2 + l_b \xi_{m,1} c_{5,7} g \\ T_7 + l_b \xi_{m,1} c_{5,7} g \end{bmatrix} \tag{A.11}$$

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.

Control

Concerning the control architecture, the distributed control scheme proposed in Section A.1.2 has been adopted to provide the ARS-LRM_v3 system with the capability to track trajectories generated by motion planning. However, the control strategy for the manipulator is simplified since the ARS-LRM_v3 system only has one robotic arm with one single joint. Fig. A.14 depicts the associated block diagram.

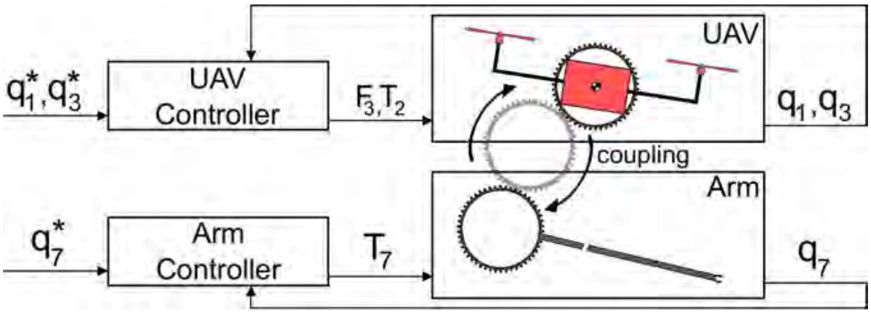

Figure A.14: Block diagram of the ARS-LRM_v3 distributed control scheme.

A.4 Hybrid robotic manipulator

A.4.1 System description

This section focuses on introducing a Hybrid Robotic Manipulator (HRM) produced by the GRVC Robotics Lab [9] in the framework of the HYFLIERS project [11]. In general terms, the HRM system is a flying-rolling inspection platform capable of landing and moving along pipe arrays to perform their inspection without wasting energy in the propellers. In this way, the robot overcomes the limitations of flying robots in terms of operation time and position accuracy when it is applied to industrial I&M tasks.

Fig. A.15 brings a general overview of the HRM system. The aerial-rolling manipulator is composed of a standard hexarotor, a customised rolling base that replaces the conventional landing gear, and a 5-DoF robotic arm supported by a 1-DoF linear guide system. The rolling base is designed to facilitate the displacement of the robot along pipe arrays. Two rollers have been integrated in order to control these displacements. Moreover, the design of the robotic manipulator enables the access to the contour of the pipes, both the side and lower parts which are the regions where the defects are usually concentrated. The prototype consists of a lightweight robotic arm with three joints for end-effector positioning, two joints for wrist orientation, and an actuated linear guide system to facilitate the deployment of the arm between pipes. The mechanical specifications of the HRM system are summarised in Table A.5.

Table A.5: HRM parameters.

Subsystem	Parameter	Value	Units
Aerial platform	mass	2.4	kg
-	dimensions	$\varnothing 1.2 \times 0.3$	m
Rolling base	mass	1.3	kg
-	roller dimensions	$\varnothing 0.05 \times 0.6$	m
-	separation between rollers	0.22	m
Robotic arm	mass	0.6	kg
-	linear-guide length	0.36	m
-	upper-arm length	0.25	m
-	forearm length	0.25	m

Figure A.15: Hybrid robotic manipulator (HRM) with aerial-rolling locomotion capabilities.

Since the actuated linear guide will be transversely moving along the gaps existing between the pipes, the most suitable configuration for inspection will be a centred position of the HRM system with respect to the array of parallel pipes that is under inspection, as suggested in Fig. A.16. This central location increases the stability and accuracy of the operation.

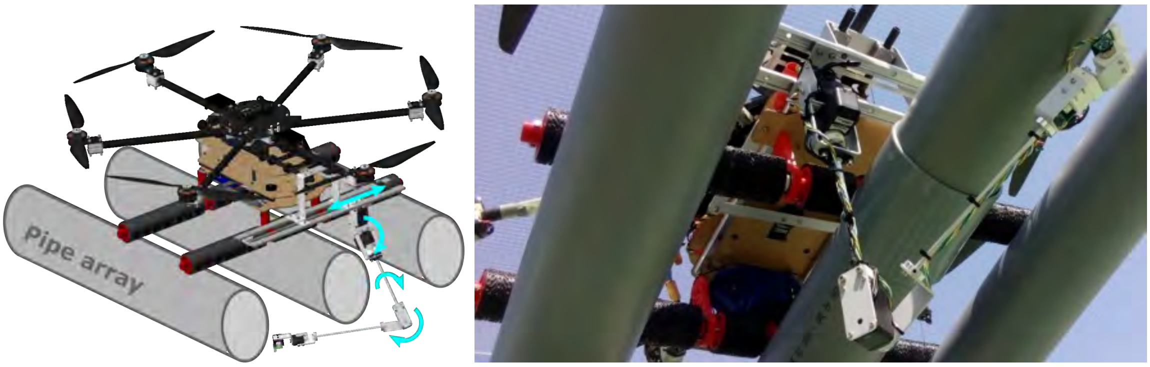

Figure A.16: HRM configuration for the inspection of pipe arrays.

Regarding software and hardware details for experimentation, the architecture of the HRM system comprises three main parts. Firstly, the multirotor platform integrates a Pixhawk autopilot with the trajectory controller as well as a Raspberry Pi 3B+ computer with the remaining software modules. These are the motion planner, the manipulator control program, and the aerial platform control interface [118]. Secondly, the rolling base is controlled by two servos using a STM32 microcontroller. Finally, the robotic manipulator integrates two groups of servos which are three standard Herkulex DRS-0101 actuators used to position the end effector, and the customised micro servos of the linear guide and the wrist joints. The motion of the HRM system is measured using a laser tracking system (robotic total station Leica MS50 [92]) that provides accurate estimations of the robot position. For more information about the HRM system, a detailed description can be found in [35].

A.4.2 Modelling and control

Modelling

The dynamic model of the HRM system will be concentrated on its behaviour in flight since this is the operation phase where the differences between reference and executed trajectories are more significant and critical. In contrast, during the rolling phase, the trajectory tracking is accurate enough to neglect the dynamic behaviour. Moreover, as the HRM manipulation is always performed with the robot landed on the pipe arrays, the robotic arm will be folded and fixed in flight and consequently, its movement will not need to be considered within the dynamic model of the HRM system. Nevertheless, the manipulator can not be disregarded completely because its mass affects the dynamic behaviour of the full HRM system in flight, even if it is not moving.

