

## An Integrated Framework for Autonomous Sensor Placement With Aerial Robots

Stephens, Brett ; Nguyen, Hai-Nguyen; Hamaza, S.; Kovac, Mirko

**DOI**

[10.1109/TMECH.2022.3202116](https://doi.org/10.1109/TMECH.2022.3202116)

**Publication date**

2022

**Document Version**

Final published version

**Published in**

IEEE/ASME Transactions on Mechatronics

**Citation (APA)**

Stephens, B., Nguyen, H.-N., Hamaza, S., & Kovac, M. (2022). An Integrated Framework for Autonomous Sensor Placement With Aerial Robots. *IEEE/ASME Transactions on Mechatronics*, 28(1), 38-49. <https://doi.org/10.1109/TMECH.2022.3202116>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.





***Green Open Access added to TU Delft Institutional Repository***

***'You share, we take care!' - Taverne project***

**<https://www.openaccess.nl/en/you-share-we-take-care>**

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

# An Integrated Framework for Autonomous Sensor Placement With Aerial Robots

Brett Stephens , *Graduate Student Member, IEEE*, Hai-Nguyen Nguyen , *Member, IEEE*, Salua Hamaza , and Mirko Kovac , *Member, IEEE*

**Abstract**—Aerial manipulators have the unique ability to cover wide-spread areas within a single mission, making them ideal for the transport and placement of sensors required to build an instrumented environment. Recent work in the field has focused on controllers for aerial interaction that account for compliance during contact-based tasks, omitting integration concerns that are critical to an automated solution. Furthermore, state-of-the-art flying base manipulators are often mechanically and computationally complex, reducing their endurance. Within this work, we present an interactive framework for autonomous sensor placement that incorporates both mechanical and software based compliance, optimised for use on a simple coplanar quadrotor. Under appropriate actuation and perception constraints, we detail the development of a control, perception, and motion planning strategy to enable sensor placement that relies solely on onboard computation and sensing, thus presenting a fully contained and accessible sensor placement approach capable of robust interaction with the environment. An extended finite-state machine is developed to facilitate automated mission planning. Extensive flight experiments are performed to validate the effectiveness of each sub-system, as well as the integrated solution. Experiments result in trajectory tracking errors under 10 mm as well as onboard mass estimation errors under 0.7% for sensors of various weights. A statistical analysis

of 162 flight experiments shows the proposed framework's ability to autonomously place sensors within 10 cm of the target with a success rate of 93.8% and 95% confidence interval of (89%, 97%), thus confirming the robustness of our approach.<sup>1</sup>

**Index Terms**—Aerial robotics, autonomous sensor placement, distributed sensor networks, unmanned aerial vehicles (UAVs).

## I. INTRODUCTION

UNMANNED aerial vehicles (UAVs) are frequently used for contactless operations that exploit visual sensing for geospatial mapping, gauging infrastructure health and security monitoring. Due to their vertical take-off and landing abilities, versatility, and accessibility, UAVs have become ubiquitous in the natural and industrial world. In more recent years, UAVs have been deployed to physically interact with the environment for tasks such as object transportation [1], [2], [3], [4], assembly tasks [5], [6], contact-based interaction [7], [8] and inspection [9], [10], valve turning [11], and marking of surfaces [12], [13]. These interactive concepts and capabilities are poised to rapidly expand the utility of UAVs for operation and maintenance, allowing for the efficiency and safety benefits of automation to be implemented within a vast new realm of applications, some of which are depicted in Fig. 1.

Within this work, we present the following contributions from our system-level, low-complexity approach to the sensor placement problem: 1) a passively adaptive manipulator design for the coplanar quadrotor, 2) the foundations of an autonomy framework composed of a admittance controller and optimized trajectory generator, and 3) an extensive set of flight experiments culminating in a statistical and failure mode analysis that validates the robustness of our proposed solution for 162 trials without reliance on external localization infrastructure.

The first contribution enables our simple aerial platform to place a sensor with a single passive degree of freedom (DoF) while allowing for deviations in target surface orientation. The second contribution focuses on producing repeatable, accurate, and precise sensor delivery in a variety of contexts. Enabled via a synergistic sensor suite, the system's state estimation camera, subject to drift errors, is paired with a depth sensor outputting absolute measurements relative to the manipulator. A bespoke

Manuscript received 8 November 2021; revised 2 June 2022; accepted 4 August 2022. Date of publication 6 October 2022; date of current version 16 February 2023. Recommended by Technical Editor Savvas G. Loizou and Senior Editor Huijun Gao. This work was supported in part by EPSRC under Award EP/R026173/1, in part by Aerial ABM under Award EP/N018494/1, in part by ORCA HUB under Award EP/R026173/1, in part by CASCADE under Award EP/R009953/1, in part by EU H2020 AeroTwin Project under Grant 810321, and in part by the Royal Society Wolfson fellowship under Grant RSWF/R1/18003. (Brett Stephens and Hai-Nguyen Nguyen contributed equally to this work.) (Corresponding author: Brett Stephens.)

Brett Stephens and Mirko Kovac are with the Aerial Robotics Laboratory, Department of Aeronautics, Imperial College London, SW7 2BU London, U.K., and also with the Swiss Federal Laboratories for Materials Science and Technology (EMPA), 8600 Dübendorf, Switzerland (e-mail: brett.stephens18@imperial.ac.uk, bsbretly@gmail.com; m.kovac@imperial.ac.uk).

Hai-Nguyen Nguyen was with the Aerial Robotics Laboratory, Imperial College London, SW7 2BU London, U.K. He is now with the LAAS-CNRS, Université de Toulouse, CNRS, 31400 Toulouse, France (e-mail: hann@ieee.org).

Salua Hamaza was with the Aerial Robotics Laboratory, Imperial College London, SW7 2BU London, U.K. She is now with the BioMorphic Intelligence Laboratory, Faculty of Aerospace Engineering, Department Control and Operations, Mekelweg 5, 2628 CD Delft, The Netherlands (e-mail: s.hamaza@tudelft.nl).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TMECH.2022.3202116>.

Digital Object Identifier 10.1109/TMECH.2022.3202116

<sup>1</sup>A video showcasing our platform can be found here: <https://youtu.be/4R8DhVpEbSQ>.



Fig. 1. Example scenarios for the proposed framework. (A) Enabling distributed sensor networks in forests for ecology monitoring. (B) Contact-based inspection of fiberglass wind turbine blades offshore. (C) Nondestructive testing of nuclear plant cooling towers.

and flexible navigation strategy for approaching an identified target optimizes the robot's motion to preserve actuation and sensing constraints. Such motion generation is paired with a suitable admittance control strategy and is governed by an extended finite-state machine (EFSM) to enable autonomous behavior.

The third contribution is a test on the validity of our proposed approach via flight experiments. The first set of experiments illustrates the controller's ability to respond to external forces in a compliant and intuitive way as well as accurately track trajectories, both in free-flight and in contact. The onboard force estimation strategy is then tested for its ability to identify when a sensor has been successfully deployed on a target surface. Lastly, for 162 sensor placement experiments, we present a rigorous failure mode analysis and statistical results for onboard computation times and sensor placement position metrics. Proposed strategies to mitigate failure are subsequently presented.

A portion of this article and the preliminary experimental results are summarized in the review workshop article [14]. In the work here, we generalize and complete the results from our previous work. We first expand upon the perception, control, and planning methodologies and then proceed with designing an integrated framework to ensure that the aerial manipulator is able to perform the sensor placement task robustly. New experimental results illustrate the effectiveness of our novel approach.

## II. RELATED WORK

The main challenges associated with aerial physical interaction include platform stability, force estimation, and control at the manipulator end-effector, accuracy, and precision during interaction and total system robustness. To date, research groups have tackled certain aspects of these challenges through a variety of methodologies. The work presented in [15] analyzes the behavior of helicopters in physical interaction and shows that compliant contact significantly aids flight stability. In [16], it is shown that the underactuation property of quadrotors can trigger undesired nonlinear behavior during interaction, such as *finite-time escape*.

Force generation at the tool-tip can be accomplished through various routes. The work in [17] employs the use of a tilted-rotor UAV with a fixed tool to deploy full-wrench (independent application of force and torque) capabilities in six dimensions. In [18], the force exerted at the tool-tip results from the coupling between the vehicle's rotational and translational dynamics by allocating part of the system's thrust as output force on a wall.

Force feedback is provided by a six-axis force/torque sensor mounted on the target surface, which gets mapped into an attitude command in the low-level attitude controller. A hybrid force/motion controller is implemented in [19] in order to allow a quadrotor to perch to nonuniform objects using a novel grapple tool-tip, highlighting the benefits of leveraging force generation to increase task robustness.

