Automation in Construction 135 (2022) 104112

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

# Tag-based visual-inertial localization of unmanned aerial vehicles in indoor construction environments using an on-manifold extended Kalman filter

Navid Kayhani [a,b,*], Wenda Zhao [b], Brenda McCabe [a], Angela P. Schoellig [b]

[a] Department of Civil and Mineral Engineering, University of Toronto, Toronto, ON M5S 1A4, Canada
[b] Dynamic Systems Lab, Institute for Aerospace Studies, University of Toronto, Toronto, ON M3H 5T6, Canada

ARTICLE INFO

*Keywords:*
Indoor localization
Unmanned aerial vehicle
Extended Kalman filter
SE(3)
On-manifold state estimation
Autonomous navigation
Building information model
Construction robotics
AprilTag

ABSTRACT

Automated visual data collection using autonomous unmanned aerial vehicles (UAVs) can improve the accessibility and accuracy of the frequent data required for indoor construction inspections and tracking. However, robust localization, as a critical enabler for autonomy, is challenging in ever-changing, cluttered, GPS-denied indoor construction environments. Rapid alterations and repetitive low-texture areas on indoor construction sites jeopardize the reliability of typical vision-based solutions. This research proposes a tag-based visual-inertial localization method for off-the-shelf UAVs with only a camera and an inertial measurement unit (IMU). Given that tag locations are known in the BIM, the proposed method estimates the UAV's global pose by fusing inertial data and tag measurements using an on-manifold extended Kalman filter (EKF). The root-mean-square error (RMSE) achieved in our experiments in laboratory and simulation, being as low as $2-5$ *cm*, indicates the potential of deploying the proposed method for autonomous navigation of low-cost UAVs in indoor construction environments.

## 1. Introduction

Over the past few decades, computer-vision-based solutions have shown promising results in automating indoor construction progress monitoring and inspection tasks [1,2]. Despite these advancements, the required visual data are still mainly captured manually [3], which is costly and tedious, especially in large/high-rise buildings. Although automated fixed cameras can be helpful for outdoor visual data collection, their effectiveness reduces on most indoor construction sites as the layout changes with construction progress [4]. Thus, there is a need for an automated mobile visual data capture solution for indoor construction environments.

Mobile robots, as sensor-carrying platforms, have attracted increasing attention in the construction community [5–12]. For instance, rotary unmanned aerial vehicles (hereafter UAVs) equipped with an onboard camera have shown great potential in automated visual data collection applications in both indoor [11,12] and outdoor construction environments [7,9]. They can provide high-resolution images from versatile locations and fields of view in a fast and cost-efficient manner [12]. However, these UAV-based solutions still rely on teleoperation for indoor navigation and data capture [11,12].

Localization is a crucial enabler for the deployment of autonomous mobile robots. Autonomous UAVs may take advantage of the Global Positioning System (GPS) for localization outdoors. However, GPS signals are unreliable in indoor settings. Specifically, indoor construction sites are cluttered and dynamically changing environments, which causes many more challenges in UAVs' localization and autonomous navigation.

A common technique for enabling autonomous navigation is incrementally mapping the environment and simultaneously localizing the platform within the map. These techniques are referred to as simultaneous localization and mapping (SLAM)-based approaches. Solving the SLAM problem requires a computational process for locally building a map while relatively localizing the agent and another parallel process for recognizing a formerly visited location for updating the map and correcting the errors, also known as loop closure. SLAM-based methods can handle unstructured and unknown environments [13]. However, these techniques are often considered memory and computationally expensive in large environments where revisiting locations for loop closure becomes a technical and practical challenge [14].

Many state-of-the-art robotic platforms relying on SLAM-based methods use high-fidelity environment maps generated upfront to

* Corresponding author.
*E-mail addresses:* navid.kayhani@mail.utoronto.ca (N. Kayhani), wenda.zhao@mail.utoronto.ca (W. Zhao), brenda.mccabe@utoronto.ca (B. McCabe), schoellig@utias.utoronto.ca (A.P. Schoellig).

https://doi.org/10.1016/j.autcon.2021.104112
Received 19 June 2021; Received in revised form 12 November 2021; Accepted 22 December 2021
Available online 10 January 2022

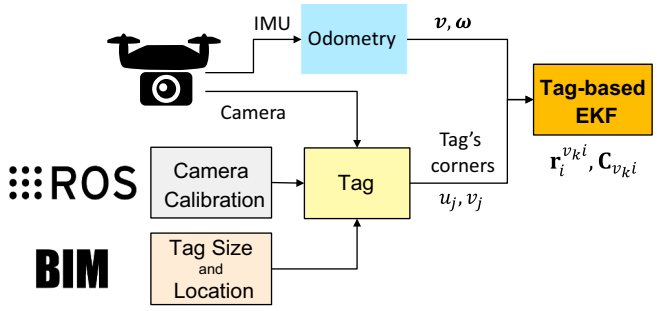



**Fig. 1.** Tag-based visual-inertial indoor localization: a schematic overview.

reduce the computations during autonomous missions. This approach requires an operator to manually navigate the mobile robot to build a map of the environment for later autonomous operations [6,10,15]. However, since construction environments alter fairly rapidly, frequent teleoperated mapping sessions may be required. Another challenge in such ever-changing environments is the potential loss of track due to dynamic and temporary objects. Moreover, maintaining large maps requires computational and storage resources, which are highly limited on aerial robots.

SLAM can be mainly divided into light-detection-and-ranging-based (LiDAR-based) and vision-based (V-SLAM) categories depending on the onboard sensors. As LiDARs are inefficient in power consumption and cost, they are hardly found on low-cost UAVs [16]. However, RGB cameras are ubiquitous. They are lightweight, inexpensive sensors with low power consumption that provide rich environmental information.

Vision-based SLAM (V-SLAM) techniques have been successfully adapted in autonomous navigation of UAVs in GPS-denied environments [13,17]. Nevertheless, they face particular challenges on construction sites. Indoor construction environments often include many low-texture and repetitive areas (e.g., white walls and studs). This factor reduces the robustness of the estimation techniques relying on natural landmarks (e.g., keypoints or textures) [14], including feature-based V-SLAM. Perceptual aliasing and feature scarcity in indoor construction environments also decrease the effectiveness of loop closing approaches [18]. Without a reliable loop closure, SLAM reduces to odometry, which drifts over time and is unreliable for consistent long-term localization.

In addition to technical challenges in indoor localization and autonomy, the deployment of many cutting-edge robotic solutions that have been proposed in academia and the industry require special consideration. One of the main factors is the cost of these solutions, limiting their scalability and applicability in practice. The majority of commercial solutions and proposed custom-built prototypes in the construction literature [5,6,19] are costly. Commercial products may cost hundreds of thousands of dollars per platform [20], while custom-built platforms require robot assembly expertise.

Despite these challenges, some characteristics in construction environments can be helpful in indoor localization and autonomous navigation. Indoor construction sites are not fully unknown, as a progressively updated 4D building information model (BIM) provides valuable prior knowledge about the actual progress and the planned layout alterations. Moreover, construction processes and practices such as frequent indoor layout surveying can be beneficial in indoor localization [21].

To enable autonomous navigation of low-priced UAVs in indoor construction environments, this research aims to present an online global localization method that: (1) is inexpensive; (2) requires low computation and storage resources; (3) can be used by multiple platforms; and (4) can handle low-texture, ever-changing construction environments. Thus, this paper proposes a low-cost, versatile, and lightweight localization method that provides online six-degree-of-freedom (6-DoF) global pose estimates for a wide range of platforms, including commercial compact UAVs, with just an RGB camera and an inertial measurement unit (IMU).

IMUs are proprioceptive sensors measuring external accelerations and angular velocities. Due to their low signal-to-noise ratio, IMUs alone are subject to accumulative errors in motion estimates. On the other hand, IMUs are scene independent, have a high output rate (up to 1000 Hz), and provide scale information [16]. Therefore, IMUs are often used in combination with cameras to produce robust state estimates. Although stereo cameras can provide depth information for nearby objects, monocular cameras are more common in aerial robots due to their low weight and power consumption [16].

As illustrated in Fig. 1, the proposed tag-based visual-inertial indoor localization method estimates the full 6-DoF global pose of a UAV with a minimum suite of onboard sensors, i.e., a monocular camera and an IMU, in real-time. Given that tag locations are known in the BIM coordinate system, it fuses inertial odometry velocity data with tag measurements using an on-manifold extended Kalman filter (EKF). The proposed solution can ultimately enable low-cost off-the-shelf UAVs to navigate autonomously in GPS-denied indoor construction environments.

The theoretical contribution and novelty of our work presented herein is the proposed on-manifold formulation for tag-based visual-inertial localization. Our effort to formulate the estimation problem at hand properly is motivated by the necessary accuracy, consistency, and stability in construction applications. The proposed formulation properly considers the manifold structure of the pose and the rotation groups in 3D and carefully deals with the representation and propagation of uncertainty over time. These are crucial theoretical aspects for achieving these goals, especially in 3D space. We also opted to incorporate tag corner measurements in our tightly coupled on-manifold formulation instead of direct camera-to-tag transforms, which results in more stability and higher accuracy of estimates.

Collecting ground truth data in a large construction setting is not a trivial task, making the validation of localization methods extremely challenging in these environments. Moreover, conducting experiments on an actual construction site raises many safety and logistics issues. The third contribution of this work is developing and deploying a BIM-enabled, photo-realistic simulation environment, which allows for safe and efficient experiments supported by absolute ground truth data.

