

Chasing Millimeters: Design, Navigation and State Estimation for Precise In-flight Marking on Ceilings

Christian Lanegger[†], Michael Pantic[†], Rik Bähnemann, Roland Siegwart and Lionel Ott

Autonomous Systems Lab, ETH Zurich, Leonhardstrasse 21, Zurich, 8092, Switzerland.

*Corresponding author(s). E-mail(s): clanegge@ethz.ch; mpantic@ethz.ch;

Contributing authors: brik@ethz.ch; rsiegwart@ethz.ch; liott@ethz.ch;

[†]These authors contributed equally to this work.

Abstract

Precise markings for drilling and assembly are crucial, laborious construction tasks. Aerial robots with suitable end-effectors are capable of markings at the millimeter scale. However, so far, they have only been demonstrated under laboratory conditions where rigid state estimation and navigation assumptions do not impede robustness and accuracy. This paper presents a complete aerial layouting system capable of precise markings on-site under realistic conditions. We use a compliant actuated end-effector on an omnidirectional flying base. Combining a two-stage factor-graph state estimator with a Riemannian Motion Policy-based navigation stack, we avoid the need for a globally consistent estimate and increase robustness. The policy-based navigation is structured into individual behaviors in different state spaces. Through a comprehensive study, we show that the system creates highly precise markings at a relative precision of 1.5 mm and a global accuracy of 5-6 mm and discuss the results in the context of future construction robotics.

Keywords: End-effector design, Sensor fusion, Riemannian Motion Policies, Construction robotics, Aerial robotics

1 Introduction

With the increased capabilities and success stories of autonomous systems, more and more applications previously considered impossible for robotic systems have come within reach. As a result, companies invest more heavily in developing robotic systems. Construction remains one of the least automatized industries, however, also in construction, there is a notable trend towards robotics technologies (Boston Consulting Group, 2018). Early mover companies have started developing ground robots for construction site tasks.

For example, Hilti recently presented a semi-autonomous drilling robot¹, and Husqvarna offers remote demolition robots².

The previously mentioned systems have a limited workspace. Due to their ground robot nature, they have limited work heights and need traversable, load-bearing paths to move. Aerial robots, on the other hand, can operate

¹"<https://www.hilti.com/content/hilti/W1/US/en/business/business/trends/jaibot.html>", last accessed on 17.02.2023

²"[HusqvarnaDXRremotedemolitionrobots](https://www.husqvarna.com/usa/en/remotedemolitionrobots)", last accessed on 17.02.2023



Fig. 1 Our aerial robot during a layouting mission on a construction site accurately marking points on a concrete ceiling.

at high altitudes and in areas difficult to traverse. However, they are susceptible to disturbances, increasing the challenge of performing exact and dexterous tasks. Commercially available Micro Aerial Vehicles (MAVs) exist for asset management and visual inspection with collision-resilient platforms³. Tasks in construction environments require contact with the environment, a regime for which these commercial systems were not designed. A recently emerging class of aerial robots are omnidirectional MAVs, which are able to exert forces and torques in arbitrary directions in a controlled manner. Early examples are the (Bodie et al, 2019) and the commercialized remote-controlled Voliro⁴. However, there exist still a wealth of technical hurdles in state estimation and planning for reliable, precise, and autonomous missions on-site. A task of interest in construction is layouting, i.e., marking points, lines, and curves on the ceiling to indicate areas where to drill or anchor components. The manual marking process is repetitive and troublesome, especially at height, and small errors accumulate, which can lead to costly remediations. The level of accuracy required from the layouting tool has yet to be achieved with common aerial robots on-site, which typically operate at centimeter accuracies.

³<https://flybotix.com/asio-drone/>, last accessed on 17.02.2023

⁴<https://www.voliro.ch>, last accessed on 17.02.2023

1.1 Related Works

In recent years, the research community investigated the usage of aerial manipulators for non-destructive contact-based inspection and manipulation (Ollero et al, 2021). Previous works have shown that aerial manipulators can accurately measure material thickness using ultrasonic probes (Watson et al, 2021) or locate material defects with eddy-current sensors (Tognon et al, 2019). Impressive progress has also been made in push-and-slide inspection, with aerial systems being able to follow curved surfaces without losing contact (Nava et al, 2020). Several works improve precision and disturbance rejection using parallel manipulators. End-effector accuracy can be improved by at least an order of magnitude with the help of a delta-manipulator (Chermprayong et al, 2019). Tzoumanikas et al (2020) showed impressive drawing precision in the millimeter range, jointly controlling the MAV and delta-arm with nonlinear model predictive control. Stephens et al (2023) showed in a remarkable number of experiments that also classical coplanar MAVs can be used to autonomously place sensors on planar surfaces within a few centimeters. All the above-presented aerial robots were operating in lab conditions. However, transitioning into real applications involves robustness to environmental uncertainties and reliable decisional autonomy (Ollero et al, 2021), which our work addresses.

Despite the great efforts from the research community, complete systems that can operate under more realistic conditions remain rare. Sanchez-Cuevas et al (2020) demonstrated an autonomous aerial robot for bridge inspection purposes equipped with multiple sensors fused in a Kalman filter to estimate the robot’s position. The authors address the problem of robustly aligning different reference frames by running a separate calibration procedure beforehand. While their sensor fusion framework can handle measurement dropouts, data collection for the calibration procedure needs to be repeated after each system restart. Additionally, there is no information on the system’s absolute accuracy; thus, it is unclear how well the system performs after longer operation times when it accumulates drift and the statically obtained alignment no longer holds. Trujillo et al (2019) presented a semi-autonomous

aerial robot for Non-destructive testing (NDT) in the oil and gas industry. The robot has two operating modes, a free flight mode and a contact-mode that keeps the robot steady with respect to the contacted surface. Switching between modes is handled by the human operator. When in contact the robot’s pose is estimated with respect to the contacted surface using image-based feature tracking. While this enables the aerial robot to robustly follow surfaces, the achieved accuracy (the authors claim an accuracy of 1.8 cm per traveled meter) is not sufficient for construction-related tasks.

1.2 Contributions

In our previous work (Lanegger et al, 2022), we presented the design and evaluation of the here-used end-effector concept for aerial layouting under laboratory conditions. In this paper, we extend our previous work and take a holistic look at all challenges present in aerial layouting under real conditions. Namely, our contributions include

- a complete, novel system capable of aerial layouting at height without the need for a motion capture system,
- a state estimation and sensing strategy for robust flight close to and in contact with structure,
- a novel navigation approach tailored to the unique setting of a flying robot with a spring-decoupled and independently actuated end-effector,
- and finally, a comprehensive experimental study of achieved accuracy and precision, and contextualization thereof in a construction setting.

To the best of our knowledge, this is the first aerial layouting system capable of precise markings on-site without the need for a motion capture system. As this paper is an extension of a conference paper, the chapters concerning the end-effector (section 2.2, section 3.1) are adapted from our previous work.