A fully actuated hexarotor platform for contact-based inspection is introduced in [20], [21], and [22], allowing the aerial manipulator base to achieve and maintain any desired pose while inspecting variable geometry surfaces with the use of an impedance-based controller. Such capabilities come at the cost of additional mechanical complexity, however, manifesting in [20] as a system mass that is  $2.5\times$  that of the simple quadrotor platform discussed here and the reliance on offboard power in [21] and [22]. The experiments performed in our work aim to investigate whether such additional complexity is required for the sensor placement task, and introduces new methods to approach the flying interactive aerial manipulator paradigm with an underactuated flying base.

While the underactuated quadrotor represents a relatively simple platform from a mechanical perspective, complexities are introduced due to the coupling of the system's dynamics (e.g., the quadrotor must pitch to translate forwards). As shown in [23], [24], and [25], such coupling hinders the field of view (FOV) of the robot when navigating its environment. Our work takes inspiration from such perception-aware approaches to produce a flexible interactive control method that does not require model parameters to be edited for specific scenarios.

Looking toward the specific case of aerial sensor placement, there are several examples that hint at the potential future direction of such an application. The first example is presented in [26] and [27], where a tricopter is constructed with an additional propeller mounted horizontally on the vehicle's frame, allowing the vehicle to produce thrust orthogonally to a vertical surface. This simple approach to sensor placement has its drawbacks; however, in that, automation capabilities are limited: reducing human-controlled sensor placement position error from a *minimum* of 12 to 7 cm. It can be seen in the statistical analysis section that our *minimum* position errors are much reduced.

A second example is presented in the works [28] and [29], where sensor placement and retrieval operations are performed using a quadcopter on both 2-D and 3-D surfaces. Force generation at the end effector is achieved through the combined action of the vehicle's pitching motion and a 1-DoF

actuated manipulator, the torque of which is controlled via a PID scheme. Results show robustness and reliability both on vertical walls and tree trunks. With such an approach, the platform has no direct control on the interaction force, relying upon inferences from the vehicle's actuated manipulator and attitude (pitch angle). Such a strategy introduces the potential for error when defining required inferences and time delays in arriving at estimated forces. Additionally, the system lacks any state estimation strategy (vision-based or otherwise) and therefore relies upon manual maneuvering to complete its task. Lastly, the open-loop attitude control of the system leads to instability when rendering high output forces at high pitch angles, reducing its applicability.

Aerial sensor placement by means of projectile launch of sensors is explored in [30], which focuses on the design and integration of a new launching mechanism on a quadrotor. However, a suitable perception, navigation, and control frameworks for such a platform are not addressed.

Based on the aforementioned efforts, this article proposes a new framework that offers a comprehensive solution to the autonomous sensor placement problem. We consider all aspects of the autonomy including perception, motion planning, and interactive control, enabling the integration of these domains with the use of onboard sensors and computation.

### III. SENSOR PLACEMENT

#### A. Task Summary

The challenge presented here is to autonomously place a sensor in a robust, accurate, and precise manner without reliance on external instrumentation. In completing such a mission, the following assumptions are made. 1) There are no obstacles present that obstruct the vehicle's motion, other than the sensor placement target itself. Such an assumption will be relaxed in future work. 2) Aerodynamic external disturbances, e.g. wind, are neglected.

The mission is executed autonomously by completing the following sequential procedures: the sensor placement surface is detected with the use of an onboard depth camera. In the case of the experiments presented in this article, a vertical plane (relative to a horizontal ground plane) is defined as the desired target surface. Targets that deviate from vertical within a  $\pm 15^\circ$  window are considered feasible due to the range of motion enabled by the tool-tip design and target detection surface normal calculation (see Section IV for details). With the placement surface identified, the desired target location is determined, e.g., at the center of the identified placement surface area.

With the target in the FOV of the robot, an optimal trajectory is generated to guide the vehicle to the target location, maintaining the target location within the FOV of the vehicle's onboard tracking camera, while obeying specific constraints imposed by the platform and the scenario. Contact with the target is sensed via onboard force estimation in the direction of travel, allowing a magnetic sensor to be robustly adhered to the target. Once robust placement is ensured, the quadrotor is commanded to return to its initial take-off location.

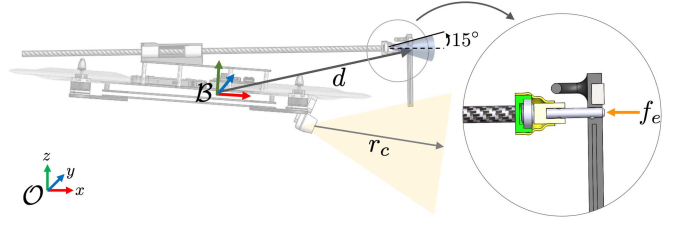


Fig. 2. Aerial system reference frames and relevant parameters.

#### B. Modeling

Let us consider a quadrotor moving in 3-D space with a tool attached at the tip as illustrated in Fig. 2. We define an inertial frame  $\mathcal{O}$ , and a body-fixed frame  $\mathcal{B}$  centered in the aircraft center-of-mass (CoM). The state vector of the vehicle in  $\mathcal{O}$  is described by  $\xi = [p, \eta]^T$  with  $p = [x, y, z]^T$  and  $\eta = [\phi, \theta, \psi]^T$  describing the vehicle CoM in the  $x, y, z$ -directions and roll, pitch, and yaw orientations, respectively. The kinematic relation between  $p$  and the tool-tip position  $p_t \in \mathbb{R}^3$ , expressed in the inertial frame  $\mathcal{O}$ , is given by

$$p_t = p + Rd \quad (1)$$

where  $R \in SO(3)$  represents the rotation of the body-fixed frame  $\mathcal{B}$  w.r.t. the inertial-frame  $\mathcal{O}$ , and  $d \in \mathbb{R}^3$  is the location of the tool-tip in frame  $\mathcal{B}$  and can be measured manually. Changes in  $d$  upon target contact are small and compensated for in our feedback control, an assumption which is validated in Section V. Therefore, by knowing  $p$ , we can control the tool-tip position  $p_t$ . It is to be noted that the sensor placement problem only requires *point-to-point* control, which is fundamentally different from the previous work presented in [16].

To control the translation and rotation of the quadrotor, we can use the well-known dynamics of the quadrotor:

$$m\ddot{p} = \lambda R e_3 - m g e_3 + f_e \quad (2)$$

$$J\dot{\omega} + \omega \times J\omega = \tau + \tau_e, \dot{R} = RS(\omega) \quad (3)$$

where  $m \in \mathbb{R}$ ,  $J \in \mathbb{R}^{3 \times 3}$  are the inertia parameters,  $\omega \in \mathbb{R}^3$  is the angular velocity of frame  $\mathcal{B}$  relative to  $\mathcal{O}$  represented in  $\mathcal{B}$ ,  $\lambda \in \mathbb{R}_+$ ,  $\tau \in \mathbb{R}^3$  are the thrust and torque input of the quadrotor, respectively,  $f_e, \tau_e \in \mathbb{R}^3$  are the external force and torque represented in  $\mathcal{O}$  and  $\mathcal{B}$ , respectively,  $g$  is the scalar gravitational acceleration constant,  $e_3 = [0, 0, 1]^T$  is the basis vector specifying the down direction in  $\mathcal{O}$ , and

$$S(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad \text{s.t.} \quad S(\omega)\nu = \omega \times \nu$$

for any  $\nu \in \mathbb{R}^3$ .

Similar to [16], the tool mechanism is designed to be light weight as to maintain a total system CoM within the body of the quadrotor. The CoM offset introduced by the inclusion of the tool is effectively suppressed by the feedback and feed-forward control, demonstrated in the experimental implementation. Moreover, it is assumed that the tool-tip is small enough so that the

interaction is reduced to a single point contact. This generates a disturbance moment  $\tau_e$  that is only proportional to the Cartesian force  $f_e$

$$\tau_e = d \times R^T f_e.$$

Such an assumption is reflected in the mechanical design presented in Section IV and allows for the direct coupling of the interaction effects with the vehicle dynamics and attitude controller.

#### IV. SENSOR PLACEMENT FRAMEWORK

##### A. Manipulator Design

The manipulator developed here is composed of a carbon fiber tubular rod encapsulated within a custom mount, allowing the rod one passive linear DoF in the direction that runs along the center of the length of the rod. Two igus drylin bearings ensure smoothness in the linear motion of the tool inside the case. A rubber hard stop is placed at the free end of the tool to prevent the rod from separating from its mount. The end effector consists of a compliant quick-release mechanism: on one end of the rod, a spherical bushing is held in place by a small pin. The spherical element allows for small angular rotations about its center, allowing the sensor to be placed on target surfaces which deviate from vertical within a  $\pm 15^\circ$  cone. Inside the spherical bushing, a quick-release pin is rigidly attached on one end. The free end of the pin has a spring-loaded ball, which is press fit inside a hole in the sensor case. When the sensor body is subjected to a force opposing the end-effector, the pin will slide out of the hole due to the force of the spring pushing the ball to the periphery. To provide additional compliance at the time of contact, and to compensate for the gravity effect on the spherical bushing, a rubber membrane encloses the entire mechanism. To adhere the sensor on the target surface, a neodymium magnet is slotted inside the sensor case, providing the required pulling force to release the sensor from the end-effector when in contact with a ferrous surface. A schematic of the system is illustrated in Fig. 2.