Therefore, the main contributions of this work can be summarized as:

1. Presenting a low-cost, lightweight, versatile, tag-based visual-inertial localization method for UAVs equipped with a minimum sensor suite of an IMU and a monocular camera using AprilTags.
2. Proposing an on-manifold extended Kalman filter formulation for tag-based visual-inertial localization that properly addresses the topological structure of the rotation and the pose groups and the associated uncertainty propagations.
3. Developing a BIM-enabled, photo-realistic simulation environment for preliminary validations and experiments.

A compact, inexpensive, commercially available UAV, *Parrot Bebop2* [58], is used for method implementation. All developments are in the Robotic Operation System (ROS) [22] as an open-source standard communication platform. Experiments are designed and conducted in a laboratory setting (an arena equipped with a motion capture system to provide ground truth data) and a developed simulation environment to evaluate the method's performance. The results demonstrate the feasibility of real-time and accurate localization of UAVs, with an IMU and an RGB camera, for autonomous navigation in indoor construction settings.

## 2. Background

### 2.1. Indoor localization of autonomous mobile robots

Autonomous mobile robots may rely on robust and accurate external

**Table 1**
Reviewed literature on automated robotic data capture solutions in GPS-denied construction environments.

| Ref. | Platform | Application | Sensor modality | Indoor | Localization method | Localization validation | Remarks |
|---|---|---|---|---|---|---|---|
| Peel et al. (2018) [40] | Custom-built ground robot | Bridge inspection | LiDAR | No | • Teleoperated mapping using Hector-SLAM.<br>• AMCL for localization. | No | No GPS |
| Adán et al. (2020) [47] | Custom-built ground robot | Semantic modeling | Vision + LiDAR | Yes | • Teleoperated mapping using SLAM<br>• AMCL for Localization. | No | Existing buildings |
| Mantha et al. (2018) [48] | Custom-built ground robot | Building retrofit performance | Vision | Yes | • AprilTags for high-level commanding and occasional drift corrections in an open-loop control approach. | No | Existing buildings |
| Kim et al. (2018) [19] | Custom-built ground robot | 3D reconstruction | LiDAR | Yes | • Teleoperated mapping and localization using Hector-SLAM. | No | |
| Asadi et al. (2018) [6] | Custom-built ground robot | Construction progress monitoring | Vision | Yes | • Teleoperated mapping and localization using ORB-SLAM. | No | |
| Xu et al. (2019) [10] | Custom-built ground robot | Real-time locating applications | Kinect | Yes | • Teleoperated mapping and localization using a modified ORB-SLAM. | 2-DoF position (at select locations) | AprilTags for localization validations. |
| Ibrahim et al. (2019) [15] | Custom-built ground robot | Construction progress monitoring | Vision + LiDAR | Yes | • Teleoperated mapping and localization using Hector-SLAM | No | |
| Asadi et al. (2020) [5] | Custom-built ground and aerial robot (blimp) | Construction progress monitoring | Vision + LiDAR | Yes | • VINS-mono for blimp localization.<br>• Teleoperated mapping and localization using RTAP-MAP. | No | Fiducials for relative position estimation comparisons. |
| Hamledari et al. (2017) [11] | Low-cost commercial aerial robot | Construction progress monitoring | Vision | Yes | N/A | N/A | Teleoperated UAV (on-site). |
| **Ours** | Low-cost commercial aerial robot | Construction progress monitoring | Vision + IMU | Yes | • Tag-based visual-inertial localization using an on-manifold EKF | 6-DoF pose (simulation and laboratory) | Lightweight, online, global localization. |

localization sources, such as GPS outdoors and motion capture systems (e.g., Vicon) indoors. Even though motion capture systems can provide a millimeter level of accuracy, they cover a limited area, require repetitive calibrations, and are incredibly costly. Although motion capture systems are usually a source of ground truth in indoor laboratory settings, they are impractical for indoor construction applications.

Wave-based localization techniques such as wireless local area networks (WLAN) [23], radio-frequency identification (RFID) [24,25], and ultra-wideband (UWB) [26,27] have their own drawbacks. Due to wave interferences with metallic materials (e.g., RFID [28] and UWB [29]), propagation condition variations (e.g., UWB [30]), and more importantly, their inadequate accuracy [31], they are still unsuitable for indoor navigation of UAVs during construction.

Autonomous robots operating in GPS-denied environments may rely on dead reckoning or odometry techniques for localization. Visual and visual-inertial odometry (VO [32] and VIO [33,34]) techniques are the most popular methods for local pose estimation in aerial robots [16]. However, without corrections, odometry estimates are subject to quick drifts and are only suitable for short-time relative localization. Revisiting locations and loop detection in SLAM-based solutions can correct this drift.

Solving the SLAM problem has been the focus of many studies in the last few decades [35]. Monocular visual SLAM was first solved using an EKF and Shi-Tomasi points by tracking key-points in subsequent images [36]. Due to their high computational cost, alternative approaches were proposed to solve the problem more efficiently and for large environments. Above all, two central notions and their variants attracted the scholars' attention, namely sliding-window filters and keyframe-based pose-graph optimization [33,37]. Sliding-window filters iterate over a window of time-steps and slide the window along. By discarding the intermediate frames, keyframe-based approaches estimate the map using a few selected frames. Some analysis showed that keyframe-based techniques are more accurate than filtering for the same computational cost [38]. In contrast, others believe that filter-based methods such as the smooth variable structure filter (SVSF) provide more stable estimates [39–41] as they are robust to modeling uncertainties and errors.

In the context of autonomous navigation, most of the existing localization solutions rely on accurate maps generated upfront via SLAM techniques [39]. The mapping process is typically performed by robot teleoperation in the workspace (e.g., [5,40]), which poses time and cost constraints in large, ever-changing indoor construction settings. For a large construction site, in addition to repetitive mapping sessions, active map maintenance is also essential. Active maintenance of these maps requires computation and storage resources, which increases the payload, reducing aerial robots' flight time. Additionally, loop closure in SLAM requires revisiting locations and accurate place recognition. However, poorly-featured or repetitive areas on indoor sites can impact the global consistency and robustness of the maps and place recognition algorithms. Thus, substantial knowledge is required to evaluate whether the generated map quality is adequate for the desired application before any autonomous mission. In contrast, the proposed tag-based visual-inertial localization approach requires no mapping sessions. Moreover, the pre-installed AprilTags allow reliable global visual measurements, enabling drift-free and lightweight global indoor localization.

### 2.2. Robotic data collection solutions in construction

In the construction community, a rapidly growing research stream has focused on the applications of autonomous mobile robots in automated data collection. There has been a considerable amount of research on the deployment of mobile robots for outdoor data collection applications. Infrastructure inspections [7,41,42], earthwork surveying [9], quality control [43], safety inspections [44,45] are some examples of the applications that have been investigated for UAV-based outdoor visual data collection. In these studies, UAVs either were remotely controlled or relied on GPS signals for autonomous flight.

Deploying autonomous ground robots for indoor environmental air quality [46], semantic modeling [47], as well as simulation and evaluation of building retrofit performance [48] are examples of automated robotic data collection solutions in existing buildings. Above all, AprilTags were used for high-level commanding and occasional drift corrections in an open-loop control approach in [48]. In the context of

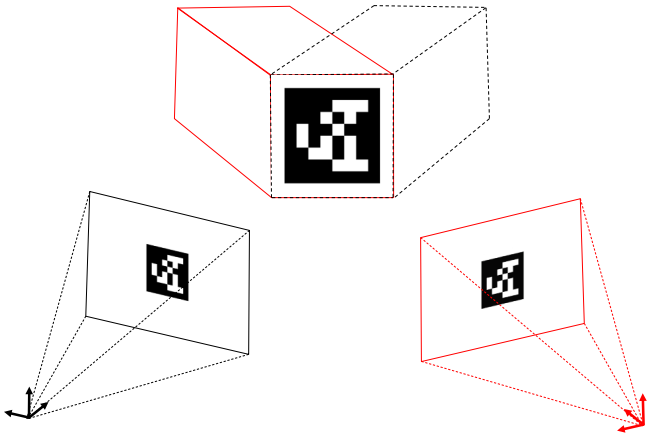


**Fig. 2.** Ambiguity in recovering the relative camera-tag orientation (solid red and dashed black cubes). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

under-construction buildings, the proposed automated indoor data collection solutions in the reviewed literature heavily relied on custom-built ground robots. In virtually all these studies [5,10,15,19,40,49] (see Table 1 for more details), the robot localization was based on a pre-built map generated using SLAM algorithms. These maps were generated using teleoperation of the platform in the navigable area. In aerial robots with limited onboard resources (i.e., storage, computation, and power), maintaining large maps reduces their functionality and effectiveness (e.g., reduced flight time). Moreover, the mapping process needs to be repeated as the construction progresses, which is time-demanding, costly, and tedious in applications where frequent data are expected (e.g., progress monitoring). Except for [10] in Table 1, in which AprilTags were used for quantitative evaluations of position estimates in select discrete locations, none of the reviewed studies provides localization error assessments against any ground truth data.

### 2.3. Tag-based localization

Visual fiducial markers (hereafter tags), such as ARTag [50], AprilTag [51,52], and CalTag [53], are planar artificial landmarks consisting of patterns. In particular, AprilTags are square-shaped payload tags that provide robust data association correspondences and can be identified even if partially occluded [51]. These paper-printable tags have an external black border and a unique inner binary code pattern for robust detections. Each detected tag provides four pairs of corner point correspondences. In theory, the relative camera pose can be uniquely obtained by finding the homography transformation between the 3D corner points in the tag reference frame and their corresponding 2D projections in the image. In practice, orientation ambiguity [54] may happen when the tag's corners are close in the image (e.g., imaging small tags or tags at a distance significantly greater than the camera's focal length), where two possible solutions may exist [55]. Fig. 2 illustrates how the same projection can come from two different relative camera poses, leading to ambiguity in orientation estimates. In ideal scenarios with noise-less corner point projections or when the difference in reprojection errors is considerable, the solution with the lowest error can be identified as the correct solution. However, due to noise and imperfections, the correct solutions cannot be guaranteed in practice when the difference in errors is small [55], leading to ambiguity and jumps in the estimated pose.