2 Method

In this section, we present the main components required for high-accuracy layouting on ceilings with an aerial robot. The robot consists of two components, an omnidirectional flying base that

can exert forces in any direction and a compliant and actively driven end-effector (see Figure 1). In our previous work (Lanegger et al, 2022), we showed that this end-effector has been crucial for achieving millimeter precision using a flying base. In order to fully utilize the capabilities of the end-effector in real-world construction environments, robust and accurate state estimation and navigation components are needed. We propose using a pose-graph-based state estimator with a dual-graph design to decouple global and local state estimation in combination with a highly modular Riemannian Motion Policy-based navigation system that exploits the compliant nature of the end-effector reactively. All main components are presented in detail in the remainder of this chapter.

2.1 Flying base

Our flying base is a custom-built hexacopter with servo-motor actuated propeller arms. It is possible to change the tilt angle of each arm individually and to effectively decouple the translational movement of the flying base from its orientation. The flying base uses six *KDE 3510XF-475* motors with 12x4.5 propellers and RPM-controlled ESCs for propulsion. Six *Dynamixel XM430-W350* control the orientation of the arms. Additionally, the robot carries an *Nvidia AGX Orin* onboard computer running Ubuntu 20.04 and an *ADIS16448B* Inertial Measurement Unit (IMU). Two onboard cameras track the flying base’s pose and the end-effector’s relative displacement, respectively. A reflective prism locates the robot’s absolute position with respect to accurate measurements from a total station. A depth camera is used to measure the relative distance to the surface of interest. Table 1 lists more detailed information on all sensors mounted on the flying base. The impedance controller developed by Bodie et al (2019) is running on the onboard computer to control the pose of the flying base.

2.2 End-Effector Design

The main goal of the end-effector is to correct the positional imprecision of the flying base mostly stemming from unforeseeable in-flight disturbances. With this in mind, our end-effector incorporates three main design choices:

Type	Model	Rate	Characteristics
IMU	Adis 16448B	200 Hz	Turn-on bias: 20 mg, 1800°/h Random walk: 187 $\mu\text{g}/\sqrt{\text{Hz}}$, 0.66°/ $\sqrt{\text{h}}$
Camera	Realsense T265	30 Hz	
Reflector	GRZ101 360° Mini Prism	15 Hz	1.5 mm pointing accuracy
Camera	See3CAM 20CUG	60 Hz	2 MP global shutter , <i>PT-01224XFL</i> lens
Depth Camera	PMD flexx 2	30 Hz	224 × 172 px resolution, 0.1 m – 4 m range

Table 1 Sensor specifications.

1. *Compliance* between the end-effector tool and the flying base to decouple disturbances acting on the base and thus increasing the precision of the marking;
2. *Multiple contact points* between the ceiling and the end-effector to increase system stability and constrain the end-effector’s attitude;
3. *Actuation* on the end-effector to allow controlled movement along the ceiling and further refinement of the tool position;

With these design objectives in mind, we developed an end-effector based on a Gough–Stewart platform (Stewart, 1965), a parallel manipulator with six degrees of freedom. Flight simulators commonly use this mechanism, where linear actuators control the cabin. Hu and Jing (2018) recently developed a passive Gough–Stewart structure by replacing the actuated legs with passive springs for vibration isolation. Similarly, our end-effector design has spring-dampers to connect the flying base with the movable end-effector tool. The design allows the flying base to push upwards and compress the spring-dampers, suppressing vibrations and effectively increasing the stability of the flying base.

The spring-dampers are off-the-shelf *Z-D0033 Dual Spring Shocks* for radio-controlled model cars. We adjusted the spring stiffness such that a force of 15 N halves the distance between the end-effector and the flying base. The required force is a trade-off between available thrust, ceiling grip and compliance. The damping fluid is water. By experimentation, it was the only fluid that allowed the upper platform to return to its maximal height with the given spring stiffness. The end-effector’s geometry is optimized such that it can be displaced by up to 2 cm in directions parallel to the flying base. The displacement is sufficient to compensate for in-flight disturbances acting on the

flying base. The typical flight control mean absolute error in all directions is less than 2 cm, as displayed in Table 4.

The end-effector is equipped with three custom-built omni-wheels to provide multiple contact points while still permitting smooth movement in any direction. Each wheel is actively driven by a servo-motor in order to compensate for tracking errors. The servos have a velocity controller that tracks a reference velocity provided by the navigation algorithm. The end-effector is additionally holding a retractable permanent marker for marking purposes. An upward-facing camera tracks the relative displacement between the flying base and the end-effector. It faces a *ChArUco* board attached to the end-effector’s bottom side. The computer vision library OpenCV (2015) provides the *ChArUco* tracker. The accuracy of the tracking system was evaluated in a static experiment using a Vicon motion capture system, reaching a tracking error of 0.8 mm in position and 0.2° in yaw on average (Lanegger et al, 2022).

2.3 Frame Definitions

The entire system has seven different coordinate frames, shown in Figure 2. In the following, we describe the frames, their relation to each other, and the determination of their transformations.

The target locations are given with reference to the globally fixed building frame \mathcal{B} . The body-fixed frame \mathcal{I} is located at the IMU’s point of percussion. It defines the pose of the aerial robot with reference to the odometry frame \mathcal{O} , which in our case, is the initial starting position of the robot. The location of the Visual Inertial Odometry (VIO) sensor and the prism is denoted as \mathcal{O}_S and \mathcal{P} , respectively, and the fixed transformation from the IMU to these sensors was obtained from the robot’s CAD model. The tool sensor frame \mathcal{T}_S

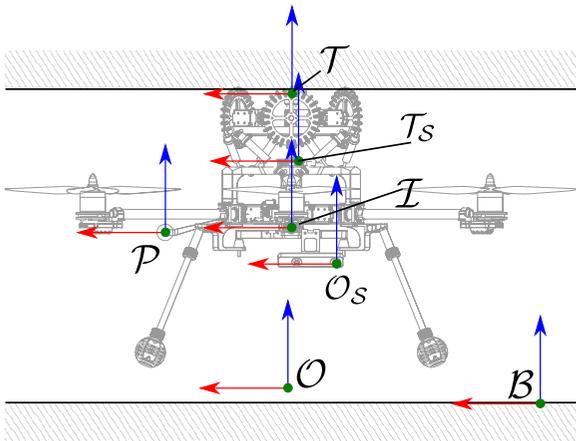


Fig. 2 Overview of the used coordinate frames. Nomenclature according to table 2.

Name	Abbr.	Parent	Fixed	Source
Building	\mathcal{B}	-	✓	-
Odometry	\mathcal{O}	\mathcal{B}		Global estimate
IMU	\mathcal{I}	\mathcal{O}		Local estimate
VIO sensor	\mathcal{O}_S	\mathcal{I}	✓	CAD model
Prism	\mathcal{P}	\mathcal{I}	✓	CAD model
Tool sensor	\mathcal{T}_S	\mathcal{I}	✓	Calibration
Tool	\mathcal{T}	\mathcal{T}_S		Tracking

Table 2 Description of all coordinate frames, their relation, and determination source.

is located at the end-effector camera, and its relative pose with respect to the IMU was obtained from an extrinsic calibration beforehand. The pose of the end-effector tool \mathcal{T} relative to \mathcal{T}_S is estimated through the end-effector camera tracking the *ChArUco* board.