##### B. Perception

1) *State Estimation*: The sensor placement platform utilizes an onboard tracking camera for full state estimates. The camera's state estimate is fused with the flight controller's inertial measurement unit (IMU) and magnetometer via an extended Kalman filter to output a final pose estimate. For our system, an external motion tracking system, Vicon, was used only as a ground truth state estimate.

2) *Target Detection*: The desired placement target is extracted from a 3-D collection of points in space, or point cloud. Such points represent an absolute measurement relative to the drone and are therefore not subject to the accumulation of prior estimation error, a fact that does not hold for visual-based state estimation strategies, such as visual inertial odometry (VIO). As such, the framework offers the capability of correcting for error accumulation (i.e., drift), a critical feature when traversing large distances. The target detection algorithm outputs a target vector

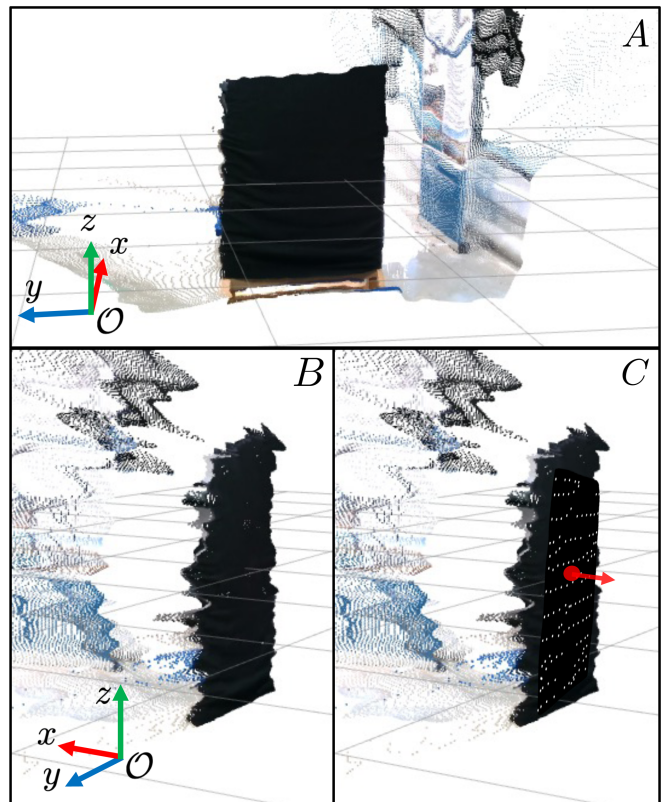


Fig. 3. Vertical black target surface is visualized as: (A) color 3-D point cloud from the viewpoint of the approaching quadrotor, (B) same 3-D point cloud as viewed from a side perspective, and (C) output of the point cloud planar segmentation algorithm visualized as a collection of small white circular points with the final desired tool-tip target  $y_d$  and normal  $\hat{n}_d$  as a large red point and arrow, respectively, superimposed on the target surface.

through the completion of two steps: 1) segmenting the incoming point cloud based on predefined properties and 2) computing normal vectors to the segmented collection of points. Each of these steps is visualized in Fig. 3.

In this article, we assume the prior knowledge of a planar target surface and, as such, complete the first step by segmenting the incoming point cloud via a 3-D  $(x, y, z)$  planar model parameterized by  $a$ ,  $b$ , and  $c$  in the form  $ax + by + cz + d = 0$ . To implement this segmentation, we use the random sample consensus (RANSAC) outlier rejection algorithm [31], chosen for its robustness and relative simplicity.

We then complete the second step of our target detection approach: computation of point normals to gain further insight into the target's geometry. The process is completed by determining the normal of a plane tangent to a given surface, implemented here as a  $k$ -nearest neighbor least-square plane fitting estimation problem for each of our segmented points from step 1 [32]. A search radius is defined a priori in order to determine the number of neighboring points. It is noted that the methodology described in these two steps can be expanded to generate targets on objects of various parameterizations and geometries.

With the above two steps completed, a target point on the planar target surface is defined as the mean of all segmented

points, with the target normal bearing as the mean of all computed point normals. The target point location and normal is then transformed into the inertial frame  $\mathcal{O}$ , generating  $p_{t,d} \in \mathbb{R}^3$  as the desired tool-tip target point and  $\hat{n}_d \in \mathbb{R}^3$  as the desired target normal. Target surface feasibility is determined a priori and is a function of the system's manipulator design: a planar target with an orientation  $\pm 15^\circ$  from vertical (in the inertial frame  $\mathcal{O}$ ) is considered feasible due to the sensor orientation maneuverability afforded to us by the tool-tip's spherical bushing.

### C. Admittance Control

Due to the coupling between the rotational and the translational dynamics of the quadrotor, direct control of the tool-tip position is challenging and requires more complex low-level attitude control (torque input), previously published in [16]. In this article, we make use of a high-level attitude controller (thrust vector input) to accomplish point-to-point motion, thus reducing the complexity and increasing the robustness of the approach.

Translational trajectory tracking is achieved with the use of a cascade control architecture: A desired thrust vector is decomposed into desired attitude commands, which in turn become the input to a low-level attitude controller. Such a strategy is implemented as follows: The desired thrust vector is defined as  $\Lambda := \lambda R e_3 \in \mathbb{R}^3$ . The control of the translational dynamics can be derived as follows:

$$\Lambda = m g e_3 + m \ddot{p}_d - k_b(\dot{p} - \dot{p}_d) - k_p(p - p_d) \quad (4)$$

with  $p_d(t) \in \mathbb{R}^3$  being the desired trajectory. Without disturbances and uncertainty, the errors in (4) will exponentially converge to zero as desired [33], Proposition 4.10).

The desired thrust is decoded into attitude commands following [34], with the thrust command computed directly as  $\lambda = \|\Lambda\|$ . To compute desired yaw, pitch, and roll angles  $[\psi_d; \theta_d; \phi_d] \in \mathbb{R}^3$ , we use the parameterized rotation matrix  $R = R_{e_3}(\psi)R_{e_2}(\theta)R_{e_1}(\phi)$  with  $R_{e_i}(\star)$  being the elementary rotation matrix about the  $e_i$ -axis. The yaw angle  $\psi_d$  can be chosen arbitrarily. For example, we can choose a yaw command to ensure that the tool is always perpendicular to the target surface. The roll and pitch commands can then be determined using the following relation:

$$R_{e_2}(\theta)R_{e_1}(\phi)e_3 = \begin{bmatrix} \sin \theta \cos \phi \\ -\sin \phi \\ \cos \theta \cos \phi \end{bmatrix} = \hat{\Lambda}(\psi)$$

with  $\hat{\Lambda}(\psi) := \frac{1}{\lambda} R_{e_3}^T(\psi)\Lambda$ . We can then choose the roll and pitch commands as

$$\phi_d = -\sin^{-1} \hat{\Lambda}_2, \quad \theta_d = \tan^{-1} \frac{\hat{\Lambda}_1}{\hat{\Lambda}_3}$$

with  $\hat{\Lambda}_i$  being the  $i$ th element of  $\hat{\Lambda}$ .

For a given desired trajectory  $p_d(t)$ , we now can compute the desired attitude  $R_d$ .

Interactions between the flying robot and environment are addressed through an admittance controller. Such an approach

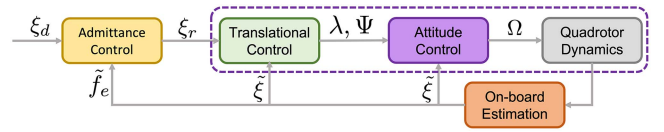


Fig. 4. Control input to the lower level attitude controller is defined as a thrust scalar and attitude command, represented by  $\lambda$  and  $\Psi = [\phi_d, \theta_d, \psi_d]^T$ , respectively. The modular nature of our control strategy is illustrated with the admittance block plugging into the existing cascade control architecture, framed here with a dashed box. Variable estimates are designated under the symbol.

has the advantage of being seamlessly incorporated into our existing motion control loop in a modular fashion as seen in Fig. 4. Desired attitude commands are relayed to an attitude controller implemented on the open source flight control software PX4, which allocates control authority amongst the four motor velocities:  $\omega_{m_i} \in \mathbb{R}$  to the  $i$ th motor:  $\Omega = (\omega_{m_1}, \omega_{m_2}, \omega_{m_3}, \omega_{m_4})$ .