Tag-based systems are composed of distinguishable planar tags as landmarks and a detection and identification algorithm. Planar tags are widely used for local estimations ranging from map initializations to UAV landing. For instance, a tag-based precision landing on a recharging station for automated energy replenishment of micro UAVs was proposed in [56]. However, few studies have focused on using tags as a significant part of their localization solution [57,58].

For instance, an offline single-camera method was proposed in [55] for creating a map of tags placed in the workspace while localizing the camera. They considered the ambiguity problem in estimating camera pose from co-planar points (for a detailed discussion on the ambiguity problem, refer to [54]). However, the method in [55] is not suitable for real-time applications such as indoor navigation. In a subsequent study, they [59] proposed a real-time solution to the problem of simultaneously localizing the camera and mapping planar tags, given that at least two tags are visible in each image. Lately, by coupling key-points and fiducial markers, they introduced UcoSLAM [60] to solve the same problem. However, these methods only rely on vision data, which may fail to provide robust estimates facing occlusion or motion blur challenges. In another research stream, by adding extra sensors [61], such as IMU [62], UWB [63], and RGB-depth sensor [64], a sensor fusion approach mainly using EKF was proposed. In a similar study [62], a generic visual-inertial EKF-SLAM based on AprilTags was proposed. Although the full pose and the velocities were incorporated into the state vector, their suggested filter-based SLAM approach can store a limited number of tags in the map to be computationally tractable in real-time. This constraint limits the effectiveness of their method in large environments, including indoor construction sites.

In summary, despite their low price, planar fiducials such as AprilTags can be easily detected and robustly differentiated from one another and among other features in the scene, which can be helpful in low-texture and low-structure areas of indoor construction sites. AprilTags may be subject to occlusions and damage in ever-changing indoor construction sites and require manual placement/replacements [21]. Therefore, the localization method cannot solely rely on tags. On the other hand, tags with known sizes are long-term visual references that provide relative pose estimates and scale in images captured by a calibrated monocular camera. Once placed in the environment, not only can tags support the localization of multiple platforms (e.g., aerial and ground robots) or handheld devices (i.e., smartphones) during data collection, they can be helpful in contextualization and data association. They can also be scanned to provide the site personnel with different forms of location-based information. Thus, adding these low-cost, easy-to-deploy, and easy-to-install tags to the site can benefit many applications in construction.

## 3. Mathematical preliminaries

In this section, a brief review of notations and essential operators in the special Euclidean group in 3D ($SE(3)$) are presented (more detailed explanations can be found in [65]). Then, the reference frames involved in the problem are introduced.

### 3.1. SE(3): a brief notation overview

Formally, the 3D special Euclidean group is a pose (i.e., translation and rotation) representation in the form of valid $4 \times 4$ transformation matrices [65]:

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \middle| \mathbf{C} \in SO(3), \mathbf{r} \in \mathbb{R}^3 \right\} \tag{1}$$

where $\mathbf{r}$ is a 3D translation ($3 \times 1$) vector, and $\mathbf{C}$ is the standard $3 \times 3$ rotation matrix in the special orthogonal group of $SO(3)$ that represent rotations in 3D and is defined as:

$$SO(3) = \left\{ \mathbf{C} \in \mathbb{R}^{3\times3} \middle| \mathbf{C}\mathbf{C}^T = \mathbf{1}, det(\mathbf{C}) = 1 \right\} \tag{2}$$

Topologically, both $SE(3)$ and $SO(3)$ can be viewed as smooth manifolds of matrix Lie groups. The matrix Lie groups of $SE(3)$ and $SO(3)$ have corresponding tangent spaces that are referred to as the Lie algebra of $se(3)$ and $so(3)$, respectively. A matrix Lie group and the

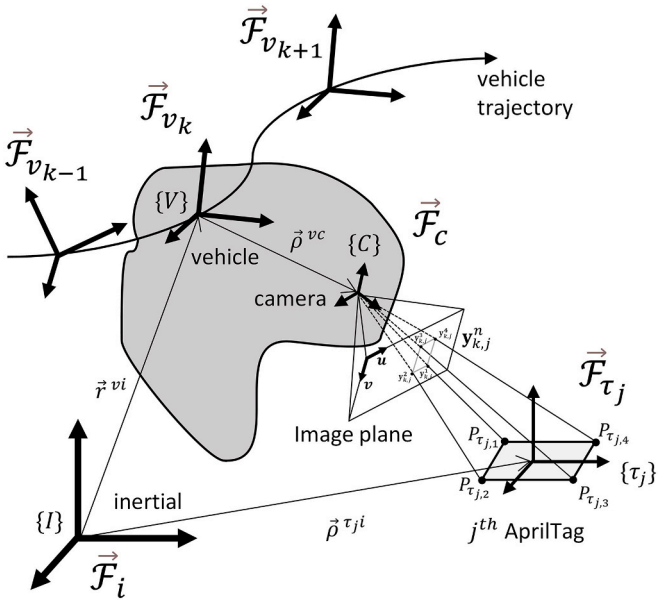



**Fig. 3.** Reference frames involved in the tag-based pose estimation problem.

corresponding Lie algebra are related through an exponential and logarithmic mapping. Following the notation used in [65], the pose of the vehicle at time $k$ with respect to a fixed inertial frame can be defined as (Fig. 3):

$$\mathbf{T}_{v_k i} = \mathbf{T}_k = \begin{bmatrix} \mathbf{C}_{v_k i} & -\mathbf{C}_{v_k i}\mathbf{r}_i^{v_k i} \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3) \tag{3}$$

which is a $4 \times 4$ transformation matrix that transforms points in the coordinate frame $\overrightarrow{\mathcal{F}}_i$ (inertial frame) to $\overrightarrow{\mathcal{F}}_{v_k}$ (vehicle frame at time $k$). This transformation matrix consists of a $3 \times 3$ rotation matrix, $\mathbf{C}_{vki} \in SO(3)$, which indicates the rotation of vehicle frame $\overrightarrow{\mathcal{F}}_{v_k}$ with respect to the inertial frame $\overrightarrow{\mathcal{F}}_i$, and $\mathbf{r}_i^{v_k i}$, indicating the translation of the vehicle frame $\overrightarrow{\mathcal{F}}_{v_k}$ with respect to the inertial frame $\overrightarrow{\mathcal{F}}_i$, expressed in inertial frame $\overrightarrow{\mathcal{F}}_i$.

Using the exponential mapping from $se(3)$ to $SE(3)$, we have:

$$\mathbf{T} = exp(\boldsymbol{\xi}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\xi}^\wedge)^n \tag{4}$$

where the translational $\boldsymbol{\rho} \in \mathbb{R}^3$ and rotational $\boldsymbol{\phi} \in \mathbb{R}^3$ components are stacked in a pose vector $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^{6\times 1}$, and $\boldsymbol{\xi}^\wedge$ is a $4 \times 4$ matrix in the tangent space $se(3)$:

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \in se(3) \tag{5a}$$

$$\boldsymbol{\phi}^\wedge = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix} \in so(3) \tag{5b}$$

where $\boldsymbol{\phi}^\wedge \in so(3)$ is the Lie algebra associated with $SO(3)$ and equivalent to the rotation vector ($\boldsymbol{\phi} \in \mathbb{R}^3$) expressed in the skew-symmetric matrix format.

We can also go in the other direction, yet not uniquely using the logarithmic map:

$$\boldsymbol{\xi} = \ln(\mathbf{T})^\vee \tag{6}$$

The left perturbed scheme [65] in $SE(3)$ is used to express an uncertain transform ($\mathbf{T}$) as a large, noise-free nominal (i.e., mean) component ($\overline{\mathbf{T}}$) and a small, zero mean, noisy, perturbation component ($\exp(\boldsymbol{\epsilon}^\wedge)$). Assuming a normal distribution for perturbation, we have:

$$\mathbf{T} = exp(\boldsymbol{\epsilon}^\wedge)\overline{\mathbf{T}}, \boldsymbol{\epsilon} \in \mathbb{R}^6 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{7}$$

Another beneficial $6 \times 6$ linear transform is the adjoint matrix of an element of $SE(3)$ [65]:

$$Ad(\mathbf{T}) = Ad\left(\begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{C} & \mathbf{r}^\wedge\mathbf{C} \\ \mathbf{0}^T & \mathbf{C} \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{8}$$

### 3.2. Reference frames

In this subsection, the reference frames involved in the problem are introduced. Fig. 3 shows the reference frames at time $k$ along with the vehicle frame $\overrightarrow{\mathcal{F}}_v$ at time $k-1$ as well as $k+1$ to indicate the vehicle trajectory. As illustrated in Fig. 3, $\overrightarrow{\mathcal{F}}_i$ is the inertial frame, which is a fixed, global frame with known coordinates in the BIM coordinates system. The vehicle's pose, $\overrightarrow{r}^{vi}$ and $\mathbf{C}_{iv}$, will be estimated with respect to this frame. Moreover, the pose of tags, $\overrightarrow{\rho}^{\tau_j i}$ and $\mathbf{C}_{i\tau j}$, are also expressed in this frame. The pose of the tags in the inertial (BIM) reference frame $\overrightarrow{\mathcal{F}}_i$ is assumed a priori.