The homogenous transformation from the odometry to the IMU frame $\mathbf{T}_{\mathcal{O}\mathcal{I}} \in SE(3)$ is estimated using only relative measurements provided by the onboard sensors. As a result, the estimated transform drifts with time. To account for this drift we additionally estimate the transformation from the building to the odometry frame $\mathbf{T}_{\mathcal{B}\mathcal{O}}$. Table 2 summarizes the abbreviation of all frames, their parent frames, and how the transformation to their parent frame was obtained.

2.4 Robust Sensor Fusion

Construction surveyors typically use total stations to stake out points of interest referenced to a globally fixed building frame \mathcal{B} . Robotic total stations can additionally track moving reflectors. Thus we use a total station to locate our robot. Unfortunately, the position of the prism is insufficient. We require a high rate estimate of $\mathbf{T}_{\mathcal{B}\mathcal{T}}$, which describes the pose of our marking tool within \mathcal{B} . The continuous estimate can be obtained by fusing multiple sensors and co-estimating their drifting calibrations.

Most approaches try to estimate the full global state of the system in a single filtering or smoothing framework (Lynen et al, 2013; Sandy et al, 2019; Indelman et al, 2013). However, systems with dense kinematic chains, multiple measurements, and states are highly non-linear and inherently hard to tune. Too many correlated states, some of which are not directly observable, need to be co-estimated and cause the optimization to get stuck in local minima. Furthermore, filtering methods are susceptible to outliers and measurement dropouts, causing them to become overconfident or diverge.

When tracking dynamic objects like aerial vehicles, total stations often lose track of the prism. With this in mind, we developed a pose-graph-based sensor fusion algorithm similar to Nubert et al (2022), based on a dual-graph design. In their work, the authors addressed the problem of measurement dropouts by switching between two different optimization problems depending on the availability of global pose measurements. We, on the other hand, propose two loosely coupled optimization problems, depicted in Figure 3.

The first optimization performs inference over a factor graph consisting only of states constrained by relative measurements. The relative nature of the constraints allows temporary dropouts and subsequent returns of measurements but comes at the cost of accumulating errors over time, causing the state estimate drift. Controlling the flying base requires only a locally consistent and smooth state estimate. The local factor graph is structured to favor robustness and local consistency over global estimate accuracy.

A second optimization uses the local state estimates and position measurements from the total station to infer the drifting pose of the odometry

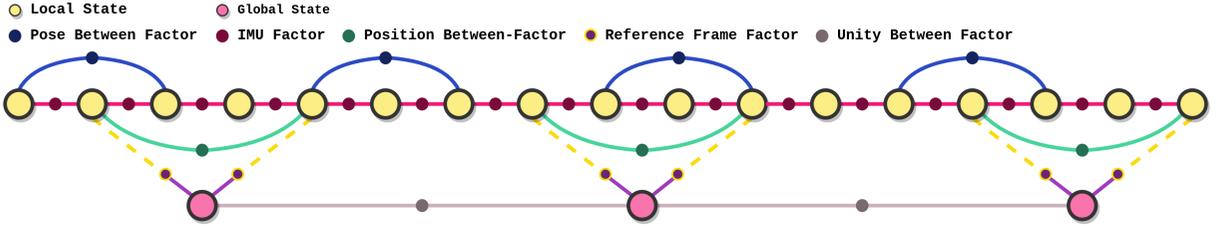


Fig. 3 Visualization of the dual-graph structure utilized in our state estimation framework. The local states are constrained purely by relative measurements in order to robustly handle measurement dropouts. The global graph independently infers the global pose using the local state estimate and position measurements provided by the total station.

(\mathcal{O}) frame origin with respect to \mathcal{B} . The global pose estimate is only used for end-effector navigation. Therefore, if measurements from the total station are lost during the operation, only the system’s global accuracy is affected. The aerial robot itself, however, remains fully operational.

The sensor fusion algorithm is implemented using the GTSAM framework (Dellaert and Contributors, 2022). In the following, we present the structure of the two different graphs in more detail.

Local State Estimation

The local state of the aerial robot at a time t is defined as

$$\mathcal{I}\mathbf{x} := [\mathbf{q}_{\mathcal{O}\mathcal{I}}, \mathcal{O}\mathbf{p}_{\mathcal{I}}, \mathcal{O}\mathbf{v}_{\mathcal{I}}, \mathcal{I}\mathbf{b}_a, \mathcal{I}\mathbf{b}_g] \quad (1)$$

with $\mathbf{q}_{\mathcal{O}\mathcal{I}} \in SO(3)$ being the rotation from \mathcal{O} to \mathcal{I} . $\mathcal{O}\mathbf{p}_{\mathcal{I}} \in \mathbb{R}^3$ and $\mathcal{O}\mathbf{v}_{\mathcal{I}} \in \mathbb{R}^3$ represent the position and velocity of \mathcal{I} relative \mathcal{O} . $\mathcal{I}\mathbf{b}_a \in \mathbb{R}^3$ and $\mathcal{I}\mathbf{b}_g \in \mathbb{R}^3$ are the biases of the IMU’s accelerometer and gyroscope modeled as integrated white noise expressed in \mathcal{I} .

Every IMU measurement adds a new state $\mathcal{I}\mathbf{x}$ to the local graph. Each of these states is connected to their previous one by an *IMU factor*. Every IMU factor is included in the optimization cost as the following additive term

$$\|\mathbf{r}_{\mathcal{I}}\|_{\Sigma_{\mathcal{I}}}^2, \text{ with } \mathbf{r}_{\mathcal{I}} := [\mathbf{r}_{\Delta\mathbf{R}}^T, \mathbf{r}_{\Delta\mathbf{v}}^T, \mathbf{r}_{\Delta\mathbf{p}}^T]^T \quad (2)$$

with covariance $\Sigma_{\mathcal{I}}$. $\mathbf{r}_{\Delta\mathbf{R}}$, $\mathbf{r}_{\Delta\mathbf{v}}$, and $\mathbf{r}_{\Delta\mathbf{p}}$ are residual errors of relative motion increments in orientation, velocity, and position, as described by Forster et al (2016, Equation 45). Odometry measurements from the VIO sensor and position measurements from the total station are added as

between factors to the local graph. These represent the error between the predicted and measured relative displacement in pose or position. Both are added at the sensor rate stated in Table 1 and always connected to the two states closest in time to the measurements.

The state estimate from the local pose graph optimization is directly used in the controller of the flying base. Therefore a high-frequent state estimate is required. We optimize the pose graph at 30 Hz and use a fixed-lag smoother with a relatively small window size of 0.5 s, trading better accuracy for lower computational cost. In between optimizations, the state is integrated using IMU measurements and provided to the controller at 200 Hz.