In this article, we assume that the interaction is point-contact. We can then form the desired virtual dynamical system as follows:

$$m_d \ddot{e}_r + b_d \dot{e}_r + k_d e_r = f_e \quad (5)$$

where  $e_r := p_d - p_r$ , with  $p_r(t) \in \mathbb{R}^3$  being the generated reference trajectory due to external force  $f_e$  and  $m_d, b_d, k_d$  are the desired virtual inertia, damping, and spring constants, respectively.

The external force  $f_e$  is estimated using onboard sensors following the acceleration-based estimation technique outlined in [35]. As shown in Section V, we found this strategy adequate for our interactive control tasks. Note that the influence of wind, which is unavoidable in real applications, can be identified and compensated [36]. In this article, our focus is on the interaction force generated during contact with the target.

### D. Trajectory Generation

As seen in Fig. 5, the automation of the sensor placement motion is enabled by user-defined time thresholds and sensed triggers. As a fail-safe, each state defaults (e.g., when no external force is sensed within a user-defined time threshold) to the *ReturnHome* phase.

We develop here a time-optimal strategy to approach our target. In completing such motion, it is advantageous for the target to remain within the FOV of the camera. Due to the under-actuated nature of the quadrotor, translational and rotational motions are coupled, resulting in movement of the camera FOV as a trajectory is traversed. Such a phenomenon may lead to the target moving out of the FOV of the camera [23]. Maintaining view of the target allows for the absolute (relative to the quadrotor) target measurement to be regularly updated as the vehicle traverses its trajectory. The benefit of this strategy becomes especially useful when navigating large distances to a sensed target. In such cases, the desired end point can be updated as frequently as desired along the approaching path, mitigating error accumulation due to prior VIO estimation drift.

The kinematic constraint (1) suggests that we utilize the translation motion  $p$  and rotation  $R$  to control the tool-tip position.

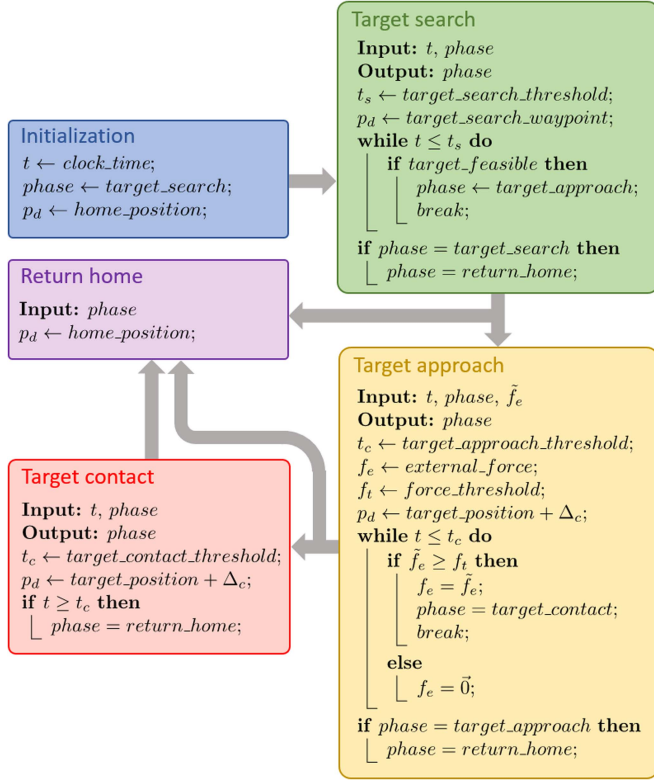


Fig. 5. EFSM logic, enabling automated transitions between sensor placement motion phases.

Our objective, therefore, is to find the desired motion  $p_d(t)$  and  $R_d(t)$  to drive the tool-tip position  $p_t(t)$  from the start location  $p_t(t_0)$  to the target location  $p_t(t_f) = p_{t,d}$  for sensor placement. Here, we address the trajectory generation problem considering constraints in the following domains: the dynamics of the robot, actuation limitations (i.e., maximum available thrust), and sensing (i.e., the camera FOV).

The following discussion, unless otherwise mentioned, is with respect to the inertial frame  $\mathcal{O}$ . We utilize Bernstein polynomials (i.e., Beziér curves) to parameterize the trajectory of the robot, optimized under the aforementioned constraints. The Bernstein polynomials were chosen for their convenient geometric properties [37] and it has been shown to be a favorable tool to approximate nonlinear optimal control problems [38].

We parameterize the trajectory  $p(t)$  of the robot over the interval  $[t_0, t_f]$  using an  $n$ th-order Bernstein polynomial  $p_\beta(t)$  as follows:

$$p(t) = p_\beta(t) = \sum_{i=0}^n \beta_{i,n} b_{i,n}(t), \quad t \in [t_0, t_f]$$

where  $\beta_{i,n} \in \mathbb{R}^3$  is the  $i$ th Bernstein coefficient (i.e., control points in the Beziér curves), and  $b_{i,n}$  is the basis Bernstein polynomial defined as

$$b_{i,n} = \frac{n!}{i!(n-i)!} \frac{(t-t_0)^i (t_f-t)^{n-i}}{(t_f-t_0)^n}, \quad i = 0, \dots, n.$$

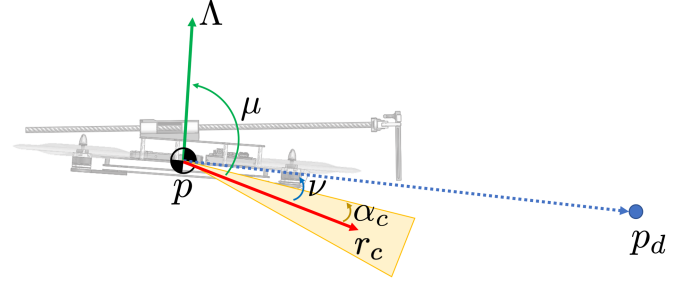


Fig. 6. Schematic of the sensing constraints imposed by the camera FOV.

Next we detail all constraints considered in this trajectory generation problem along with the relevant Bernstein polynomial properties leveraged for parameterization.

**Actuation constraint:** The quadrotor dynamics from (2) defines a constraint between the translation and rotation of the quadrotor (i.e., a nonholonomic constraint):

$$\Lambda(t) = \lambda \text{Re}_3 = mg(\ddot{p}_\beta(t) + e_3). \quad (6)$$

Looking at the rightmost side of (6), we invoke the *derivatives* property for  $\ddot{p}_n(t)$  in conjunction with the *degree elevation* and *arithmetic operations* for the addition of  $e_3$ , thus allowing us to approximate the thrust vector  $\Lambda(t)$  as a 3-D Bernstein polynomial.

The quadrotor's thrust bound can be formulated as  $\|\Lambda\| \leq \bar{\lambda}$ , where  $\bar{\lambda}$  is the maximum available thrust. By squaring both sides of the aforementioned thrust bound inequality, we can form the following relation:

$$\Lambda^T \Lambda = m^2 g^2 (\ddot{p}_\beta(t) + e_3)^T (\ddot{p}_\beta(t) + e_3) \leq \bar{\lambda}^2. \quad (7)$$

With the thrust vector parameterized as a Bernstein polynomial [i.e., (6)], we again invoke the *arithmetic operations* property in stating that the left-hand side of the inequality in (7) is also a 1-D Bernstein polynomial.

**Sensing constraint:** Consider the geometry of the robot with the front-view camera as shown in Fig. 6. We assume the distance from the quadrotor CoM to camera center to be small and negligible. While traversing the trajectory, we would like to maintain the target position  $p_d$  within the camera FOV. In order to facilitate this, we develop a relation between the thrust vector  $\Lambda$  and the camera center vector  $r_c \in \mathbb{R}^3$  as

$$r_c = R_\Lambda(\mu) \Lambda \quad (8)$$

where  $\mu$  is the constant angle between the two vectors and  $R_\Lambda(\cdot)$  is the rotation matrix required to rotate the thrust vector  $(\cdot)$  degrees about the body  $y$ -axis. Using the Bernstein polynomial property of *arithmetic operations*, it becomes clear that  $r_c$  is itself a 3-D Bernstein polynomial. We can then take the dot product of the vectors  $r_c$  and  $(p_d - p)$

$$r_c \cdot (p_d - p) = \cos(\nu) \|r_c\| \|p_d - p\|. \quad (9)$$

It can be seen in Fig. 6 that in order for the target point  $p_d$  to remain within the FOV of the camera, the following inequality must hold:  $\nu \leq \alpha_c$ , with  $2\alpha_c$  being the camera view angle. For small enough view angles, we can then arrive at the following



constraint:  $\cos^2 \nu \geq \cos^2 \alpha_c$ . By combining this constraint into (9) and squaring both sides, we arrive at

$$r_c^T(p_d - p_\beta) \cdot r_c^T(p_d - p_\beta) \geq \cos^2(\alpha_c) \|r_c\|^2 \|p_d - p_\beta\|^2. \quad (10)$$

Lastly, (8) can be substituted into (10) in order to have a direct relation for thrust. Again using the Bernstein property of *arithmetic operations*, it becomes clear that (10) itself takes the form of a Bernstein polynomial. Note here that, to utilize a Bernstein polynomial, we use a constraint on  $\cos^2 \nu$  instead of  $\nu$  itself.