The tag size for a square planar tag is defined as its side length. For instance, for AprilTags, the tag size is the distance between the outer black edges, including the inner binary pattern and the external black border. Given the $j-th$ tag's size ($s_{\tau j}$) is known, the 3D coordinates of its corners in the tag frame $\overrightarrow{\mathcal{F}}_{\tau_j}$ can be written as:

$$P_{\tau_{j,1}} = \begin{bmatrix} -\frac{s_{\tau_j}}{2} & -\frac{s_{\tau_j}}{2} & 0 \end{bmatrix}^T \tag{9a}$$

$$P_{\tau_{j,2}} = \begin{bmatrix} \frac{s_{\tau_j}}{2} & -\frac{s_{\tau_j}}{2} & 0 \end{bmatrix}^T \tag{9b}$$

$$\mathrm{P}_{\tau_{j,3}} = \begin{bmatrix} \frac{\mathrm{s}_{\tau_j}}{2} & \frac{\mathrm{s}_{\tau_j}}{2} & 0 \end{bmatrix}^T \tag{9c}$$

$$\mathrm{P}_{\tau_{j,4}} = \begin{bmatrix} -\frac{\mathrm{s}_{\tau_j}}{2} & \frac{\mathrm{s}_{\tau_j}}{2} & 0 \end{bmatrix}^T \tag{9d}$$

The vehicle frame $\overrightarrow{\mathcal{F}}_v$ is rigidly attached to the vehicle base-link, where the IMU is located. The camera frame $\overrightarrow{\mathcal{F}}_c$ is attached to the vehicle's onboard camera. The fixed transformation between the vehicle frame and the camera frame is already determined by calibration.

## 4. Methodology

In this section, the overall methodology and derivations for the backbone of our estimation scheme are given. First, a formal problem formulation is presented using the notations introduced in the previous section. Then, the canonical EKF formulation for the proposed tag-based 6-DoF pose estimation method is delivered by providing the details for the motion and measurement models as well as error calculations.

### 4.1. Problem formulation

As a localization problem, the state we aim to estimate is the vehicle's poses along the entire trajectory. Similar to [65], we have:

$$\mathbf{x} = \left\{ \left\{ \mathbf{r}_i^{v_0 i}, \mathbf{C}_{v_0 i} \right\}, \left\{ \mathbf{r}_i^{v_1 i}, \mathbf{C}_{v_1 i} \right\}, \ldots, \left\{ \mathbf{r}_i^{v_K i}, \mathbf{C}_{v_K i} \right\} \right\} = \{\mathbf{T}_0, \mathbf{T}_1, \ldots, \mathbf{T}_K\} \tag{10}$$

where $\mathbf{T}_k \in SE(3)$ is defined in Eq. (3). Since localization can be seen as a pose tracking problem, along with the initial state $\mathbf{T}_0$, we assume the translational and rotational velocities (vehicle's twist $\boldsymbol{\varpi}$)) as the system inputs $\mathbf{v}$. The twist can be derived from the vehicle's IMU data or directly from other odometry sources (e.g., visual-inertial odometry). From now on, we assume it comes from a source of visual-inertial odometry for consistency. We have:

$$\mathbf{v} = \left\{ \check{\mathbf{T}}_0, \boldsymbol{\varpi}_1, \ldots, \boldsymbol{\varpi}_K \right\} \tag{11a}$$

where:

$$\boldsymbol{\varpi}_k = \begin{bmatrix} \mathbf{v}_{v_k}^{v_k i} \\ \boldsymbol{\omega}_{v_k}^{v_k i} \end{bmatrix}, \; k = 1, \ldots, K \tag{11b}$$

The set of all measurements at each time step over the entire trajectory can be encapsulated in **y:**

$$\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\} \tag{12a}$$

Although we may or may not observe all the tags in the workspace at time $k$, given the total number of tags is $M$, we can write:

$$\mathbf{y}_k = \left[\mathbf{y}_{k,1}, \mathbf{y}_{k,2}, \ldots, \mathbf{y}_{k,M}\right]^T \tag{12b}$$

where the measurement of tag $\tau_j$ at time step $k$ contains all four corner points' pixel coordinates on the image plane:

$$\mathbf{y}_{k,j} = \left[\mathbf{y}_{k,j}^1, \mathbf{y}_{k,j}^2, \mathbf{y}_{k,j}^3, \mathbf{y}_{k,j}^4\right]^T = \left[u_{k,j}^1, v_{k,j}^1, \ldots, u_{k,j}^4, v_{k,j}^4\right]^T \tag{12c}$$

### 4.2. Motion model

Given the perturbation scheme in Eq. (7) and an additive perturbation for the vehicle's twist, we can write the nominal and perturbation kinematics as follows [65]:

Nominal kinematics for propagating the mean:

$$\overline{\mathbf{T}}_k = \Xi_k \, \overline{\mathbf{T}}_{k-1} \tag{13a}$$

Perturbation kinematics for propagating the covariance:

$$\delta\boldsymbol{\xi}_k = \mathrm{Ad}(\Xi_k) \, \delta\boldsymbol{\xi}_{k-1} + \mathbf{w}_k, \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \tag{13b}$$

Here we have:

$$\Xi_k = \overline{\mathbf{T}}_{v_k v_{k-1}} = \begin{bmatrix} \boldsymbol{\Psi}_{v_k v_{k-1}} & -\boldsymbol{\Psi}_{v_k v_{k-1}} \mathbf{d}_{v_{k-1}}^{v_k v_{k-1}} \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3) \tag{14}$$

where [65]:

$$\mathbf{d}_{v_{k-1}}^{v_k v_{k-1}} = \mathbf{v}_{v_{k-1}}^{v_{k-1} i} \Delta t_k + \delta\mathbf{d}_k \tag{15}$$

$$\Delta \mathrm{t}_k = \mathrm{t}_k - \mathrm{t}_{k-1} \tag{16}$$

$$\boldsymbol{\Psi}_{v_k v_{k-1}} = \cos\psi_k \mathbf{1} + (1 - \cos\psi_k) \left(\frac{\boldsymbol{\psi}_k}{\psi_k}\right) \left(\frac{\boldsymbol{\psi}_k}{\psi_k}\right)^T - \sin\psi_k \left(\frac{\boldsymbol{\psi}_k}{\psi_k}\right)^\times \tag{17a}$$

$$\boldsymbol{\psi}_k = \boldsymbol{\omega}_{v_{k-1}}^{v_{k-1} i} \Delta \mathrm{t}_k + \delta\boldsymbol{\psi}_k \tag{17b}$$

$$\psi_k = |\boldsymbol{\psi}_k| \tag{17c}$$

Translational and rotational velocities come from an onboard visual-inertial odometry source as discussed earlier:

$\mathbf{v}_{v_k}^{v_k i}$: Translational velocity of vehicle frame with respect to inertial frame, expressed in vehicle frame.

$\boldsymbol{\omega}_{v_k}^{v_k i}$: Rotational velocity of vehicle frame with respect to inertial frame, expressed in vehicle frame.

$\boldsymbol{\Psi}_{v_k v_{k-1}}$: Rotation (matrix) of vehicle frame with respect to the previous time step.

$\mathbf{d}_{v_{k-1}}^{v_k v_{k-1}}$: Translation (vector) of vehicle frame with respect to the previous time step, expressed in the previous time step frame.

$\delta\mathbf{d}_k$: Translational component of the process noise.

$\delta\boldsymbol{\psi}_k$: Rotational component of the process noise.

$$\mathbf{Q}_k = Var\left(\begin{bmatrix} \delta\mathbf{d}_k \\ \delta\boldsymbol{\psi}_k \end{bmatrix}\right) = \Delta t_k{}^2 \begin{bmatrix} \boldsymbol{\sigma}_v^2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma}_\omega^2 \end{bmatrix} \tag{18}$$

where the diagonal values $\boldsymbol{\sigma}_v{}^2$ and $\boldsymbol{\sigma}_\omega{}^2$ can be estimated from the twist error estimation based on ground-truth data in advance. It is also possible to leave these quantities as tunning parameters.

### 4.3. Measurement model

Tag reading measurements are incorporated to update (correct) predictions from the twist inputs (dead reckoning). As mentioned earlier, the detector provides a 6-DoF pose estimate of tag with respect to camera frame from a single image. However, instead of directly using the relative camera-tag pose as measurements, we reproject the tag corners onto the frontal image plane and consider the four corresponding pixel coordinates as pixel-level measurements. The main gain of this approach is that the noise can be applied to the pixel location of detected tag corners [62]. Since noise behavior highly depends on the geometry of the projected tag in the image, fitting a proper noise model to the relative pose provided by the detector is not a trivial task. Modeling the noise becomes more complex when the tag image is small, and the corners are too close to one another (e.g., when the tag is rotated about its $y$-axis and located far away relative to the camera focal length). The noise magnitude depends on the relative location and orientation of tags in the camera frame as well as the camera lens parameters (i.e., camera intrinsics). However, the pixel-level noise on the reprojected tag corners can be assumed the same for all tag measurements and is less dependent on the camera-tag relative configuration [62]. Instead of directly using the relative pose provided by the detector, using corner points as measurements leads to a tightly coupled fusion approach, generally leading to better estimation results [17].