The configuration variables selected as system generalised coordinates are the position of the centre of mass $\mathbf{p} = [x, y, z]^T$ in an Earth-fixed frame and the orientation angles $\boldsymbol{\eta} = [\phi, \theta, \psi]^T$ (roll, pitch, yaw). These configuration variables lead to the following kinematic differential equations where $u_x, u_y, u_z, u_\phi, u_\theta, u_\psi$ are the generalised velocities associated with the configuration variables:

$$\begin{gathered}
[\dot{x}, \dot{y}, \dot{z}]^T = [u_x, u_y, u_z]^T \\
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \left(u_\phi \cos\psi - u_\theta \sin\psi\right) / \cos\theta \\ u_\phi \sin\psi + u_\theta \cos\psi \\ u_\psi - \tan\theta \left(u_\phi \cos\psi - u_\theta \sin\psi\right) \end{bmatrix}
\end{gathered} \tag{A.12}$$

Regarding forces and torques exerted on the HRM system, the rotors generate a resultant lifting force F_x applied at the multirotor centre of mass as well as a torque $\mathbf{T} = [T_x, T_y, T_z]^T$ applied to the rigid body. Application of Kane's method leads to the following dynamic differential equations for translation and rotation:

$$\begin{bmatrix} m\dot{u}_x \\ m\dot{u}_y \\ m\dot{u}_z \\ I_{xx}\dot{u}_\phi \\ I_{yy}\dot{u}_\theta \\ I_{zz}\dot{u}_\psi \end{bmatrix} = \begin{bmatrix} F_x \sin\theta \\ -F_x \sin\phi \cos\theta \\ F_x \cos\phi \cos\theta - mg \\ T_x - (I_{zz} - I_{yy})\, u_\theta u_\psi \\ T_y + (I_{zz} - I_{xx})\, u_\phi u_\psi \\ T_z - (I_{yy} - I_{xx})\, u_\phi u_\theta \end{bmatrix} \tag{A.13}$$

where m is the mass of the full HRM system, I_{xx}, I_{yy}, I_{zz} are its principal moment of inertia with the robotic arm folded, and $g = 9.81 m/s^2$ is the gravity acceleration.

Control

Similar to the modelling, the control strategy will be focused on the flying phase. Consequently, the derivation of a controller for the manipulator is intentionally omitted here. Concerning the control architecture selected for the aerial platform, the control scheme proposed in Section A.1.2 for the multirotor has been adopted. However, the scheme has been extended to all the HRM position and yaw coordinates to provide the system with the capability to track in flight 3D trajectories generated by motion planning. Fig. A.14 schematises the associated block diagram.

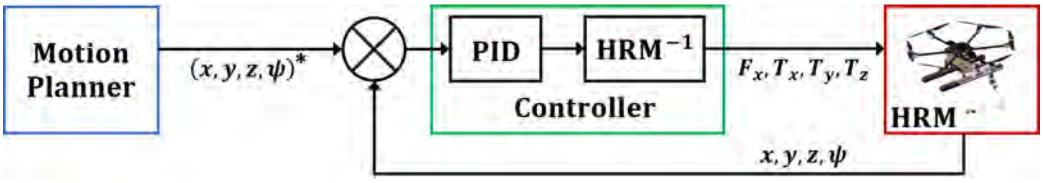

Figure A.17: Block diagram of the HRM control scheme.

A.5 Aerial robotic manipulator with linear actuator

A.5.1 System description

This section introduces an Aerial Robotic Manipulator endowed with a Linear Actuator (ARM-LA) that is devoted to the installation of clip-type bird diverters on power lines. Presented in Fig. A.18, the robot is produced by the GRVC Robotics Lab [9] in the framework of the AERIAL-CORE project [12] and allows the fast and autonomous installation in flight of the two models of diverters shown in Fig. A.18 (bottom corners). In contrast to the few state-of-the-art solutions available for this operation while flying, the ARM-LA system can apply the high forces required for the installation of these diverters, without the need to use a heavy aerial vehicle. Moreover, although the ARM was initially conceived to install one single diverter per flight, an extended version that is under development will enable the deployment of up to four devices.

The ARM-LA system consists of a quadrotor platform and a manipulator acting as installation mechanism. At the same time, such mechanism includes a high-force (up to 90 kg) linear actuator that exerts a pushing force directly on the diverter for its installation. The diverter is held by a clamp system that transmits the reaction forces to the power line, isolating in this way the aerial platform. Additionally, the manipulator is attached to the quadrotor on its bottom part through a passive spherical joint that avoids the propagation of wrench disturbances from the manipulator to the aerial platform during the execution of the operation in flight. The mechanical specifications of the ARM-LA system are summarised in Table A.6. For more information about the robot, a detailed description can be found in [41].

Table A.6: ARM-LA parameters.

Subsystem	Parameter	Value	Units
Aerial platform	mass	6.5	kg
-	dimensions	$1.2 \times 1.2 \times 0.8$	m
Manipulator	mass	1.7	kg
-	length	0.45	m

Figure A.18: Aerial Robotic Manipulator with Linear Actuator (ARM-LA) for the installation on power lines of clip-type bird diverters (bottom corners).

Appendix B

Characterisation of aerodynamic effects

This annex describes an experimental approach that allows characterising the aerodynamic phenomena that arise when a rotor operates close to a surface; these are, the ground effect, the ceiling effect and the wall effect. The resulting models were developed and presented previously in [119] as a contribution of its author, and are essential to formulate Aerodynamics Awareness in the planning process.

B.1 Test bench

In order to characterise aerodynamic phenomena, a test bench that allows the analysis of the thrust behaviour while a rotor is operating close to a surface 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 Fig. B.1). 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:

$$F_3 = F_{lc}\frac{d_h}{d_v} \tag{B.1}$$

where d_h and d_v are the dimensions depicted in Fig. B.1. 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 PWM (Pulse Width Modulation) 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 computer with MATLAB [7] that offers a graphical interface to process, display and save all the data collected by the sensors.