**Task requirements:** The tool-tip's final position is defined as  $p_t(t_f) = p_{t,d}$  with the requirement that the approaching motion must be perpendicular to the target surface. These requirements can be formulated as

$$\begin{aligned} p_t(t_f) &= p_{t,d}, \\ n^T R(t_f) e_2 &= 0, \quad n^T \Lambda(t_f) = 0, \end{aligned} \quad (11)$$

where  $p_{t,d} \in \mathbb{R}^3$  is the desired target location and  $n \in S^3$  is the direction normal to the surface, both expressed in the inertial frame  $\mathcal{O}$ , with  $e_i \in \mathbb{R}^3$  being the  $i$ th standard basis vector.

The examination of (11) suggests that at the specified end time  $t_f$ , we can compute the desired CoM position  $p_d(t_f)$  and the rotation  $R_d(t_f)$  of the quadrotor. For a given  $p_d, n$ , there is a family of solutions with different desired roll angles. We then choose  $\phi_d(t_f) = 0$  and solve (11) for an unique  $p_d(t_f), R_d(t_f)$ , which can then be set as boundary conditions for the trajectory generation.

**Time-optimal trajectory:** To find the time-optimal trajectory while respecting the actuation constraint, sensing constraint and also the task requirements, we formulate trajectory generation problem as the following optimization problem:

$$\begin{aligned} \min_{\beta_{i,n}, t_f} \quad & t_f \\ \text{subject to} \quad & p_\beta(t_0) = p(t_0), \quad p_\beta(t_f) = p_d(t_f), \\ & \dot{p}_\beta(t_0) = \dot{p}(t_0), \quad \dot{p}_\beta(t_f) = \dot{p}(t_f), \\ & n^T m g (\ddot{p}_\beta + e_3) = n^T R_d(t_f), \\ & \text{equations (7) and (10)}. \end{aligned}$$

Note here that the coupling between longitudinal ( $x$ -translation) translation and pitch  $\theta$  in addition to changes in the quadrotor's altitude ( $z$ -translation) will affect the location of the target within the camera's FOV. Motion in the  $(x, y)$  plane and roll  $\phi$  motion, however, will not affect the target's location within the FOV. The yaw motion, which can be controlled arbitrarily, is utilized to compensate for view of the object in the  $(x, y)$  plane. The above optimization then can be reduced to the  $x, z$  dimensions, i.e., by using  $p(t) = [x(t) \ z(t)]^T \in \mathbb{R}^2$ .

## V. EXPERIMENTS AND SIMULATIONS

### A. System Architecture

The aerial platform developed for each of the following experiments is based on the Lumenier QAV400 quadrotor, with a Pixhawk Pixracer flight controller unit (FCU) as seen in

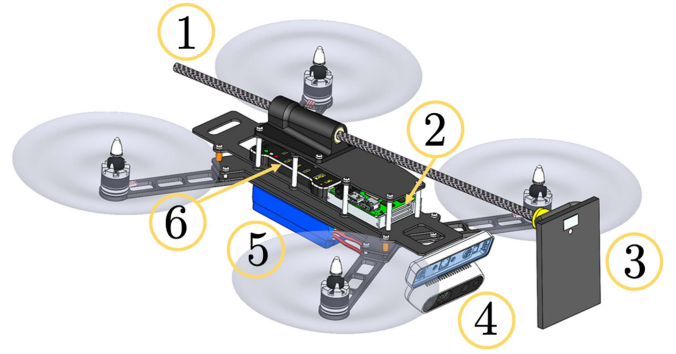


Fig. 7. Lumenier QAV400 quadrotor CAD schematic. Moving clockwise: 1) 1-DoF tool-tip, 2) Intel NUC i7, 3) magnetic sensor, 4) Intel T265 (top) and D435i (bottom), 5) battery, and 6) Pixhawk Pixracer FCU.

Fig. 7. The onboard computer is the Intel NUC i7<sup>2</sup>, performing perception, control, and motion planning schemes within the Robot Operating System suite. Two Intel vision-based sensors are mounted forward facing at the front of the vehicle for depth sensing and visual odometry estimates, the RealSense D435i and T265, respectively. Onboard electronics are powered by a 4-s 3700 mAh battery, with a maximum flight time of approximately 20 min. The passive manipulator is attached on the top plate of the quadrotor and extends 35 cm in the forward direction of travel. The sensor houses a magnet and weighs 56 g. The placement mechanism weighs 90 g, contributing to less than 5% of the total mass. The system has a total mass of 1.9 kg.

### B. Admittance Control Performance

To examine the motion performance and stability of the robot during interaction, we disturb the system with an unknown external force. In Fig. 8, the vehicle is exposed to such a force by manually perturbing the quadrotor via a cable attached to its CoM. The response of the admittance controller is predicated by the desired virtual parameters  $(m_d, b_d, k_d)$ : for low-stiffness parameters, e.g.,  $k_d = 5$ , the vehicle follows the cable force closely and for critical damping,  $m_d = 1, b_d = 10$ , and  $k_d = 25$  the system returns to the desired motion quickly when the cable is released.

Note here that since the translation and rotation of the quadrotor are coupled, we can manipulate the virtual compliance of a physical interaction via the translational or rotational dynamics. We chose the translation layer after observation of the desirable system response to external perturbations, namely the ability to shape the interactive behavior of the quadrotor in accordance with the properties (e.g., stiffness) of a target surface.

### C. State Estimation and Trajectory Tracking Performance

A graphic of the system performing the sensor placement sequence is illustrated in Fig. 9. The placement pipeline is segmented into four core phases. Each phase is distinguished in Fig. 10, overlaid with the system's position.

<sup>2</sup>Early prototypes are equipped with Intel UP Core Atom.

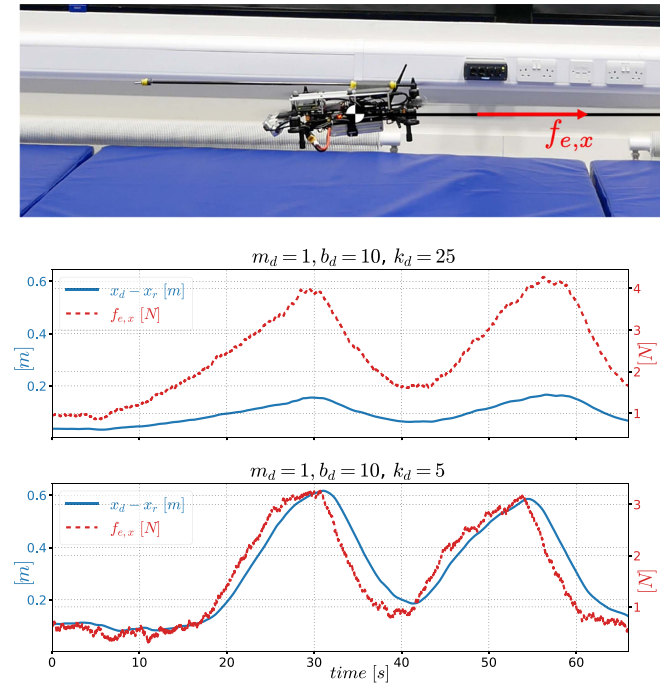


Fig. 8. Response of the admittance controller with low-stiffness parameters  $m_d, b_d, k_d = (1, 10, 5)$  and critical damping  $(1, 10, 25)$ . The external disturbance is induced by manually pulling a cable attached to the vehicle's CoM.