Global State Estimation

The global state

$$\mathcal{O}\mathbf{x} := [\mathbf{q}_{\mathcal{B}\mathcal{O}}, \mathcal{B}\mathbf{p}_{\mathcal{O}}] \quad (3)$$

defines the drift in orientation $\mathbf{q}_{\mathcal{B}\mathcal{O}}$ and position $\mathcal{B}\mathbf{p}_{\mathcal{O}}$ of \mathcal{O} wrt. \mathcal{B} . The state estimate of the local graph and position measurements from the total station are matched based on their timestamp and added as a *reference frame factor* to the graph. This factor is included in the optimization as the following position residual

$$\|\mathbf{r}_{\mathcal{G}}\|_{\Sigma_{\mathcal{G}}}^2 = \|\mathcal{B}\mathbf{p}_{\mathcal{P}} - (\mathcal{B}\mathbf{p}_{\mathcal{O}} + \mathbf{q}_{\mathcal{B}\mathcal{O}}(\mathcal{O}\mathbf{p}_{\mathcal{P}}))\|_{\Sigma_{\mathcal{G}}}^2 \quad (4)$$

with covariance $\Sigma_{\mathcal{G}}$, the measurement from the total station $\mathcal{B}\mathbf{p}_{\mathcal{P}}$ and $\mathcal{O}\mathbf{p}_{\mathcal{P}}$ obtained from the pose estimate of the local graph

$$\mathcal{O}\mathbf{p}_{\mathcal{P}} = \mathcal{O}\mathbf{p}_{\mathcal{I}} + \mathbf{q}_{\mathcal{O}\mathcal{I}}(\mathcal{I}\mathbf{p}_{\mathcal{P}}). \quad (5)$$

The position of the prism with respect to the IMU $\mathcal{I}\mathcal{P}_{\mathcal{P}}$ is obtained from the CAD model of the system.

As this factor provides a residual solely on the position, the orientation $\mathbf{q}_{\mathcal{B}\mathcal{O}}$ is only indirectly observable through the lever arm between \mathcal{I} and \mathcal{P} (see Figure 2) and sufficiently distinctive measurements. Hence, the global state $\mathcal{O}\mathbf{x}$ is only observable when the aerial robot is in motion. We, therefore, only add new states with every 15th measurement to the graph, corresponding roughly to one state per second. A unity *between factor* enforces consistency between two consecutive global transformation states. The identity assumption acts as a prior for the expected drift.

The rate at which the local state estimate accumulates drift is slow compared to the rest of the system’s dynamics. We, therefore, optimize the global pose graph only at 5 Hz. This allows us to optimize over a larger sliding window of 20 s while keeping the computational cost low.

2.5 Reactive Riemannian navigation

The flying base and the end-effector are controlled independently and linked through a compliant suspension. While necessary for accuracy, this design adds several unusual degrees of freedom, e.g., the body-tool offset, which we define as the distance and relative orientation between the flying base and the end-effector. Due to the physical nature of aerial interaction, unpredictable forces affect the system, such as air-flow disturbances, irregular contact, or ceiling effects. To address these unique challenges, we propose to use a modular and reactive navigation architecture based on Riemannian Motion Policies (Ratcliff et al, 2018). The well-defined geometric-mathematical structure of Riemannian Motion Policies (RMPs) allows the formulation of motion constraints in multiple different, potentially diverging, coordinate frames. Compared to optimization- or sampling-based algorithms, we do not need to commit to a single global and consistent overall state estimate and can stay robust in the presence of drift and sensing inaccuracies. The reactive nature of RMPs is ideal for coping with the unpredictability of aerial manipulation and layouting.

Riemannian Motion Policies

In the following, we summarize the most important characteristics and operators of RMPs. Please refer to (Ratcliff et al, 2018) for more details. The main idea behind RMPs is to decouple a navigation problem into small individual policies defined in the task manifold \mathcal{X} with dimensionality $d_{\mathcal{X}}$ where a given problem is easiest to solve. Each policy defines a state-dependent acceleration $f \in \mathbb{R}^{d_{\mathcal{X}}}$ and Riemannian metric $A \in \mathbb{R}^{(d_{\mathcal{X}} \times d_{\mathcal{X}})}$, which are then locally mapped to a common configuration manifold \mathcal{C} (e.g. $SE(3)$). The metric allows the weighting of individual policies, relative to others, directionally or axis-wise. The mapping, called the pull-back operator, from manifold \mathcal{X} to \mathcal{C} of the policy (f, A) is defined as

$$pull((f, A)_{\mathcal{X}}) = ((J^T A J)^+ J^T A f, J^T A J)_{\mathcal{C}}, \quad (6)$$

where J is the local Jacobian relating the two manifolds at a given position. The mapped policies add up into a single metric-weighted sum according to Equation (7).

$$(\bar{f}, \bar{A}) = \left(\left(\sum_i A_i \right)^+ \sum_i A_i f_i, \sum_i A_i \right) \quad (7)$$

Finally, the robot executes the resulting acceleration \bar{f} .

We structure the overall navigation approach into five policies that drive the complete system in a coordinated, robust, and exact manner. Table 3 provides an overview of the different policies, their metrics, and operation manifolds. The general mode of operation is to approach the ceiling through a depth-servoing policy and push against the wall to provide friction for the end-effector by activating a spring-loading policy. Once firmly in contact with the ceiling, the end-effector navigation policy aggressively drives the end-effector to the desired target. Due to the mechanical design, the range of independent movement of the end-effector is constrained to about 2 cm. Thus, the end-effector following policy continuously centers the flying base below the end-effector. To summarize, in free flight, the flying base operates as a normal aerial robot, but once in contact, the end-effector operates independently, and the flying base follows it to provide stability. Finally, the

prism tracking policy constrains the platform yaw such that the prism remains in direct line-of-sight to the total station. Generally, we consider our configuration manifold \mathcal{C} to be $SE(3)$ for the flying base and $SE(2)$ for the end-effector.

Most policies follow the generic attractor scheme from (Ratliff et al, 2018), which defines f as an acceleration based on a soft-maxed error function.

$$f = \alpha \cdot \mathbb{S}(x - x_0) - \beta \quad (8)$$

where α and β are tuning parameters, and \mathbb{S} is the soft-normalization function

$$\mathbb{S}(z) = \frac{z}{z + \gamma \log(1 + \exp(\gamma z))} \quad (9)$$

with tuning parameter γ . The corresponding metric A is often constructed as a non-directional diagonal matrix where individual axes can be weighted. The following subsections give an intuitive explanation of each policy’s function f . We provide a comprehensive set of equations to facilitate reproducibility in the supplementary material (Online Resource 2).

