# Towards Autonomous Bridge Inspection: Sensor Mounting Using Aerial Manipulators

Ivanovic, Antun; Markovic, Lovro; Car, Marko; Duvnjak, Ivan; Orsag, Matko

*Article*

# Towards Autonomous Bridge Inspection: Sensor Mounting Using Aerial Manipulators

Antun Ivanovic [1,*], Lovro Markovic [1], Marko Car [1], Ivan Duvnjak [2] and Matko Orsag [1]

[1] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia; lovro.markovic@fer.hr (L.M.); marko.car@fer.hr (M.C.); matko.orsag@fer.hr (M.O.)

[2] Faculty of Civil Engineering, University of Zagreb, Fra Andrije Kacica-Miosica 26, 10000 Zagreb, Croatia; ivan.duvnjak@grad.unizg.hr

[*] Correspondence: antun.ivanovic@fer.hr

**Featured Application: The main idea of this paper was to deploy a team of unmanned aerial vehicles (UAVs) to attach a sensor to a bridge using a two-component adhesive in order to perform an inspection. Constant pressure must be applied for several minutes to form a bond between two adhesives. Therefore, one UAV sprays the colored component of an adhesive while the aerial manipulator transports the sensor, detects the contact point and attaches the sensor to it. A trajectory planning algorithm was developed around the dynamic model of the UAV and the manipulator attached to it, ensuring that the end-effector is parallel to the wall normal. Finally, the aerial manipulator achieves and maintains contact with a predefined force through an adaptive impedance control approach.**

**Abstract:** Periodic bridge inspections are required every several years to determine the state of a bridge. Most commonly, the inspection is performed using specialized trucks allowing human inspectors to review the conditions underneath the bridge, which requires a road closure. The aim of this paper was to use aerial manipulators to mount sensors on the bridge to collect the necessary data, thus eliminating the need for the road closure. To do so, a two-step approach is proposed: an unmanned aerial vehicle (UAV) equipped with a pressurized canister sprays the first glue component onto the target area; afterward, the aerial manipulator detects the precise location of the sprayed area, and mounts the required sensor coated with the second glue component. The visual detection is based on an Red Green Blue - Depth (RGB-D) sensor and provides the target position and orientation. A trajectory is then planned based on the detected contact point, and it is executed through the adaptive impedance control capable of achieving and maintaining a desired force reference. Such an approach allows for the two glue components to form a solid bond. The described pipeline is validated in a simulation environment while the visual detection is tested in an experimental environment.

**Keywords:** aerial robotics; inspection and maintenance; aerial manipulation; multirotor control

## 1. Introduction

The world of unmanned aerial vehicles (UAVs) has been rapidly growing in recent years. As their design and control are perfected, these aerial vehicles have become more and more available. Nowadays, off-the-shelf ready-to-fly UAVs can be found and bought in shops, which makes them available to virtually anybody. This, in turn, has sparked a great deal of public interest in UAVs since their potential can be found in applications such as agriculture, various inspections (bridges, buildings, wind turbines), geodetic terrain mapping, the film industry, and even for hobby enthusiasts to fly and record videos from a first-person perspective. The vast majority of commercially available UAVs are equipped with a camera, while more specialized vehicles for terrain mapping or crop spraying offer a more diverse sensor suite.

All of the aforementioned systems primarily observe and gather data about the environment, while having little to no ability to interact with and change the environment. One way to augment these vehicles for physical interaction is to attach a lightweight manipulator to their body, which is the main interest of the aerial manipulation field. Although such vehicles are more complex for both modeling and control, their benefit lies in performing versatile tasks that require interaction with the environment.

In general, there are three types of bridge inspections: periodic, special and damage inspections. Periodic bridge inspections differ from country to country according to national standards, and are usually performed at least once every two to three years. Special inspections are typically used to monitor the condition of deficient elements at specific locations based on predefined requirements. Damage inspections are usually performed after events that have occurred due to environmental impacts or human actions. The aim of a bridge inspection is to evaluate and assess structural safety and reliability. Current techniques are based on traditional visual inspection with a combination of non-destructive methods (NDTs). Traditional visual inspection is performed by experienced (trained) engineers and using specialized trucks equipped with the cranes and basket, that allow inspectors to review the conditions underneath the bridge. During the inspection, the engineers are equipped with various NDT [1] tools to detect construction faults and defects such as corrosion, cracks, voids, weakening connections, and concrete delamination. Some of these NDTs require mounting small sensors to collect data, such as accelerometers, strain gauges, tilt meters and various transducers for acoustic or pressure measurements. Afterwards, the bridge is excited with vibrations, sound waves, tapping, etc., and mounted sensors record responses to these specific excitations. Furthermore, there are usually requirements for performing measurements during the bridge inspection, such as the short- and long-term monitoring of vibrations, strains, displacements, etc. Mainly, these inspections offer valuable information about the current bridge conditions, but there are a number of disadvantages. The use of trucks during inspections requires total or temporary road closures, which at the same time require safety measures to keep traffic flowing as freely as possible. In addition, inspectors often encounter challenges in reaching all portions or elements in narrow areas, such as tight spaces between girders, beams and vaults. The aforementioned significantly increases the time and overall cost of the inspection. An aerial robot, with the potential to reach these challenging locations on the bridge, could significantly reduce the time and cost of these inspections and improve worker safety. Moreover, we note that the aforementioned sensors are relatively lightweight, which makes them suitable for transportation and mounting with an aerial robot.

*1.1. Concept*

We envision a team of robots working together to attach sensors to bridges and similar grade separation infrastructure. In theory, such a task could be accomplished with a single aerial robot, at the cost of a complex mechanical design. The proposed team shown in Figure 1 consists of two drones. One drone applies the adhesive material, and the other attaches sensors. We envision a two-stage process using two-component adhesives which form a solid bond from two separate reactive components: the "resin" and the "hardener". The first UAV applies the resin by spraying it onto the surface, while the second one attaches the sensor with the hardener already applied before the flight.

It is important to follow the prescribed ratio of the resin and the hardener to achieve the desired physical properties of the adhesive. Only when mixed together do the two components form the adhesive. The reaction typically begins immediately after the two components are mixed and the bond strength depends both on maintaining the contact and the viscosity of the mixed adhesive during the process. Manufacturers can control the cure rate to achieve various working times (worklife) until final bond strength is achieved, ranging from minutes to weeks. Resin bases are usually more viscous than their respective hardener and are generally applied by brush, roller, applicator or spray. In this work, we propose attaching a canister of pressurized resin to the UAV, and spray it through a

nozzle onto the infrastructure surface. In this scenario, the spray needs to be softer and less turbulent to reduce the amount of material lost due to bouncing and it must be colored for the detection in the second stage. Spraying with drones is not a novel concept [2,3], so without loss of generality, we will omit the details of this design and instead focus on detecting, navigating to, and sustaining contact with the sprayed surface.

In typical applications, the assemblies are usually kept in contact until the sufficient strength of the bond is achieved. When fully cured, two-component adhesives are typically tough and rigid with good temperature and chemical resistance. We rely on the robotic arm attached to the second aerial vehicle to apply a controlled contact force between the sensor and the surface. Maintaining this fixed assembly contact through the impedance control system enables us to achieve a successful curing process and create a permanent bond between the sensor and the infrastructure. After the first UAV sprays the resin onto the surface, the second aerial robot finds the sprayed part and applies the contact with the sensor's surface. Before takeoff, the surface of the sensor is brushed with a hardener. Once contact is made, it is maintained for the prescribed curing time, after which the aerial robot disembarks and leaves the sensor attached to the surface.



**Figure 1.** Two aerial robots working together to attach sensors to different parts of a bridge and similar grade separation infrastructure. The one on the left is used to spray the resin onto the surface, while the aerial robot on the right maintains contact to the surface with the sensor attached to its end-effector.

*1.2. Contributions*

This paper focuses on developing a method for mounting sensors on a bridge wall using an aerial manipulator. The first contribution is augmenting the model-based motion planning with the adaptive impedance controller. The motion planning method accounts for the underactuated nature of the multirotor UAV and corrects the end-effector configuration for an appropriate approach. This method also relies on the dexterity analysis which keeps the manipulator configuration within its optimal region, ensuring that the manipulator is never fully extended or contracted while mounting a sensor. The second contribution is the visual blob detection which locates and tracks the appropriate sensor mounting point. The blob detection has been experimentally verified in an indoor environment, yielding the reliable and robust tracking of the mount location, as well as the blob plane orientation. Finally, the third contribution is the simulation analysis of the system's performance, conducted on a straight and inclined wall approach. The simulation concentrates on testing the motion planning together with the impedance controller, performing a repeatability analysis and ensuring that the desired contact force is achieved.

## 2. Related Work

In the world of aerial inspections, a number of UAV-based solutions are being proposed by researchers. In [4], a technical survey for bridge inspections is given. Researchers in [5] present the project AERIAL COgnitive Integrated Multi-task Robotic System with Extended Operation Range and Safety (AERIAL-CORE) which focuses on power lines inspection, maintenance and installing bird diverters and line spacers. Most of these approaches are based in conjunction with new technologies to ensure faster and cheaper inspections. Nowadays, UAVs use high-resolution cameras for visual inspections and employ point cloud methods based on digital photogrammetry [6], Light Detection And Ranging (LiDAR)-based methods [7], digital image correlation [8], etc. There are also reports for visual compensation during aerial grasping [9], aerial grasping in strong winds [10], and the development of a fully actuated aerial manipulator for performing inspections underneath a bridge [11]. According to the experimental testing of contact-based bridge inspections, there is a need to develop a solution for mounting application sensors (such as accelerometers, strain gauges and tilt meters) on a bridge using a UAV. It is expected that a sophisticated system with the possibility of automatic sensor mounting will increase the frequency of measurements without interrupting traffic, ensure the safety of inspectors as well as reduce inspection time and overall costs.