The 3D coordinates of the $n$-th corner point of $j$-th tag expressed in camera frame $\mathbf{p}_{c_k}^{p_{\tau j}, {}^{n}c_k}$ can be written as:

$$\mathbf{p}_{c_k}^{p_{\tau_j,n} c_k} = \mathbf{D}^T \mathbf{T}_{cv} \mathbf{T}_{v_k i} \mathbf{p}_{\tau_j,n} \tag{19}$$

where:

$$\mathbf{p}_{\tau_j,n} = \mathbf{T}_{i\,\tau_j} \mathbf{P}_{\tau_j,n} = \mathbf{T}_{\tau_j\,i}^{-1} \mathbf{P}_{\tau_j,n} \tag{20}$$

$$\mathbf{P}_{\tau_j,n} = \begin{bmatrix} P_{\tau_{j,n}} \\ 1 \end{bmatrix} \tag{21}$$

$$\mathbf{T}_{cv} = \begin{bmatrix} \mathbf{C}_{cv} & -\mathbf{C}_{cv} \rho_v^{cv} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{22}$$

$$\mathbf{D}^T = [\mathbf{1}_3 | \mathbf{0}_{3\times 1}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{23}$$

The $j$-th tag pose in the inertial frame and accordingly the corresponding transformation matrix $\mathbf{T}_{\tau j}$ is known as a priori. In this derivation, we preserve the ideal assumption that $\mathbf{T}_{\tau j}$ is not subject to any uncertainty. Additionally, $\mathbf{T}_{cv}$ is determined from calibration in advance. Finally, $\mathbf{D}^T$ is a dilated identity matrix padded with a column of zeros on the right to refine the matrix dimensions.

If we define $z_k^{\tau_j,n}(\boldsymbol{x}) = \mathbf{p}_{c_k}^{p_{\tau_j,n} c_k} = [X \quad Y \quad Z]^T$, where $\boldsymbol{x}$ is the state to be estimated ($\mathbf{T}_{vki}$) and $s(.)$ is a pinhole camera model that projects $\mathbf{p}_{ck}^{p_{tj},\,{}^{n}c_k}$ into a rectified image (lens distortion is previously handled and the input image is already rectified), we have:

$$\mathbf{y}_{k,j}^n = s\left(z_k^{\tau_j,n}(\boldsymbol{x})\right) = \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{D}_\mathbf{p} \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \frac{1}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \delta\boldsymbol{n}_{k,j}^n, \delta\boldsymbol{n}_{k,j}^n \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{k,j}\right) \tag{24}$$

where $\mathbf{D}_\mathbf{p} = [\mathbf{1}_2 | \mathbf{0}_{2\times 1}]$ is a dilated identity matrix that maps the 3D point

coordinates back to 2D pixels.

$\mathbf{y}_{k,j}^{n}$: Pixel coordinates of the $n$-th corner point of tag $j$, $P_{\tau_j,n}$, observed at time $k$, projected onto the frontal image plane of the pinhole camera model $(u,v)$

$\delta\boldsymbol{n}_{k,j}^{n}$: Additive measurement noise at pixel-level

$c_u$, $c_v$: Horizontal and vertical optical offsets from the top left corner of the image [pixels]

$f_u$, $f_v$: Horizontal and vertical camera focal lengths [pixels]

### 4.4. Pose filtering: linearization and EKF formulation

The EKF algorithm follows a prediction and a correction step in a recursive manner. The prediction step projects the current state estimate and error covariance (uncertainties) forward in time, while the correction step incorporates the measurements in the estimates and the associated uncertainties. To obtain the canonical EKF formulation, we first need to linearize the non-linear motion and measurement models about their mean as the operating point. Using the left perturbation scheme defined in Eq. (7) and the first-order Taylor expansion, we can linearize the measurement model (Eq. (24)). The measurement model can be viewed as a combination of two non-linearities. Using the chain rule, we can write:

For the first non-linearity $z_k^{\tau_j,n}$ in Eq. (19), we have:

$$z_k^{\tau_j,n}(\boldsymbol{x}) = \mathbf{D}^T\mathbf{T}_{cv}\mathbf{T}_{v_ki}\mathbf{p}_{\tau_j,n} \tag{25a}$$

$$z_k^{\tau_j,n}(\boldsymbol{x}) = \mathbf{D}^T\mathbf{T}_{cv}exp\left(\delta\boldsymbol{\xi}_k^{\wedge}\right)\overline{\mathbf{T}}_{v_ki}\mathbf{p}_{\tau_j,n} \tag{25b}$$

$$\Rightarrow^{First-order} z_k^{\tau_j,n}(\boldsymbol{x}) \approx \mathbf{D}^T\mathbf{T}_{cv}\left(1+\delta\boldsymbol{\xi}_k^{\wedge}\right)\overline{\mathbf{T}}_{v_ki}\mathbf{p}_{\tau_j,n} \tag{25c}$$

$$z_k^{\tau_j,n}(\boldsymbol{x}) = \mathbf{D}^T\mathbf{T}_{cv}\overline{\mathbf{T}}_{v_ki}\mathbf{p}_{\tau_j,n} + \mathbf{D}^T\mathbf{T}_{cv}\delta\boldsymbol{\xi}_k^{\wedge}\overline{\mathbf{T}}_{v_ki}\mathbf{p}_{\tau_j,n} \tag{25d}$$

If we define the dot operator, $(.)^{\odot}$, as $\boldsymbol{\xi}^{\wedge}\mathbf{p} \equiv \mathbf{p}^{\odot}\boldsymbol{\xi}$ [65] where $\mathbf{p}$ is expressed in the homogeneous coordinates, we have:

$$z_k^{\tau_j,n}(\boldsymbol{x}) = \underbrace{\mathbf{D}^T\mathbf{T}_{cv}\overline{\mathbf{T}}_{v_ki}\mathbf{p}_{\tau_j,n}}_{z_k^{\tau_j,n}(\bar{\boldsymbol{x}})} + \underbrace{\mathbf{D}^T\mathbf{T}_{cv}\left(\overline{\mathbf{T}}_{v_ki}\mathbf{p}_{\tau_j,n}\right)^{\odot}}_{Z_k^{\tau_j,n}}\delta\boldsymbol{\xi}_k \tag{25e}$$

$$z_k^{\tau_j,n}(\boldsymbol{x}) = z_k^{\tau_j,n}(\bar{\boldsymbol{x}}) + Z_k^{\tau_j,n}\delta\boldsymbol{\xi}_k \tag{25f}$$

The second non-linearity $s(.)$ comes from the sensor model (Eq. (24)):

$$g_{k,j}^{n}(\boldsymbol{x}) = \mathbf{y}_{k,j}^{n} = s\left(z_k^{\tau_j,n}(\boldsymbol{x})\right) + \delta\boldsymbol{n}_{k,j}^{n} \tag{26a}$$

$$\overset{(Eq.(25f))}{\Rightarrow} \mathbf{y}_{k,j}^{n} \approx s\left(z_k^{\tau_j,n}(\bar{\boldsymbol{x}}) + \underbrace{Z_k^{\tau_j,n}\delta\boldsymbol{\xi}_k}_{small}\right) + \delta\boldsymbol{n}_{k,j}^{n} \tag{26b}$$

If we define $S_k^{\tau_j,n} = \left.\frac{\partial s}{\partial \mathbf{z}_k^{\tau_j,n}}\right|_{z_k^{\tau_j,n}(\bar{\boldsymbol{x}})} = \mathbf{D}_{\mathrm{p}}\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix}\begin{bmatrix} \frac{1}{Z} & 0 & -\frac{X}{Z^2} \\ 0 & \frac{1}{Z} & -\frac{Y}{Z^2} \\ 0 & 0 & 0 \end{bmatrix}$,

we have:

$$\overset{First-order}{\Rightarrow} \mathbf{y}_{k,j}^{n} \approx \underbrace{s\left(z_k^{\tau_j,n}(\bar{\boldsymbol{x}})\right)}_{g_{k,j}^{n}(\bar{\boldsymbol{x}})} + \underbrace{S_k^{\tau_j,n}Z_k^{\tau_j,n}}_{\mathbf{G}_k^{\tau_j,n}}\delta\boldsymbol{\xi}_k + \delta\boldsymbol{n}_{k,j}^{n} \tag{26c}$$

$$\mathbf{y}_{k,j}^{n} \approx g_{k,j}^{n}(\bar{\boldsymbol{x}}) + \mathbf{G}_k^{\tau_j,n}\delta\boldsymbol{\xi}_k + \delta\boldsymbol{n}_{k,j}^{n} \tag{26d}$$

If we stack the quantities together, we can write:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_{k,1} \\ \vdots \\ \mathbf{y}_{k,M} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k,1}^{1\,T} & \cdots & \mathbf{y}_{k,1}^{4\,T} & \cdots & \mathbf{y}_{k,M}^{1\,T} & \cdots & \mathbf{y}_{k,M}^{4\,T} \end{bmatrix}^T,$$

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{G}_{k,1} \\ \vdots \\ \mathbf{G}_{k,M} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_k^{\tau_1,1^T} & \cdots & \mathbf{G}_k^{\tau_1,4^T} & \cdots & \mathbf{G}_k^{\tau_M,1^T} & \cdots & \mathbf{G}_k^{\tau_M,4^T} \end{bmatrix}^T, \tag{26e}$$

$$\mathbf{R}_k = diag\left(\underbrace{\mathbf{R}_{k,1},\ldots,\mathbf{R}_{k,1}}_{4\ corners\ of\ \tau_1},\ldots,\underbrace{\mathbf{R}_{k,M},\ldots,\mathbf{R}_{k,M}}_{4\ corners\ of\ \tau_M}\right)$$

For the motion model, as indicated in Eq. 13b, we already obtained an equation linear in $\delta\boldsymbol{\xi}$. Using the adjoint transform defined in Eq. 8, we can rewrite Eq. 13b as:

$$\delta\boldsymbol{\xi}_k = \underbrace{\mathbf{F}_{k-1}}_{\mathrm{Ad}(\boldsymbol{\Xi}_k)}\delta\boldsymbol{\xi}_{k-1} + \mathbf{w}_k, \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \tag{27}$$