End-effector navigation

The end-effector navigation policy is an attractor that, when activated, drives the end-effector to the specified marking target location in the building frame \mathcal{B} . The current position of the end-effector is calculated through a concatenation of all necessary transforms from \mathcal{B} to \mathcal{T} . This policy is executed independently of the flying base on the end-effector wheels.

End-effector following

A simple $2d$ attractor that drives the flying base to be exactly below the independently moving end-effector.

Depth servoing

A $1d$ attractor moves the flying base towards the ceiling until the end-effector is in contact. This policy operates based on the output of a time-of-flight depth camera.

Spring loading

The spring policy exploits the underlying impedance controller to drive the flying base

upwards until a desired spring extension is reached. The end-effector spring extension can be measured based on the estimate between \mathcal{T} and \mathcal{T}_S . This is equivalent to controlling the pushing force without needing a force sensor. For safety, the metric of the spring loading policy decays exponentially if the desired spring load is exceeded.

Prism tracking

The prism tracking policy influences the yaw axis based on the current distance between the prism and an imaginary, gravity-aligned plane spanned by the total station and the flying base center. Keeping the prism in this plane ensures visibility.

Each policy can be disabled by temporarily multiplying the corresponding metric with zero. We use simple state-based rules or operator buttons to enable individual policies. All policies also have tuning parameters. However, tuning has proven to be robust and can be done for each policy independently, simplifying the tuning process significantly. As all involved manifolds are equivalent to $SE(3)$, the Jacobians between them are simple rotation matrices.

The evaluations and summations of policies are executed at the controller frequency of 200 Hz, with a CPU usage below 10% of one core. Compared to a sampling-based or optimization-based planner, our proposed stack is able to react fast and without delay to any disturbance or deviation at negligible compute cost. Due to the decomposition in multiple policies, each behavior can be tested and executed independently. Except for the end-effector navigation policy, which can be disabled if needed, all policies seamlessly cope with drifting odometry due to their body-frame formulations.

3 Results

We evaluate individual parts as well as the complete system, with a focus on accuracy, precision, and robustness. The contribution of the mechanical end-effector design is evaluated through a set of flight experiments under simplified conditions. The most important characteristics of the state estimation and navigation algorithms are evaluated individually. Finally, the complete system is

Policy	End-effector navigation	Spring loading	Depth servoing	End-effector following	Prism tracking
Task Manifold	\mathcal{T}	\mathcal{T}_S	\mathcal{I}	\mathcal{T}_S	\mathcal{I}
$diag(A)$	[1, 1, 0, 0, 0, 0]	[0, 0, 1, 0, 0, 0]	[0, 0, 1, 0, 0, 0]	[1, 1, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 1]
Conf. Manifold	\mathcal{B}	$\mathcal{I} \rightarrow \mathcal{O}$	\mathcal{O}	$\mathcal{I} \rightarrow \mathcal{O}$	\mathcal{O}
Executed on	EE	FB	FB	FB	FB

Table 3 Overview of the used policies. Task manifold is the space in which the policies f function is defined, whereas configuration manifold is the space the policy is mapped into for execution. Some policies are mapped through a chain of spaces, indicated by an arrow. The last row indicates on which part of the robot the policy is executed. EE stands for end-effector and FB for flying base. For the formulation of the f function of each policy, please refer to the supplementary material (Online Resource 2). The axis order in the metric matrix is $x, y, z, roll, pitch, yaw$.

demonstrated, and its precision and accuracy are evaluated in a laboratory setting and on a construction site. For the final evaluation, only x and y position errors, i.e., errors in the plane parallel to the ceiling, were considered as the ceiling constrains the end-effector in height and attitude.

3.1 End-effector

To evaluate the end-effector design, we performed an ablation study in which we removed individual features from the end-effector and compared the precision of the different configurations. During the experiments, the aerial robot followed a circular trajectory with a radius of 250 mm. The maximum velocity and acceleration were limited to 5 cm/s and 2.5 cm/s², respectively. All experiments were conducted using a Vicon motion capture system, providing accurate and non-drifting pose estimates used by the controller of the aerial robot. This removes error sources unrelated to the end-effector design. The results presented in Table 4, therefore, represent the maximum achievable precision of the system.

As a baseline, the aerial robot tracked the reference trajectory in free-flight, reaching an average precision of 22.6 mm. This corresponds roughly to the tracking performance of the flying base. To remove the compliance and actuation of the end-effector one at a time, we used rigid rods and dummy servos with frictionless bearings. By adding frictionless contact points between the ceiling and the end-effector (experiment ②) the aerial robot can almost halve the tracking error. This demonstrates that contact points with the manipulated surface can be exploited to stabilize the flying base. However, adding compliance to such a system is not beneficial, as ③ shows. The stiffness of the end-effector’s compliant structure is

not large enough to keep the end-effector above the flying base, and without additional actuation, the end-effector just gets dragged behind. Surprisingly, the main performance gain can be obtained by using a rigid end-effector with multiple actuated contact points, improving the precision of any previous experiment by order of magnitude (experiment ④). In addition to the multiple contact points stabilizing the flying base, the actuated wheels increase friction in unwanted directions while still keeping the friction low in the direction of the trajectory. These results also suggest that compliance is not strictly necessary to reach high precision. Nevertheless, adding compliance to the end-effector still reduces the tracking error, most likely because the end-effector can better compensate for orientation errors of the flying base (experiment ⑤). All subsequent experiments do not use the motion capture system.

3.2 Sensor fusion

This section presents a more qualitative evaluation of our sensor fusion framework. We tested the robustness of our local state estimation against sensor dropouts in an offline experiment and studied the convergence rate of our global state estimator. For a thorough evaluation of the system’s accuracy, the reader is referred to Section 3.4. The robustness of the local state estimator was evaluated by artificially removing sensor measurements that were recorded during an actual layouting experiment. In Figure 4 the output of the local estimator using this sparsified data is compared to an estimator provided with the full set of measurements. Green and orange bands represent periods in which position, respectively, pose measurements were missing completely. From the graphs, it is

Experiment	#	Contact	Wheels	Compliance	Actuation	End-effector xy-error [mm]			Flying Base xyz-error [mm]		
						MAE	STD	P _{90%}	MAE	STD	P _{90%}
Free-Flight	①					22.6	13.0	43.5	21.6	9.0	33.8
Friction-less	②	✓	✓			14.5	7.9	25.0	13.2	5.8	20.8
Spring-Dampers	③	✓	✓	✓		33.1	15.3	54.5	15.4	6.2	23.6
Actuation	④	✓	✓		✓	1.3	0.8	2.4	7.7	1.5	9.8
Full System	⑤	✓	✓	✓	✓	1.0	0.5	1.7	12.2	3.4	15.6

Table 4 Mean Absolute Error (MAE), standard deviation, and the 90th percentile for the end-effector and aerial vehicle for every individual design validation experiment.