As mentioned earlier, the second UAV needs to be aware of the position of the sprayed adhesive which is applied in a blob-like pattern. For this purpose, an Red Green Blue - Depth (RGB-D) camera is used due to its favorable dimensions and weight. It provides image and depth information about the environment which proves useful for object localization and UAV navigation. Such cameras were commonly found on UAVs and unmanned ground vehicles (UGVs) present at the recent MBZIRC 2020 competition. In [12,13], RGB-D information is used for color-based brick detection and localization for the wall-building challenge using UAVs and UGVs, respectively, while in [14] the authors use a Convolutional Neural Network (CNN)-based UAV detection and tracking method for the intruder UAV interception challenge. Furthermore, visual sensors proved useful in [15], where the authors performed a contact-based inspection of a flat surface with an aerial manipulator. The surface position and orientation was obtained by applying random sample consensus (RANSAC) on the RGB-D information. A thorough survey of 2D object detection methods from UAVs was given in [16]. In this paper, the authors present a modular framework for object detection in which a simple contour-based blob detector is implemented. The goal is to use RGB-D information to enable an autonomous inspection workflow. The blob position is obtained by segmenting the depth data at the points where the object is detected in the image, while RANSAC [17] is used to determine its orientation.

After the successful detection of a blob-like pattern, it is necessary to attach the inspection sensor. The first phase of the sensor attachment is achieving contact, and the second is maintaining that contact to allow for the two adhesive components to form a bond. Generally, the contact can be achieved with or without force measurements. In [18], contact with the wall is performed and maintained. Researchers in [19] performed wall contact and aerial writing experiments. The work presented in [20] modeled and exploited the effects of the ceiling effect to perform an inspection underneath a bridge. The common denominator in the former approaches is maintaining the contact without any force feedback. Although mounting a force sensor on a UAV increases both mechanical and control complexity, an immediate benefit is the ability to maintain precise contact force regardless of the environment. In [21], the researchers used a force/torque sensor to achieve compliant control while pulling a rope and a semi-flexible bar. A fully actuated UAV with a manipulator has been employed in [22] to compare force feedback control with and without the force/torque sensor. Researchers in [23] used a single degree of freedom manipulator with a force sensor mounted at the end-effector to press an emergency switch.

Relying on the blob-like pattern detection and the impedance control, a trajectory for achieving contact is required to steer the aerial manipulator towards the contact point. While mounting the sensor, it is essential that the approach and contact are perpendicular

to the wall plane. This can be considered as a task constraint imposed on the planner which the aerial manipulator has to satisfy. Researchers in [24] propose a task constrained planner for a redundant robotic manipulator that enables them to do everyday tasks such as opening drawers or picking up objects. In [25], a task-constrained planner was developed for underactuated manipulators. Since multirotor UAVs are typically underactuated systems, it is necessary to address dynamics and kinematics while planning the end-effector trajectory. Aerial manipulator 6D end-effector trajectory tracking based on the differential flatness principle was presented in [26]. The underactuated nature of multirotor UAVs can cause unexpected deviations in end-effector configuration. Researchers in [27] address this particular problem by including the dynamic model of the system into the planning procedure. In our previous work [28], a trajectory planning method based on the full dynamic model of an aerial manipulator was developed. In this paper, we further augmented this method to plan for the desired force required by the impedance controller.

## 3. Mathematical Model

In this section, the mathematical model of the aerial manipulator is presented. The coordinate systems convention is depicted in Figure 2. Furthermore, an analysis for the manipulator dexterity and reach was performed.

### 3.1. Kinematics

The inertial frame is defined as $L_W$. The body-fixed frame $L_B$ is attached to the center of gravity of the UAV. The position of the UAV in the world frame is given with $\mathbf{p}_W^B = \begin{bmatrix} x & y & z \end{bmatrix}^T \in$, and the attitude vector is $\boldsymbol{\Theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$. Combining the position and attitude vectors defined the generalized coordinates of the UAV as $\mathbf{q}_B = \begin{bmatrix} (\mathbf{p}_W^B)^T & \boldsymbol{\Theta}^T \end{bmatrix}^T \in \mathbb{R}^6$. Written in a matrix form $T_W^B$, the transformation contains both the position and orientation of the UAV obtained through an on-board sensor fusion or through an external positioning system (i.e., GPS). The notation $T_a^b \in \mathbb{R}^{4 \times 4}$ was used to denote a homogeneous transformation matrix between frames $a$ and $b$.

A rigid attachment between the body of the UAV and the base of the manipulator $L_0$ was considered, denoted with the transformation matrix $T_B^0$. The manipulator used in this work was a $M = 3$ degree-of-freedom (DoF) serial chain manipulator with the end-effector attached to the last joint. The DH parameters of the arm are given in Table 1. Using this notation, one can write a transformation matrix $T_0^{ee}$ between the manipulator base and its end-effector as a function of joint variables $q_1, q_2$ and $q_3$. For brevity, the expression for the entire matrix $T_0^{ee}$ is left out and only the end-effector position and its approach vector equations are written using the well-known abbreviation $\cos(q_1 + q_2) := C_{12}$:

$$\mathbf{p}_0^{ee} = \begin{bmatrix} a_1(C_1 + C_{12}) + d_3 C_{123} \\ a_1(S_1 + S_{12}) + d_3 S_{123} \\ 0 \end{bmatrix}, \mathbf{z}_0^{ee} = \begin{bmatrix} C_{123} \\ S_{123} \\ 0 \end{bmatrix} \quad (1)$$

**Table 1.** DH parameters of the 3-DoF manipulator attached to the UAV. A virtual joint $q_4^*$ is added to fully comply with the DH convention. Link sizes $a_1$ and $d_3$ are omitted for clarity.

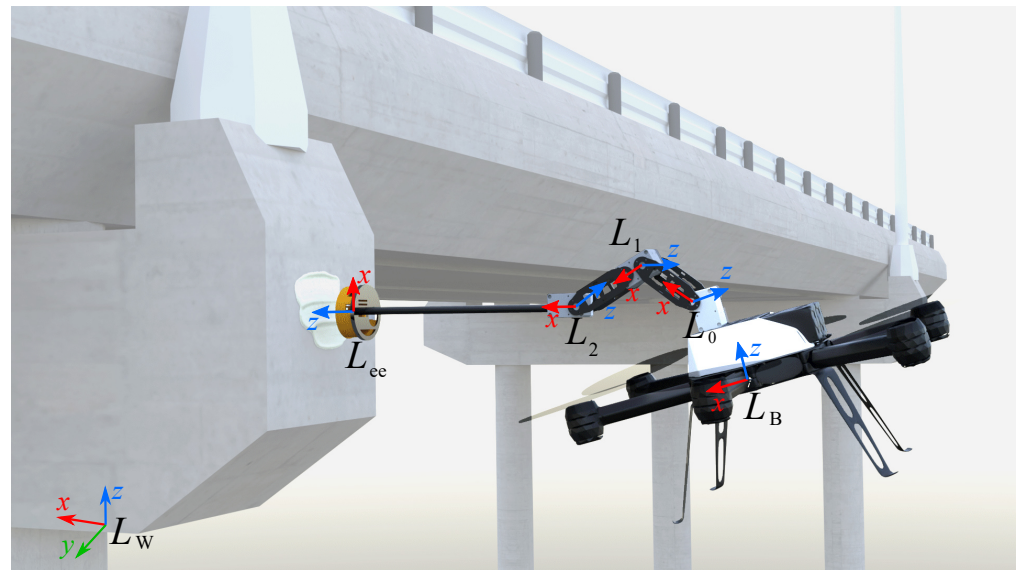| Joint | $\theta$ | $d$ | $\alpha$ | $a$ |
|-------|----------|-----|----------|-----|
| $q_1$ | 0 | 0 | 0 | $a_1$ |
| $q_2$ | 0 | 0 | 0 | $a_1$ |
| $q_3$ | $\frac{\pi}{2}$ | 0 | $\frac{\pi}{2}$ | 0 |
| $q_4^*$ | 0 | $d_3$ | 0 | 0 |

**Figure 2.** Coordinate systems of the world, UAV and the 3-DoF manipulator.

Putting it all together, the full kinematic chain of the aerial manipulator can be constructed as

$$T_W^{ee} = T_W^B \cdot T_B^0 \cdot T_0^{ee},\tag{2}$$

combining the fixed transformation $T_B^0$ with $T_W^B$ and $T_0^{ee}$ depend on UAV and manipulator motion. Since there is obvious coupling between the motion of the body and the manipulator arm, a $\beta \in [0,1]$ parameter is introduced to distribute the end-effector motion commands either to the UAV global position control or manipulator joint position control. To this end, the following distribution relationship is used:

$$\Delta \mathbf{P}_{UAV} = \beta \cdot \Delta \mathbf{P}$$
$$\Delta \mathbf{P}_{arm} = (1 - \beta) \cdot \Delta \mathbf{P},\tag{3}$$

where $\Delta \mathbf{P}$ is used to denote the desired aerial manipulator displacement expressed as the following combination body and arm motion:

$$\Delta \mathbf{P} = \Delta \mathbf{P}_{UAV} + \Delta \mathbf{P}_{arm}$$
$$= \beta \cdot \Delta \mathbf{P} + (1 - \beta) \cdot \Delta \mathbf{P}.\tag{4}$$

The manipulator displacement is denoted by $\Delta \mathbf{P}_{arm}$ and the UAV displacement by $\Delta \mathbf{P}_{UAV}$, where both $\Delta \mathbf{P}_{arm}$ and $\Delta \mathbf{P}_{UAV}$ are expressed in the coordinate system $L_0$. With $\beta = 1$, the UAV motion is used to control the position of the end-effector. When $\beta = 0$, the situation is reversed and the manipulator motion is used to move the end-effector. For every other $\beta$, the end-effector motion is obtained in part by the UAV body and the manipulator arm motion.

There are obvious advantages in combining the motion of the UAV and the manipulator arm. The UAV can move in 3D space beyond the reach of the arm; however, the motion of the UAV is not as precise and dynamically decoupled. The kinematics of the arm enable the end-effector to obtain the desired approach angle $\mathbf{z}_0^{ee} = [\cos(\delta), \sin(\delta), 0]^T$, which under the hovering assumption, becomes equal to the global approach vector $\mathbf{z}_W^{ee}$ pointing towards the contact point on the infrastructure. The straightforward mathematical manipulation of Equation (1) allows for writing the constraint equation:

$$q_3 = \delta - q_1 - q_2,\tag{5}$$

which ensures that the manipulator points in the right direction, where $\delta$ is the desired manipulator inclination in the body $x$–$z$ plane.