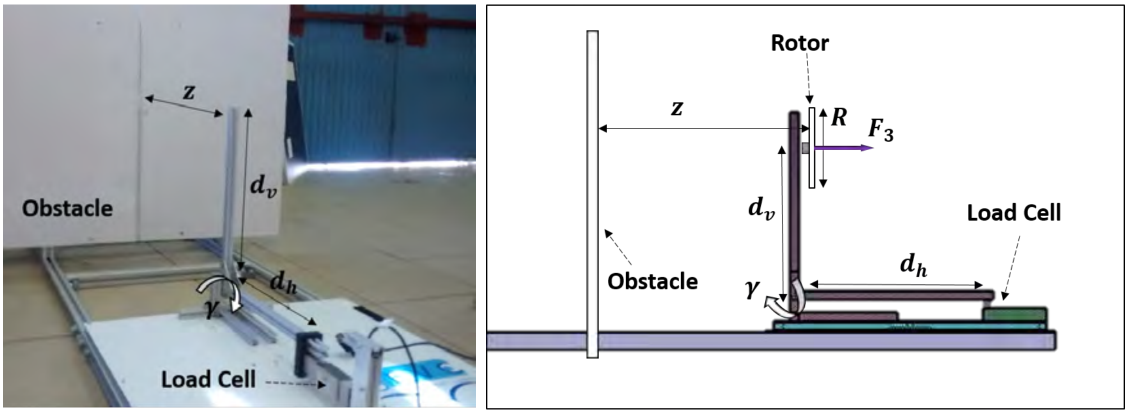

Figure B.1: Test bench used for the characterisation of the aerodynamic effects.

B.2 Experimental procedure

As it was advanced before, 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 Fig. B.1) 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.

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 Fig. B.2. In this figure, the aerodynamic modification of the 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):

$$\tau = \frac{F_{3,IAE}}{F_{3,OAE}} \tag{B.2}$$

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 Fig. B.2 above) and denominator is the rotor radius; that is, z/R for the ground and ceiling effects or x/R for the wall effect. Finally, since several tests are carried out for each distance under study, it is possible to have statistical characterisations like the standard deviations represented in Fig. B.2 using an error-bar format.

Ground effect

The experimental results concerning the ground effect are consistent with the classical approach presented in [120] (dotted black curve in the left column of Fig. B.2). 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 (B.3) 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

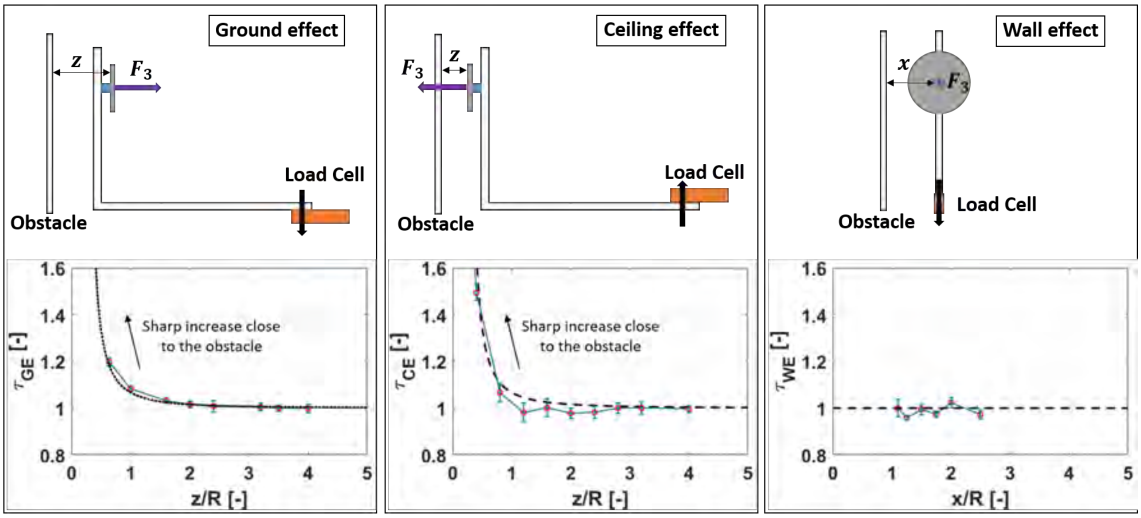

Figure B.2: Characterisation of aerodynamic effects (below) and associated configuration of the test bench (above): ground, ceiling and wall effects. Experimental results (red dots and blue error bars) and analytical models (dotted black curve for the theoretical model and dashed black curves for regression models).

potential aerodynamics.

$$\tau_{GE} = \frac{1}{1 - \frac{1}{16}\left(\frac{R}{z}\right)^2} \tag{B.3}$$

Ceiling effect

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:

$$\tau_{CE} = \frac{1}{1 - \frac{1}{k_1}\left(\frac{R}{z+k_2}\right)^2} \tag{B.4}$$

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 Fig. B.2 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.

Wall effect

Regarding the wall effect, the experimental results shown in the right column of Fig. B.2 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.

Maps of aerodynamic effects

The complete range of effects produced by the aerodynamic phenomena that have been studied 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 (B.3) and (B.4). One example of this graphical representation can be seen in Fig. B.3 for a rectangular-shaped obstacle. These maps are used in Chapter 4 to implement the motion planner with Aerodynamics Awareness.

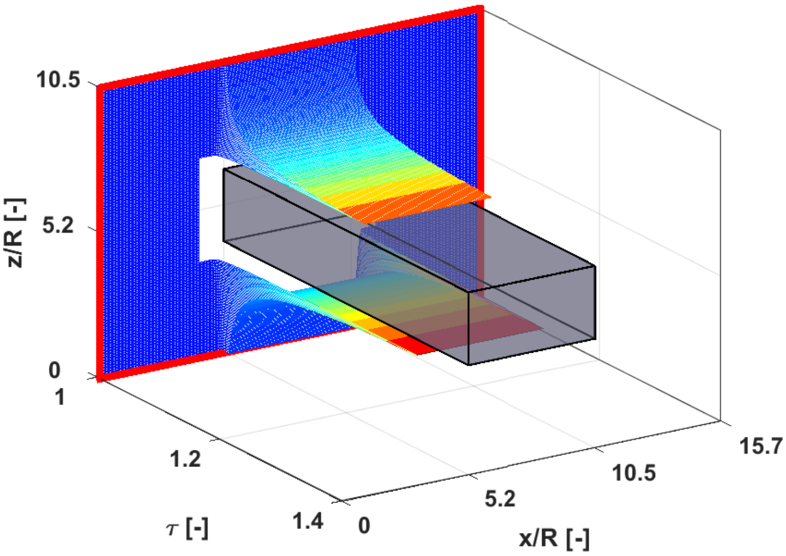

Figure B.3: 3D map corresponding to the aerodynamic effects associated with a rectangular-shaped obstacle. The colour scale represents the magnitude of the ratio τ of aerodynamic modification (from lower values in blue to higher values in red) while the white areas are associated with non-flyable regions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

Appendix C

Fundamentals of signal temporal logic for motion planning

This annex summarises the fundamentals of STL (Signal Temporal Logic) for motion planning based on the description in [6]. The motion planner presented in Chapter 6 for the installation of bird diverters on power lines is built over them.

