

Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms

Alvaro Caballero¹, Manuel Bejar², Angel Rodriguez-Castaño¹
and Anibal Ollero¹

Abstract

Human activities in maintenance of industrial plants pose elevated risks as well as significant costs due to the required shutdowns of the facility. An aerial robotic system with two arms for long reach manipulation in cluttered environments is presented to alleviate these constraints. The system consists of a multirotor with a long bar extension that incorporates a lightweight dual arm in the tip. This configuration allows aerial manipulation tasks even in hard-to-reach places. The objective of this work is the development of planning strategies to move the aerial robotic system with two arms for long reach manipulation in a safe and efficient way for both navigation and manipulation tasks. The motion planning problem is addressed considering jointly the aerial platform and the dual arm in order to achieve wider operating conditions. Since there exists a strong dynamical coupling between the multirotor and the dual arm, safety in obstacle avoidance will be assured by introducing dynamics awareness in the operation of the planner. On the other hand, the limited maneuverability of the system emphasizes the importance of energy and time efficiency in the generated trajectories. Accordingly, an adapted version of the optimal Rapidly-exploring Random Tree algorithm has been employed to guarantee their optimality. The resulting motion planning strategy has been evaluated through simulation in two realistic industrial scenarios, a riveting application and a chimney repairing task. To this end, the dynamics of the aerial robotic system with two arms for long reach manipulation has been properly modeled, and a distributed control scheme has been derived to complete the test bed. The satisfactory results of the simulations are presented as a first validation of the proposed approach.

Keywords

Unmanned aerial vehicle, aerial manipulation, motion planning, Rapidly-exploring Random Tree, planning with dynamics awareness

Date received: 31 August 2017; accepted: 4 March 2018

Topic: Special Issue – Mobile Robots

Topic Editor: Lino Marques

Associate Editor: Michal Kelemen

Introduction

Among the numerous applications in which unmanned aerial vehicles (UAVs) can be used, aerial manipulation is arousing much interest. Potential applications in this field include instrument deployment, maintenance operation, and contact inspection in industrial sites in which the access is very dangerous or costly. The motivation is to decrease risks and operational costs in these scenarios with the

¹Robotics, Vision and Control Group (GRVC), University of Seville, Seville, Spain

²Robotics, Vision and Control Group (GRVC), University Pablo de Olavide, Seville, Spain

Corresponding author:

Alvaro Caballero, Robotics, Vision and Control Group (GRVC), University of Seville, Camino de los descubrimientos s/n, 41092, Seville, Spain.

Email: alvarocaballero@us.es



support of aerial manipulation 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 operations without shutdowns of the facilities (the mandatory safety policy in case of human operation) and without the use of scaffolding or cranes.

These new promising applications of aerial robotic systems for manipulation tasks also bring new challenges. On the one hand, it is necessary to develop new manipulation 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 for operating autonomously the UAV and the manipulators should be extended for the integrated system. In this respect, one of the most challenging issues is the development of new methods that consider both the UAV and the manipulator dynamics when planning the motion of the complete system. When moving between different locations inside a dense industrial installation, this planning will be essential for the generation of accurate and collision-free movements close to obstacles. In addition, it will also enable rapid and agile maneuvers (e.g. using the arm to let the aerial manipulator turn quickly) aiming to approach to the goal location avoiding waste of battery.

Many research works about aerial manipulation have been recently published. Mellinger et al.¹ present the design of several lightweight, low-complexity grippers that allow quadrotors to grasp and perch on branches or beams and pick up and transport payloads. In a very different system scale, Kondak et al.² propose a system for aerial manipulation, composed of a helicopter and an industrial manipulator. The usage of an industrial manipulator is motivated by practical applications which were identified in different cooperation projects with the industry. In another valuable contribution, Naldi et al.³ present a control strategy for aerial manipulators that allow both position and orientation tracking by end effector. Another interesting research topic is the usage of cables for aerial manipulation. Along with this line, Manubens et al.⁴ study aerial six-dimensional manipulation using flying robots. To this end, the authors propose a motion planning approach for the reliable six-dimensional quasi-static manipulation with an aerial towed-cable system.

However, among the different contributions focused on aerial manipulation, very few 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 pieces (to avoid swinging movements), application of torques, or execution of different tasks simultaneously. Thus, Orsag et al.⁵ and Korpela et al.⁶ propose a dual arm aerial manipulator to turn a valve that requires a tightly integrated control scheme between aircraft and both manipulators. The arm-aircraft system for valve turning is validated through flight tests. On the other hand, in Suarez et al.,⁷ a human-size and lightweight dual

arm manipulator is integrated in a multirotor platform and tested in outdoor flights. Concerning theoretical contributions, Yüksel et al.⁸ introduce a generic planar aerial manipulator with any number of arms attached at the center of mass of a UAV. The authors prove that this kind of systems are differentially flat regardless of the number of joints of each arm and their kinematic and dynamic parameters. This theory is validated by simulating object grasping and transportation tasks.

On the other hand, although a large amount of works have been focused on the development of control techniques for the system integrating the aerial vehicle and the manipulator devices, not many of them deal with the associated motion planning problem. Yu et al.⁹ propose an optimal planning strategy for a quadrotor with two arms that minimizes the interaction between the aerial platform and the arms. Furthermore, the existing contributions like Ragel et al.¹⁰ usually assume a strong simplification by addressing the planning problem in a decoupled way, that is, adopting independent planners for the UAV and the manipulators that switch their operation according to the mission phase. This means that during the navigation phase, the arm configuration is assumed to be fixed and hence the UAV planner is in charge of planning the motion. In contrast, the manipulation phase is resolved by using the manipulators planners and assuming that the aerial platform is not moving.

Concerning motion planning problems where dynamics cannot be neglected, there also exists some contributions aiming at finding admissible and collision-free trajectories for such systems. Elbanhawi and Simic¹¹ present an exhaustive review in this respect. In a relevant contribution, LaValle and Kuffner Jr.¹² propose 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, which implies high computational costs. Alternatively, Koyuncu and Inalhan¹³ and Richter et al.¹⁴ 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 the geometry of 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—the system is replaced by its bounding sphere—since it could complicate the existence of a collision-free path. Boeuf et al.^{15,16} 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 contribution, 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

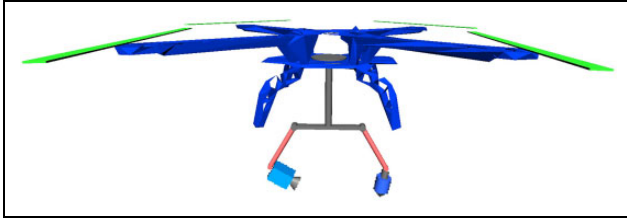


Figure 1. Aerial robotic system with two arms for long reach manipulation.

technique to avoid generating local trajectories that violate kinodynamic constraints. These contributions have been validated through simulation.

This article extends previous work of the authors.¹⁷ This research line explores dual arm configurations that guarantee long reach manipulation in those scenarios where the target is far from the operation area of the UAV. In order to meet these requirements, a new aerial robotic system with two arms for long reach manipulation (ARS-LRM) has been proposed. More precisely, the system consists of a multirotor with a long bar extension that incorporates a lightweight dual arm in the tip (see Figure 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 the single arm configurations existing in the literature.

Concerning the motion planning problem, this article investigates strategies for both navigation and manipulation tasks in cluttered environments. To this end, the aerial platform and the dual arm device are considered jointly within the planner operation, which constitutes a remarkable difference to previous contributions where the 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 and safer operating conditions. On the one hand, there exists a strong dynamical coupling between the multirotor and the dual arm. Then, it is necessary to introduce dynamics awareness (DA) in the planner for robust obstacle avoidance. Accordingly, the expansion of the search tree is based on the behavior of the controlled system. Figure 2 illustrates an example of the undesired dynamic effects that must be avoided. Regarding the operation basis of the planner, an adapted version of the optimal Rapidly-exploring Random Tree (RRT*) algorithm that optimizes energy and time performance has been developed.

The work begins with the “Modeling and control” section presenting the structure of the integrated platform, the corresponding multi-body dynamical model and finally the distributed control approach derived for the system. Then, in the “Motion planner with DA” section, the proposed planning algorithm is explained in detail. In order to better illustrate its benefits, the “Application scenarios” section

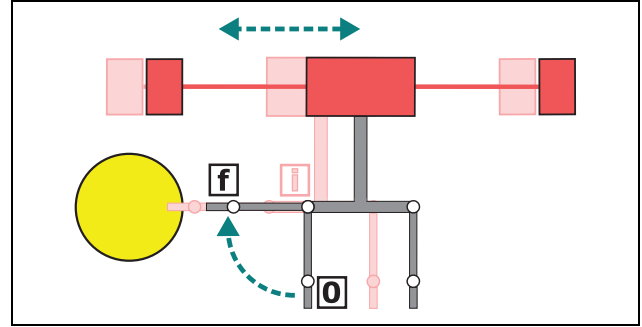


Figure 2. ARS-LRM spreading the left arm from initial position 0 to final position f . The UAV oscillation (shaded intermediate position i) produces a collision with the yellow obstacle. ARS-LRM: aerial robotic system with two arms for long reach manipulation. UAV: unmanned aerial vehicle.