Bringing together all the above, we obtain the recursive filter update steps. The first step is a prediction step that projects the current state mean and covariance estimates forward in time (Eq. 28a and 28b). The estimated covariance at time $k$, $\check{\mathbf{P}}_k$, is calculated satisfying Eq. 28a, where $\mathbf{F}_{k-1}$ is the Jacobian of motion transformation between time $k-1$ and $k$ ($\mathrm{Ad}(\boldsymbol{\Xi}_k)$), and $\mathbf{Q}_k$ is the process noise covariance (Eq. 18). This is followed by a correction step where tag measurements are incorporated (Eq. 28d and 28e). Note that the corrections are imposed using the difference between the actual and expected measurements, also known as innovation ($\mathbf{y}_k - \check{\mathbf{y}}_k$). Finally, we can write the canonical form [65] of an EKF as:

Predictor:

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_k \tag{28a}$$

$$\check{\mathbf{T}}_k = \boldsymbol{\Xi}_k\hat{\mathbf{T}}_{k-1} \tag{28b}$$

Kalman gain:

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{G}_k^T\left(\mathbf{G}_k\check{\mathbf{P}}_k\mathbf{G}_k^T + \mathbf{R}_k\right)^{-1} \tag{28c}$$

Corrector:

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{G}_k)\check{\mathbf{P}}_k \tag{28d}$$

$$\hat{\mathbf{T}}_k = \exp\left(\left(\mathbf{K}_k\left(\mathbf{y}_k - \check{\mathbf{y}}_k\right)\right)^{\wedge}\right)\check{\mathbf{T}}_k \tag{28e}$$

where $\hat{(\cdot)}$ represents posterior (estimated) quantities, $\check{(\cdot)}$ shows prior quantities, and $\mathbf{K}_k$ is the *Kalman gain*. The *Kalman gain* weighs the innovation's contribution to the estimates (compared to the prediction).

In the end, to evaluate the estimator performance, the translational and rotational estimation errors at time $k$ are computed as:

$$\delta\mathbf{r}_k = \begin{bmatrix} \delta r_{x,k} \\ \delta r_{y,k} \\ \delta r_{z,k} \end{bmatrix} := \hat{\mathbf{r}}_i^{v_ki} - \mathbf{r}_i^{v_ki} \tag{29}$$

$$\delta\boldsymbol{\theta}_k^{\wedge} = \begin{bmatrix} \delta\theta_{x,k} \\ \delta\theta_{y,k} \\ \delta\theta_{z,k} \end{bmatrix}^{\wedge} := \mathbf{1} - \hat{\mathbf{C}}_{v_ki}\,\mathbf{C}_{v_ki}^T \tag{30}$$

where quantities with $\hat{(\cdot)}$ are estimated values, while those without are ground truth.

## 5. Implementations and experimental setups

### 5.1. Implementations

The aerial robotic platform chosen in this work is a low-cost, commercially available UAV, *Parrot Bebop2* [66], with no hardware modifications. This compact, off-the-shelf UAV has an onboard flight controller as well as the following sensory set: (1) an IMU for inertial measurements; (2) a sonar and a vertical camera for height measurements used in the onboard controller; and (3) a forward-looking camera. Nonetheless, this choice was entirely arbitrary and only for validation purposes, as any other platform that outputs IMU/odometry data, as well as a camera image stream, could have been used. We use the well-known Robotic Operation System (ROS) [22] as a structured communication layer among the heterogeneous cluster of the processing