To find the optimal manipulator pose during contact, the dexterity $\mathfrak{D}$ and the reach $\mathfrak{R}$ of the pose were taken into account, while considering that the joints are as far as possible from their physical limits $\mathfrak{L}$. Since the motion of the arm is constrained with its approach axis condition, a reduced form of a Jacobian matrix was used $\mathbf{J} = \left[ \frac{\delta \mathbf{p}_0^{ee}}{\delta q_1}, \frac{\delta \mathbf{p}_0^{ee}}{\delta q_2} \right]$ to derive the pose dexterity index $\mathfrak{D} = \left| \mathbf{J}^T \cdot \mathbf{J} \right|$ and determine how far the current pose is from the null space of the manipulator [29]. The reach of the pose $\mathfrak{R} = (\mathbf{p}_0^{ee})^T \cdot \mathbf{p}_0^{ee}$ was also taken into account, since the goal was to keep the end-effector and the contact point as far away from the UAV body. Finally, the following equation is defined:

$$\mathfrak{L} = \frac{(q_1^2 - Q_{1max}^2)(q_2^2 - Q_{2max}^2)}{Q_{1max}^2 Q_{2max}^2}, \tag{6}$$

to measure how far away the given configuration is (i.e., $q_1, q_2$) from the joint limits $Q_{1max}, Q_{2max}$. Normalizing $\mathfrak{D}, \mathfrak{R}$ and $\mathfrak{L}$ enables combining the three conditions into a single manifold $\mathfrak{M} = \mathfrak{D} \cdot \mathfrak{R} \cdot \mathfrak{L}$ and find the optimal configuration $\mathbf{q}_M^* = \begin{bmatrix} q_1^* & q_2^* & q_3^* \end{bmatrix}^T$ for the desired approach angle $\delta$. The described method is depicted in Figure 3 for the specific case of the approach angle $\delta = 0°$, but can be extended to any value of the approach angle through Equation (5).

As a side note, the manipulator attachment on the top of the UAV body was chosen to be able to reach surfaces underneath the bridge. Although this shifts the center of gravity upwards, the stability of the system is not compromised since the manipulator is constructed of lightweight materials.
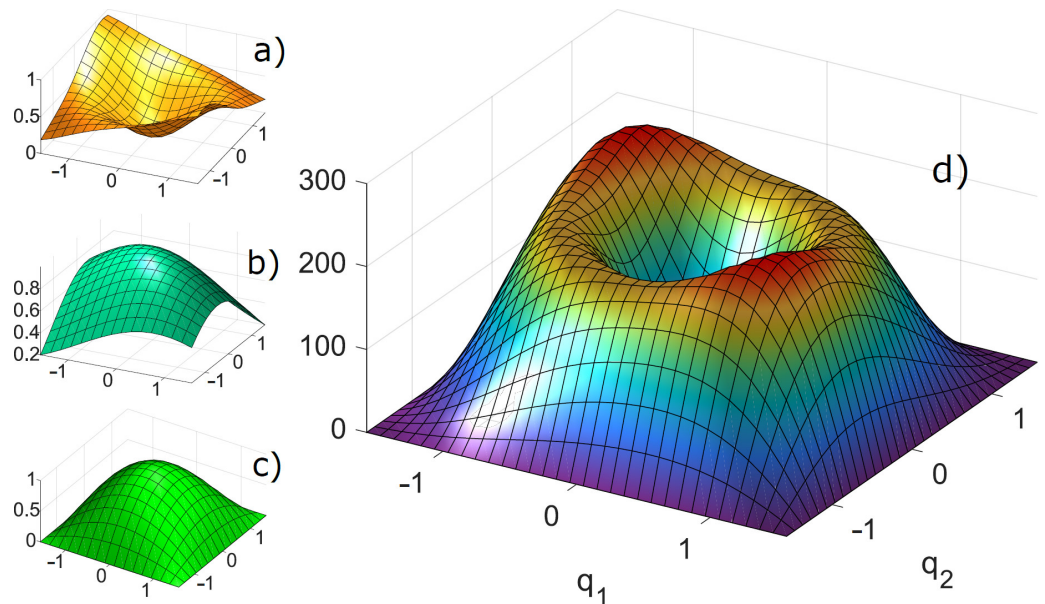


**Figure 3.** The visual decomposition of dexterity $\mathfrak{D}$, reach $\mathfrak{R}$, limit $\mathfrak{L}$ and the overall combined surface. This analysis is performed for $\delta = 0°$: (**a**) the dexterity $\mathfrak{D}$ surface shows the measure of how far the manipulator is from the null space. Values around zero are closer to the null space; (**b**) the reach $\mathfrak{R}$ surface shows how far the end-effector can move in a certain configuration. This value tends towards zero as the arm approaches a folded configuration; (**c**) the limit $\mathfrak{L}$ depicts how far a certain configuration is from the physical limits of the manipulator joints; and (**d**) the combined manifold $\mathfrak{M}$ of the formerly described surfaces. Higher values offer better trade-off between dexterity, reach and limit, defining the optimal manipulator configuration $\mathbf{q}_M^*$.

*3.2. Dynamics*

The most complicated task of the aerial manipulator is attaching the sensor to a wall and maintaining the required force reference while the two-component adhesive hardens.

To successfully perform such a task, the coupled UAV-manipulator system dynamics have to be addressed for the precise end-effector configuration planning.

Considering the UAV dynamics only, the derivative of the generalized coordinates can be defined as $\dot{\mathbf{q}}_{\mathrm{B}} = \left[ (\dot{\mathbf{p}}_{\mathrm{W}}^{\mathrm{B}})^T \quad (\boldsymbol{\omega}_{\mathrm{W}}^{\mathrm{B}})^T \right]^T \in \mathbb{R}^6$. Here, the $(\dot{\mathbf{p}}_{\mathrm{W}}^{\mathrm{B}})^T$ is the linear velocity of the body of the UAV in the world frame and the $(\boldsymbol{\omega}_{\mathrm{W}}^{\mathrm{B}})^T$ represents the angular velocity of the UAV in the world frame. The UAV's propulsion system consists of $n_p$ propellers rigidly attached to the body. Each propeller produces force and torque along the $z_{\mathrm{B}}$ axis. The vector of the propeller rotational velocities is simply defined as

$$\boldsymbol{\Omega}_{\mathrm{UAV}} = \begin{bmatrix} \Omega_1 & \dots & \Omega_{n_p} \end{bmatrix}^T \in \mathbb{R}^{n_p}. \tag{7}$$

Force and torque produced by each propeller are non-linear functions depending on the rotational velocity $\boldsymbol{\Omega}_{\mathrm{UAV}}$. Rather than using the rotational velocities as control inputs, they can be mapped to a more convenient space. Namely, the mapped control input space can be written as

$$\mathbf{u}_{\mathrm{UAV}} = \mathbf{K} \cdot diag(\boldsymbol{\Omega}_{\mathrm{UAV}}) \cdot \boldsymbol{\Omega}_{\mathrm{UAV}}, \tag{8}$$

where $\mathbf{K} \in \mathbb{R}^{4 \times n_p}$ is the mapping matrix and $\mathbf{u}_{\mathrm{UAV}} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T$, where $u_4$ represents the net thrust and $u_1$, $u_2$ and $u_3$ are moments around the body frame axes.

As stated earlier, the manipulator consists of three rotational DoFs. Therefore, the joint positions of the manipulator are defined as $\mathbf{q}_{\mathrm{M}} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$. The rotational velocity of each joint is a time derivative of joint positions $\dot{\mathbf{q}}_{\mathrm{M}} = d\mathbf{q}_{\mathrm{M}}/dt$. The torque of each joint is considered the control input of the manipulator $\mathbf{u}_{\mathrm{M}} = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$.

The resulting generalized coordinates of the aerial manipulator can be written as $\mathbf{q} = \begin{bmatrix} \mathbf{q}_{\mathrm{UAV}} & \mathbf{q}_{\mathrm{M}} \end{bmatrix}^T \in \mathbb{R}^9$, and the velocities can be obtained in the same manner as $\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_{\mathrm{UAV}} & \dot{\mathbf{q}}_{\mathrm{M}} \end{bmatrix}^T \in \mathbb{R}^9$. The resulting control inputs of the system can be expressed as $\mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathrm{UAV}} & \mathbf{u}_{\mathrm{M}} \end{bmatrix}^T \in \mathbb{R}^7$. Finally, the full system dynamics can be written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \tag{9}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{9 \times 7}$ is the inertia matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^7$ is the vector of centrifugal and Coriolis forces, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^7$ is the gravitational term.

## 4. Control System

The overall control of the aerial manipulator consists of several nested control loops. The complete controller overview, with motion planning and blob detection blocks, is depicted in Figure 4.

### 4.1. Aerial Manipulator Control

At the inner most level, the UAV is controlled through cascade attitude and rate controllers. The input to these controllers is the desired orientation and based on the state, the output is the vector of the rotors' angular velocities. The second level of control, which uses the inner attitude control loop, consists of two additional cascades, the position and the velocity control. These controllers receive a referent position and velocity feed-forward value to generate the desired vehicle orientation and thrust. The manipulator joints are controlled through standard Proportional, Integral, Derivative (PID) controllers; however, in a real-world setting, servo motors with integrated control are typically used.