C.1 Preliminaries

Firstly, a continuous-time dynamical system and its discrete-time version $\mathbf{x_{k+1}} = f(\mathbf{x_k}, \mathbf{u_k})$ are considered, where $\mathbf{x_k}, \mathbf{x_{k+1}} \in \mathcal{X} \in \mathbb{R}^n$ represent the current and the next state of the system, respectively, and $\mathbf{u_k} \in \mathcal{U} \in \mathbb{R}^m$ is the control input. Also, $f\colon \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is assumed to be differentiable in both arguments. Hence, given an initial state $\mathbf{x_0} \in \mathcal{X}_0 \in \mathbb{R}^n$, and the time vector $\mathbf{t} = (t_0, \ldots, t_N)^\top \in \mathbb{R}^{N+1}$, the finite control input sequence $\mathbf{u} = (\mathbf{u_0}, \ldots, \mathbf{u_{N-1}})^\top \in \mathbb{R}^N$ can be defined as the input to be given to the system to obtain the unique sequence of states $\mathbf{x} = (\mathbf{x_0}, \ldots, \mathbf{x_N})^\top \in \mathbb{R}^{N+1}$. Finally, the k-th element of the sequences $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{t}$ is denoted with $\mathbf{x_k}$, $\mathbf{u_k}$, and t_k, respectively.

Particularising for a multirotor from here on, the state $\mathbf{x}$ and control input $\mathbf{u}$ sequences can be defined as $\mathbf{x_k} = (p_k^{(1)}, v_k^{(1)}, p_k^{(2)}, v_k^{(2)}, p_k^{(3)}, v_k^{(3)})^\top$ and $\mathbf{u_k} = (a_k^{(1)}, a_k^{(2)}, a_k^{(3)})^\top$, where $p_k^{(j)}$, $v_k^{(j)}$, and $a_k^{(j)}$ represent the position, velocity, and acceleration of the vehicle at time instant t_k along the j-axis ($j = \{1, 2, 3\}$) of the inertial reference frame, respectively.

C.2 Signal temporal logic

Signal temporal logic was first-time introduced in [97] to monitor the behaviour of real-valued signals. Such logic allows the description of complex system behaviours in a succinct and unambiguous way, encoding requirements and specifications into a single STL formula φ. The full description of STL syntax and semantics can be found in [97, 121] and is not reported here for the sake of brevity. In short, an STL formula φ is defined over a set of predicates; these are, atomic prepositions that yield simple operations, such as belonging to a region or comparisons with real values. These predicates are combined together using boolean ($\neg$, $\wedge$, $\vee$) and temporal ($\Diamond$, $\Box$, $\mathcal{U}$) operators, and the resulting STL formula φ is considered to be valid if the expression takes a true ($\top$) logic value, and invalid ($\perp$) otherwise.

C.3 Robust signal temporal logic

The presence of system uncertainties, a dynamic environment, and unforeseen events can affect the satisfaction of an STL formula φ. In an attempt to have a manoeuvrability margin of satisfaction of φ, measuring how well (badly) a given specification is met, the concept of *robustness* ρ for the formula φ; this is ρ_φ, can be formulated. This robustness can be leveraged to guide an optimisation problem towards the best feasible solution to attain the mission satisfaction. Such value ρ relative to the system state $\mathbf{x}$ at time instant t_k can be formally described by using the equations (C.1), where $t_k + \mathbf{I}$ is meant here as the Minkowski sum between the scalar t_k and the time interval $\mathbf{I}$, and $\mu_i(\mathbf{x}, t_k)$ represents the generic i-th predicate p_i evaluated at time t_k. Further details can be found in [97, 121, 122]. In this case, $\mathbf{x}$ satisfies the STL formula φ at time t_k if $\rho_\varphi(\mathbf{x}, t_k) > 0$, and $\mathbf{x}$ violates φ if $\rho_\varphi(\mathbf{x}, t_k) \leq 0$.

$$\begin{array}{rcl}
\rho_{p_i}(\mathbf{x}, t_k) & = & \mu_i(\mathbf{x}, t_k) \\
\rho_{\neg\varphi}(\mathbf{x}, t_k) & = & -\rho_{\varphi}(\mathbf{x}, t_k) \\
\rho_{\varphi1\wedge\varphi2}(\mathbf{x}, t_k) & = & \min\left(\rho_{\varphi_1}(\mathbf{x}, t_k), \rho_{\varphi_2}(\mathbf{x}, t_k)\right) \\
\rho_{\Box_{\mathbf{I}}\varphi}(\mathbf{x}, t_k) & = & \min\limits_{t'_k\in[t_k+\mathbf{I}]} \rho_{\varphi}(\mathbf{x}, t'_k) \\
\rho_{\Diamond_{\mathbf{I}}\varphi}(\mathbf{x}, t_k) & = & \max\limits_{t'_k\in[t_k+\mathbf{I}]} \rho_{\varphi}(\mathbf{x}, t'_k) \\
\rho_{\varphi1\mathcal{U}\varphi2}(\mathbf{x}, t_k) & = & \max\limits_{t'_k\in[t_k+\mathbf{I}]} \Big(\min\left(\rho_{\varphi_2}(\mathbf{x}, t'_k)\right), \min\limits_{t''_k\in[t_k, t'_k]} \left(\rho_{\varphi_1}(\mathbf{x}, t''_k)\right)\Big)
\end{array} \tag{C.1}$$

Thus, the control inputs $\mathbf{u}$ can be computed by maximising the robustness over the set of finite-state and input sequences $\mathbf{x}$ and $\mathbf{u}$, respectively. The obtained sequence $\mathbf{u}^\star$ associated with $\mathbf{x}^\star$ is valid if $\rho_{\varphi}(\mathbf{x}^\star, t_k)$ is positive for every value of time t_k, where $\mathbf{x}^\star$ and $\mathbf{u}^\star$ obey the dynamical system. The larger $\rho_{\varphi}(\mathbf{x}^\star, t_k)$ is, the more robust the behaviour of the system is.