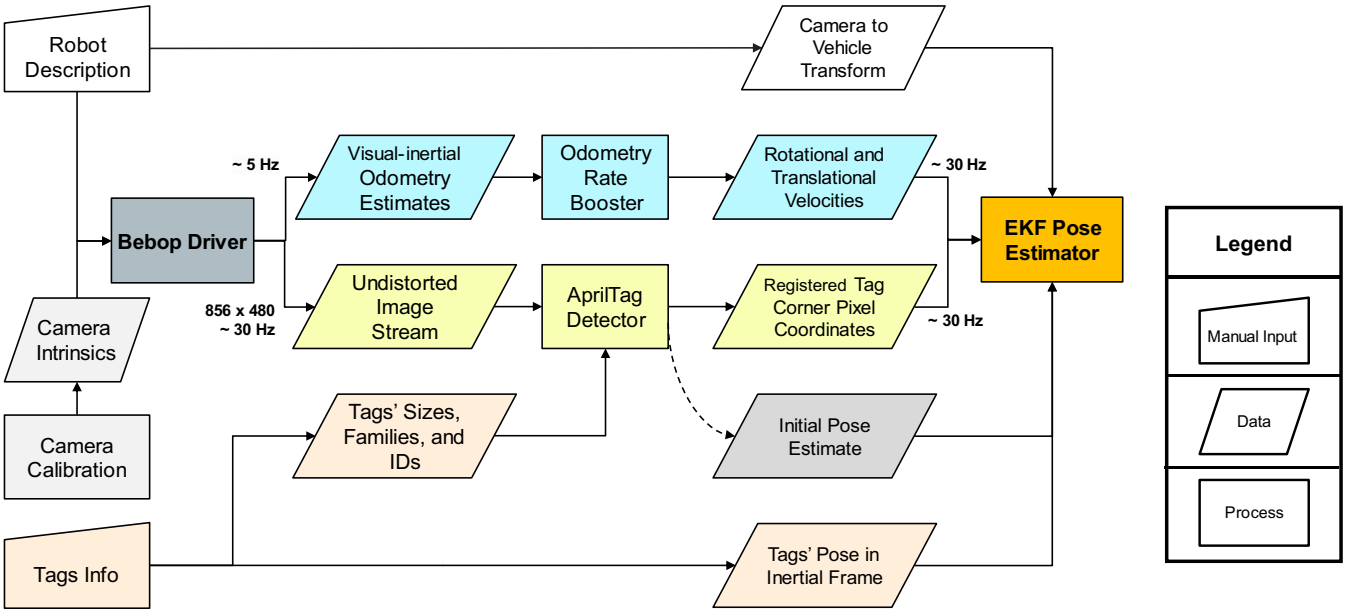



**Fig. 4.** Tag-based visual-inertial localization: implementation on Parrot Bebop2.

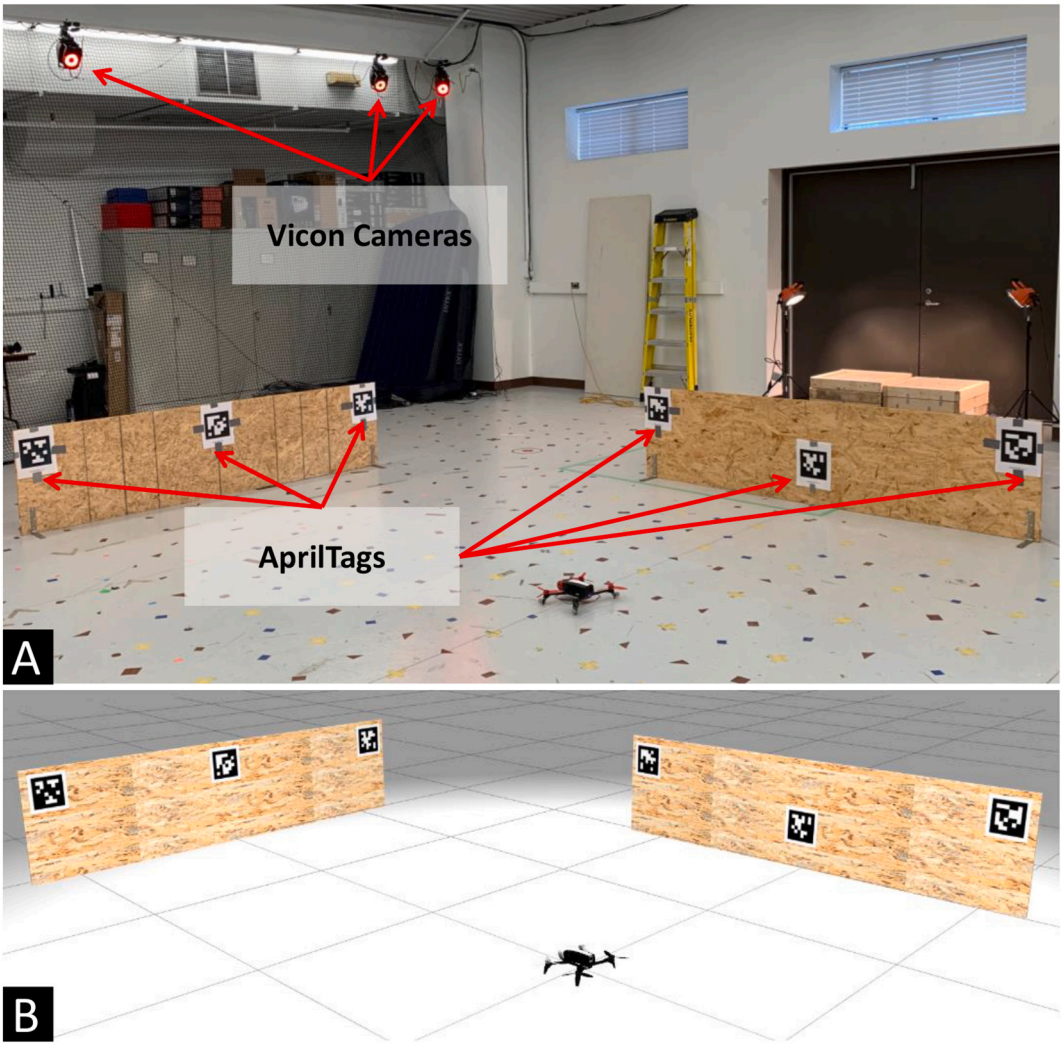



**Fig. 5.** – Test-bench settings: (A) Laboratory; (B) Simulation.

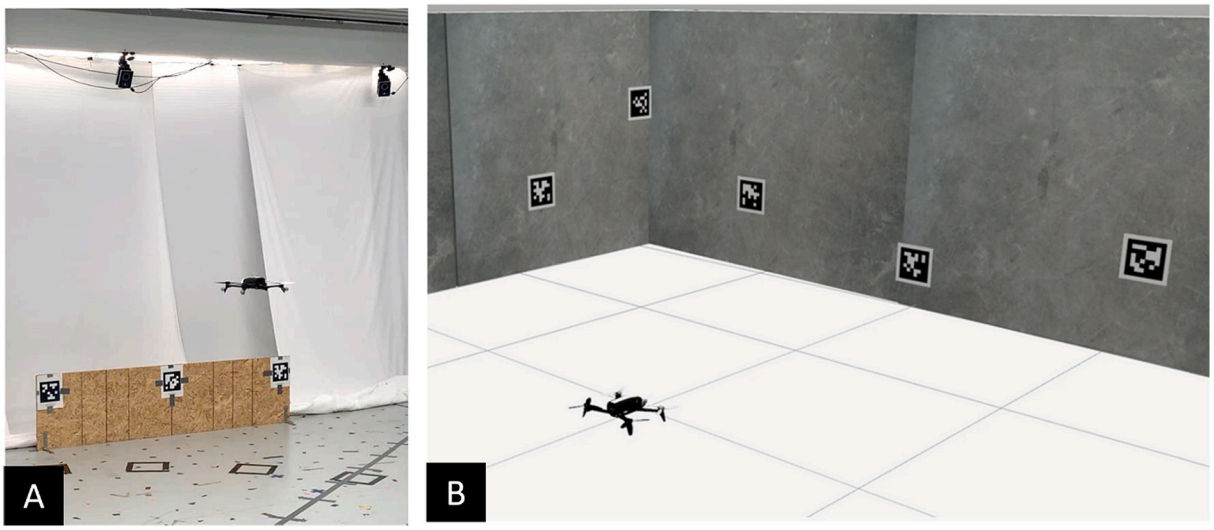



**Fig. 6.** Parrot Bebop2 and AprilTags: (A) during a laboratory experiment (B) in a simulated indoor construction environment.

modules. The *Bebop2* UAV's open-source ROS driver *bebop_autonomy* [67] with necessary modifications was used for interaction with the platform. For AprilTag detections, the publicly available code implemented by *AprilRobotics* [68] was modified and deployed. Their implementation [68] includes a faster detector with improved detection rates on small tags and supports flexible tag layouts [69].

#### 5.1.1. System setup

As shown in the process diagram in Fig. 4, the robot description, camera intrinsics, and tag information (i.e., size, family, IDs, and pose) are assumed to be known a priori. The robot description contains the robot's transform tree (e.g., camera to vehicle transform). The camera intrinsics are obtained from calibration, while tags' family, size, and pose information are manual inputs.

The EKF pose estimator module fuses two sources of information from the UAV: (a) system inputs, i.e., rotational and translational velocities, and (b) the image stream from the onboard forward-looking camera. System inputs come from the onboard odometry estimations of the UAV (~5 Hz) and are being boosted up to the image rate (~30 Hz). The camera publishes rectified 856 × 480 images at the same rate of 30 Hz, fed into the modified AprilTag detector module. It then provides the system with the registered 2D pixel coordinates of tag corner points in the image, i.e., the measurements. The proposed tag-based localization module also gets an initial estimate of the vehicle pose as input. The initial pose can be given manually or estimated based on raw pose estimates provided by the AprilTag detector. We validate our method using two test-bench environments: laboratory (Fig. 5 A) and simulation (Fig. 5 B).

#### 5.1.2. Simulation environment

Simulation helps solve real-world problems safely and efficiently. A reliable simulation environment provides a valuable method of analysis that is easily verified, communicated, and understood. Once configured, it can serve as a common-use module to validate the core research ideas and avoid struggling with hardware complications, which may prolong. A well-designed simulation environment facilitates the testing and validation processes in numerous scenarios where ground truth data collection is challenging or safety is critical in the actual workspace, e.g., indoor construction sites. Accordingly, a photo-realistic BIM-enabled simulation environment (see Fig. 6) was developed and configured with the principal capabilities of (1) integration with the ROS ecosystem; (2) automatic generation of simulated *worlds* (e.g., indoor construction environments); (3) automatic generation and spawn of AprilTags in the simulated *worlds*; (4) simulating onboard sensors (e.g., IMU, ultrasound, and vertical camera); (5) video streaming using a virtual front camera; (6) connecting to any controller for sending commands to the UAV (e.g., keyboard); (7) simulating battery. Fig. 6 (B) shows a snapshot of an instance of a simulated indoor construction environment. The simulation environment is mainly based on a publicly available simulation tool called *Parrot-Sphinx* [70]. The manufacturer initially designed this tool to facilitate their software developments. The official simulator of ROS, *Gazebo*, is tightly coupled with *Parrot-Sphinx* to simulate the physical and visual surroundings of the platform. Once connected to the simulation environment via Wi-Fi or virtual ethernet, one can use the codes and algorithms implemented in *C++* or *Python* programming languages in ROS. Both *Gazebo* and command-line scripting capabilities are incorporated. One or more quadrotor products by *Parrot* can be spawned in the *world* environment.

**Table 2**
A brief description of custom-designed experiments in laboratory and simulation environments.

| Summary of experiments | | | | |
|---|---|---|---|---|
| ID | Name | Sim/ Lab | Tag-blind zone | Description |
| 1 | Straight line | Sim | No | A back-and-forth straight-line trajectory (3 $m$ × 4) |
| 2 | Mixed maneuvers | Sim | No | A trajectory of arbitrary maneuvers in an under-construction residential unit while tags are always in view. |
| 3 | Planar | Sim | Yes | A planar trajectory with a combination of rotational and translational maneuvers while tags may be out of sight. |
| 4 | Straight line | Lab | No | The same test as "1" in a laboratory setting. |
| 5 | 3D circular | Lab | No | A 3D circular trajectory of radius one meter. |

### 5.2. Experimental setups

The proposed tag-based localization method is validated via custom-designed experiments in a laboratory and a simulation environment (Fig. 5 and Fig. 6), a summary of which is provided in Table 2. The remainder of this section elaborates on the main components in each test-bench environment.

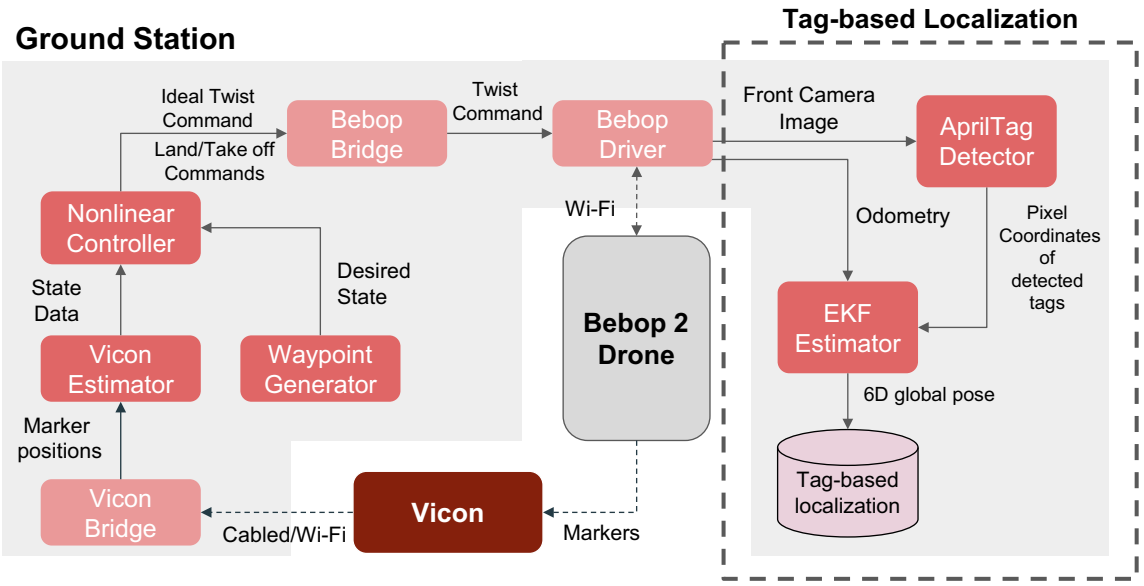



**Fig. 7.** Laboratory (Vicon-based) test-bench architecture including a ground station, Parrot Bebop2 platform, and the Vicon system for providing ground truth data.