As mentioned earlier, it is important to track the desired force after contact with a wall is achieved. To accomplish this, an adaptive impedance controller is employed to generate an appropriate setpoint for the position controller. This controller receives a trajectory

supplied by the mission planner, which steers the aerial manipulator towards the sensor mounting target on the bridge.
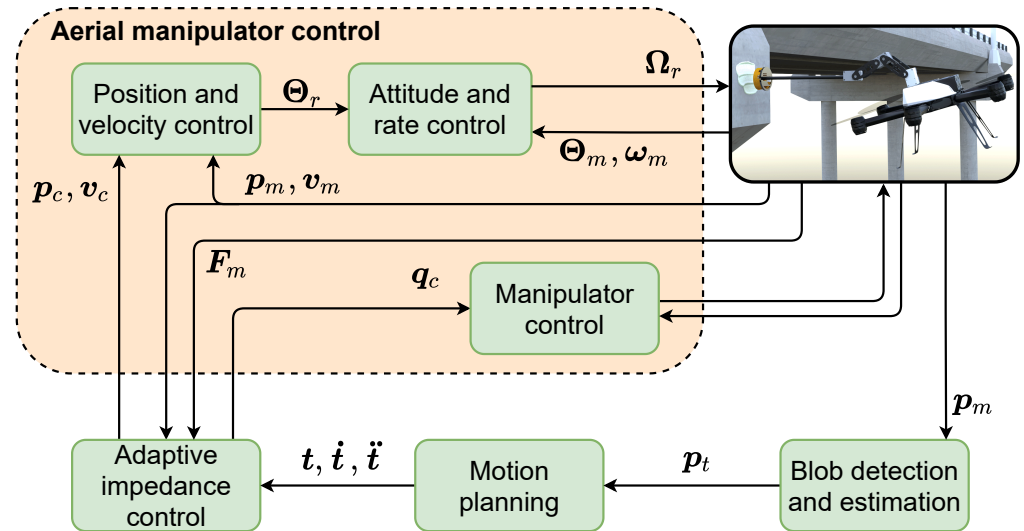


**Figure 4.** The overall functional schematic of the system. The aerial manipulator control subsystem is necessary for the controlling position and attitude of the UAV, and manipulator joints. On top of this controller, the adaptive impedance control is employed in order to track the desired force. Motion planning generates an appropriate trajectory based on the target point supplied by the blob detection algorithm.

*4.2. Adaptive Impedance Control*

The objective of the adaptive impedance controller is to ensure a stable physical interaction between the aerial manipulator and the environment [30]. As mentioned earlier, the standard UAV control scheme is based on position and attitude controllers. When interacting with the environment, the desired contact force must be considered. The position controlled system can be extended to follow the desired force by introducing an impedance filter. The design of such a filter is explained here for a single DoF.

The behavior of the system is defined by the target impedance as

$$e(t) = m(\ddot{x}_c(t) - \ddot{x}_r(t)) + b(\dot{x}_c(t) - \dot{x}_r(t)) + k(x_c(t) - x_r(t)), \qquad (10)$$

where $m$, $b$ and $k$ are constants, $x_r(t)$ is the referent position, provided to the impedance filter as an input, and $x_c(t)$ is the output of the impedance filter representing the position command. The filter is designed as a linear second-order system with a dynamic relationship between the position and the contact force tracking error $e(t)$ so that it mimics a mass-spring–damper system. The contact force tracking error is defined as follows:

$$e(t) = f_r(t) - f(t), \qquad (11)$$

where $f_r(t)$ is the other filter input defining the referent force, and $f(t)$ is the measured (exerted) contact force. If the environment is modeled as a first-order elastic system (equivalent spring system) with unknown stiffness $k_e$, the measured force can be approximated as

$$f(t) = k_e(x(t) - x_e(t)), \qquad (12)$$

where $x(t)$ is the position of the manipulator and $x_e(t)$ is the position of the environment in an unexcited state. By substituting Equation (12) in Equation (11), the position of the aerial manipulator can be expressed as follows:

$$x(t) = \frac{f_r(t) - e(t)}{k_e} + x_e(t). \qquad (13)$$

Assuming that the commanded position value can be achieved by the aerial manipulator, i.e., $x = x_c$, the substitution of Equation (13) in Equation (10), the system in the steady state can be described as follows:

$$e(t) = \frac{k \cdot k_e}{k + k_e} \left( \frac{f_r(t)}{k_e} + x_e(t) - x_r(t) \right). \tag{14}$$

For a contact force error of zero in the steady state, the following must hold:

$$x_r(t) = \frac{f_r(t)}{k_e} + x_e(t). \tag{15}$$

In other words, the position setpoint has to be designed in such a way that it compensates for the displacement of the environment due to the exerted contact force. To ensure this, a value of the unknown environment stiffness $k_e$ is needed. Furthermore, $k_e$ plays a fundamental role in the stability of the impedance filter Equation (10), which ultimately affects the stability of the aerial manipulator while in contact with the environment. A stable contact between the aerial manipulator and the environment can be ensured using the Hurwitz stability criterion, by designing the system with $b/m > 0$ and $(k + k_e)/m > 0$. However, since $k_e$ is unknown, an adaptation law for the position setpoint that guarantees the contact stability while compensating for this hidden, unknown parameter is proposed.

The adaptation law is derived starting from Equation (15). An adaptive parameter $\kappa(t)$ is introduced so that:

$$x_r(t) = \kappa(t)f_r(t) + x_e(t). \tag{16}$$

It can be shown using the Lyapunov stability analysis that the following adaptation dynamics equation for $\kappa(t)$ will yield a stable system response:

$$k\dot{\kappa}(t) + b\ddot{\kappa}(t) + m\dddot{\kappa}(t) = -\gamma\sigma(t) + \gamma_d\dot{\sigma}(t). \tag{17}$$

We refer the interested reader to the proof which can be found in the Appendix A.

## 5. Motion Planning

As discussed in Section 1.1, the main concept of this paper was to use a team of two UAVs, each applying one component of the adhesive. To apply the "resin" component, the UAV has to plan a collision-free trajectory and position itself in front of the target area to start spraying. This is fundamentally different from mounting a sensor coated with "hardener". In the latter case, apart from planning a collision-free trajectory, the manipulator-endowed UAV has to apply pressure for a certain amount of time for the two components to mix.

From the perspective of motion planning, the planner needs to be augmented to include a manipulator with three degrees of freedom, contact force and the weighing parameter $\beta$. To successfully maintain the pressure, the planner relies on the impedance controller described in Section 4.2. Furthermore, one of the requirements when mounting the sensor on the wall is for the sensor to be perpendicular to the wall. Therefore, it is necessary to take the underactuated nature of the multirotor UAVs into account during the motion planning. Namely, the errors in the planned end-effector configuration were mainly induced due to the roll and pitch angles while executing the planned motion. In our previous work [28], we developed a model-based motion planner for aerial manipulators that is capable of correcting the aforementioned end-effector deviations. In this paper, the idea from [28] was extended to consider the impedance control when obtaining the full state of the aerial manipulator.

### 5.1. Waypoint Configuration

When dealing with an aerial manipulator, exerting some contact force inevitably yields a high dimensional waypoint configuration. We define a single waypoint as a set of UAV and joint poses, together with the force reference and motion distribution factor $\beta$:

$$\mathbf{w} = \begin{bmatrix} \mathbf{q}_\mathrm{B}^T & \mathbf{q}_\mathrm{M}^T & \mathbf{f}_r^T & \beta \end{bmatrix}^T \in \mathbb{R}^{13}, \tag{18}$$

where $\mathbf{q}_\mathrm{B} \in \mathbb{R}^6$ and $\mathbf{q}_\mathrm{M} \in \mathbb{R}^3$ are generalized coordinates of the UAV and the manipulator defined in Section 3.1. The force reference vector $\mathbf{f}_r = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \in \mathbb{R}^3$ and weighing scalar parameter $\beta$ are required by the impedance controller. Furthermore, the impedance controller assumes a step change of these values. Ideally, the change should occur at the moment of contact since no force can be exerted without contact. Therefore, these values are only changed at the final waypoint.

Apart from the desired force and the parameter $\beta$, the final waypoint must contain the UAV position and orientation, as well as the manipulator joint configuration. Specifying these values relies on the blob detection algorithm presented in Section 6. Namely, the algorithm outputs the position and orientation of the detected blob in the world frame. Following the manipulator dexterity and reach analysis described in Section 3.1, the optimal manipulator configuration $\mathbf{q}_\mathrm{M}^*$ is obtained based on the provided plane normal. The optimal manipulator configuration is then used as the desired configuration for the final waypoint. This way, during operation, the manipulator never reaches a fully extended or contracted pose, which allows the impedance controller to command both the arm and the UAV to achieve and maintain the desired force.

### 5.2. Trajectory Planning

There are three phases in the trajectory planning procedure. First, an initial trajectory is planned based on the provided waypoints. Second, the initial trajectory is sent to a simulated model in order to obtain the full state of the aerial manipulator during the trajectory execution. Third, the end-effector configuration is corrected based on the full state of the vehicle, and the final trajectory is sent to the target aerial manipulator.

Initial Trajectory

To execute a smooth motion towards the desired waypoint, we use a suitable time-optimal path planning by reachability analysis (TOPP-RA) trajectory planner [31]. The TOPP-RA algorithm searches for the time-optimal trajectory and is based on a "bang-bang" principle on the generalized torque of each DoF. The planner is capable of receiving the input waypoints of an arbitrary dimension and output a smooth trajectory. Each DoF has to be provided with dynamical constraints in terms of velocity and acceleration, which are respected during the trajectory generation process.

As mentioned, the input to the TOPP-RA trajectory is the path of a set of $n \geq 2$ waypoints:

$$\mathscr{P} = \left\{ \mathbf{w}_i \mid \mathbf{w}_i \in \mathbb{R}^{13}, i \in (0, 1, \dots, n) \right\}. \tag{19}$$

Based on the dynamical constraints, the output of the TOPP-RA planner is a sampled trajectory:

$$\mathscr{T}_s = \left\{ \mathbf{t}(kT_s) \mid \mathbf{t}(kT_s) \in \mathbb{R}^{3 \times 13}, k \in (0, \dots, n_t) \right\}, \tag{20}$$

where $\mathbf{t} = \begin{bmatrix} (\mathbf{w})^T & (\dot{\mathbf{w}})^T & (\ddot{\mathbf{w}})^T \end{bmatrix}^T \in \mathbb{R}^{3 \times 13}$ is a single sampled trajectory point consisting of position, velocity and acceleration; $T_s$ is the sampling time; and $n_t$ is the number of points in the sampled trajectory. Note that each trajectory point contains both roll and pitch angles. Although these angles can be planned through the TOPP-RA algorithm,

they are omitted at this point because of the underactuated nature of the multirotor UAV. Nevertheless, they are used later in the paper when the model corrections are applied.