C.4 Smooth approximation

Considering $\lambda > 0$ as a tunable parameter, the smooth approximation [123] of the q-array maximum (max) and minimum (min) is:

$$\begin{aligned}
\max(\rho_{\varphi_1}, \ldots, \rho_{\varphi_q}) &\approx \frac{\sum_{i=1}^{q} \rho_{\varphi_i} e^{\lambda\rho_{\varphi_i}}}{\sum_{i=1}^{q} e^{\lambda\rho_{\varphi_i}}} \\
\min(\rho_{\varphi_1}, \ldots, \rho_{\varphi_q}) &\approx -\frac{1}{\lambda}\log\left(\sum_{i=1}^{q} e^{-\lambda\rho_{\varphi_i}}\right)
\end{aligned} \tag{C.2}$$

This approximation is asymptotically complete and smooth everywhere, as the widely known Log-Sum-Exponential (LSE) approximation [121], and also sound as it does not over approximate the maximum. The larger λ is, the greater the approximation of true robustness ρ is.

C.5 STL motion planner

Starting from mission specifications encoded as an STL formula φ, and replacing its robustness ρ_φ with the smooth approximation $\tilde{\rho}_\varphi$, the trajectory generation for multirotors can be formulated according to [6] as the set of three optimisation problems, one for each j-axis ($j = \{1, 2, 3\}$) of the inertial reference frame:

$$\begin{aligned} \underset{p_k^{(j)}, v_k^{(j)}, a_k^{(j)}}{\text{maximise}} \quad & \tilde{\rho}_\varphi(p_k^{(j)}, v_k^{(j)}, t_k) \\ \text{s.t.} \quad & |v_k^{(j)}| \leq v_{max}^{(j)}, |a_k^{(j)}| \leq a_{max}^{(j)} \\ & \mathbf{S_k^{(j)}}, \forall k = \{0, 1, \ldots, N-1\} \end{aligned} \tag{C.3}$$

where $v_{max}^{(j)}$ and $a_{max}^{(j)}$ are the desired maximum values of velocity and acceleration along the motion, respectively, and $\mathbf{S_k^{(j)}}(p_k^{(j)}, v_k^{(j)}, a_k^{(j)}) = (p_{k+1}^{(j)}, v_{k+1}^{(j)}, a_{k+1}^{(j)})^\top$ are the motion primitives presented in [99]. For more information about STL motion planning for multirotors, a detailed description can be found in [6].

Bibliography

[1] Google Maps. Reinforced concrete bridge. `https://www.google.com/maps/@37.4071952,-5.9489918,3a,90y,24.23h,109.91t/data=!3m6!1e1!3m4!1sC2Hg4rLXeYW_zI7hC1jjFg!2e0!7i13312!8i6656`. Accessed: 2022-08-31.

[2] A. Suarez, P. R. Soria, G. Heredia, B. C. Arrue, and A. Ollero. Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 992–997, 2017.

[3] C. Korpela, M. Orsag, and P. Oh. Towards valve turning using a dual-arm aerial manipulator. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pages 3411–3416. IEEE, 2014.

[4] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero. Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2107–2112. IEEE, 2014.

[5] Multi-Robot Systems Group (MRS). Czech Technical University in Prague (CTU). `http://mrs.felk.cvut.cz/`. Accessed: 2022-08-31.

[6] G. Silano, T. Baca, R. Penicka, D. Liuzza, and M. Saska. Power Line Inspection Tasks With Multi-Aerial Robot Systems Via Signal Temporal Logic Specifications. *IEEE Robotics and Automation Letters*, 6(2):4169–4176, 2021.

[7] MathWorks. MATLAB. `https://www.mathworks.com/`. Accessed: 2022-08-31.

[8] N. Koenig and A. Howard. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. IEEE, 2004.

[9] GRVC Robotics Lab. University of Seville (Spain). `https://grvc.us.es/`. Accessed: 2022-08-31.

[10] AEROARMS project, “AErial RObotic system integrating multiple ARMS and advanced manipulation capabilities for inspection and maintenance”. European Commission under the H2020 Framework Programme (grant agreement No 644271). `https://aeroarms-project.eu/`. Accessed: 2022-08-31.

[11] HYFLIERS project, “HYbrid FLying-rollIng with-snakE-aRm robot for contact inSpection”. European Commission under the H2020 Framework Programme (grant agreement No 779411). `https://www.oulu.fi/hyfliers/`. Accessed: 2022-08-31.

[12] AERIAL-CORE project, “AERIAL COgnitive integrated multi-task Robotic system with Extended operation range and safety”. European Commission under the H2020 Framework Programme (grant agreement No 871479). `https://aerial-core.eu/`. Accessed: 2022-08-31.

[13] M. A. Trujillo, J. R. Martinez-de Dios, C. Martin, A. Viguria, and A. Ollero. Novel aerial manipulator for accurate and robust industrial NDT contact inspection: A new tool for the oil and gas inspection industry. *Sensors*, 19(6):1305, 2019.

[14] M. Tognon, H. A. T. Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés, and A. Franchi. A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants. *IEEE Robotics and Automation Letters*, 4(2):1846–1851, 2019.

[15] P. Chermprayong, K. Zhang, F. Xiao, and M. Kovac. An integrated delta manipulator for aerial repair: A new aerial robotic system. *IEEE Robotics Automation Magazine*, 26(1):54–66, 2019.

[16] A. E. Jimenez-Cano, P. J. Sanchez-Cuevas, P. Grau, A. Ollero, and G. Heredia. Contact-based bridge inspection multirotors: Design, modeling, and control considering the ceiling effect. *IEEE Robotics and Automation Letters*, 4(4):3561–3568, 2019.

[17] A. E. Jimenez-Cano, J. Braga, G. Heredia, and A. Ollero. Aerial manipulator for structure inspection by contact from the underside. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1879–1884. IEEE, 2015.

[18] A. E. Jimenez-Cano, G. Heredia, and A. Ollero. Aerial manipulator with a compliant arm for bridge inspection. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 1217–1222. IEEE, 2017.