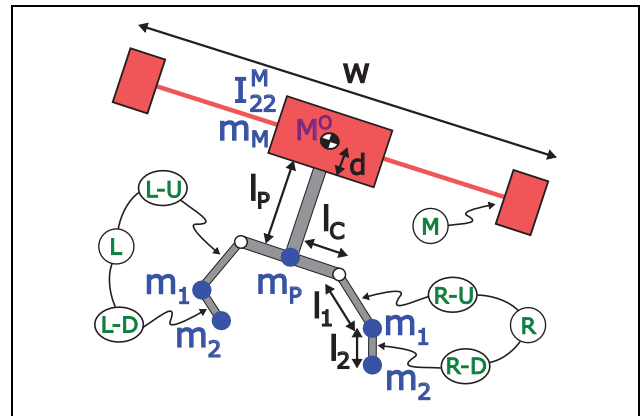


Figure 3. Geometry and mass distribution of the ARS-LRM. ARS-LRM: aerial robotic system with two arms for long reach manipulation.

defines two realistic industrial scenarios given by a riveting application and a chimney repairing task. After presenting the complete system as well as the motivating scenarios, the “Simulation results” section includes several simulations of the controlled ARS-LRM when following the planned trajectories to endorse the validity of the proposed algorithm. Finally, the ‘Conclusions’ section is devoted to conclusions and future work.

Modeling and control

System description

As can be seen in Figure 3, the proposed ARS-LRM consists of a multirotor with a long bar extension that incorporates a lightweight dual arm in the tip. This configuration allows aerial manipulation tasks in hard-to-reach places increasing considerably the safety distance between rotors and manipulated objects. Furthermore, the dual arm offers extended manipulation capabilities with respect to the single arm configurations existing in the literature. In this first prototype of the system, each separate arm is composed of

Table 1. ARS-LRM parameters.

| | Parameter | Value | Units |
|------------------|------------|-------|-------------------|
| Mass and inertia | m_M | 6.5 | kg |
| | I_{22}^M | 0.093 | 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 |

ARS-LRM: aerial robotic system with two arms for long reach manipulation.

two links, corresponding the lower one to the end effector, but further extensions of the manipulation chain are considered in future work.

A planar characterization of the system will serve for establishing a first proof of concept for the ARS-LRM setup. This simplified approach eases the modeling and control developments while maintaining the operation basis of the system. Following this assumption, the multirotor is characterized by a mass m_M , a principal moment of inertia I_{22}^M , and dimensions $2d \times w$. Regarding the long bar, its longitude is given by l_P and it is assumed to be aligned with the UAV center of mass M^O at a distance d . The crosspiece in the tip is defined by a length of $2l_C$. The total mass of the long bar and the crosspiece is m_P and will be treated as the punctual mass located where the long bar and the crosspiece intersects for simplicity purposes. Finally, the two arms are characterized, 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 the aforementioned parameters are shown in Table 1.

Modeling

According to Kondak et al.,¹⁸ the dynamics of a multirotor under 20 kg is mostly determined by its mechanical model. This article embraces the same assumption, and consequently the behavior of the ARS-LRM platform will be described by means of the mechanical model of the complete system. To this end, specific methodologies for multi-body systems will be applied below.

Several approaches can be found in the literature to derive equations of motion for mechanical systems. However, Kane's method¹⁹ has proved in Sandino et al.²⁰ to hold some unique advantages over other traditional approaches when addressing multi-body robotic systems like the ARS-LRM under study in this article. One of the

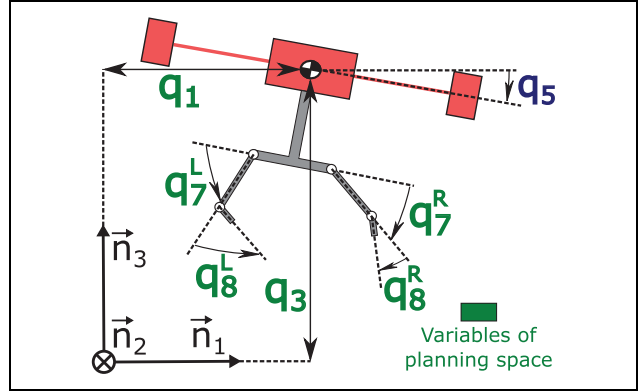


Figure 4. Configuration variables of the ARS-LRM model. In green, the variables selected later for the planning space. ARS-LRM: aerial robotic system with two arms for long reach manipulation.

most remarkable advantages is that the adoption of generalized speeds enables the derivation of a compact model in first-order differential equations that are uncoupled in the generalized speed derivatives. Other important features are the easy computerization as well as the computational efficiency of the resulting equations of motion.

The configuration variables selected as system generalized coordinates are the longitudinal q_1 and vertical q_3 positions of the UAV center of mass M^O in the inertial reference frame N , the multirotor pitch angle q_5 and the joint angles both for left L and right R arms q_7^L, q_8^L, q_7^R , and q_8^R (see Figure 4). Generalized speeds u_i are defined as

$$\begin{aligned}
 {}^M\omega^{R-U} &= u_7^R \mathbf{n}_2 \\
 {}^N\mathbf{v}^{M^O} &= u_1 \mathbf{n}_1 + u_3 \mathbf{n}_3 & {}^{R-U}\omega^{R-D} &= u_8^R \mathbf{n}_2 \\
 {}^N\omega^M &= u_5 \mathbf{n}_2 & {}^{M^O-L-U}\omega &= -u_7^L \mathbf{n}_2 \\
 & & {}^{L-U}\omega^{L-D} &= -u_8^L \mathbf{n}_2
 \end{aligned} \quad (1)$$

where ${}^N\mathbf{v}^{M^O}$ is the velocity of the UAV center of mass M^O with respect to the inertial reference frame N and ${}^i\omega^j$ is the angular velocity of the element j with respect to the element i (see Figure 3 to identify the different elements i, j). Previous equations lead to the following kinematic differential equations

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

Regarding forces and torques exerted on the ARS-LRM (see Figure 5), the rotors generate a resultant lifting force $F_3 \mathbf{a}_3$ applied at the multirotor center of mass M^O as well as a torque $T_2 \mathbf{a}_2$ applied to rigid body M . On the other hand, 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 software²¹ leads to the following dynamic differential

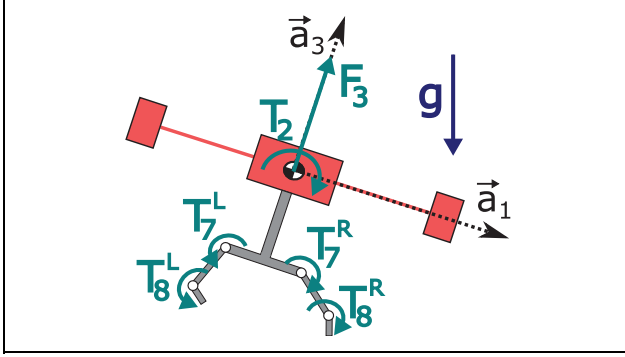


Figure 5. Forces and torques applied to the ARS-LRM. ARS-LRM: aerial robotic system with two arms for long reach manipulation.

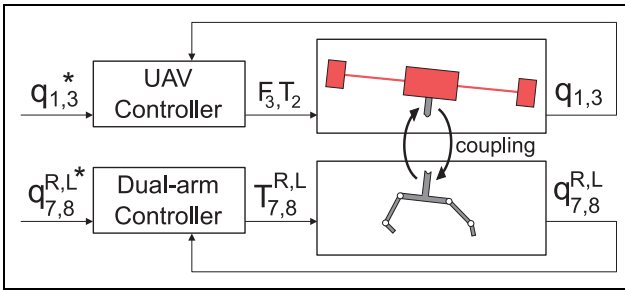


Figure 6. Block diagram of the distributed control scheme.

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 , and q_8^L and the system parameters defined in Table 1, and g is the gravity acceleration

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

Control

After modeling the ARS-LRM, a distributed control scheme (see Figure 6) has been derived to provide the system with the capacity of executing navigation and manipulation maneuvers. The objective is the completion of the simulation environment that will allow the investigation of new planning strategies for the ARS-LRM platform. A basic control structure that makes use of nonlinear control strategies based on model inversion shall suffice to complete the test bed.