The impedance controller expects a step change in the force and weighing parameter $\beta$ referent values. To satisfy this requirement, large constraints for velocity and acceleration are imposed for these DoFs. However, because other DoFs have constraints below their physical limit, the overall force and $\beta$ trajectory has a slower, dynamically smooth profile. These profiles also have overshoots and undershoots which are not acceptable because they are not within the hard constraints required for $\beta$. To tackle this problem, a simple piecewise constant velocity interpolation was applied to the force and $\beta$. This way, a large velocity constraint produces a step change which is a suitable input to the impedance controller. A visual example of the difference between the TOPP-RA and piecewise constant velocity interpolation is depicted in Figure 5.
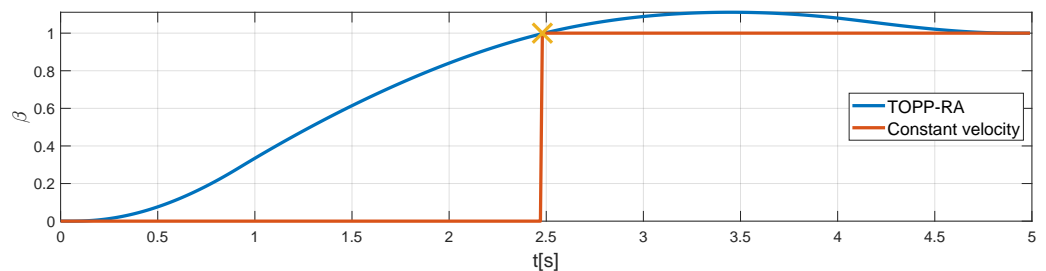


**Figure 5.** Visual comparison between TOPP-RA and the piecewise constant velocity interpolation. Waypoints for both trajectories are kept the same, and around $t = 2.5$ s, the middle waypoint(yellow cross) is reached. Although the dynamical constraints are the same, TOPP-RA takes other degrees of freedom into account and produces a trajectory with overshoot which is not suitable for parameter $\beta$.

### 5.3. Model-Based Corrections

The initial trajectory from Equation (20) is planned without any consideration about the underactuated nature of the multirotor UAV. To obtain the unknowns, namely roll and pitch angles, the initial trajectory can be executed in a simulation environment. The chosen simulation environment is, in our case, *Gazebo*, because it is realistic and supports the robotics operating system (ROS), which is the backbone of our implementation. The simulated aerial manipulator is based on the mathematical model described in Section 3. The standard cascade PID controllers are employed for low-level attitude and high-level position control. The impedance controller is built on top of the position controller and provides a position reference based on the input trajectory. More details about the simulation environment are provided in Section 7.

The first step is executing the initial trajectory in the aforementioned simulation environment. While executing, the roll and pitch angles are recorded as they are needed for obtaining the full state of the UAV. Rearranging Equation (2) and plugging the unknown roll and pitch angles in the full state of the UAV, the transform of the end-effector in the manipulator base frame can be obtained:

$$T_{L_0}^{ee} = (T_{\mathrm{B}}^{L_0})^{-1} \cdot (T_{\mathrm{W}}^{\mathrm{B}})^{-1} \cdot T_{\mathrm{W}}^{ee}. \tag{21}$$

Usingthe inverse kinematics of the manipulator, joint values $\mathbf{q}_{\mathrm{M}}$ for the desired end-effector configuration are obtained. This way, the null space of the aerial manipulator is used for the end-effector correction. Note that due to the configuration of the manipulator, an exact solution of the inverse kinematics will not always exist. In such a case, an approximate closest solution is used instead.

The final trajectory is constructed by replacing the initial $\mathbf{q}_{\mathrm{M}}$ with the corrected values. This trajectory is afterwards sent to the target aerial manipulator.

The careful reader should note that the developed three DoF manipulator operates on the $x$ and $z$ position in the body frame, as well as the pitch angle. This allows the impedance controller to maintain the orientation perpendicular to the wall, while compensating for the

UAV body motion in the $x$ and $z$ axes. However, the system will experience disturbances and control errors which will act on the roll and pitch angle, and the lateral movement along the body $y$ axis. We can address these issues either with mechanical dampers or by adding additional degrees of freedom to the manipulator, which will be explored in future work.

## 6. Blob Detection

This section presents the methods we propose to detect the hardener blob position and orientation. A modular object detection framework, as shown in Figure 6, is designed to ensure a reliable blob pose detection. Since the detection is to be done on board the UAVs, RGB-D cameras are selected. Therefore, the inputs to the framework are images and organized point clouds obtained from the visual sensor. The remainder of this section introduces the individual components of the framework and adds implementation details where necessary.
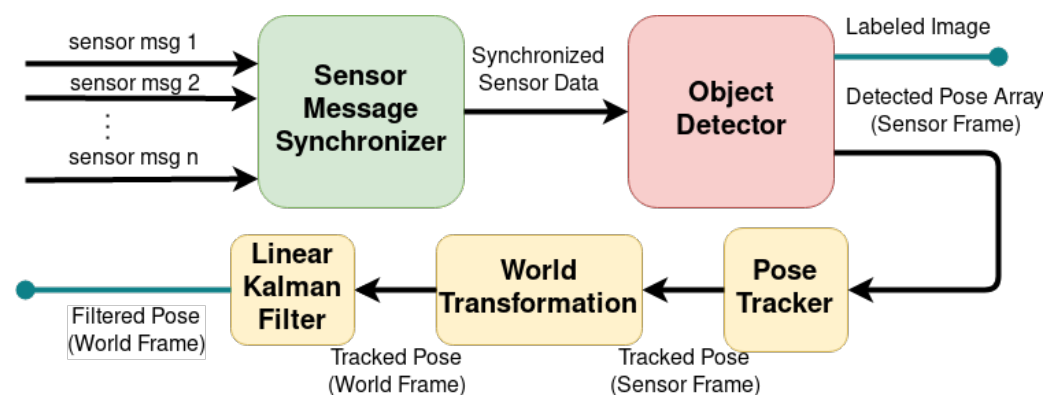


**Figure 6.** The pipeline for the modular object detection framework. Inputs to the system are an arbitrary number of sensor messages (image, depth, point cloud, etc.) along with sensor–world frame transformations. Output is the detected blob pose in the world frame. The synchronizer and detector are modular components, while the linear Kalman filter, world transformation and pose tracker stay invariant.

The sensor message synchronizer is responsible for the time-based synchronization of the given sensor message streams. In the case of blob detection, a module that synchronizes images and organized pointclouds from an RGB-D camera is derived. This is necessary since the algorithm detects the blob in both 2D image space and 3D point clouds, which are not necessarily sampled simultaneously. The underlying implementation uses ROS libraries to synchronize messages with an approximate time policy.

An object detector attempts to find a set of object poses using synchronized sensor data. The module used in this paper detects blob poses and is implemented in the following way. First, all the blob positions and radii are found in the image frame using the standard blob detection functionality found in the OpenCV libraries. Second, the depth information corresponding to the detected blobs is isolated from the organized point cloud. Finally, blob positions are calculated as centroids of the corresponding depth positions, while the orientation is obtained through the random sample consensus (RANSAC) algorithm from the Point Cloud Library (PCL).

The remaining framework components are independent from synchronizer and detector modules. The pose tracker is used to track the obtained object's poses through multiple frames based on the closest Euclidean distance criterion. This component solves the issue of multiple objects being visible, as it always outputs the pose of the currently tracked object. Moreover, it increases the robustness of the system since it remembers the object poses for a certain number of frames, which allows some leniency with the detector.

The goal of the world transformation component is to transform the tracked pose from the sensor to the world frame using the estimated odometry from an external source

that any UAV should have access to. Additionally, since the blob poses are to be sent as references to the trajectory planner, it is important to correctly compute the blob orientation. Since the blob is a flat surface, there are two equally correct possible orientations that can be detected. Therefore, the blob orientation is chosen as follows:

$$R_{\text{blob}} = \begin{cases} R_{\text{blob}} & \text{if } \mathbf{r}_{1B} \cdot \mathbf{r}_{1\text{blob}} \geq 0 \\ R_{\text{blob}} \cdot R_{180} & \text{otherwise} \end{cases}, \tag{22}$$

where $\mathbf{r}_{1B}$ is the heading component of the UAV rotation matrix expressed in world coordinates $R_B = \begin{bmatrix} \mathbf{r}_{1B} & \mathbf{r}_{2B} & \mathbf{r}_{3B} \end{bmatrix}$ and $\mathbf{r}_{1\text{blob}}$ is the heading component of the blob rotation matrix expressed in world coordinates $R_{\text{blob}} = \begin{bmatrix} \mathbf{r}_{1\text{blob}} & \mathbf{r}_{2\text{blob}} & \mathbf{r}_{3\text{blob}} \end{bmatrix}$ and $R_{180} = diag(-1, -1, 1)$.

Finally, a linear Kalman filter with a constant velocity model is used to further increase the robustness of the system and provide smoother blob position estimates. The constant velocity model for each axis is given as follows:

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{w}_k, \qquad F_k = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \tag{23}$$

where $T_s$ is the discretization step, $\mathbf{x}_k \in \mathbb{R}^2$ is the state vector containing the position and velocity along the corresponding axis and $\mathbf{w}_k \in \mathbb{R}^2$ is the process noise. The observation model along a single axis is given as follows:

$$z_k = H_x \mathbf{x}_k + v_k, \qquad H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{24}$$

where $z_k \in \mathbb{R}$ is the position observation along the corresponding axis and $v_k \in \mathbb{R}$ is the measurement noise.

If the detector is unable to provide measurements and the pose tracker removes the pose from the tracking set, the linear Kalman filter is still able to provide blob position estimates.

Experimental validation of the described methods is performed in an indoor Optitrack environment with an Intel Realsense D435 RGB-D camera. To ensure ground truth is available for detection validation, reflective markers are attached to both the camera and the blob. In order to determine the transformation between the camera optical frame and the reflective markers attached to the camera, an optimization-based calibration approach is used as described in [32].