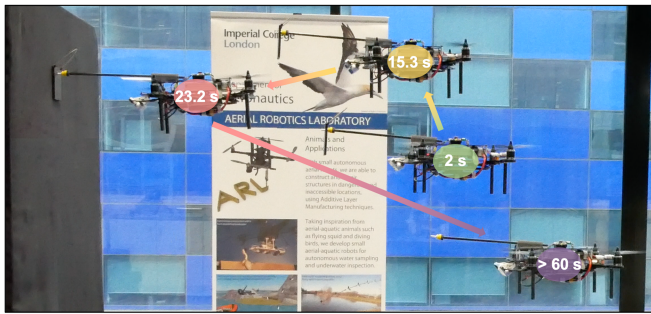
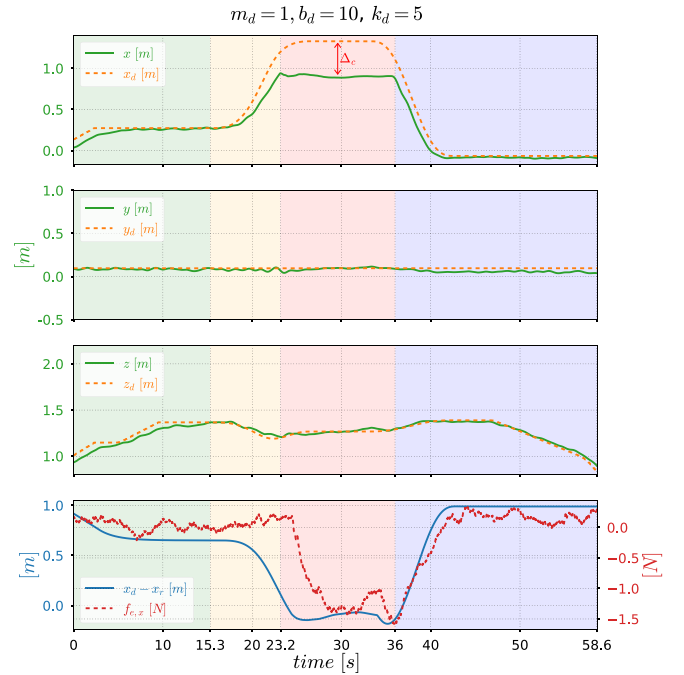


Fig. 9. Example sequence of the autonomous sensor placement task: the vertical placement target surface can be seen on the far left of the image.

1) **Target Search:** Moving forward and up in the  $x$ - and  $z$ -directions, respectively, the vehicle searches for the vertical target surface. Potential targets run through a validation step in order to ensure motion planning feasibility: the target must be forward of the front of the robot (a positive body-frame  $x$ -coordinate) and the target normal must be orthogonal ( $\pm 15^\circ$ ) to the robot's tool-tip.

2) **Target Approach:** Once the target surface is detected at 15.3 s, a feasible trajectory, considering our platform's actuation constraints and camera FOV, is generated following the methodology described in Section IV-D toward the target location. In practice, a target position is set at a "buffer" distance behind the target ( $\Delta_c$  in Fig. 10) in order to ensure that the vehicle leverages the admittance controller to establish consistent contact with the target surface.



Trajectory Tracking Mean Squared Error [cm]			
	x	y	z
MSE	0.845	0.093	0.129

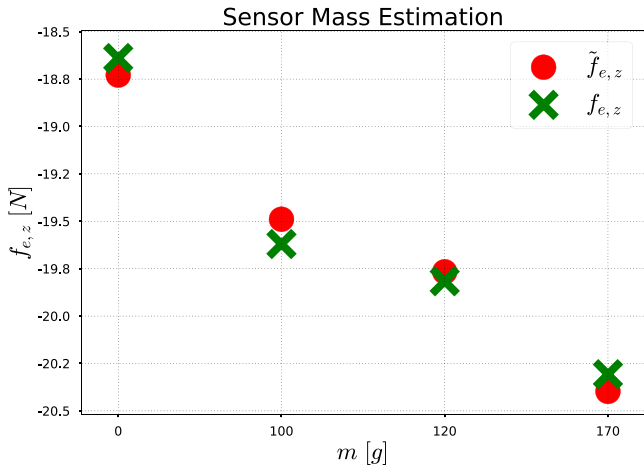
Fig. 10. Trajectory tracking performance and force estimation for the sensor placement task with desired virtual parameters  $m_d, b_d, k_d = (1, 10, 5)$ .

3) **Target Contact:** At 23.2 s, the vehicle makes contact with the vertical target surface. The contact phase is recognized via force thresholding in the direction of travel: the  $x$ -direction in this example. If such a force exceeds a predefined empirically derived threshold value, the admittance controller is engaged in order to ensure a virtually "soft" interaction with the target surface. In a similar fashion to the force threshold, a contact-phase time threshold value is defined in order to ensure that the sensor has been secured to the target. In this experiment, the sensor placement contact phase persists until 36 s.

4) **Return Home:** Once the contact-phase time has expired, the robot retreats from the target and returns to home, defined here as the vehicle's take-off position. The retreat phase spans until the robot is safely landed at final time  $T: 36 \text{ s} > t > T$ .

The force estimation subplot included in Fig. 10 facilitates a comparison to the admittance controller performance experiment. It can be clearly seen that in both experiments, the reference position deviation  $e_{r,x} = x_d - x_r$  and external perturbation  $f_{e,x}$  both follow similar shapes as desired. It is worth noting, however, that a discrepancy between experiments can be noticed in what appears as a delay in  $f_{e,x}$  relative to  $e_{r,x}$  during the *target approach* phase.

Such a phenomenon is due to the difference in experimental procedures: In the admittance controller performance experiment, the vehicle was commanded to hover at a fixed position



Sensor Mass Estimation Error			
0g	100g	120g	170g
0.48%	0.67%	0.25%	0.45%

Fig. 11. Expected,  $f_{e,z}$ , and estimated,  $\tilde{f}_{e,z}$ , external forces are plotted for a variety of sensor masses. Sensor mass estimation error is quantified below.

and was subsequently perturbed, while in this experiment, the vehicle is commanded to fly toward the target, contact the target (experiencing a perturbation), and, subsequently, retreat from the target. As a result, reference position deviation  $e_{r,x}$  begins to increase in magnitude prior to the sensed perturbation. Once target contact is initiated, however,  $e_{r,x}$  maintains the desirable compliant behavior observed in the admittance controller performance experiment: vehicle motion is “shaped” by the external perturbation.

Trajectory tracking metrics reported in Fig. 10 for a single sample trial validate our control and motion methodology within the most salient metrics of the sensor placement task: highly accurate (e.g., subcentimeter) trajectory following paired with the ability to maintain compliance to external perturbation.

#### D. Evaluation of Onboard Force Estimation

In order to validate onboard force estimation capabilities, a hover test with three different sensor masses was performed. Each sensor mass (100, 120, and 170 g) was placed on the end of the tool-tip and compared to the case where no mass is present. The intent of this experiment is to not only understand the degree to which we are able to estimate external force, but more importantly determine if we are able to infer when a sensor has been deployed on a target.

As the weight of each respective sensor acts primarily in the  $z$ -direction,  $f_{e,z}$  was recorded and a regression analysis was performed to fit estimated forces to expected forces  $f_{e,z}$ . The results of this experiment are quantified in Fig. 11. Results indicate that we are able to identify external force, and therefore sensor mass, with adequate enough resolution ( $< 0.7\%$  error) to enable further autonomy of our framework, namely the ability

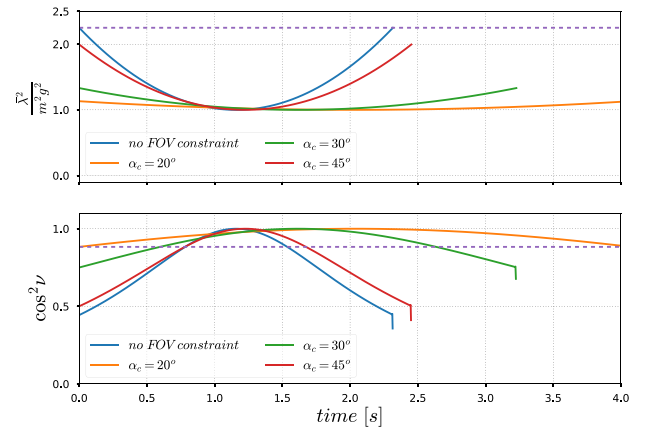


Fig. 12. Trajectory generation under varying sensing constraints ranging from no constraint to  $20^\circ$ ,  $30^\circ$ , and  $45^\circ$  FOV. The top subplot describes the commanded thrust constraint on  $\lambda$ , parameterized as Beziér curve as in (6) with the purple dotted line at  $1.5 \times$  the weight of the robot. The bottom subplot highlights the camera angle constraint on  $\nu$ , again parameterized as in (10) with the purple dotted line highlighting the constraint on  $\nu$  for the single case of  $\alpha_c = 20^\circ$ .

to know when the sensor is no longer being transported by the manipulator due to successful deployment. While other approaches, such as momentum-based estimation with gyroscopic information [39], can provide a higher level of precision in estimating the sensor weight, we find that our acceleration-based approach provides sufficiently accurate and responsive force estimation for the range of sensor weights we are interested in.