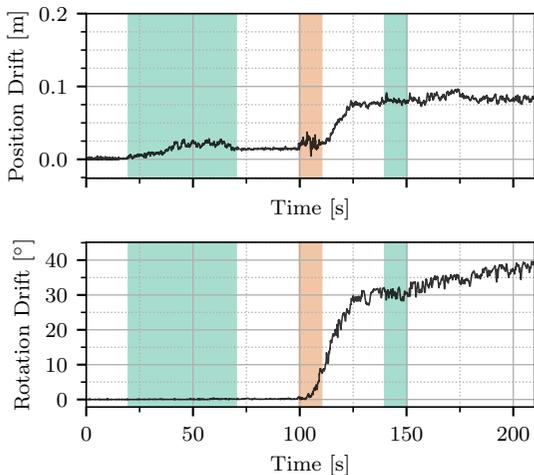


Fig. 4 The difference in absolute position and angle of rotation between a local state estimator without measurement dropouts and one with. The green and orange bars represent the loss of position and pose measurements, respectively.

visible that the state estimator can handle measurement dropouts. While slowly accumulating some drift, the local estimate stays locally consistent in both position and rotation and does not jump when sensor measurements return. It shows that the local estimator can provide the required robustness for operating aerial robots under realistic conditions where sensor measurements are not constantly available. It is worth noting that the loss of pose measurements results in a more considerable drift, especially in orientation. In such cases, the orientation is estimated purely by IMU measurements rendering the yaw unobservable and causing the orientation and position

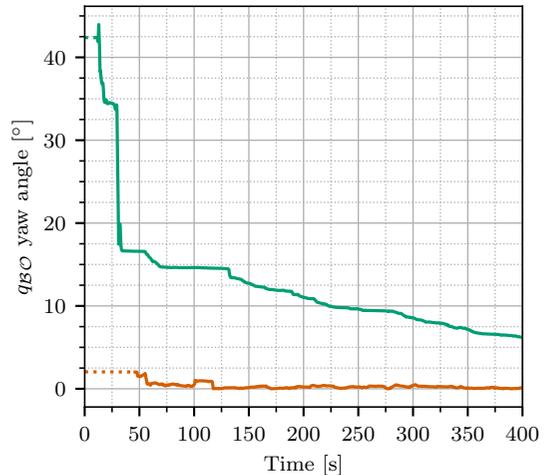


Fig. 5 The yaw angle estimate of the local odometry frame for two different experiments, one with a good initial guess (orange) and the other with a bad one (green). The dotted line indicates the time before take-off.

to drift slowly ($\approx 1.2^\circ/\text{s}$ and 0.004m/s). While errors in the global estimate do not affect the operability of the aerial robot, they do directly impact on the system’s absolute accuracy. A well-converged global estimate is, therefore, crucial for accurate end-effector positioning. Due to the nature of the available measurements, only the position and attitude of the global transform $\mathbf{T}_{\mathcal{B}\mathcal{O}}$ are observable. The yaw angle can only be indirectly observed from motion and, thus, is usually the largest source of error. Figure 5 shows that the global estimator can converge to a consistent solution in yaw. However, if the initial value is wrong by a significant amount, sufficient motion and time are needed to achieve convergence. The setup time

and convergence rate could be further improved by parameter tuning or by providing better initial guesses.

3.3 Navigation

Amongst all policies, the end-effector navigation policy has the largest influence on the final marking accuracy. In order to evaluate the convergence of the policy, we obtain 40 in-flight time series, starting when the end-effector is in firm contact, and the policy is activated and ending at the release of the marker pen. Figure 6 visualizes the millimeter xy plane deviations from the tool to the target location, as estimated by the state estimator for all tries, at four different times from policy activation to marking. The first plot corre-

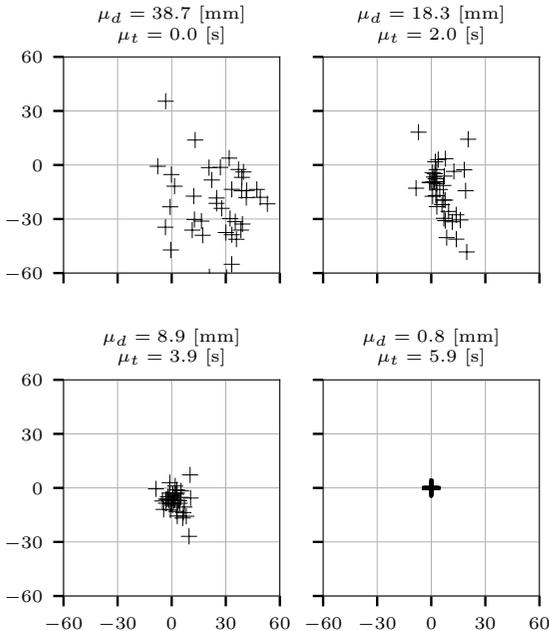


Fig. 6 Deviation of the tool tip to the believed target location according to state estimation, in global frame millimeters. Top left is before enabling the policy, bottom right is at the moment of pen actuation. The remaining two panels correspond to 33.3% resp. 66.6% time passed between start to marking. μ_d is the average deviation and μ_t the average time since enabling the policy.

sponds to the situation when the policy is enabled – basically, the free-flight error after attaching and stabilizing to the ceiling (38.7 mm). As is visualized, the end-effector policy drove the tool location

to below a millimeter deviation for all tries, taking 5.9s on average. The friction between the end-effector wheels and the ceiling varied between trials. On some surfaces, such as smooth paper, there was a significant amount of wheel slippage, which posed no problem to the reactive planning methodology.

The deviation reported in this experiment is seen from the robot in flight, i.e., the deviation to the target as output by state estimation. As the tool always converged to less than 1 mm error as estimated by state estimation, most of the global accuracy error we present in the next section comes from state estimation itself and calibration.

3.4 Full system

We perform a complete end-to-end accuracy study of the whole system. No external evaluation system, such as the VICON mocap system, could satisfactorily provide sub-millimeter accurate repeatable measurements in realistic conditions. Instead, we test the accuracy of single-point markings and the relative precision of a square grid of markings using pen on paper. An A3 paper equipped with pre-printed fiducials is mounted to a rigid, flat ceiling, and its absolute center spot location is determined using a total station (Nova MS60, Leica Geosystems). We evaluated the repeatability of the measurement to be below 1 mm. The aerial robot is then commanded to mark the center spot and a pattern of 4 markings in a 100 mm \times 100 mm square while using the total station as external tracking input as described Section 2.4. The pen release mechanism is tuned to draw as small and precisely locatable points as possible. After the experiment, we scan the paper with markings at 600 DPI, corresponding to a resolution of ≈ 26.3 pixels per mm. Furthermore, we rectify the scanned image using a homography calculated based on the pre-printed, known fiducials and thus obtain a metrically accurate, distortion-free representation of the marked points.