[19] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2668–2673. IEEE, 2011.

[20] R. Naldi, P. Pounds, S. De Marco, and L. Marconi. Output tracking for quadrotor-based aerial manipulators. In *American Control Conference (ACC), 2015*, pages 1855–1860. IEEE, 2015.

[21] S. Hamaza, I. Georgilas, M. Fernandez, P. Sanchez, T. Richardson, G. Heredia, and A. Ollero. Sensor installation and retrieval operations using an unmanned aerial manipulator. *IEEE Robotics and Automation Letters*, 4(3):2793–2800, 2019.

[22] Y. S. Sarkisov, M. J. Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak. Development of SAM: cable-Suspended Aerial Manipulator. In *2019*

International Conference on Robotics and Automation (ICRA), pages 5323–5329, 2019.

[23] M. Orsag, C. Korpela, S. Bogdan, and P. Oh. Valve turning using a dual-arm aerial manipulator. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 836–841. IEEE, 2014.

[24] A. Suarez, A. E. Jimenez-Cano, V. Vega, G. Heredia, A. Rodriguez-Castaño, and A. Ollero. Lightweight and human-size dual arm aerial manipulator. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 1778–1784. IEEE, 2017.

[25] A. Suarez, P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Aerial physical interaction in grabbing conditions with lightweight and compliant dual arms. *Applied Sciences*, 10(24), 2020.

[26] B. Yüksel, G. Buondonno, and A. Franchi. Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-elastic-joints. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 561–566. IEEE, 2016.

[27] R. Zufferey, A. O. Ancel, C. Raposo, S. F. Armanini, A. Farinha, R. Siddall, I. Berasaluce, H. Zhu, and M. Kovac. Sailmav: design and implementation of a novel multi-modal flying sailing robot. *IEEE Robotics and Automation Letters*, 4(3):2894–2901, 2019.

[28] S. F. Armanini, R. Siddall, and M. Kovac. Modelling and simulation of a bioinspired aquatic micro aerial vehicle. In *AIAA Aviation 2019 Forum*, page 3115, 2019.

[29] J. R. Page and P. E. Pounds. The Quadroller: Modeling of a UAV/UGV hybrid quadrotor. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4834–4841. IEEE, 2014.

[30] A. Kalantari and M. Spenko. Design and experimental validation of HyTAQ, a hybrid terrestrial and aerial quadrotor. In *2013 IEEE International Conference on Robotics and Automation*, pages 4445–4450. IEEE, 2013.

[31] C. J. Dudley, A. C. Woods, and K. K. Leang. A micro spherical rolling and flying robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5863–5869. IEEE, 2015.

[32] R. J. Bachmann, R. Vaidyanathan, and R. D. Quinn. Drive train design enabling locomotion transition of a small hybrid air-land vehicle. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5647–5652. IEEE, 2009.

[33] L. Daler, J. Lecoeur, P. B. Hählen, and D. Floreano. A flying robot with adaptive morphology for multi-modal locomotion. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1361–1366. IEEE, 2013.

[34] W. C. Myeong, K. Y. Jung, S. W. Jung, Y. Jung, and H. Myung. Development of a drone-type wall-sticking and climbing robot. In *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 386–389. IEEE, 2015.

[35] A. Suarez, A. Caballero, A. Garofano, P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Aerial manipulator with rolling base for inspection of pipe arrays. *IEEE Access*, 8:162516–162532, 2020.

[36] A. Lopez-Lora, P. J. Sanchez-Cuevas, A. Suárez, A. Garofano-Soldado, A. Ollero, and G. Heredia. MHYRO: Modular HYbrid RObot for contact inspection and maintenance in oil & gas plants. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1268–1275. IEEE, 2020.

[37] Preformed Line Products (PLP). UAV-Installed Bird Diverters. `https://preformed.com/energy/distribution/wildlife-protection/uav-installed-bird-diverters`. Accessed: 2022-08-31.

[38] Fulcrum Air. Robotic Installation of Bird Flight Diverters. `https://fulcrumair.com/operational-services/`. Accessed: 2022-08-31.

[39] J. Cacace, S. M. Orozco-Soto, A. Suarez, A. Caballero, M. Orsag, S. Bogdan, G. Vasiljevic, E. Ebeid, J. A. Acosta, and A. Ollero. Safe Local Aerial Manipulation for the Installation of Devices on Power Lines: AERIAL-CORE First Year Results and Designs. *Applied Sciences*, 11(13):6220, 2021.

[40] A. Suarez, H. Romero, R. Salmoral, J. A. Acosta, J. Zambrano, and A. Ollero. Experimental Evaluation of Aerial Manipulation Robot for the Installation of Clip Type Bird Diverters: Outdoor Flight Tests. In *2021 Aerial Robotic Systems Physically Interacting with the Environment*, pages 1–7, 2021.

[41] A. Rodriguez-Castaño, S. R. Nekoo, H. Romero, R. Salmoral, J. A. Acosta, and A. Ollero. Installation of Clip-Type Bird Flight Diverters on High-Voltage Power Lines with Aerial Manipulation Robot: Prototype and Testbed Experimentation. *Applied Sciences*, 11(16), 2021.

[42] N. Iversen, A. Kramberger, O. B. Schofield, and E. Ebeid. Pneumatic-Mechanical Systems in UAVs: Autonomous Power Line Sensor Unit Deployment. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 548–554, 2021.

[43] E. Cuniato, J. Cacace, M. Selvaggio, F. Ruggiero, and V. Lippiello. A hardware-in-the-loop simulator for physical human-aerial manipulator cooperation. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 830–835. IEEE, 2021.

[44] R. Ragel, I. Maza, F. Caballero, and A. Ollero. Comparison of motion planning techniques for a multi-rotor UAS equipped with a multi-joint manipulator arm. In *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2015 Workshop on*, pages 133–141. IEEE, 2015.

[45] Y. Yu, X. Ding, and J. J. Zhu. Dynamic modeling and control for aerial arm-operating of a multi-propeller multifunction aerial robot. *Advanced Robotics*, 31(13):665–679, 2017.

[46] H. Kim, H. Seo, J. Kim, and H. J. Kim. Sampling-based motion planning for aerial pick-and-place. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7402–7408. IEEE, 2019.