#### E. Trajectory Generation Simulation

Here, we present simulation results of our trajectory generation methodology from Section IV-D with  $\mu = 90^\circ$ , plotted for a variety of camera FOV constraints. The maximum available thrust is set to  $1.5 \times$  the weight of the robot, as shown in Fig. 12 on the top subplot as a purple dotted line. In the bottom subplot, the camera angle  $\nu$  is plotted. To illustrate our ability to generate trajectories while abiding by the defined constraints, we plot the minimum allowable  $\nu$  for the target to remain within the camera FOV as a dotted purple line for a single camera FOV case (i.e., when  $\alpha_c = 20^\circ$ ). As intuition would suggest, the angle  $\nu$  increases as the robot pitches forward to follow the trajectory. The robot is commanded to travel from an initial point of  $(x, y, z) = (0, 0, 1)$  to a target point of  $(1, 0, 1)$  on a vertical surface; with initial and final velocities set to 0. The order of the Bernstein polynomial is set to 6, while the optimization problem is solved using the optimizer in [40].

Looking at Fig. 12, we can see that without the sensing constraint defined by (10), the robot will aggressively translate forwards at the maximum available thrust, reaching the target in less than 2.5 s ( $\approx 4$  m/s) to achieve minimum time. Such aggressive motion, however, potentially leads to the target to deviate from the camera’s FOV. Furthermore, we can see that the smaller the camera FOV  $\alpha_c$ , the more conservative the motion of the robot is. For example, given a camera with an FOV of  $20^\circ$ , the robot requires 4 s to complete the motion while a camera

**TABLE I**  
EXPERIMENT STATISTICAL METRICS

Experiment Summary					
$n$	Failure (%)	95 % Success rate CI			
162	6.17	(89 %, 97 %)			
Failure mode instances					
	Mode 1	Mode 2	Mode 3		
	1	2	7		
Approach trajectory computation time [s]					
	Mean	Median	SD	IQR	
	0.49	0.48	0.03	0.03	
Sensor placement position [cm]					
Dim.	Mean	Median	SD	IQR	95 % CI
$y$	-0.45	-0.41	1.65	2.20	(-0.71, -0.19)
$z$	-0.22	0.18	3.25	3.57	(-0.74, 0.29)

with a FOV of  $45^\circ$  is able to reach the target in just less than 2.5 s.

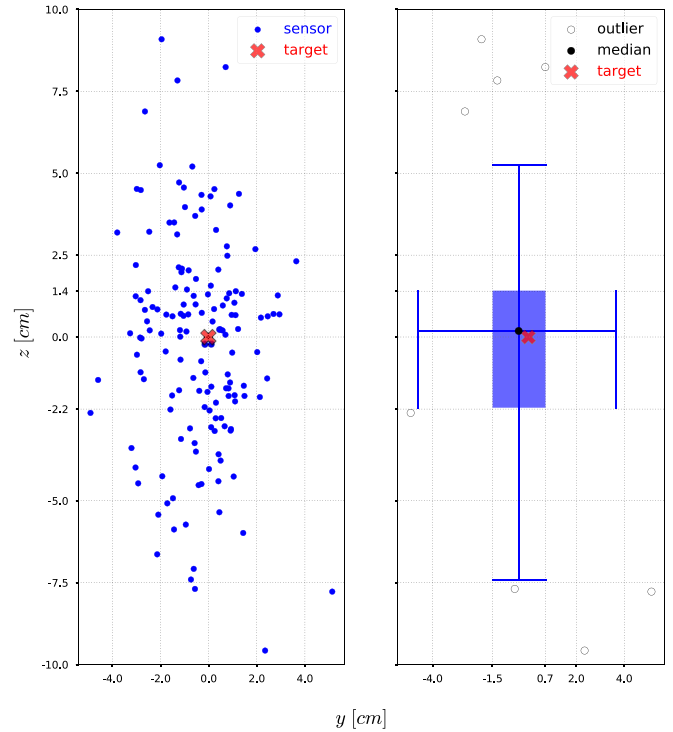
Lastly, it is noted that there is a sharp drop in the angle  $\cos^2 \nu$  once the target is reached in the bottom subplot: Once the camera center reaches the target end point, the sensing constraint is no longer well-defined and  $\cos^2 \nu$  is set to 0.

## VI. SENSOR PLACEMENT STATISTICAL ANALYSIS

A statistical investigation into the performance of the autonomous sensor placement framework was performed in laboratory conditions using onboard sensing for a total of 162 placement trials. The intent of this statistical analysis is to quantify our framework's performance during compliant interaction with the target while holding all other variables constant. As a result, a simplified version of our trajectory generation methodology was used for these tests: an empirically derived constraint on the quadrotor's velocity, acceleration, and jerk was imposed in order to preserve both actuation and camera FOV limitations while traversing a cubic Bézier curve. The same time-optimal Bézier curve trajectory was generated and traversed for all 162 placement trials. Further trajectory generation implementation results expanding upon our preliminary evaluation presented in Section V-E are left for future work.

For all statistical metrics in Table I, any position offset bias in the  $y$  and  $z$  directions that existed prior to the generation of the *target approach* trajectory is corrected for. Such a step was completed in order to quantify the baseline performance of our control and motion planning methodology independent of inaccuracies resulting from platform specific anomalies.

The robustness of our proposed pipeline is quantified in Table I as a failure rate percent. Here, a failure is defined as a sensor placement error in either the  $y$  or  $z$  directions in excess of 10 cm and occurred as a result of one of the following reasons.



**Fig. 13.** Successful placement results ( $n = 152$ ) are summarized from left to right in a scatter and box-and-whisker plot, respectively.

- 1) State estimate drift due to lack of visual features from the onboard tracking camera, manifested as position estimate error.
- 2) Communication latency between the command center computer and onboard computer manifesting in nonsmooth approach trajectory generation.
- 3) Inaccurate thrust mapping as battery voltage decreases, manifesting as a  $z$ -direction error.

Each respective failure mode can be mitigated as follows: mode 1) by updating the robot's relative position to the target position (kept within the camera FOV) during the approach phase, mode 2) by improving the range of the communication network between the command and onboard companion computers, and mode 3) by correcting for the commanded thrust scalar error due to battery energy decline during flight by monitoring the battery voltage and applying appropriate compensation.

The placement failure rate for the experiment performed with our proposed framework compares favorably to the direct sensor placement strategies described in [28] and [29], which cite a failure rate of 15.2% during the completion of 33 placement trials. Metrics highlighting the onboard computation time of the approach trajectory for each trial are included in Table I. The approach trajectory was generated once for each sensor placement trial. Lastly, all of the 152 successful placements performed using our framework are presented in Fig. 13.

## VII. CONCLUSION

In this article, we have presented an integrated solution for autonomous sensor placement carried out by a quadrotor. The

proposed strategy makes use of a standard and readily available platform to elaborate on the high-level path planning and force estimation required to complete the sensor placement task on a vertical surface. Within our integrated approach, we account for platform underactuation and shape the interaction trajectory and force in such way to achieve compliant contact with the environment via admittance control. The trajectory generation during target approach considers the visual requirements of the onboard cameras while observing salient platform constraints, contributing to the overall robustness of the autonomy pipeline.

We demonstrate our proposed solution in flight without reliance on external sensors or instrumentation, illustrating our framework's ability to reliably navigate through an unmodeled environment with subcentimeter tracking error while maintaining compliance to external perturbation. A statistical analysis is performed, revealing our system's robustness, accuracy, and precision when performing the sensor placement task. Completion of 162 trials confirms our system's ability to deliver a sensor within 10 cm of the target with less than a 3.3 cm deviation in each dimension on a 2-D vertical target surface while maintaining a success rate of 93.8%. Moreover, it can be seen that the failure rate can be drastically decreased by addressing the dominant failure mode, namely inaccurate thrust mapping resulting in a  $z$ -direction error.

Future work will investigate alternative applications for our proposed autonomy framework, as well as methods for autonomous sensor retrieval operations, paving the way for the deployment of aerial manipulators in a variety of real-world environments: both man-made and natural.