Regarding the multirotor, the control scheme is inspired by Kondak et al.¹⁸ and consists in linearizing the system through model inversion and applying applying

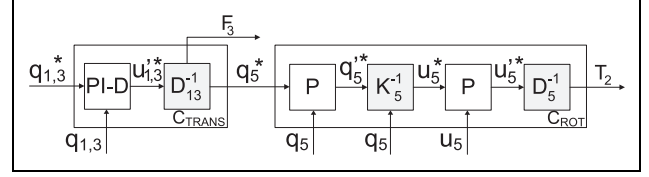


Figure 7. Block diagram of the UAV controller. UAV: unmanned aerial vehicle.

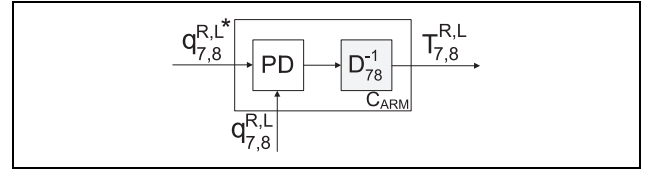


Figure 8. Block diagram of the arm controller.

Proportional -Integral -Derivative (PID) control laws to the resultant dynamics. 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 error. A general overview of the control scheme is shown in Figure 7, where D_{13}^{-1} , K_5^{-1} , and D_5^{-1} blocks represent, respectively, the inversions of the translational dynamics, rotational kinematics, and rotational dynamics.

The control strategy selected for each arm is again based on linearization through model inversion and applying Proportional -Derivative (PD) control, which yields a non-linear control law capable of commanding the link positions of both arms. The schematic representation of this approach is shown in Figure 8, where D_{78}^{-1} represents the block in charge of inverting arm dynamics, and torques T_7^R , T_8^R , T_7^L , and T_8^L correspond to the output signals of the 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.

Motion planner with DA

According to Tang et al.²² and Elbanhawi and Simic¹¹, sampling-based planners like the family of RRT algorithms²³ have demonstrated high potential in finding fast solutions for high-dimensional robots. Furthermore, some of these methods bring the possibility of generating motion plans that optimize certain cost functions, as for the case of RRT* variations.²⁴ This makes it possible to find an optimal solution in terms of a specific metric. Taking all these considerations into account, and recalling the limited maneuverability of the ARS-LRM, an RRT*-based algorithm that optimizes energy and execution time of the planned motion has been selected for the ARS-LRM.

Another determining factor for planner performance is the planning space considered when exploring the different possibilities of motion. In this work, the planner explores jointly the configuration variables of the aerial platform (with the exception of pitch angle q_5) and the dual arm, which corresponds to the variables in green color in Figure 4. 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 intermediate links.

As was previously advanced, the strong dynamical coupling between the multirotor and the dual arm makes it necessary considering the dynamics of the ARS-LRM within the motion planner for robust obstacle avoidance in cluttered environments. With this purpose, the standard RRT* approach is transformed into a more advanced RRT* algorithm that incorporates DA (RRT*-DA). To this end, the expansion of the search tree is based on the behavior of the controlled system, which means that collision detection is calculated through close-loop simulations of the controlled system instead of using geometrical interpolation between states. This guarantees that the resultant planned trajectories are both free of obstacles and compatible with the dynamic constraints previously mentioned. The approach being investigated for the dynamical analysis of the tree extension is based on a root-to-candidate validation. Thus, not only the dynamical feasibility of the new possible branch reaching the candidate node is analyzed, but also the complete path from the tree root.

The pseudocode of the planning algorithm that has been developed in this work is shown in algorithm 1. It mainly corresponds to the common structure of the RRT* algorithm but some of the intermediate functionalities have been customized for the problem under study giving rise to the RRT*-DA algorithm presented in this article. These particular developments will be dealt with in detail hereafter.

Discretization of the planning space

Due to the high dimension of the planning space integrating both UAV and manipulator states that is considered by the ARS-LRM planner (see Figure 4), a continuous treatment of the variable ranges considered in the sampling operation would require excessively elevated execution times for achieving convergent solutions. The former suggests the adoption of discretization patterns that guarantee bounded execution times for the planner. However, this discretization must be accomplished carefully since an excessively reduced set of data could 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.

Algorithm 1. RRT*-DA algorithm.

Input: $map, param$
Output: $trajectory$

```

1:  $Tree \leftarrow INITIALIZATION(map, param)$ 
2: for  $i = 1$  to  $iter_{max}$  do
3:    $x_{rand} \leftarrow SAMPLE()$ 
4:    $x_{nearest} \leftarrow NEAREST(Tree, x_{rand})$ 
5:    $x_{new} \leftarrow STEER(x_{nearest}, x_{rand})$ 
6:   if  $\sim COLLISION(x_{nearest}, x_{new}, map)$  then
7:      $x_{near} \leftarrow NEAR(Tree, x_{new})$ 
8:      $Tree \leftarrow ADD(x_{nearest}, x_{near}, x_{new})$ 
9:      $Tree \leftarrow REWIRE(x_{near}, x_{new})$ 
10:  end if
11: end for
12:  $trajectory \leftarrow TRAJECTORY(Tree)$ 

```

Computation of the nearest node

The $NEAREST(Tree, x_{rand})$ function finds the nearest node $x_{nearest}$ to the random state x_{rand} generated in the sampling-based exploration of the planning space. Since nodes include state information both for multirotor and dual arm accordingly with the integrated operation basis of the planner, there will be two different measurements for calculating the nearest node: the difference in position for the multirotor and the difference in angle for the arm joints. Thus, there appears the need of defining a homogenizing metric. The reference velocities u_{ref} (for the UAV) and w_{ref} (for the joints) have been defined with this purpose of transforming the heterogeneous measurements into a common metric given by the time magnitude required for each system component to move between the configurations associated with the nodes under analysis. The equations corresponding to this normalization approach are presented as follows

$$t_{UAV} = \frac{\sqrt{(\Delta q_1)^2 + (\Delta q_3)^2}}{u_{ref}}$$

$$t_{ARMS} = \frac{\max(|\Delta q_7^R|, |\Delta q_8^R|, |\Delta q_7^L|, |\Delta q_8^L|)}{w_{ref}} \quad (4)$$

$$x_{nearest} = \min_{x \in Tree} \left(\max(t_{UAV}|_x, t_{ARMS}|_x) \right)$$

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 dual arm device.

Collision checking

The $COLLISION(x_{nearest}, x_{new}, map)$ function checks if the branch that would link two nodes produces some 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 investigated to see if any part of the system 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 ARS-LRM planner that allows differentiating equivalent configurations in terms of final effector positions according to the positions of both the multirotor and the intermediate links. The consideration of the different geometries of the system components, together with joint exploration of the planning space for both system components, is crucial in this respect. Concerning the former, simplified models that alleviate the computational burden of collision checking, but maintaining at the same time their capability to express the heterogeneity existing in the geometry of the different parts, are the desirable option. To this end, the multirotor has been considered rectangularly shaped while the dual arm and the long bar extension are modeled by rectilinear bars with negligible section. Thus, the collision checking operation is performed with higher precision than considering only the bounding sphere associated with the ARS-LRM. This granularity eases the planner convergence to feasible trajectories in cluttered environments. 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 reduce the complexity of the collision checking algorithm.

Another aspect that requires further consideration is the algorithm selected for detecting the collisions. In the case of the multirotor, the approach is straightforward since it only requires checking whether the position of the center of mass is within the limits of the rectangular region that produces collisions with the obstacle (see Figure 9).

In contrast, the collision management for the extension bar and the dual arm 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 bar and the arm 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 ARS-LRM. It mainly consists of translating the collision condition to the angular space as shown in Figure 10. In this way, the obstacles are characterized in terms of the minimum and maximum bar or link angle that may produce a collision. Then, taking into account also the distance to the obstacle, it is possible to check the collision with a considerable reduction in the computational load. Figure 11 shows two examples of this angular approach. The collision management consists of checking if a point

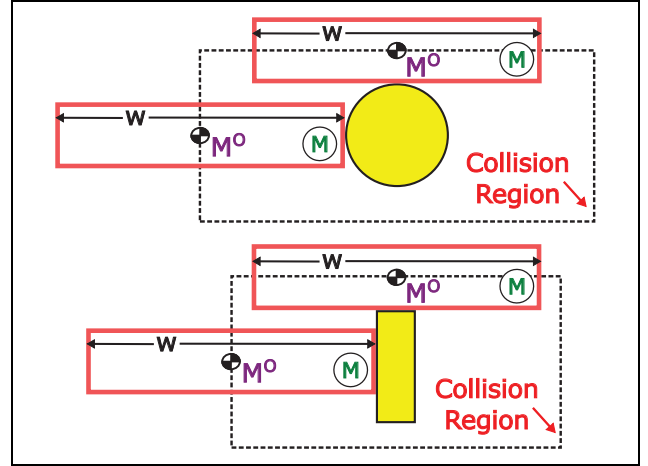


Figure 9. Collision checking for the multirotor based on the position of the center of mass M^o . Round obstacle (above) and rectangular obstacle (below).