Results are shown in Figures 7 and 8. The experiments are performed with the UAV in constant motion while looking at the general direction of the painted blob. Figure 7 shows a relative difference between the ground truth UAV motion in the world frame and the UAV motion as observed from the detected blob frame. Figure 8 presents the comparison of ground truth and detected blob positions expressed in the world frame. It is important to note that camera calibration errors can manifest themselves as static offsets between the detected and ground truth blob positions in Figure 8. However, in this case, the visual detection provided a reliable blob tracking results which is a direct consequence of careful camera calibration.
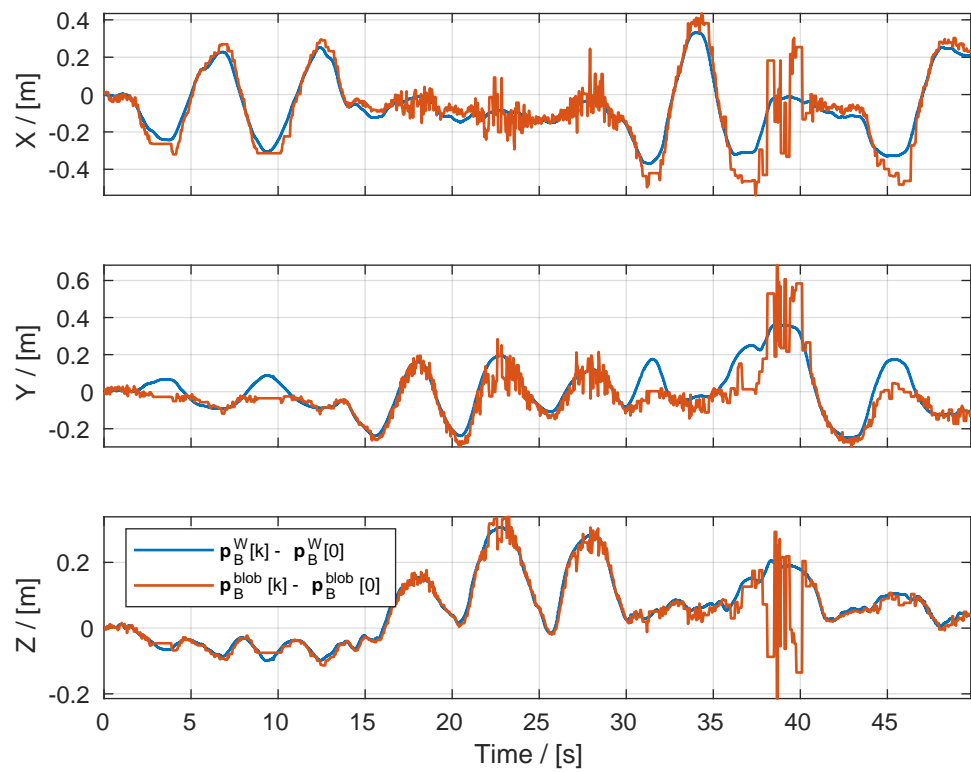
**Figure 7.** This figure shows the comparison of normalized motion of the UAV body frame as observed from the Optitrack world frame and as observed from the detected blob frame labeled $\mathbf{p}_B^W[k] - \mathbf{p}_B^W[0]$ and $\mathbf{p}_B^{blob}[k] - \mathbf{p}_B^{blob}[0]$, respectively.
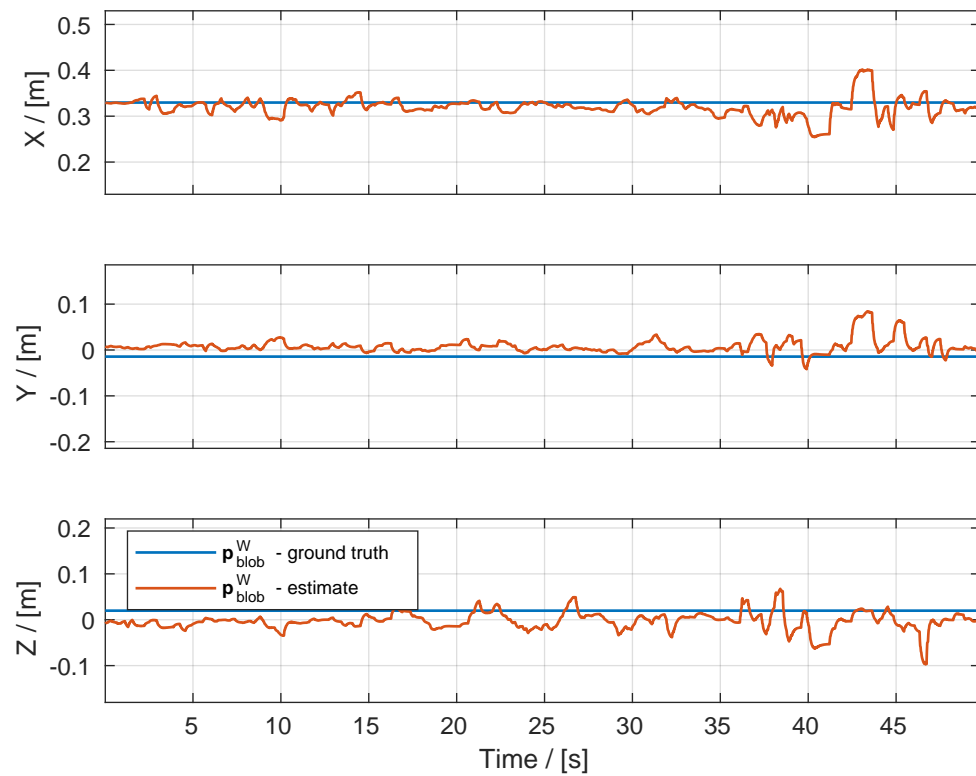


**Figure 8.** This figure shows the comparison of the ground truth blob position and the blob position estimate expressed in the Optitrack world frame. The root mean squared error across all three axes is 4.6 cm.

## 7. Simulation

The environment used for simulating the UAV and manipulator dynamics, as well as the contact with the environment, is the widely accepted Gazebo simulator. It is realistic and highly modular, with a large community and a support for the robot operating system (ROS), which is also the primary implementation environment for impedance control, motion planning and blob detection. Through ROS, Gazebo has a large variety of developed plugins realistically simulating various sensors and actuators. All simulations were conducted with Linux Ubuntu 18.04 operating system and ROS Melodic middleware installed.

The UAV is modeled as a single rigid body with $n_p$ propellers mounted at the end of each arm. As propulsion units, these propellers generate thrust along the $z$ axis of the UAV body. To simulate the propeller dynamics, the `rotors_simulator` package is used. It contains a plugin that models thrust based on the user-provided propeller parameters [33]. Furthermore, to obtain the UAV attitude and position, IMU and odometry plugins are mounted on the vehicle. The manipulator was mounted on the body of the UAV and consists of three joints connected with links. A rod type tool is mounted as the end-effector, with a force-torque sensor required by the impedance controller. Furthermore, a monocular camera with an infrared projector is also mounted for the blob detection.

### 7.1. End-Effector Motion Distribution Analysis

Given some end-effector configuration, the inverse kinematics is responsible for finding the UAV position and yaw angle, as well as the manipulator joint values that satisfy the desired configuration. The parameter $\beta$ from Equation (3) defines a ratio of how much the manipulator joints and UAV position and orientation contribute to achieving the desired end-effector configuration, as described in Section 3.1. Recalling the values, $\beta = 1$ only moves the UAV in the direction of the desired end-effector configuration; and $\beta = 0$ uses the inverse kinematics of the manipulator to achieve the desired configuration.

To determine the influence of $\beta$ on the overall system, an analysis was conducted with different $\beta$ values. The desired end-effector configuration was chosen to be in contact with a plane perpendicular to the bridge wall which required the force reference along the $x$ axis. The waypoints for the trajectory planner were kept the same across all trials, and only $\beta$ was changed. The results of this analysis are depicted in Figure 9. As can be observed, all trials produced very similar results with oscillating force upon contact and eventually reaching the desired reference, providing us with no obvious conclusion regarding how to select the optimal $\beta$. However, following the dexterity analysis from Section 3.1, and only relying on the manipulator motion might drive the system close to its limits due to the UAV body movement. On the other hand, the motion of the UAV induces disturbances in the end-effector pose control. The manipulator is therefore responsible for compensating errors introduced by the motion of the UAV body. Taking all of the aforementioned into account, the value is chosen as $\beta = 0.5$ so that both the manipulator and the UAV are simultaneously used to maintain a steady contact force.
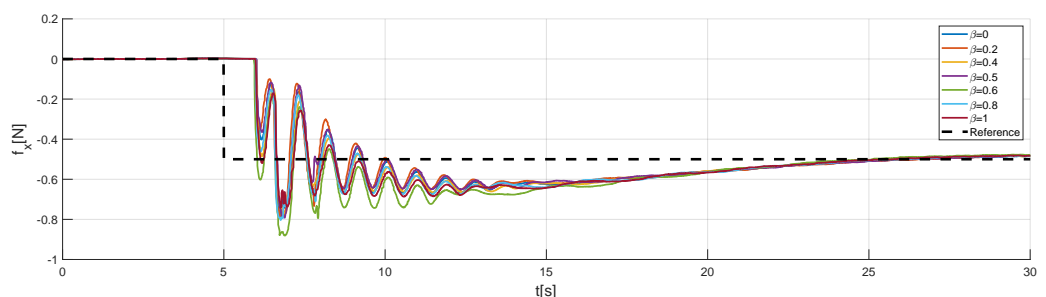


**Figure 9.** Force response comparison for different values of the parameter $\beta$. The analysis was conducted on a plane perpendicular to the ground where the force reference along the $x$ axis is required.

### 7.2. Bridge Sensor Mounting

Since the concept of this paper was to mount inspection sensors on a bridge, the simulation trials were tailored in the same direction. After spraying the first component, it is necessary to achieve and maintain a stable contact while the second adhesive component on the sensor dries. Since the manipulator is attached above the propellers, the workspace of the manipulator is limited to contact above the UAV or on the plane perpendicular to the ground.