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

[48] H. Kim, H. Seo, C. Y. Son, H. Lee, S. Kim, and H. J. Kim. Cooperation in the air: A learning-based approach for the efficient motion planning of aerial manipulators. *IEEE Robotics & Automation Magazine*, 25(4):76–85, 2018.

[49] H. Lee, C. Y. Son, and H. J. Kim. Collision-free path planning for cooperative aerial manipulators under velocity and curvature constraints. *IEEE Access*, 7:171153–171162, 2019.

[50] M. Manubens, D. Devaurs, L. Ros, and J. Cortés. Motion planning for 6-D manipulation with aerial towed-cable systems. In *Robotics: Science and Systems (RSS)*, page 8p, 2013.

[51] D. Hirano, H. Kato, and T. Saito. Online path planning and compliance control of space robot for capturing tumbling large object. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2909–2916, 2018.

[52] M. H. Korayem, H. R. Nohooji, and A. Nikoobin. Path planning of mobile elastic robotic arms by indirect approach of optimal control. *International Journal of Advanced Robotic Systems*, 8(1):10–20, 2011.

[53] M. Zhao, F. Shi, T. Anzai, K. Chaudhary, X. Chen, K. Okada, and M. Inaba. Flight motion of passing through small opening by dragon: Transformable multilinked aerial robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4735–4742, 2018.

[54] J. Zhang, C. Hu, R. G. Chadha, and S. Singh. Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation. *Journal of Field Robotics*, 37(8):1300–1313, 2020.

[55] G. Chen, D. Sun, W. Dong, X. Sheng, X. Zhu, and H. Ding. Computationally Efficient Trajectory Planning for High Speed Obstacle Avoidance of a Quadrotor With Active Sensing. *IEEE Robotics and Automation Letters*, 6(2):3365–3372, 2021.

[56] M. Elbanhawi and M. Simic. Sampling-based robot motion planning: A review. *IEEE Access*, 2:56–77, 2014.

[57] S. M. LaValle and J. J. Kuffner Jr. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[58] E. Koyuncu and G. Inalhan. A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments. In *Intelligent Robots and Systems (IROS), 2008 IEEE/RSJ International Conference on*, pages 815–821. IEEE, 2008.

[59] C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pages 649–666. Springer, 2016.

[60] A. Boeuf, J. Cortés, R. Alami, and T. Siméon. Planning agile motions for quadrotors in constrained environments. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pages 218–223. IEEE, 2014.

[61] A. Boeuf, J. Cortés, R. Alami, and T. Siméon. Enhancing sampling-based kinodynamic motion planning for quadrotors. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2447–2452. IEEE, 2015.

[62] M. Tognon, E. Cataldi, H. A. T. Chavez, G. Antonelli, J. Cortés, and A. Franchi. Control-aware motion planning for task-constrained aerial manipulation. *IEEE Robotics and Automation Letters*, 3(3):2478–2484, 2018.

[63] J. Welde, J. Paulos, and V. Kumar. Dynamically feasible task space planning for underactuated aerial manipulators. *IEEE Robotics and Automation Letters*, 6(2):3232–3239, 2021.

[64] L. Danjun, Z. Yan, S. Zongying, and L. Geng. Autonomous landing of quadrotor based on ground effect modelling. In *34th Chinese Control Conference (CCC)*, pages 5647–5652. IEEE, 2015.

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

[66] P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Multirotor uas for bridge inspection by contact using the ceiling effect. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 767–774. IEEE, 2017.

[67] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming. A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems. *IEEE Transactions on Vehicular Technology*, 65(12):9585–9596, 2016.

[68] A. Lakas, B. Belkhouche, O. Benkraouda, A. Shuaib, and H. J. Alasmawi. A framework for a cooperative UAV-UGV system for path discovery and planning. In *2018 International Conference on Innovations in Information Technology (IIT)*, pages 42–46. IEEE, 2018.

[69] F. Ropero, P. Muñoz, and M. D. R-Moreno. TERRA: A path planning algorithm for cooperative UGV–UAV exploration. *Engineering Applications of Artificial Intelligence*, 78:260–272, 2019.

[70] T. Petrovic, T. Haus, B. Arbanas, M. Orsag, and S. Bogdan. Can UAV and UGV be best buddies? Towards heterogeneous aerial-ground cooperative robot system

for complex aerial manipulation tasks. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, volume 1, pages 238–245. IEEE, 2015.

[71] B. Arbanas, A. Ivanovic, M. Car, T. Haus, M. Orsag, T. Petrovic, and S. Bogdan. Aerial-ground robotic system for autonomous delivery tasks. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 5463–5468. IEEE, 2016.

[72] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan. Decentralized planning and control for UAV–UGV cooperative teams. *Autonomous Robots*, 42(8):1601–1618, 2018.

[73] D. D. Fan, R. Thakker, T. Bartlett, M. B. Miled, L. Kim, E. Theodorou, and A. Agha-mohammadi. Autonomous hybrid ground/aerial mobility in unknown environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3070–3077. IEEE, 2019.

[74] R. Zhang, Y. Wu, L. Zhang, C. Xu, and F. Gao. Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles. *IEEE Robotics and Automation Letters*, 7(2):3008–3015, 2022.

[75] S. M. LaValle. *Rapidly-exploring random trees: A new tool for path planning.* Citeseer, 1998.

[76] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations research*, 52(5):723–738, 2004.

[77] Y. Liu, J. Shi, Z. Liu, J. Huang, and T. Zhou. Two-layer routing for high-voltage powerline inspection by cooperated ground vehicle and drone. *Energies*, 12(7):1385, 2019.

[78] F. Nekovář, J. Faigl, and M. Saska. Multi-Tour Set Traveling Salesman Problem in Planning Power Transmission Line Inspection. *IEEE Robotics and Automation Letters*, 6(4):6196–6203, 2021.

[79] A. Madridano, A. Al-Kaff, D. Martín, and A. de la Escalera. Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*, 173:114660, 2021.

[80] C. E. Luis, M. Vukosavljev, and A. P. Schoellig. Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, 5(2):604–611, 2020.

[81] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access*, 7:146264–146272, 2019.

[82] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian. Trajectory Planning for Quadrotor Swarms. *IEEE Transactions on Robotics*, 34(4):856–869, 2018.

[83] Y. Chen, X. C. Ding, A. Stefanescu, and C. Belta. Formal Approach to the Deployment of Distributed Robotic Teams. *IEEE Transactions on Robotics*, 28(1):158–171, 2012.