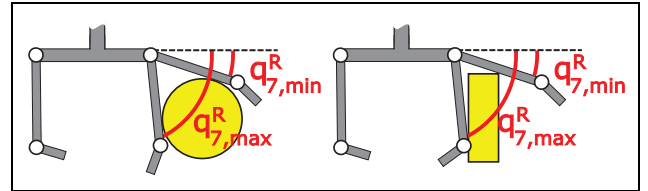


Figure 10. Collision checking for the right upper link. Round obstacle (left) and rectangular obstacle (right).

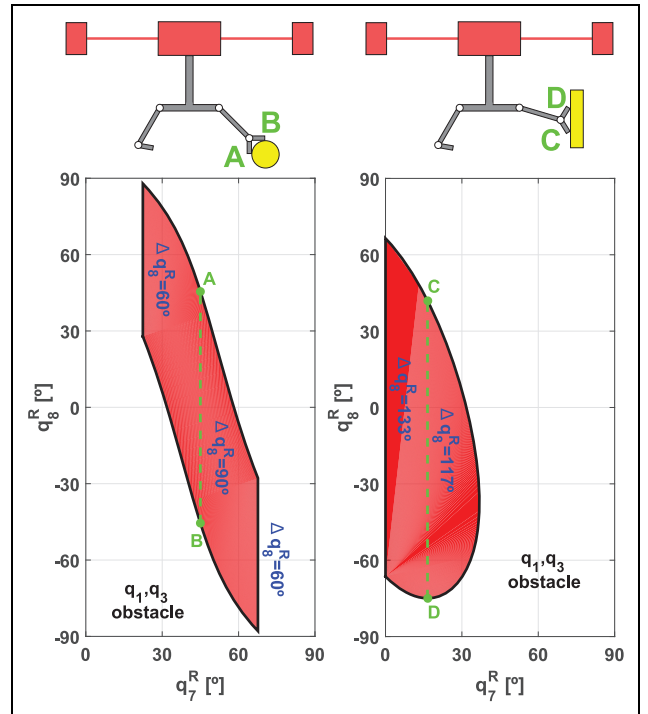


Figure 11. Regions in which the right arm collides with an obstacle for fixed UAV and obstacle positions. Round obstacle (left) and rectangular obstacle (right). The shape of the collision regions varies with the relative position between the multirotor and the obstacle. UAV: unmanned aerial vehicle.

representing certain configuration in the angular space falls inside the collision regions defined through the minimum–maximum angular characterization 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.

Precomputation of collisions

In order to further reduce the execution time devoted to collision management, a precomputation of collision conditions can be implemented. For that purpose, a static environment with known-location obstacles is assumed. Then, a representative set of states for the ARS-LRM in the application scenario is selected for implementing the collision precomputation. The results are stored in a binary matrix that will allow checking collisions in two simple steps. First, the most similar element within the aforementioned set of representative states to the configuration under collision checking is searched for. Second, 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 precomputation will be executed only once for each environment, a dense discretization 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_{ARMS} . This definition can be expressed mathematically as follows

$$\begin{aligned} \rho_{UAV} &= \sqrt{(\Delta q_1)^2 + (\Delta q_3)^2} \\ \rho_{ARMS} &= \max(|\Delta q_7^R|, |\Delta q_8^R|, |\Delta q_7^L|, |\Delta q_8^L|) \\ x_{near} &= x \in Tree / \begin{cases} \rho_{UAV}|_x \leq \gamma_{UAV} \\ \rho_{ARMS}|_x \leq \gamma_{ARMS} \end{cases} \end{aligned} \quad (5)$$

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* optimization 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 system (CF_T), and the energy measurement given by the linear and angular

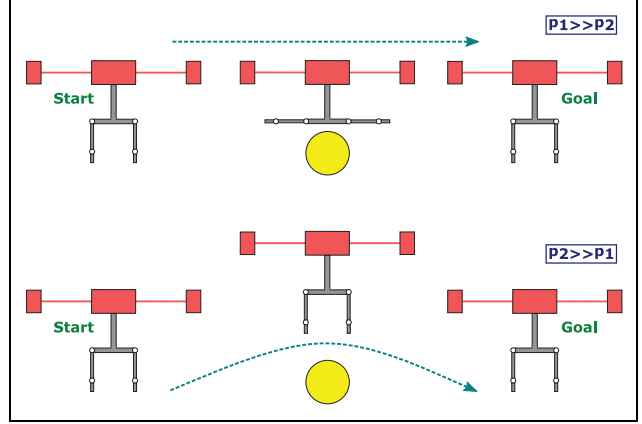


Figure 12. Effect of the weighting parameters $p_{1,2}$ in the cost function CF_E .

displacements produced in the multirotor and the arm joints, respectively (CF_E). These cost indices can be formulated as follows

$$\begin{aligned} CF_T &= \max(t_{UAV}, t_{ARMS}) \\ CF_E &= p_1 \cdot \rho_{UAV} + p_2 \cdot \sigma_{ARMS} \end{aligned} \quad (6)$$

where t_{UAV} and t_{ARMS} were defined in equations (4); ρ_{UAV} was defined in equations (5); $\sigma_{ARMS} = |\Delta q_7^R| + |\Delta q_8^R| + |\Delta q_7^L| + |\Delta q_8^L|$ 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 allows the prioritization of movements with minimum displacements in the multirotor or in the dual arm. Thus, $p_1 > p_2$ prioritizes trajectories in which the UAV displacement is minimum while $p_2 > p_1$ prioritizes the contrary. Figure 12 illustrates the effect of these weighting parameters.

Dynamics awareness

The developments presented in previous subsections guarantee planned trajectories that are collision free as well as time- and energy-efficient. However, the strong dynamical coupling that exists in the ARS-LRM between the multirotor and the dual arm requires further attention since it provokes considerable differences between planned and executed trajectories. The collision risk introduced by these differences is specially critical for cluttered environments like industrial sites.

In order to solve the aforementioned problem, it is necessary to introduce DA in the planner. The resulting algorithm, RRT*-DA (RRT* with DA), guarantees robust obstacle avoidance by modifying the $COLLISION(x_{nearest}, x_{new}, map)$ function described in the ‘‘Collision checking’’ subsection. Instead of checking collisions in a set of geometrically intermediate configurations that are generated using lineal interpolation between the nodes under analysis (RRT* green dashed

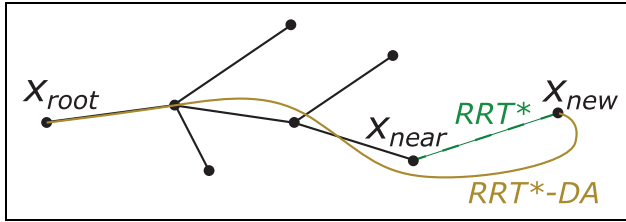


Figure 13. Operation basis of the $\text{COLLISION}(x_{\text{nearest}}, x_{\text{new}}, \text{map})$ function: green dashed line for the standard RRT*-based algorithm and yellow solid line for the advanced RRT*-DA algorithm. RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness.

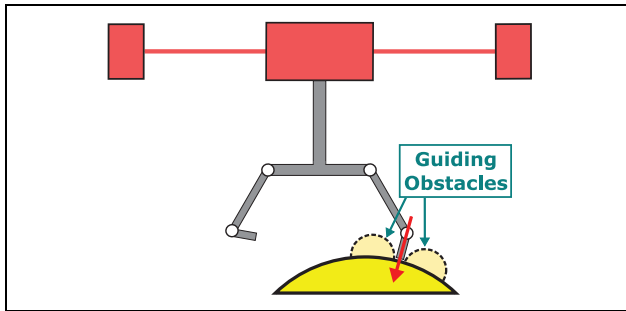


Figure 14. Guiding obstacles enforcing a rectilinear movement perpendicular to the contact surface.

line in Figure 13), the checking will be now carried out through dynamically intermediate configurations that belong to the trajectory obtained after simulating the close-loop dynamics of the controlled ARS-LRM (RRT*-DA yellow solid line in Figure 13). Furthermore, the application of previous paradigm for tree extension without collisions is based on a root-to-candidate validation as can be seen in Figure 13. Thus, the dynamical feasibility of the new possible branch will depend on the absence of collisions in the complete dynamical trajectory from the root node. The consequence of the application of this approach is the derivation of a search tree that is based on the behavior of the controlled system and therefore the resultant planned trajectories are both compatible with dynamics constraints and free of obstacles.