Naturally, the first set of simulation trials were conducted by holding the desired force on a plane perpendicular to the ground. In this case, the contact force only acts along the $x$ axis and the response is depicted in Figure 10. The time delay between the planned and executed contact is present due to the impedance filter which slows down the dynamics of the referent trajectory. After the initial contact, there are some oscillations and an overshoot which diminish over time and the desired force reference is achieved.
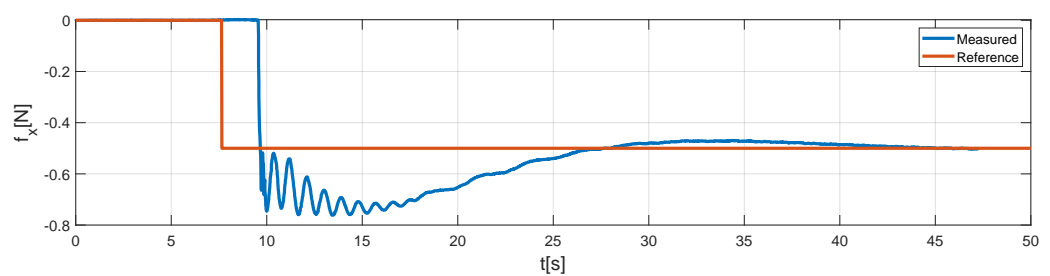


**Figure 10.** Force response in case of a contact plane perpendicular to the ground, $\delta = 0°$.

The second set of simulation trials included an inclined contact plane. This requires the UAV approach from below the plane and achieving contact perpendicular to the plane. Since the plane is inclined for $\delta = 68°$, the planned force referent values have components in both the $x$ and $z$ axes, as shown in Figure 11. Similarly to the previous example, the force response has some oscillations around the instance of contact, but it eventually settles and reaches the desired force reference.
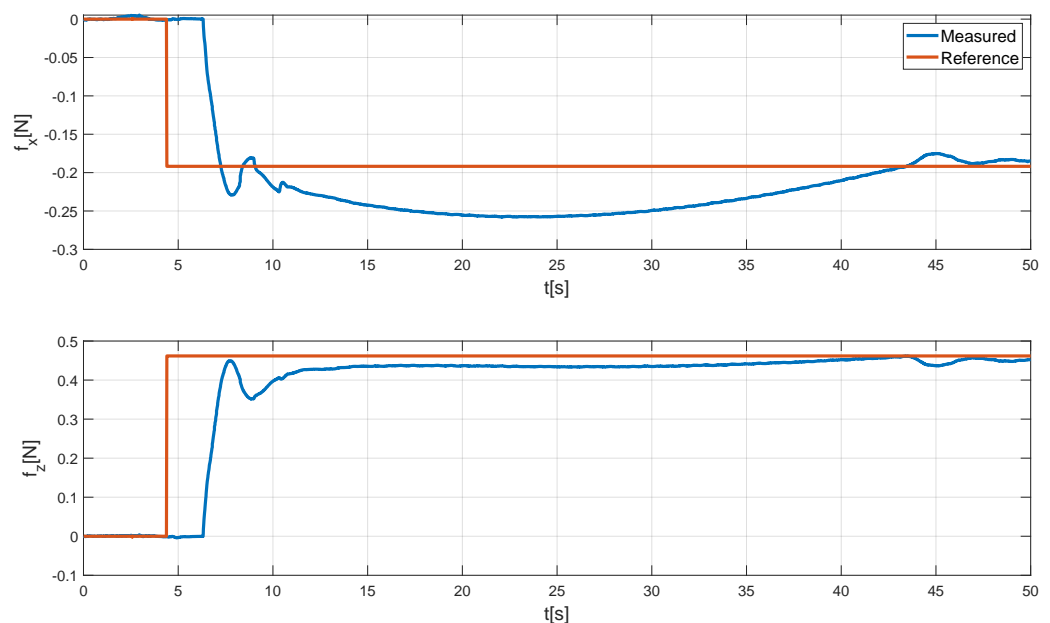


**Figure 11.** Force response in the case of the contact plane inclined for $\delta = 68°$.

The simulation tests for $\delta = 0°$ and $\delta = 68°$ were performed $n = 10$ times for each case, as depicted in Figure 12. The left portion of the figure is a dot product between the normal of the blob $\mathbf{r}_t$ and the end-effector orientation vector $\mathbf{r}_{ee}$. If the value of the dot product $\mathbf{r}_t \cdot \mathbf{r}_{ee} = 1$, the two vectors are parallel which results in a successful approach. For

both angles, the dot product is very close to 1 and the orientation error is negligible. On the right, the distance between the center of the target and contact point is shown. The error distance is in both cases less than 0.1 m, which ensures the relatively high precision of sensor mounting, well within margins for the bridge inspection. The accompanying video of simulation tests can be found on our YouTube channel [34].
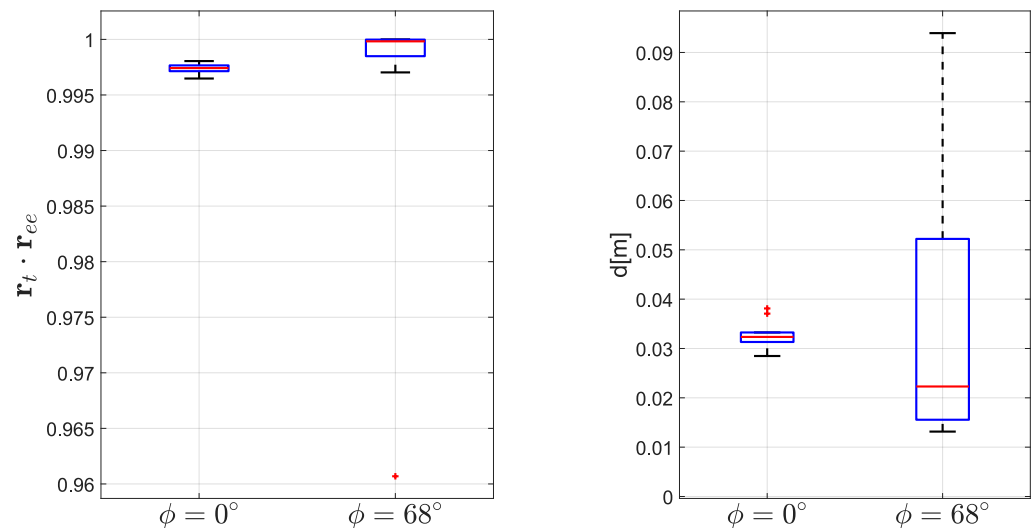


**Figure 12.** (**Left**): box and whiskers plot of the dot product between the blob plane normal vector $\mathbf{r}_t$ and the end-effector orientation vector $\mathbf{r}_{ee}$; and (**Right**): box and whiskers plot of the distance from the target point after contact.

## 8. Conclusions

This paper presents a step towards autonomous bridge inspection by investigating the possibility of mounting various inspection sensors using an aerial manipulator. Currently, inspectors use specialized trucks with cranes and baskets in order to access the area underneath the bridge. This inevitably leads to road closure which poses an inconvenience for both inspectors and traffic. To alleviate this problem, the aforementioned aerial manipulators can be used to access difficult-to-reach areas of the bridge. As mounting sensors require forming a bond between the wall and sensor, we envision using a two-component adhesive with a short cure time. Since the aerial manipulator has to achieve and maintain contact with the sensor mount point, short cure times are desirable because of te limited flight time of these platforms. Nevertheless, current flight times of outdoor multirotors reach up to 30 min, which ensures enough time for the two adhesive components to form the bond.

Although preliminary, the results of this paper seem promising. The visual detection was extensively tested and reliably tracks the blob position. The adaptive impedance controller is capable of maintaining the required force. Even though there are some oscillations and settling times in the force response, in practical use, it does not make much difference since the curing time of the adhesive is at least several minutes. The trajectory planner was augmented to plan in the force space which allows for setting the force reference step change before the contact. The simulation results show the high repeatability of the overall system which gives us the confidence to perform experiments in a real-world environment.

Our first step in future work was to perform experiments in a controlled laboratory environment. The outdoor environment poses a different set of challenges including the lower accuracy positioning system and unpredictable disturbances, i.e., wind gusts. Since these factors will inevitably reflect on the overall end-effector accuracy, we are looking into augmenting the manipulator to be able to compensate for lateral movements, as well as roll and yaw angles. To further increase the system's accuracy, the developed visual

tracker will be used to improve feedback around the tracked blob on the bridge wall in real-world experiments.

**Appendix A**

Here, we show detailed proof that the adaptation law (17) is stable. Assuming that THE referent force $f_r(t)$ is constant, $f_r(t) = F_r$, the time derivatives of Equation (16) are:

$$\dot{x}_r(t) = \dot{\kappa}(t)F_r + \dot{x}_e(t),$$

$$\ddot{x}_r(t) = \ddot{\kappa}(t)F_r + \ddot{x}_e(t),$$

(A1)

while the derivatives of Equation (13) yield:

$$\dot{x}(t) = \frac{-\dot{e}(t)}{k_e} + \dot{x}_e(t),$$

$$\ddot{x}(t) = \frac{-\ddot{e}(t)}{k_e} + \ddot{x}_e(t).$$

(A2)

By susbstituting Equations (A1) and (A2) into Equation (10), the dynamics of the contact force error can be obtained as

$$m\ddot{e}(t) + b\dot{e}(t) + (k + k_e)e(t) = g(t),$$

(A3)

where:

$$g(t) = F_r[k(1 - k_e\kappa(t)) - k_e(b\dot{\kappa}(t) + m\ddot{\kappa}(t))],$$

(A4)

and $x = x_c$. The adaptation law thus determines the dynamics of the adaptation parameter $\kappa(t)$ and defines the dynamics of the contact force error. Formally, the adaptation law should enforce $g(t) \to g^*$ such that $e(t) \to 0$ and $\kappa(t) \to 1/k_e$.