Figure 7 shows an example scan of one experiment. We estimate the resolution of the absolute accuracy evaluation to be on the order of ≈ 0.5 mm, while the relative precision evaluation is accurate to within ≈ 0.04 mm. Figure 8 visualizes the results of the absolute accuracy tests performed. Of the marked points, 75% are within

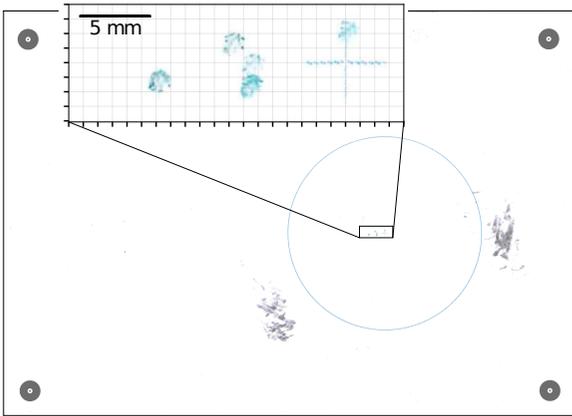


Fig. 7 Example of fiducials and results on A3 paper. The magnified rectangle shows the rectified cropped image with markings and the pre-printed center point cross. Grid spacing is 1mm.

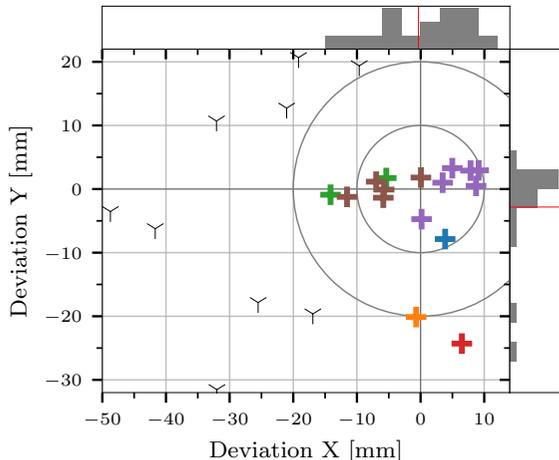


Fig. 8 In-plane global accuracy deviations of markings, colored by flight. The state estimator was reset between flights, leading to slightly different converged states. To illustrate the effect of calibration, the Y-shaped markings show markings affected by slightly suboptimal camera calibrations or prism-IMU calibrations.

a 10 mm radius around the true position, with an average error of 6.36 mm. The achievable global accuracy, as visualized in Figure 10, is between 5 – 6mm.

We attribute most of the remaining error to state estimation and calibration inaccuracies. As discussed in Section 3.3, the end-effector policy could always drive the end-effector to what the robot believes to be the true absolute global position. Especially the global yaw estimate has a large

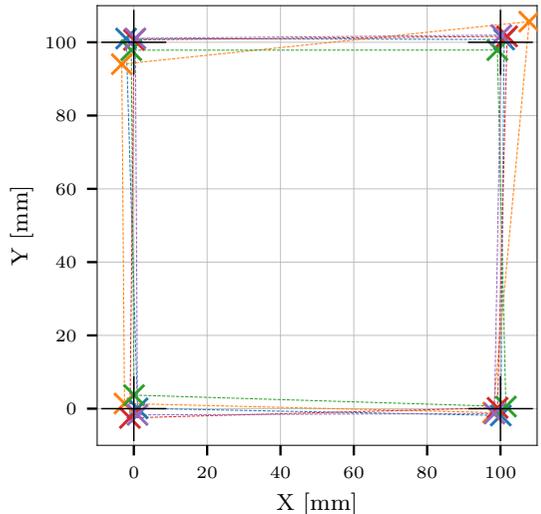


Fig. 9 Visualization of five repetitions of aligned square markings, colored by repetition. One try has been omitted as one marked point was not visible on paper, likely due to mechanical jamming of the pen release.

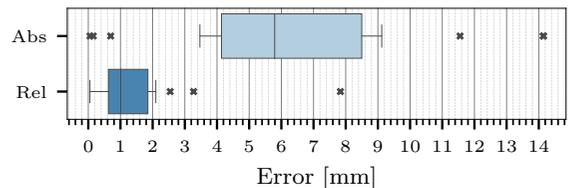


Fig. 10 Comparison and statistics of end-to-end absolute accuracy (Abs) and relative precision (Rel).

influence on accuracy, as it is not directly measurable – an estimation error of 1° yields a global marking error of ≈ 3.5 mm. The yaw error also likely explains the larger variance of errors in the x direction in Figure 8, as it roughly corresponds to yaw misalignment in this experimental setup.

Next, we evaluate the relative precision by commanding the robot to mark a 100 mm \times 100 mm square grid of four corner markings. The resulting marked pattern is then non-deformable least-squares aligned to the nominal corner coordinates spanning from (0, 0) to (100, 100), and the deviations measured for each marked point. This mimics the precision requirements for tasks such as mounting a bracket with a square hole pattern.

Figure 9 visualizes five repetitions of the square marking experiment. Overall, the average point-wise deviation from the true 100 mm \times 100 mm pattern is 1.49 mm. This corresponds to

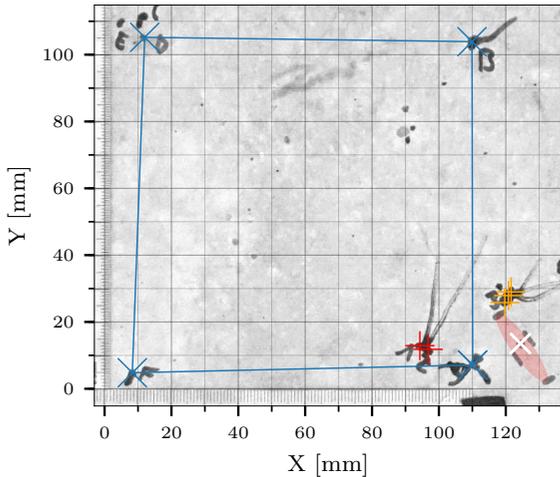


Fig. 11 Qualitative illustration of on-site results under realistic operation conditions. The 100 mm square grid is marked in blue, two different absolute precision tests (3 and 2 sequential markings) in red respectively orange. The measured target location is marked using a white cross. The red ellipse represents the laser beam divergence from the total station under the given distance and incidence angle conditions.

the tolerance available when using M8 bolts in 11 mm holes, a typical scenario for hole patterns in this size category (neglecting drilling tolerances). The larger deformation of the trial marked in orange in Figure 9 can be attributed to the convergence of the state estimator between individual point markings.

Figure 10 compares the achieved absolute accuracy and relative precision statistics. The demonstrated relative precision is close to the limits of the used hardware, as shown in our previous work (Lanegger et al (2022)), allowing for consistent marking of patterns with high precision. The larger variance and mean error of the absolute accuracy can be attributed to a stack up of state estimation and calibration inaccuracies, which can have stronger or weaker impacts depending on estimator convergence and robot orientation.

3.5 On-site demonstration

In addition to the laboratory environment experiments, we performed an on-site experiment on a mock construction site, as shown in Figure 1. Figure 11 gives a qualitative overview of the results. The slightly worse on-site absolute precision can be attributed to an older calibration used in the on-site experiment, as it was flown before

the laboratory trials. A video of on-site trials with annotations about active policies is available in the supplementary material (Online Resource 1).