Guiding obstacles to enforce manipulation patterns

In certain 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. Figure 14 depicts an example of the usage of these guiding obstacles. In that scenario, the two guiding obstacles allow the right end effector to approach the yellow surface following a rectilinear movement that is perpendicular to the contact surface. In this way, other undesired trajectories

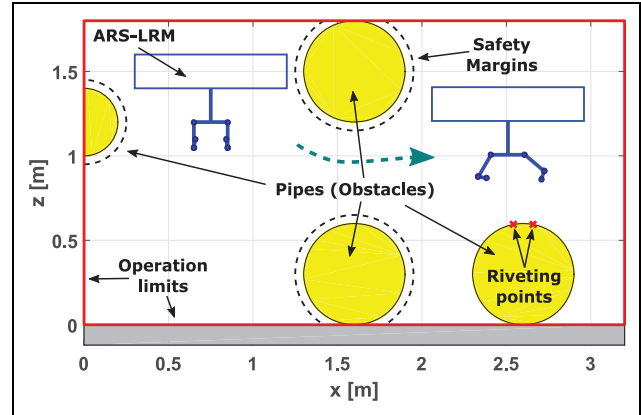


Figure 15. Application scenario given by a riveting task.

that could make the end effector slides onto the surface are conveniently discarded.

Application scenarios

In order to demonstrate the validity of the motion planning strategy presented in previous section, the algorithm will be tested in two realistic industrial scenarios, a riveting application and a chimney repairing task. Both scenarios require advanced motion planning of the ARS-LRM in order to perform robustly the desired operations in the associated cluttered environments.

Riveting application

The schematic description of this scenario is shown in Figure 15, where colored 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 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 camera integrated as end effector (see Figure 1). In this first proof of concept, the riveting operations will assume ideal conditions, that is, absence of interaction forces.

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

1. Navigation phase: This phase corresponds to the system 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 maneuver that will be accomplished during the manipulation phase.
2. Manipulation phase: This phase covers the different maneuvers involved in the manipulation task under

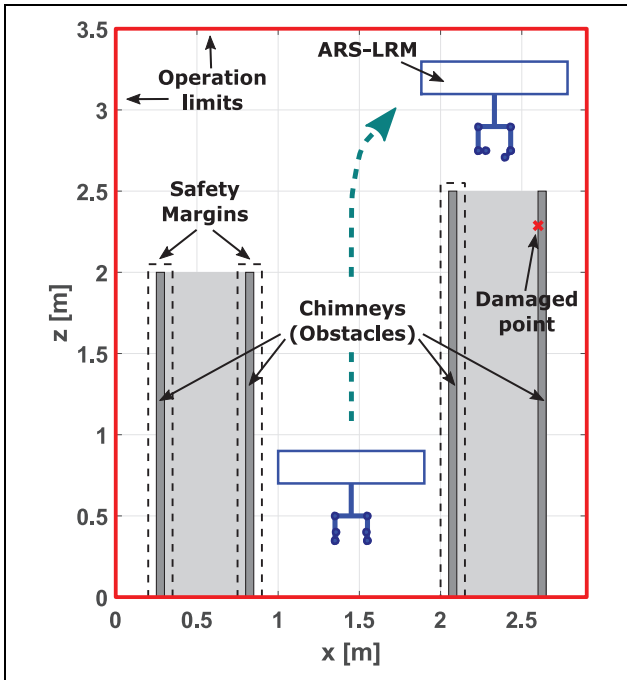


Figure 16. Application scenario given by a chimney repairing task.

consideration, the riveting operation. In all the sub-phases described below, it is assumed that the left arm will adapt its configuration to optimize the visual feedback provided by its integrated camera.

- (a) Rivet placement: Approaching to the target point in the perpendicular direction to the target point plane by the riveting effector integrated in the right arm.
- (b) Release: Opposite maneuver to the rivet placement in which the riveting effector leaves the target point, again following the perpendicular direction to the target point plane.
- (c) Switching: Maneuver of the complete ARS-LRM to switch between the ready-to-go configurations for riveting points 1 and 2.

Chimney repairing task

This scenario is composed of two chimneys represented by four dark gray rectangles (see Figure 16). The two light gray rectangles correspond to the free space inside the chimneys and the dotted surrounding areas denote safety regions whose violations will be treated as collisions. As depicted in the figure, the ARS-LRM will be commanded to repair a crack inside the 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 Figure 1). It is worth noting that, thanks to the bar extension of the ARS-LRM, 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 presented both inside the chimney as well as in its surrounding space. Again in this first proof of concept for this scenario, the repairing operation will assume ideal conditions in the chimney surface, that is, absence of interaction forces.

Similar 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 maneuvers:

1. Navigation phase: This phase corresponds to the system 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 maneuver that will be accomplished during the manipulation phase.
2. Manipulation phase: This phase covers the different maneuvers involved in the manipulation task under consideration, the chimney repairing operation. In all the subphases described below, it is assumed that the left arm will adapt its configuration to optimize 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.
 - (b) Release: Opposite maneuver to the repair sub-phase in which the tool effector leaves the target point, again following the perpendicular direction to the chimney surface.

Simulation results

This section analyzes the results corresponding to the application of the RRT*-DA algorithm in the scenarios previously described. The simulations carried out for both scenarios have been organized around two main lines. On the one hand, the justification for the need to employ the RRT*-DA algorithm for robust obstacle avoidance in cluttered environments. To this end, the performance limitations of the basic RRT* algorithm without DA have been brought to light. On the other hand, a detailed analysis of the RRT*-DA performance in terms of the similarity between planned and executed trajectories has been also presented.

The index selected for optimization has been the cost function CF_E defined in the “Cost functions” subsection. Concerning the discretizing approach presented in the “Discretization of the planning space” subsection, Table 2 shows the discretization grids adopted for both

Table 2. Discretization of the planning space.

| Riveting task | | |
|------------------------|-----------------------------|-----------------------------|
| Phase | Navigation | Manipulation |
| Variable | Discretization pattern | |
| q_1 | [45, 50, ..., 275] cm | [245, 250, ..., 275] cm |
| q_3 | [30, 35, ..., 170] cm | [80, 85, ..., 120] cm |
| q_7^R | [0, 45, 90] $^\circ$ | [30, 40, ..., 150] $^\circ$ |
| q_8^R | Fixed value of 170 $^\circ$ | [-45, 0, 45] $^\circ$ |
| q_7^L | [0, 45, 90] $^\circ$ | Fixed value of 60 $^\circ$ |
| q_8^L | Fixed value of 170 $^\circ$ | Fixed value of 110 $^\circ$ |
| Chimney repairing task | | |
| Phase | Navigation | Manipulation |
| Variable | Discretization Pattern | |
| q_1 | [45, 50, ..., 245] cm | [205, 210, ..., 245] cm |
| q_3 | [30, 35, ..., 340] cm | [230, 235, ..., 340] cm |
| q_7^R | [0, 45, 90] $^\circ$ | [30, 40, ..., 110] $^\circ$ |
| q_8^R | Fixed value of 170 $^\circ$ | [-70, -60, 20] $^\circ$ |
| q_7^L | [0, 45, 90] $^\circ$ | Fixed value of 90 $^\circ$ |
| q_8^L | Fixed value of 170 $^\circ$ | Fixed value of 90 $^\circ$ |

navigation and manipulation phases in the different scenarios. As can be seen, different patterns have been used in order to optimize the planner performance in each phase. Finally, it is worth mentioning that two guiding obstacles have been placed on either side of the target points. This addition will ensure a perpendicular approach of the right end effector to the target points avoiding slides onto the manipulation surfaces.

Considering all the above information, the basic RRT* algorithm as well as the more advanced RRT*-DA motion planner of the ARS-LRM have been executed for both scenarios. The resultant plans (represented in the figures with light blue lines for the RRT* algorithm and with light green lines for the RRT*-DA algorithm) have been also provided to the controlled ARS-LRM in order to analyze the close-loop behavior of the system (represented in the figures with dark blue lines for the RRT* algorithm and with dark green lines for the RRT*-DA algorithm) when following the planned trajectories.

It should be clarified that the planned trajectories are calculated independently for each phase described in the ‘‘Application scenarios’’ section. However, the resulting trajectories are then integrated into a single one that will be given to the controller as commanded reference. Consequently, there are no different phases from the point of view of the system/control execution. The simulation work has been carried out in a MATLAB–Simulink framework that provides the graphical evolution of the system variables as well as the corresponding virtual reality animations. Both graphical outputs will be used throughout this section to illustrate the obtained results.