## REFERENCES

- [1] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2491–2498.
- [2] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 2668–2673.
- [3] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Proc. Distrib. Auton. Robot. Syst.*, 2013, pp. 545–558.
- [4] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Dexterous aerial robots—mobile manipulation using unmanned aerial systems," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1453–1466, Dec. 2017.
- [5] F. Augugliaro et al., "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Syst. Mag.*, vol. 34, no. 4, pp. 46–64, Aug. 2014.
- [6] A. E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 4916–4921.
- [7] M. Kovac, "Learning from nature how to land aerial robots," *Science*, vol. 352, no. 6288, pp. 895–896, 2016.
- [8] P. Sareh, P. Chermprayong, M. Emmanuelli, H. Nadeem, and M. Kovac, "Rotorigami: A rotary origami protective system for robotic rotorcraft," *Sci. Robot.*, vol. 3, no. 22, p. eaah5228, 2018.
- [9] M. Fumagalli et al., "Developing an aerial manipulator prototype: Physical interaction with the environment," *Robot. Automat. Mag.*, vol. 21, pp. 41–50, 2014.
- [10] T. Ikeda et al., "Wall contact by octo-rotor UAV with one DOF manipulator for bridge inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 5122–5127.
- [11] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Valve turning using a dual-arm aerial manipulator," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2014, pp. 836–841.
- [12] S. Hamaza, I. Georgilas, and T. Richardson, "2D contour following with an unmanned aerial manipulator: Towards tactile-based aerial navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3664–3669.
- [13] A. S. Vempati et al., "PaintCopter: An autonomous UAV for spray painting on three-dimensional surfaces," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 2862–2869, Oct. 2018.
- [14] S. Hamaza, H.-N. Nguyen, and M. Kovac, "Sensor delivery in forests with aerial robots: A new paradigm for environmental monitoring," in *Proc. IEEE IROS Workshop Percep. Plan. Mobility Forestry Robot.*, 2020.
- [15] P. E. Pounds and A. M. Dollar, "Stability of helicopters in compliant contact under PD-PID control," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1472–1486, Dec. 2014.
- [16] H. Nguyen, C. Ha, and D. J. Lee, "Mechanics, control and internal dynamics of quadrotor tool operation," *Automatica*, vol. 61, pp. 289–301, 2015.
- [17] M. Ryll et al., "6D interaction control with aerial robots: The flying end-effector paradigm," *Int. J. Robot. Res.*, vol. 38, no. 9, pp. 1045–1062, 2019.
- [18] H. W. Wopereis, J. Hoekstra, T. Post, G. A. Folkertsma, S. Stramigioli, and M. Fumagalli, "Application of substantial and sustained force to vertical surfaces using a quadrotor," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 2704–2709.
- [19] H. Nguyen, R. Siddall, B. Stephens, A. Navarro-Rubio, and M. Kovač, "A passively adaptive microspine grapple for robust, controllable perching," in *Proc. IEEE Int. Conf. Soft Robot.*, 2019, pp. 80–87.
- [20] K. Bodie et al., "An omnidirectional aerial manipulation platform for contact-based inspection," *Proc. Robotics: Sci. Syst. XV*, 2019, pp. 1–9.
- [21] R. Rashad, D. Bicego, R. Jiao, S. Sanchez-Escalonilla, and S. Stramigioli, "Towards vision-based impedance control for the contact inspection of unknown generically-shaped surfaces with a fully-actuated UAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 1605–1612.
- [22] M. Kamel et al., "The voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle," *IEEE Robot. Automat. Mag.*, vol. 25, no. 4, pp. 34–44, Dec. 2018.
- [23] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette, "Vision-based minimum-time trajectory generation for a quadrotor UAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6199–6206.
- [24] M. Jacquet and A. Franchi, "Motor and perception constrained NMPC for torque-controlled generic aerial vehicles," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 518–525, Apr. 2021.
- [25] J. Mao, G. Li, S. Nogar, C. Kroninger, and G. Loianno, "Aggressive visual perching with quadrotors on inclined surfaces," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5242–5248.
- [26] D. R. McArthur, A. B. Chowdhury, and D. J. Cappelleri, "Design of the interacting-boomcopter unmanned aerial vehicle for remote sensor mounting," *J. Mechanisms Robot.*, vol. 10, no. 2, pp. 1–8, 2018.
- [27] D. R. McArthur, A. B. Chowdhury, and D. J. Cappelleri, "Autonomous control of the interacting-boomcopter UAV for remote sensor mounting," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5219–5224.
- [28] S. Hamaza et al., "Sensor installation and retrieval operations using an unmanned aerial manipulator," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 2793–2800, Jul. 2019.
- [29] S. Hamaza, I. Georgilas, G. Heredia, A. Ollero, and T. Richardson, "Design, modeling, and control of an aerial manipulator for placement and retrieval of sensors in the environment," *J. Field Robot.*, vol. 37, no. 7, pp. 1224–1245, 2020.
- [30] A. Farinha, R. Zufferey, P. Zheng, S. F. Armanini, and M. Kovac, "Unmanned aerial sensor placement for cluttered environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6623–6630, Oct. 2020.
- [31] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [32] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Comput. Sci. Dept., Technische Universitaet Muenchen, Germany, Oct. 2009.
- [33] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 2017.
- [34] H. Nguyen, S. Park, J. Park, and D. Lee, "A novel robotic platform for aerial manipulation using quadrotors as rotating thrust generators," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 353–369, Apr. 2018.
- [35] T. Tomic, C. Ott, and S. Haddadin, "External wrench estimation, collision detection, and reflex reaction for flying robots," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1467–1482, Dec. 2017.
- [36] T. Tomic and S. Haddadin, "Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 5290–5296.

- [37] R. T. Farouki, "The Bernstein polynomial basis: A centennial retrospective," *Comput. Aided Geometric Des.*, vol. 29, no. 6, pp. 379–419, 2012.
- [38] V. Cichella, I. Kammer, C. Walton, and N. Hovakimyan, "Optimal motion planning for differentially flat systems using Bernstein approximation," *IEEE Contr. Syst. Lett.*, vol. 2, no. 1, pp. 181–186, Jan. 2018.
- [39] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance control of VTOL UAVs with a momentum-based external generalized forces estimator," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 2093–2099.
- [40] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, pp. 261–272, 2020.



**Brett Stephens** (Graduate Student Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of California, Davis, CA, USA, and the M.Sc. degree in robotics, systems, and control from the Swiss Federal Institute of Technology in Zurich (ETHZ), Zurich, Switzerland, in 2012 and 2018, respectively. He is currently working toward the Ph.D. degree in aeronautical engineering with the Department of Aeronautical Engineering, Aerial Robotics Laboratory, Imperial College

London, London, U.K.

Between earning the B.Sc. and M.Sc. degrees, Brett worked as a Research and Development Engineer with Ford Motor Company, Dearborn, MI, USA. His research interests include perception, control, and planning strategies for interactive autonomous systems.



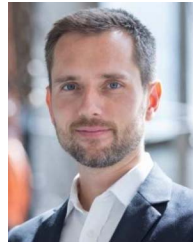
**Hai-Nguyen (Hann) Nguyen** (Member, IEEE) received the B.Sc. degree in mechatronics and the M.Sc. degree in engineering mechanics from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2008 and 2011, respectively, and the Ph.D. degree in mechanical and aerospace engineering from Seoul National University, Seoul, South Korea, in 2018.

From 2009 to 2012, he was a Permanent Researcher with the Institute of Mechanics, Vietnam Academy of Science and Technology, Hanoi, Vietnam. In 2018, he joined the Aerial Robotics Laboratory, Imperial College London as a Postdoc. He was a Senior Robotics Engineer with Vimaan Robotics, Santa Clara, CA, USA, in 2020, working on autonomous indoor robots. Since 2021, he has been a Contract Research Scientist with Laboratoire d'Analyse et d'Architecture des Systemes (LAAS), Centre National de la Recherche Scientifique (CNRS), Paris, France. His research interests include dynamics, control, and planning problems related to aerial manipulation.



**Salua Hamaza** received the B.Sc. and M.Sc. degrees in mechanical engineering from the University of Bologna, Bologna, Italy, in 2012 and 2014, respectively. The M.Sc. degree was completed in collaboration with the Delft University of Technology (TU Delft), Delft, Netherlands. She worked toward the Ph.D. degree in robotics and autonomous systems with Bristol Robotics Laboratory, University of Bristol, Bristol, U.K.

After the M.Sc., she joined the Robotics and Mechatronics Group (RAM) as a Research Engineer with the University of Twente, Enschede, Netherlands. In 2018, she was a visiting Ph.D. student with the Group of Robotics, Vision and Control (GRVC), University of Seville, Seville, Spain. She joined the Aerial Robotics Laboratory, Imperial College London, London, U.K., in 2019 as a Postdoc. She is currently an Assistant Professor in aerial robotics and Director of the BioMorphic Intelligence Lab, Faculty of Aerospace Engineering, TU Delft. Professor Hamaza's research interests include compliant design, interaction control, and embodied intelligence for aerial robots.



**Mirko Kovac** (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering from the Swiss Federal Institute of Technology in Zurich (ETHZ), Zurich, Switzerland, in 2005, and the Ph.D. degree in bio-inspired mobile robots from the Swiss Federal Institute of Technology in Lausanne (EPFL), Vaud, Switzerland, in 2010.

He pursued a Postdoc with Harvard University, Cambridge, MA, USA, until 2012. He is currently the Director of the Aerial Robotics Laboratory, full Professor with Imperial College London, London, U.K., and Royal Society Wolfson Fellow. He is also heading the Materials and Technology Centre of Robotics, Swiss Federal Laboratories for Materials Science and Technology (EMPA), Zurich, Switzerland. His research group focuses on the development of novel, aerial robots for distributed sensing, and autonomous manufacturing in complex natural and man-made environments. His specialization is in robot design, hardware development, and multimodal robot mobility.