For the Lyapunov candidate:

$$V(t) = \frac{1}{2}\left[p_1 e(t)^2 + p_2 \dot{e}^2(t)\right] + \frac{1}{2\gamma}[g(t) - g^*]^2,$$

(A5)

with $p_1$, $p_2$ and $\gamma$ as positive parameters, the condition $\dot{V}(t) \leq 0$ yields:

$$\frac{2}{\gamma}g(t)\dot{g}(t) + 2g(t)[p_1 e(t) + p_2 \dot{e}(t)] - \frac{2}{\gamma}\dot{g}(t)g^* \leq 0.$$

(A6)

After reordering, we obtain:

$$\dot{g}(t)[g(t) - g^*] \leq -\gamma \sigma(t) g(t), \tag{A7}$$

where $\sigma(t) = [p_1 e(t) + p_2 \dot{e}(t)]$. By choosing:

$$\dot{g}(t) = -\gamma \sigma(t) + \gamma_d \dot{\sigma}(t), \tag{A8}$$

where $\gamma_d$ is a positive constant, Lyapunov condition Equation (A7) becomes:

$$g(t) \leq g^*, \tag{A9}$$

i.e., for the adaptation law to be stable, $g(t)$ should be bounded. Since $x_r, \dot{x}_r$ and $\ddot{x}_r$ are bounded, so are $e, \dot{e}$ and $\ddot{e}$. Therefore, $g(t)$ is also bounded, i.e., the condition in Equation (A9) is satisfied. The adaptation law is finally obtained by taking the derivative of Equation (A3), and substituting $\dot{g}(t)$ with Equation (A8), yields the (17). Parameters $\gamma$ and $\gamma_d$ dictate the adaptation dynamics. Based on the measured contact force, the error adaptation law Equation (17) estimates the adaptation parameter $\kappa$ (reciprocal value of the environment stiffness), which is then used in Equation (16) for calculating the referent position $x_r$.

## References

1. Kashif Ur Rehman, S.; Ibrahim, Z.; Memon, S.A.; Jameel, M. Nondestructive test methods for concrete bridges: A review. *Constr. Build. Mater.* **2016**, *107*, 58–86. [CrossRef]
2. Hunt, G.; Mitzalis, F.; Alhinai, T.; Hooper, P.A.; Kovac, M. 3D printing with flying robots. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 4493–4499. [CrossRef]
3. Vempati, A.S.; Kamel, M.; Stilinovic, N.; Zhang, Q.; Reusser, D.; Sa, I.; Nieto, J.; Siegwart, R.; Beardsley, P. PaintCopter: An Autonomous UAV for Spray Painting on Three-Dimensional Surfaces. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2862–2869. [CrossRef]
4. Jeong, E.; Seo, J.; Wacker, J. Literature Review and Technical Survey on Bridge Inspection Using Unmanned Aerial Vehicles. *J. Perform. Constr. Facil.* **2020**, *34*, 04020113. [CrossRef]
5. Cacace, J.; Orozco-Soto, S.M.; Suarez, A.; Caballero, A.; Orsag, M.; Bogdan, S.; Vasiljevic, G.; Ebeid, E.; Rodriguez, J.A.A.; Ollero, A. Safe Local Aerial Manipulation for the Installation of Devices on Power Lines: AERIAL-CORE First Year Results and Designs. *Appl. Sci.* **2021**, *11*, 6220. [CrossRef]
6. Zollini, S.; Alicandro, M.; Dominici, D.; Quaresima, R.; Giallonardo, M. UAV Photogrammetry for Concrete Bridge Inspection Using Object-Based Image Analysis (OBIA). *Remote Sens.* **2020**, *12*, 3180. [CrossRef]
7. Bolourian, N.; Hammad, A. LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection. *Autom. Constr.* **2020**, *117*, 103250. [CrossRef]
8. Reagan, D.; Sabato, A.; Niezrecki, C. Unmanned aerial vehicle acquisition of three-dimensional digital image correlation measurements for structural health monitoring of bridges. In *Proceedings of the Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure, Portland, OR, USA, 25–29 March 2017*; Wu, H.F., Gyekenyesi, A.L., Shull, P.J., Yu, T.Y., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, DC, USA, 2017; Volume 10169, pp. 68–77. [CrossRef]
9. Chen, H.; Quan, F.; Fang, L.; Zhang, S. Aerial Grasping with a Lightweight Manipulator Based on Multi-Objective Optimization and Visual Compensation. *Sensors* **2019**, *19*, 4253. [CrossRef] [PubMed]
10. Zhang, G.; He, Y.; Dai, B.; Gu, F.; Yang, L.; Han, J.; Liu, G. Aerial Grasping of an Object in the Strong Wind: Robust Control of an Aerial Manipulator. *Appl. Sci.* **2019**, *9*, 2230. [CrossRef]
11. Sanchez-Cuevas, P.J.; Gonzalez-Morgado, A.; Cortes, N.; Gayango, D.B.; Jimenez-Cano, A.E.; Ollero, A.; Heredia, G. Fully-Actuated Aerial Manipulator for Infrastructure Contact Inspection: Design, Modeling, Localization, and Control. *Sensors* **2020**, *20*, 4708. [CrossRef] [PubMed]
12. Baca, T.; Penicka, R.; Stepan, P.; Petrlik, M.; Spurny, V.; Hert, D.; Saska, M. Autonomous Cooperative Wall Building by a Team of Unmanned Aerial Vehicles in the MBZIRC 2020 Competition. *arXiv* **2020**, arXiv:2012.05946.
13. Vatavuk, I.; Polic, M.; Hrabar, I.; Petric, F.; Orsag, M.; Bogdan, S. Autonomous, Mobile Manipulation in a Wall-building Scenario: Team LARICS at MBZIRC 2020. *J. Field Robot.* **2021**, submitted for publication.
14. Barišić, A.; Petric, F.; Bogdan, S. Brain over Brawn—Using a Stereo Camera to Detect, Track and Intercept a Faster UAV by Reconstructing Its Trajectory. *arXiv* **2021**, arXiv:2107.00962.
15. Markovic, L.; Car, M.; Orsag, M.; Bogdan, S. Adaptive stiffness estimation impedance control for achieving sustained contact in aerial manipulation. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021.

16. Cazzato, D.; Cimarelli, C.; Sanchez-Lopez, J.L.; Voos, H.; Leo, M. A Survey of Computer Vision Methods for 2D Object Detection from Unmanned Aerial Vehicles. *J. Imaging* **2020**, *6*, 78. [CrossRef]

17. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]

18. Fumagalli, M.; Naldi, R.; Macchelli, A.; Carloni, R.; Stramigioli, S.; Marconi, L. Modeling and control of a flying robot for contact inspection. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 3532–3537. [CrossRef]

19. Alexis, K.; Darivianakis, G.; Burri, M.; Siegwart, R. Aerial Robotic Contact-based Inspection: Planning and Control. *Auton. Robot.* **2016**, *40*, 631–655. [CrossRef]

20. Jimenez-Cano, A.E.; Sanchez-Cuevas, P.J.; Grau, P.; Ollero, A.; Heredia, G. Contact-Based Bridge Inspection Multirotors: Design, Modeling, and Control Considering the Ceiling Effect. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3561–3568. [CrossRef]

21. Cataldi, E.; Muscio, G.; Trujillo, M.A.; Rodriguez, Y.; Pierri, F.; Antonelli, G.; Caccavale, F.; Viguria, A.; Chiaverini, S.; Ollero, A. Impedance Control of an aerial-manipulator: Preliminary results. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 3848–3853. [CrossRef]

22. Nava, G.; Sablé, Q.; Tognon, M.; Pucci, D.; Franchi, A. Direct Force Feedback Control and Online Multi-Task Optimization for Aerial Manipulators. *IEEE Robot. Autom. Lett.* **2020**, *5*, 331–338. [CrossRef]

23. Meng, X.; He, Y.; Li, Q.; Gu, F.; Yang, L.; Yan, T.; Han, J. Contact Force Control of an Aerial Manipulator in Pressing an Emergency Switch Process. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2107–2113. [CrossRef]

24. Stilman, M. Global Manipulation Planning in Robot Joint Space With Task Constraints. *IEEE Trans. Robot.* **2010**, *26*, 576–584. [CrossRef]

25. Cefalo, M.; Oriolo, G. Task-constrained motion planning for underactuated robots. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2965–2970. [CrossRef]

26. Yu, Y.; Lippiello, V. 6D Pose Task Trajectory Tracking for a Class of 3D Aerial Manipulator From Differential Flatness. *IEEE Access* **2019**, *7*, 52257–52265. [CrossRef]

27. Tognon, M.; Cataldi, E.; Chavez, H.A.T.; Antonelli, G.; Cortés, J.; Franchi, A. Control-Aware Motion Planning for Task-Constrained Aerial Manipulation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2478–2484. [CrossRef]

28. Ivanovic, A.; Car, M.; Orsag, M.; Bogdan, S. Exploiting Null Space in Aerial Manipulation through Model-In-The-Loop Motion Planning. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 686–693. [CrossRef]

29. Zargarbashi, S.; Khan, W.; Angeles, J. The Jacobian condition number as a dexterity index in 6R machining robots. *Robot. Comput.-Integr. Manuf.* **2012**, *28*, 694–699. [CrossRef]

30. Car, M.; Ivanovic, A.; Orsag, M.; Bogdan, S. Position-based adaptive impedance control for a UAV. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA 12–15 June 2018; pp. 957–963. [CrossRef]

31. Pham, H.; Pham, Q. A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis. *IEEE Trans. Robot.* **2018**, *34*, 645–659. [CrossRef]

32. Maric, B.; Polic, M.; Tabak, T.; Orsag, M. Unsupervised optimization approach to in situ calibration of collaborative human–robot interaction tools. In Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 14–16 September 2020; pp. 255–262. [CrossRef]

33. Furrer, F.; Burri, M.; Achtelik, M.; Siegwart, R. Robot Operating System (ROS): The Complete Reference (Volume 1). In *Studies in Computational Intelligence*; Chapter RotorS—A Modular Gazebo MAV Simulator Framework; Springer International Publishing: Cham, Switzerland, 2016; pp. 595–625. [CrossRef]

34. LARICS. Aerial Manipulation Approach for Autonomous Bridge Inspection. Available online: https://youtu.be/bBaxPMZEV04 (accessed on 23 July 2021).