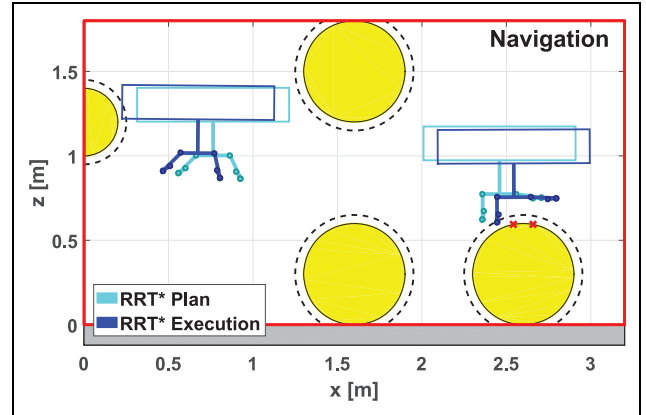


Figure 17. Riveting scenario. Navigation phase planned with the RRT* algorithm. The planned trajectory is collision-free but not the trajectory executed by the controlled ARS-LRM. RRT* algorithm: optimal Rapidly-exploring Random Tree algorithm; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

Results using the RRT* algorithm

The basic RRT* algorithm has certain performance limitations since it does not consider the dynamics of the system. For instance, dynamic effects like oscillations are not accounted for during the collision-checking phase, 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 RRT* algorithm including both the planned and the close-loop trajectories have been considered for the two scenarios under study (see the corresponding animations²⁵). According to these simulations, the basic RRT* algorithm plans efficient trajectories for navigation and manipulation phases in both scenarios. However, the corresponding close-loop trajectories described by the controlled ARS-LRM do not satisfy the desired collision-free and slide-free properties. For the riveting scenario, Figure 17 shows how the ARS-LRM violates safety margins during the navigation phase, whereas Figure 18 reveals a slide onto the pipe surface during the manipulation phase. Similarly, for the chimney scenario, Figure 19 illustrates a collision of the right end effector with chimney surface.

Results using the RRT*-DA algorithm

The previous subsection motivated the need to consider the dynamics of the system within the motion planning problem when operating in cluttered environments. To solve the above, the RRT*-DA algorithm has been proposed in this article. This section is devoted to analyze the performance of such algorithm in the same simulation scenarios that were used in previous subsection with the basic RRT* algorithm (see the corresponding animations^{26,27}). To this end, again both the planned and the close-loop trajectories will be studied.

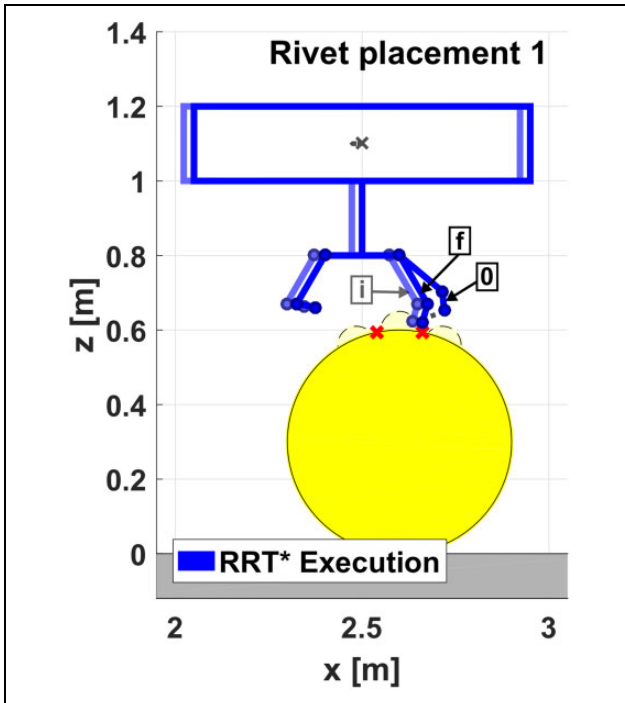


Figure 18. Riveting scenario. Manipulation phase (rivet placement 1) planned with the RRT* algorithm. The dotted lines represent the simulated movements of both the multirotor center of mass and the right end effector. The trajectory executed by the controlled ARS-LRM (from initial configuration 0 to final configuration f through shaded intermediate configuration i) violates a guiding obstacle and consequently the right end effector slides onto the pipe surface. RRT* algorithm: optimal Rapidly-exploring Random Tree algorithm; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

The results corresponding to the navigation phase of both scenarios are presented together, Figures 20 and 21 for the riveting scenario, whereas Figures 22 and 23 corresponds to the chimney scenario. The trajectory followed by the ARS-LRM is illustrated by the dotted line representing the movement of the multirotor center of mass M^O from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,\dots}$. As can be observed, in both scenarios, the planned trajectory commands safely and efficiently the controlled ARS-LRM through the navigation phase.

Regarding the manipulation phase, Figures 24 and 25 present the achieved results for the riveting scenario. As in the navigation phase, Figure 24 shows a schematic representation of the maneuvers associated with the manipulation phase where the dotted lines represent the simulated movements of both the multirotor center of mass M^O and the right end effector from initial configurations 0 to final configurations f . Similarly, in Figure 25, the evolution of the planning-space variables, for both the planned trajectory (light green line) and the close-loop executed trajectory (dark green line), has been represented for this

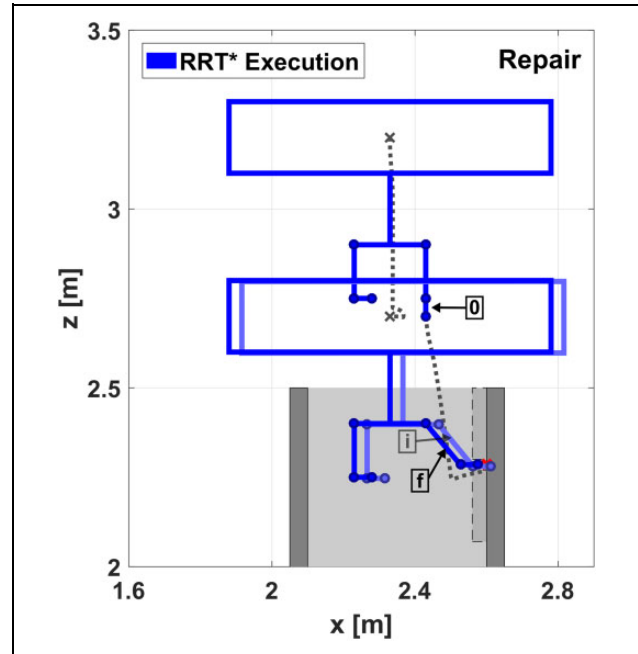


Figure 19. Chimney scenario. Manipulation phase (repair) planned with the RRT* algorithm. The dotted lines represent the simulated movements of both the multirotor center of mass and the right end effector. The trajectory executed by the controlled ARS-LRM (from initial configuration 0 to final configuration f through shaded intermediate configuration i) violates a guiding obstacle and consequently the right end effector collides with the chimney surface. RRT* algorithm: optimal Rapidly-exploring Random Tree algorithm; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

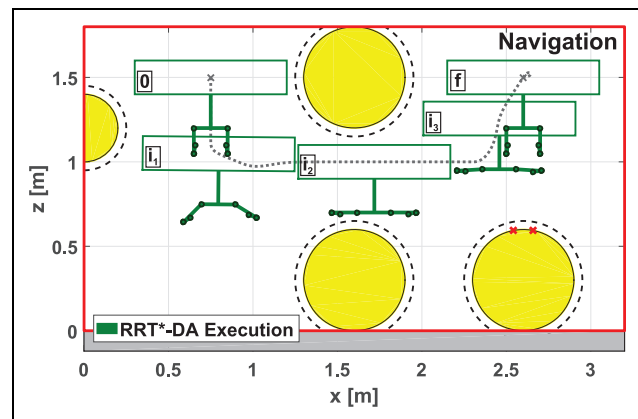


Figure 20. Riveting scenario. Navigation phase planned with the RRT*-DA algorithm and executed by the controlled ARS-LRM. The dotted line represents the simulated movement of the multirotor center of mass M^O from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2,3}$. The ARS-LRM navigates through the obstacles following an efficient and collision-free trajectory. RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

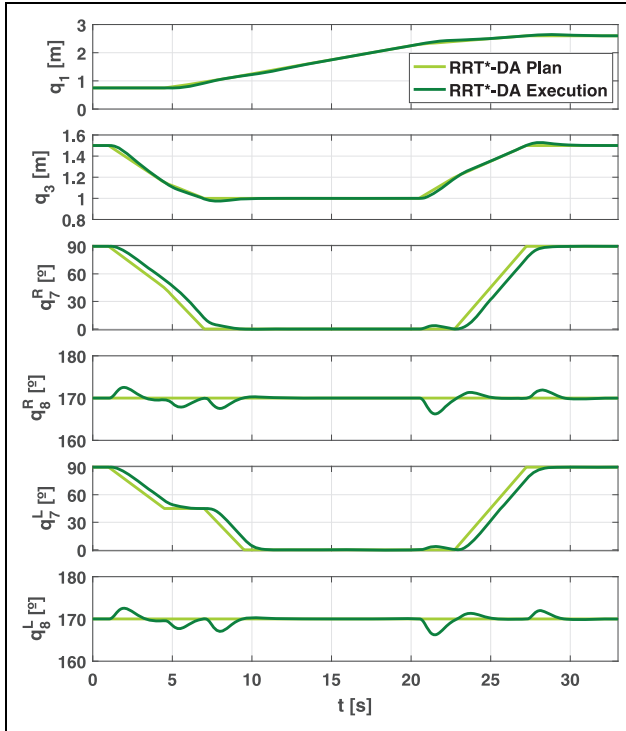


Figure 21. Riveting scenario. Navigation phase planned with the RRT*-DA algorithm (light green) and executed by the controlled ARS-LRM (dark green). RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

manipulation phase. Once again the planned trajectory succeeds in commanding efficiently the controlled ARS-LRM through the different manipulation maneuvers involved in the riveting task. In contrast to the results using the basic RRT* 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, Figure 24 (switching) illustrates how the joint consideration of the planning space for the multirotor and the dual arm allows the optimization of the switching maneuver 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 maneuver of the riveting effector in a more efficient way. Turning now to the chimney repairing scenario, Figures 26 and 27 show the results corresponding to the manipulation scenario. Once more, the planned trajectory commands the ARS-LRM through an efficient and collision-free trajectory.