4 Discussion

The precision and accuracy are comparable to what a human worker can do on-site with modern tooling. In order to reach the quality shown here, care has to be taken during the calibration and tuning of the system. Whenever possible, we performed least-squares fitting from datasets, e.g., for intrinsics and extrinsics of cameras. More problematic are calibrations that need to be taken from CAD, such as prism-to-imu or tool frames. As an example, the tolerance of the prism mounting axis to the optical center is given as 1.5 mm, and the exact center of percussion of the used ADIS16448B IMU is determined only to within a few millimeters (as each axis has a slightly different origin) - all these slight tolerances compound quickly. Furthermore, the state-of-the-art total station used exhibits typical beam divergence (roughly 0.4 mrad), further degrading measurements of arbitrary points at a distance.

However, one naturally wonders how precision and accuracy could be improved even more and what efforts and benefits this would entail? After all, millimeters are tiny at the construction site scale - even the seasonal temperature fluctuation alone can cause a 10 m concrete slab to contract/expand by up to 4 mm, while typical building codes specify as-built tolerances of 5 to 10 mm to be within limits (Schweizerischer Ingenieur- und Architektenverein, 2016, Norm 414). A simple way to increase absolute accuracy would be to change the mechanical design such that the prism can be mounted closer and ideally rigidly to the end-effector, which is not easily possible due to visibility constraints. Likely the biggest improvement to both precision and accuracy is increasing the global yaw-estimate quality - but this could entail additional means such as fiducials, pre-scanning the construction site for SLAM, or similar, which is not always feasible.

5 Conclusion

We presented one of the first robust and precise aerial layouting systems capable of operating

under realistic conditions. In addition to our previous work, we contribute novel state estimation and navigation approaches that decouple local and global estimation and motion policies. The chosen approach has been resilient over hours of flight time, even when individual sensors failed. Through a comprehensive high-precision evaluation, we showed that the system marks at a very high relative precision of 1.5 mm and an absolute accuracy of 5.5 mm. The presented approach is suitable for highly precise and robust applications on real construction sites.

References

- Bodie K, Brunner M, Pantic M, et al (2019) An omnidirectional aerial manipulation platform for contact-based inspection. In: *Robotics: Science and Systems (RSS)*, <https://doi.org/10.15607/rss.2019.xv.019>
- Boston Consulting Group (2018) Getting ready for robotics in property development and building. Retrieved January 8, 2023, from <https://www.bcg.com/publications/2018/robotics-property-development-building>
- Chermprayong P, Zhang K, Xiao F, et al (2019) An integrated delta manipulator for aerial repair: A new aerial robotic system. *IEEE Robotics & Automation Magazine* 26(1):54–66. <https://doi.org/10.1109/MRA.2018.2888911>
- Dellaert F, Contributors G (2022) borglab/gtsam. <https://doi.org/10.5281/zenodo.5794541>, URL <https://github.com/borglab/gtsam>
- Forster C, Carlone L, Dellaert F, et al (2016) On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics* 33(1):1–21. <https://doi.org/10.1109/TRO.2016.2597321>
- Hu F, Jing X (2018) A 6-DOF passive vibration isolator based on Stewart structure with X-shaped legs. *Nonlinear Dynamics* 91:157–185. <https://doi.org/10.1007/s11071-017-3862-x>
- Indelman V, Williams S, Kaess M, et al (2013) Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems* 61(8):721–738. <https://doi.org/https://doi.org/10.1016/j.robot.2013.05.001>
- Lanegger C, Ruggia M, Tognon M, et al (2022) Aerial Layouting: Design and Control of a Compliant and Actuated End-Effector for Precise In-flight Marking on Ceilings. In: *Proceedings of Robotics: Science and Systems (RSS)*, <https://doi.org/10.15607/RSS.2022.XVIII.073>
- Lynen S, Achtelik MW, Weiss S, et al (2013) A robust and modular multi-sensor fusion approach applied to mav navigation. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp 3923–3929, <https://doi.org/10.1109/IROS.2013.6696917>
- Nava G, Sablé Q, Tognon M, et al (2020) Direct force feedback control and online multi-task optimization for aerial manipulators. *IEEE Robotics and Automation Letters* 5(2):331–338. <https://doi.org/10.1109/LRA.2019.2958473>
- Nubert J, Khattak S, Hutter M (2022) Graph-based multi-sensor fusion for consistent localization of autonomous construction robots. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp 10,048–10,054, <https://doi.org/10.1109/ICRA46639.2022.9812386>
- Ollero A, Tognon M, Suarez A, et al (2021) Past, present, and future of aerial robotic manipulators. *IEEE Transactions on Robotics* pp 1–20. <https://doi.org/10.1109/TRO.2021.3084395>
- OpenCV (2015) Open source computer vision library. URL <https://opencv.org/>
- Ratliff ND, Issac J, Kappler D, et al (2018) Riemannian motion policies. *arXiv preprint* <https://doi.org/10.48550/ARXIV.1801.02854>
- Sanchez-Cuevas PJ, Gonzalez-Morgado A, Cortes N, et al (2020) Fully-actuated aerial manipulator for infrastructure contact inspection: Design, modeling, localization, and control. *Sensors* 20(17). <https://doi.org/10.3390/s20174708>

- Sandy T, Stadelmann L, Kerscher S, et al (2019) Confusion: Sensor fusion for complex robotic systems using nonlinear optimization. *IEEE Robotics and Automation Letters* 4(2):1093–1100. <https://doi.org/10.1109/LRA.2019.2894168>
- Schweizerischer Ingenieur- und Architektenverein (2016) Sia norm 414/2:2016. Obtainable at <https://www.sia.ch>
- Stephens B, Nguyen HN, Hamaza S, et al (2023) 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>
- Stewart D (1965) A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers* 180(1):371–386. https://doi.org/10.1243/PIME_PROC_1965_180_029_02
- Tognon M, Tello-Chavez HA, Gasparin E, et al (2019) A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants. *IEEE Robotics and Automation Letters* 4(2). <https://doi.org/10.1109/LRA.2019.2895880>
- Trujillo MA, Martínez-De Dios JR, Martín C, et al (2019) Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry. *Sensors* 19(6). <https://doi.org/10.3390/S19061305>
- Tzoumanikas D, Graule F, Yan Q, et al (2020) Aerial manipulation using hybrid force and position nmpc applied to aerial writing. In: *Robotics: Science and Systems (RSS)*, <https://doi.org/10.15607/RSS.2020.XVI.046>
- Watson R, Kamel M, Zhang D, et al (2021) Dry coupled ultrasonic non-destructive evaluation using an over-actuated unmanned aerial vehicle. *IEEE Transactions on Automation Science and Engineering* pp 1–16. <https://doi.org/10.1109/TASE.2021.3094966>