[84] L. Lindemann and D. V. Dimarogonas. Barrier Function Based Collaborative Control of Multiple Robots Under Signal Temporal Logic Tasks. *IEEE Transactions on Control of Network Systems*, 7(4):1916–1928, 2020.

[85] K. Leahy, Z. Serlin, C. Vasile, A. Schoer, A. M. Jones, R. Tron, and C. Belta. Scalable and Robust Algorithms for Task-Based Coordination From High-Level Specifications (ScRATCHeS). *IEEE Transactions on Robotics*, 2021. In Press.

[86] H. Baik and J. Valenzuela. Unmanned aircraft system path planning for visually inspecting electric transmission towers. *Journal of Intelligent & Robotic Systems*, 95(3):1097–1111, 2019.

[87] S. H. Tang, W. Khaksar, N. B. Ismail, and M. K. A. Ariffin. A review on robot motion planning approaches. *Pertanika Journal of Science and Technology*, 20(1):15–29, 2012.

[88] Advanced Center for Aerospace Technologies (CATEC). La Rinconada, Seville (Spain). `http://www.catec.aero/en`. Accessed: 2022-08-31.

[89] Vicon. Motion Capture Systems. `https://www.vicon.com/`. Accessed: 2022-08-31.

[90] G. J. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2006.

[91] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2016.

[92] Leica Geosystems. Leica MS50 robotic total station. `https://leica-geosystems.com/`. Accessed: 2022-08-31.

[93] S. Küfeoğlu. *Economic Impacts of Electric Power Outages and Evaluation of Customer Interruption Costs*. PhD thesis, Aalto University, Helsinki (Finland), 2015.

[94] K. Hunting. *A Roadmap for PIER Research on Avian Collisions with Power Lines in California. Commission Staff Report*. California Energy Commission, 2002.

[95] M. Ferrer, V. Morandini, R. Baumbusch, R. Muriel, M. De Lucas, and C. Calabuig. Efficacy of different types of “bird flight diverter” in reducing bird mortality due to collision with transmission power lines. *Global Ecology and Conservation*, 23:e01130, 2020.

[96] IEEE Guide for Maintenance Methods on Energized Power Lines. *IEEE Std 516-2021 (Revision of IEEE Std 516-2009)*, pages 1–159, 2022.

[97] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004.

[98] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics (NRL)*, 54(8):811–819, 2007.

[99] M. W. Mueller, M. Hehn, and R. D'Andrea. A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation. *IEEE Transactions on Robotics*, 31(6):1294–1310, 2015.

[100] M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming. `http://cvxr.com/cvx/`. Accessed: 2022-08-31.

[101] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[102] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[103] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska. The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*, 102(26):1–28, 2021.

[104] D. Hert, T. Baca, P. Petracek, V. Kratky, V. Spurny, M. Petrlik, M. Vrba, D. Zaitlik, P. Stoudek, V. Walter, et al. MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments. In *Unmanned Aircraft Systems (ICUAS), 2022 International Conference on*, pages 1303–1312. IEEE, 2022.

[105] L. Bauersfeld and D. Scaramuzza. Range, endurance, and optimal speed estimates for multicopters. *IEEE Robotics and Automation Letters*, 7(2):2953–2960, 2022.

[106] J. Pan and D. Manocha. GPU-based parallel collision detection for fast motion planning. *The International Journal of Robotics Research*, 31(2):187–200, 2012.

[107] D. Devaurs, T. Siméon, and J. Cortés. Parallelizing RRT on distributed-memory architectures. In *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pages 2261–2266. IEEE, 2011.

[108] D. Devaurs, T. Siméon, and J. Cortés. Parallelizing RRT on large-scale distributed-memory architectures. *IEEE Transactions on Robotics*, 29(2):571–579, 2013.

[109] C. R. de Cos, J. A. Acosta, and A Ollero. Relative-pose optimisation for robust and nonlinear control of unmanned aerial manipulators. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 999–1005. IEEE, 2017.

[110] J. Song, P. J. Sanchez-Cuevas, and M. Olivares-Mendez. Towards online system identification: Benchmark of model identification techniques for variable dynamics uav applications. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 590–598. IEEE, 2022.

[111] A. Garofano-Soldado, P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Numerical-experimental evaluation and modelling of aerodynamic ground effect for small-scale tilted propellers at low reynolds numbers. *Aerospace Science and Technology*, page 107625, 2022.

[112] P. J. Sanchez-Cuevas, V. Martín, G. Heredia, and A. Ollero. Aerodynamic effects in multirotors flying close to obstacles: modelling and mapping. In *Iberian Robotics conference*, pages 63–74. Springer, 2019.

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

[114] T. R. Kane and D. A. Levinson. *Dynamics, theory and applications*. McGraw-Hill, 1985.

[115] L. A. Sandino, M. Bejar, and A. Ollero. A survey on methods for elaborated modeling of the mechanics of a small-size helicopter. Analysis and comparison. *Journal of Intelligent & Robotic Systems*, 72(2):219–238, 2013.

[116] T. R. Kane. Motiongenesis. `http://www.motiongenesis.com/`. Accessed: 2022-08-31.

[117] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[118] F. Real, A. Torres-González, P. Ramon-Soria, J. Capitán, and A. Ollero. Unmanned aerial vehicle abstraction layer: An abstraction layer to operate unmanned aerial vehicles. *International Journal of Advanced Robotic Systems*, 17(4):1729881420925011, 2020.

[119] P. J. Sánchez Cuevas. *Modelling And Control Of Aerial Manipulators Considering Aerodynamic Effects*. PhD thesis, University of Seville (Spain), 2020.

[120] I. C. Cheeseman and W. E. Bennett. The effect of ground on a helicopter rotor in forward flight. 1955.

[121] A. Donzé and O. Maler. Robust satisfaction of temporal logic over real-valued signals. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 92–106. Springer, 2010.

[122] G. Pola and M. D. Di Benedetto. Control of Cyber-Physical-Systems with logic specifications: A formal methods approach. *Annual Reviews in Control*, 47:178–192, 2019.

[123] Y. Gilpin, V. Kurtz, and H. Lin. A Smooth Robustness Measure of Signal Temporal Logic for Symbolic Control. *IEEE Control Systems Letters*, 5(1):241–246, 2021.