Conclusions

This article extends previous works of the authors in the field of motion planning strategies for aerial robotic systems. The most relevant contribution is the inclusion of DA

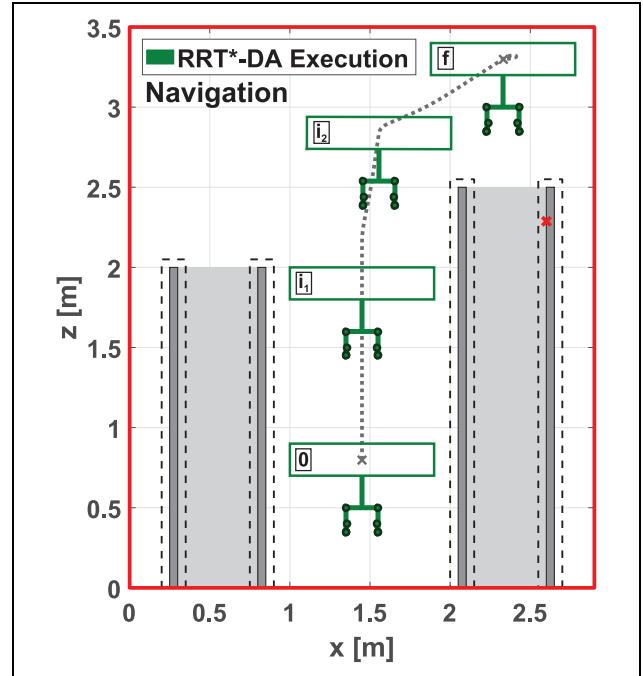


Figure 22. Chimney scenario. Navigation phase planned with the RRT*-DA algorithm and executed by the controlled ARS-LRM. The dotted line represents the simulated movement of the multirotor center of mass M^0 from initial configuration 0 to final configuration f through intermediate configurations $i_{1,2}$. The ARS-LRM navigates through the obstacles following an efficient and collision-free trajectory. RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

in the motion planner derived for an ARS-LRM since this feature guarantees robust obstacle avoidance in cluttered environments.

In order to evaluate the algorithms under consideration, a simulation environment that characterizes the relevant system behaviors for planner operation was required. Consistently with this requirement, the ARS-LRM platform has been described in detail together with its potential benefits: a considerable increment in the safety distance between rotors and manipulated objects, the capability to execute tasks in hard-to-reach places, and the extended manipulation capabilities offered by the dual arm. Taking this description as reference, the dynamics of the system has been modeled with specific methodologies for multi-body systems. Furthermore, a distributed control scheme that makes use of nonlinear control strategies based on model inversion has been derived to complete the test bed.

With respect to the presented RRT*-DA planning approach, several features justify the relevance of this contribution. The aerial platform and the dual arm device have been considered jointly within the planner operation. In this way, it is possible to achieve wider and safer operating conditions since equivalent configurations in terms of final

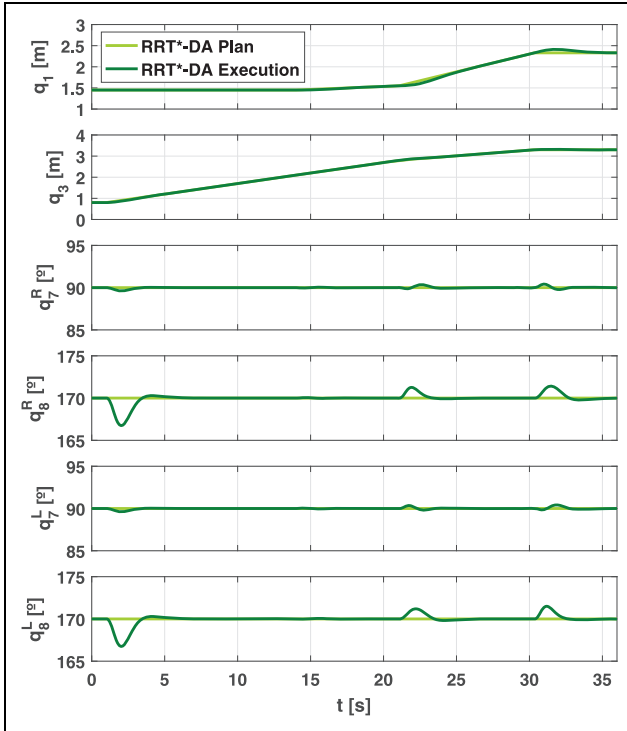


Figure 23. Chimney scenario. Navigation phase planned with the RRT*-DA algorithm (light green) and executed by the controlled ARS-LRM (dark green). RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

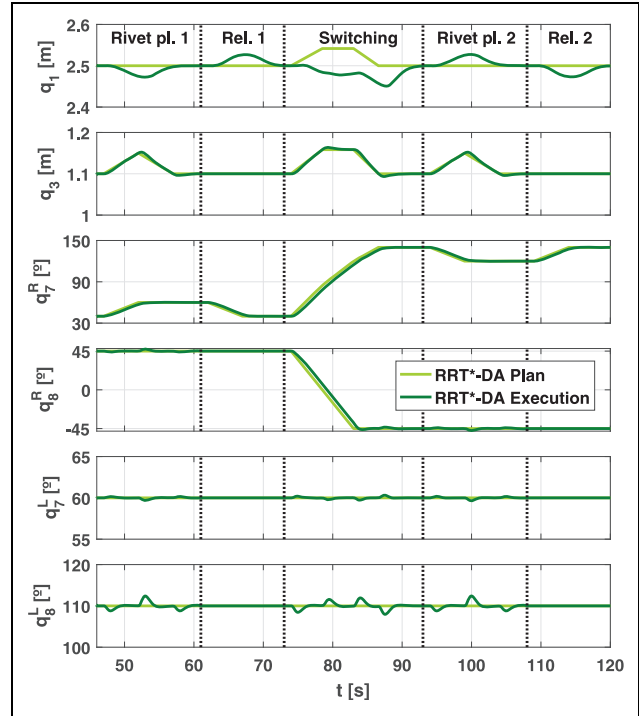


Figure 25. Riveting scenario. Manipulation phase planned with the RRT*-DA algorithm (light green) and executed by the controlled ARS-LRM (dark green). RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

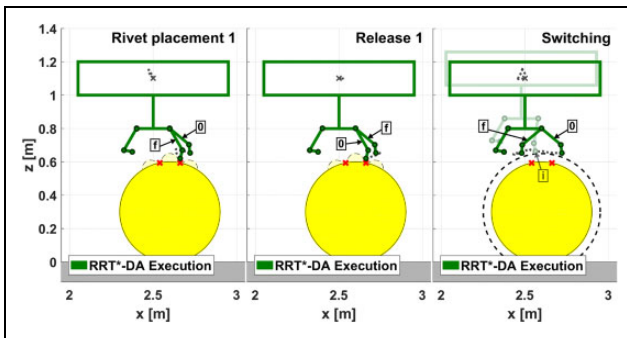


Figure 24. Riveting scenario. Manipulation phase planned with the RRT*-DA algorithm and executed by the controlled ARS-LRM. The dotted lines represent the simulated movements of both the multirotor center of mass M^0 and the right end effector from initial configurations 0 to final configurations f . The ARS-LRM places the first rivet and then switches efficiently (see the shaded intermediate configuration i) between the ready-to-go configurations. RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

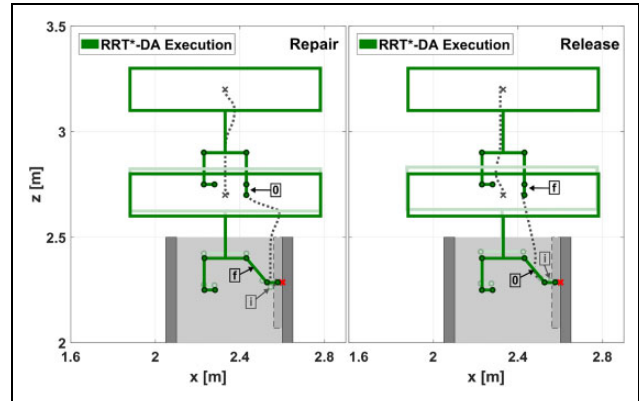


Figure 26. Chimney scenario. Manipulation phase planned with the RRT*-DA algorithm and executed by the controlled ARS-LRM. The dotted lines represent the simulated movements of both the multirotor center of mass M^0 and the right end effector from initial configurations 0 to final configurations f through intermediate configuration i . The ARS-LRM approaches and moves away the right end effector to the target point in the chimney surface following perpendicular movements. RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

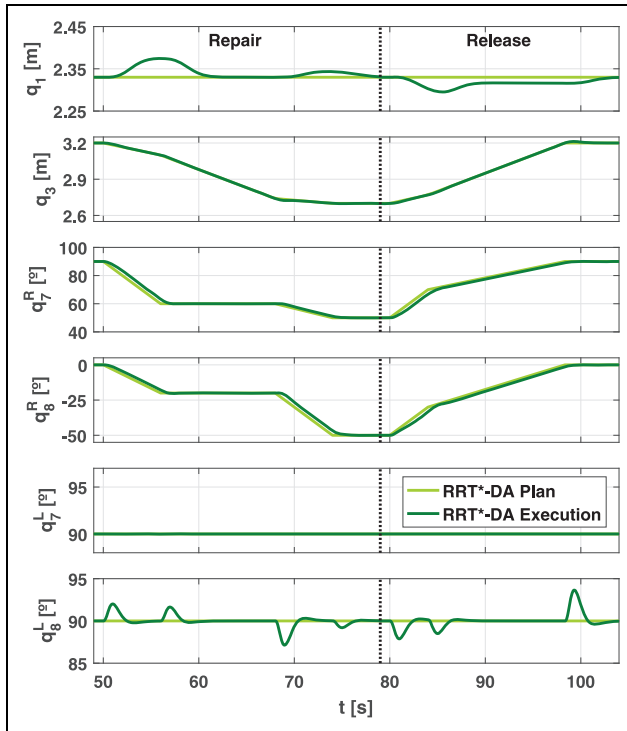


Figure 27. Chimney scenario. Manipulation phase planned with the RRT*-DA algorithm (light green) and executed by the controlled ARS-LRM (dark green). RRT*-DA algorithm: optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness; ARS-LRM: aerial robotic system with two arms for long reach manipulation.

effector positions can be differentiated according to the positions of both the multirotor and the intermediate links. On the other hand, the planner operation is driven by an RRT*-based algorithm that optimizes energy and execution time in cluttered environments for both navigation and manipulation tasks. Finally, the aforementioned feature of DA guarantees robust obstacle avoidance.

With the purpose of demonstrating the validity of the motion planning strategy presented, the RRT*-DA algorithm has been tested in two realistic industrial scenarios, a riveting application and a chimney repairing task. As was discussed in the simulation section, the planned trajectories succeed in commanding efficiently the controlled ARS-LRM through navigation and manipulation phases without producing collisions with the obstacles existing in the scenarios. The latter has been proved not to be possible using the basic RRT* algorithm without DA.

Regarding future extensions of the work, the presented results will be completed with a three-dimensional characterization of the ARS-LRM. Then, experimental validation will be carried out by extending the platform prototype presented by Suarez et al.⁷ Finally, due to the relevance of the aerodynamic effects when an aerial platform operates close to other surfaces in cluttered environments, the motion planning strategies will be extended to account for such aerodynamic effects.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This work has been supported by the European Project AEROARMS, funded by the Horizon 2020 research and innovation program of the European Commission under grant agreement no. 644271 and by the AEROMAIN project of the Spanish RETOS Program DPI2014-59383-C2-1-R.

References

1. Mellinger D, Lindsey Q, Shomin M, et al. Design, modeling, estimation and control for aerial grasping and manipulation. In: *IEEE/RSJ international conference on, intelligent robots and systems (IROS)*, San Francisco, CA, USA, September 2011, pp. 2668–2673. IEEE.
2. Kondak K, Huber F, Schwarzbach M, et al. Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In: *IEEE international conference on robotics and automation (ICRA)*, Hong Kong, China, May 2014, pp. 2107–2112. IEEE.
3. Naldi R, Pounds P, De Marco S, et al. Output tracking for quadrotor-based aerial manipulators. In: *American control conference (ACC)*, Chicago, IL, USA, July 2015, pp. 1855–1860. IEEE.
4. Manubens M, Devaurs D, Ros L, et al. Motion planning for 6-D manipulation with aerial towed-cable systems. In: *Proceedings of the Robotics: Science and Systems Conference (RSS)*, Berlin, Germany, June 2013, pp. 1–8.
5. Orsag M, Korpela C, Bogdan S, et al. Valve turning using a dual-arm aerial manipulator. In: *International conference on unmanned aircraft systems (ICUAS)*, Orlando, FL, USA, May 2014, pp. 836–841. IEEE.
6. Korpela C, Orsag M, and Oh P. Towards valve turning using a dual-arm aerial manipulator. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Chicago, IL, USA, September 2014, pp. 3411–3416. IEEE.
7. Suarez A, Jimenez-Cano AE, Vega V, et al. Lightweight and human-size dual arm aerial manipulator. In: *international conference on unmanned aircraft systems (ICUAS)*, Miami, FL, USA, June 2017, pp. 1778–1784. IEEE.
8. Yüksel B, Buondonno G, and Franchi A. Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-/elastic-joints. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Daejeon, South Korea, October 2016, pp. 561–566. IEEE.
9. Yu Y, Ding X, and Zhu JJ. Dynamic modeling and control for aerial arm-operating of a multi-propeller multifunction aerial robot. *Advanced Robotics* 2017; 31(13): 665–679.
10. Ragel R, Maza I, Caballero F, et al. Comparison of motion planning techniques for a multi-rotor UAS equipped with a

- multi-joint manipulator arm. In: *Workshop on research, education and development of unmanned aerial systems (RED-UAS)*, Cancun, Mexico, November 2015, pp. 133–141. IEEE.
11. Elbanhawi M and Simic M. Sampling-based robot motion planning: a review. *IEEE Access* 2014; 2: 56–77.
 12. LaValle SM and Kuffner Jr JJ. Randomized kinodynamic planning. *The International Journal of Robotics Research* 2001; 20(5): 378–400.
 13. Koyuncu E and Inalhan G. A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Nice, France, September 2008, pp. 815–821. IEEE.
 14. Richter C, Bry A, and Roy N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: Inaba M and Corke P (eds) *Robotics Research. Springer Tracts in Advanced Robotics*, Vol. 114. Cham, Switzerland: Springer.
 15. Boeuf A, Cortés J, Alami R, et al. Planning agile motions for quadrotors in constrained environments. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Chicago, IL, USA, September 2014, pp. 218–223. IEEE.
 16. Boeuf A, Cortés J, Alami R, et al. Enhancing sampling-based kinodynamic motion planning for quadrotors. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, Germany, September 2015, pp. 2447–2452. IEEE.
 17. Caballero A, Bejar M, Rodriguez-Castaño A, et al. Motion planning for long reach manipulation in aerial robotic systems with two arms. In: *European conference on mobile robots (ECMR)*, Paris, France, September 2017, pp. 1–7.
 18. Kondak K, Bernard M, Meyer N, et al. Autonomously flying VTOL-robots: modeling and control. In: *2007 IEEE international conference on robotics and automation (ICRA)*, Roma, Italy, April 2007, pp. 736–741. IEEE.
 19. Kane TR and Levinson DA. *Dynamics, theory and applications*. New York, NY, USA: McGraw-Hill, 1985.
 20. Sandino LA, Bejar M, and Ollero A. A survey on methods for elaborated modeling of the mechanics of a small-size helicopter. Analysis and comparison. *Journal of Intelligent & Robotic Systems* 2013; 72(2): 219–238.
 21. Motiongenesis Kane 5.x, <http://www.motiongenesis.com/>.
 22. Tang S, Khaksar W, Ismail N, et al. A review on robot motion planning approaches. *Pertanika J Sci Technol* 2012; 20(1): 15–29.
 23. LaValle SM. *Rapidly-exploring random trees: a new tool for path planning*. TR 98-11, Computer Science Dept., Iowa State University, October 1998.
 24. Karaman S and Frazzoli E. Sampling-based algorithms for optimal motion planning. *Int J Robot Res* 2011; 30(7): 846–894.
 25. Simulation results using the basic RRT* algorithm. 2D Animations, <https://youtu.be/Rh7yNgloOoY> and <https://youtu.be/NvbPZvUf12A>.
 26. Simulation results using the RRT*-DA algorithm. 2D Animations, <https://youtu.be/WwQ4E0fxKg8> and <https://youtu.be/qH8Q3q3ePsI>.
 27. Simulation results using the RRT*-DA algorithm. 3D Animations, <https://youtu.be/R73q35jXhn0> and <https://youtu.be/D5kTg6NScak>.