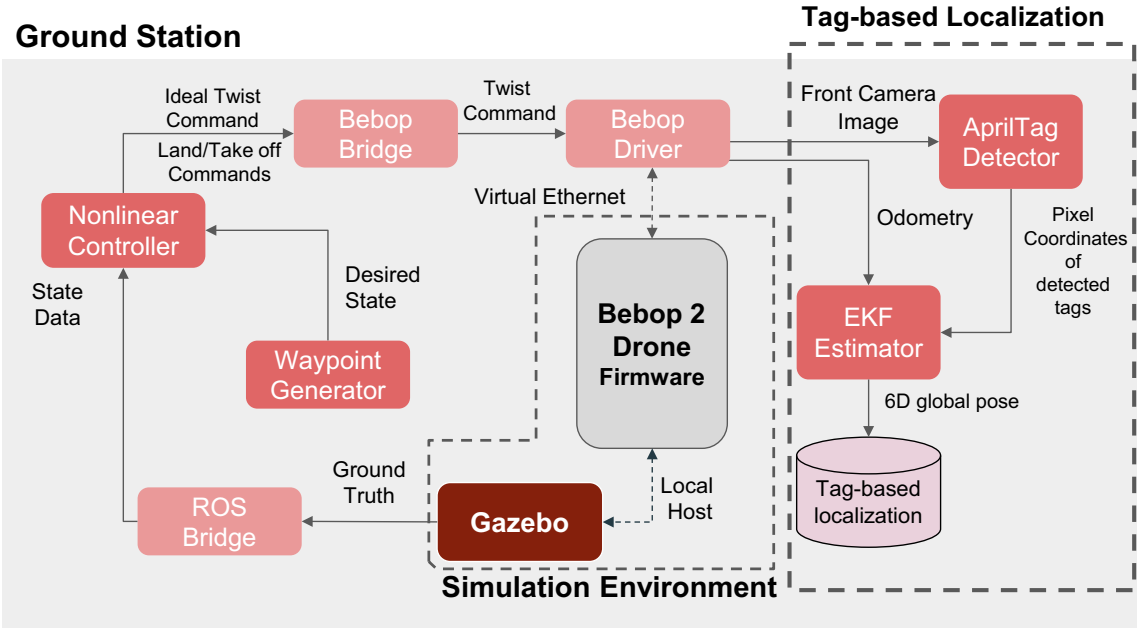



**Fig. 8.** - Simulation test-bench architecture where Parrot Bebop2 platform and the Vicon system are abstracted as two communicating components in the simulation environment.

*5.2.1. Laboratory setup*

Fig. 7 depicts the general architecture of the test-bench environment used to assess the tag-based localization accuracy in a laboratory setting. The lab test-bench setting consists of three major components, *Bebop2* hardware, a constellation of motion capture cameras (Vicon cameras), and a ground station (the gray area) (e.g., a laptop).

A set of retroreflective markers is attached to the UAV to enable the motion capture system to track the platform (Fig. 5 (a)). The Vicon system estimates the pose of the UAV with sub-millimeter accuracy and high frequency ($>$ 200 *Hz*), given that it is already calibrated. Hence, Vicon data is used as the ground truth to be compared against our estimates. The communication between the Vicon system and the ground station is either cabled or through Wi-Fi. The state data from *Vicon Estimator* and the desired state from *Waypoint Generator* are sent to *Nonlinear Controller*. Based on these two states, the corresponding commands from *Nonlinear Controller* are sent to the platform through two intermediary processes, and the loop is closed for autonomous flight. The communication between the UAV and the ground station is handled via Wi-Fi connections. In parallel, *Bebop Driver* sends data to the tag-based localization module. The data received by this module includes the image stream from the front camera and the odometry velocity data (body-fixed rotational and translational velocities). *AprilTag Detector* processes the front camera images, and the tag-based pose estimates are sent to *EKF Estimator*. These two types of data are fused in *EKF Estimator* and stored as tag-based localization outputs.

*5.2.2. Simulation setup*

In the simulation test-bench architecture, the actual platform and the

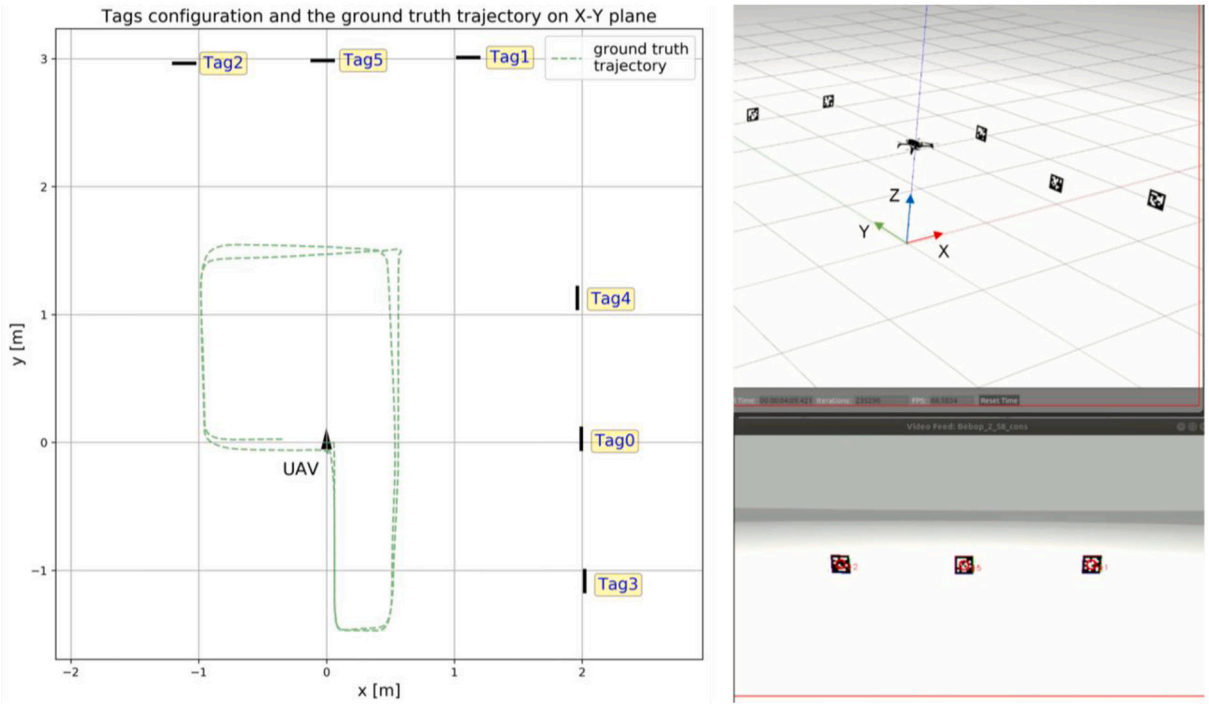



**Fig. 9.** - Tags' configuration and UAV's actual trajectory on $X-Y$ plane (dashed line) (left). A third-person view of the UAV taken off (top-right), the front camera image stream, and the detected tags with the corresponding tag IDs (bottom-right) in the simulation environment. (video available at [71]).

**Table 3**
– Tag properties in the experiments.

| Tag configuration summary | | |
|---|---|---|
| Tag property | Laboratory | Simulation |
| Type | AprilTag | AprilTag |
| Family | 36h11 | 36h11 |
| Size | 0.165 m × 0.165 m | 0.165 m × 0.165 m |
| Number of tags | 6 | 6 |
| Global pose (in BIM reference frame) | Known a priori | Known a priori |

Vicon system were replaced and incorporated in *Ground Station* (the area with gray background in Fig. 8). These replacements were abstracted as two simulated components: the UAV's firmware and *Gazebo*. *Gazebo* runs on localhost and handles the graphics and the simulation of dynamic interactions of the robot (i.e., UAV) and the environment based on its physical properties (e.g., mass and moments of inertia).

The absolute ground truth data published in the *Gazebo*'s publisher/subscriber communication network was linked to the ROS network using *ROS Bridge,* replacing *Vicon* (Fig. 8). In this case, the state data required for the control module is published through *ROS Bridge*. A virtual ethernet was defined to communicate data with firmware and *Bebop Driver*. The rest of the architecture remains untouched.

## 6. Results

The proposed tag-based localization method was validated in simulation and laboratory settings via custom-designed experiments, as described in Table 2. The same tag configuration (Fig. 9) and properties (summarized in Table 3) are deployed for consistency in experiments. The UAV autonomously tracks a pre-planned path by following intermediate waypoints. The state estimates fed in the control loop come from the ground truth sources (*Vicon* and *ROS bridge* in the laboratory and simulation environments, respectively).

This section first provides the full details for two experiments: (1) the planar trajectory test, conducted in the simulation environment; (2) the 3D circular trajectory test, carried out in the laboratory setting. The former investigates the estimator's performance in a planar motion when tags are not always visible, while the latter studies a circular 3D translational maneuver where tags remain in view throughout the whole flight. Each experiment is discussed in a separate sub-section, including thorough qualitative and quantitative performance analyses. Next, a quantitative analysis is provided for all the experiments described in Table 2.

### 6.1. Simulation: the planar trajectory

Fig. 9 provides a snapshot of the simulation environment while the UAV has taken off. On the left, the tag locations and the UAV's actual trajectory on the X − Y plane (dashed line) are shown. The green dashed line representing the entire flight trajectory is based on ground truth data and just for visualization purposes. Moreover, a third-person view of the UAV while taken off, the BIM reference frame (top-right), and the front camera image stream, including the detected tags with the corresponding tag IDs (bottom-right), are visible in Fig. 9.

In the planar trajectory test, a combination of rotational and translational motions is involved. As depicted in Fig. 9, each set of tags, i.e., *2–5–1* and *4–0–3*, contains three *36h11* AprilTags (0.165 $m$ × 0.165 $m$). The UAV starts at $O(0,0,0)$, takes off and fixes its altitude facing tag set *2–5–1*. Then, it immediately rotates 90 degrees clockwise to face *4–0–3* tags. This case purposefully starts with a rotational motion so that the drift in dead reckoning becomes easier to observe (see Fig. 10). The minimum and maximum camera-to-tag distances during the flight are 1.5 $m$ and 4.5 $m$, respectively. The UAV follows a straight path parallel to *4–0–3* tags until the tags are out of sight (a tag-blind zone). While no tags are visible at the top right corner of the trajectory (no corrections in EKF), it rotates until facing *2–5–1* tags. These tags remain in view until the UAV returns to its starting point and follows the same path one more time (for more visualization and more details, please refer to the provided video available at [71]).

Fig. 10 visualizes the localization performance of the proposed

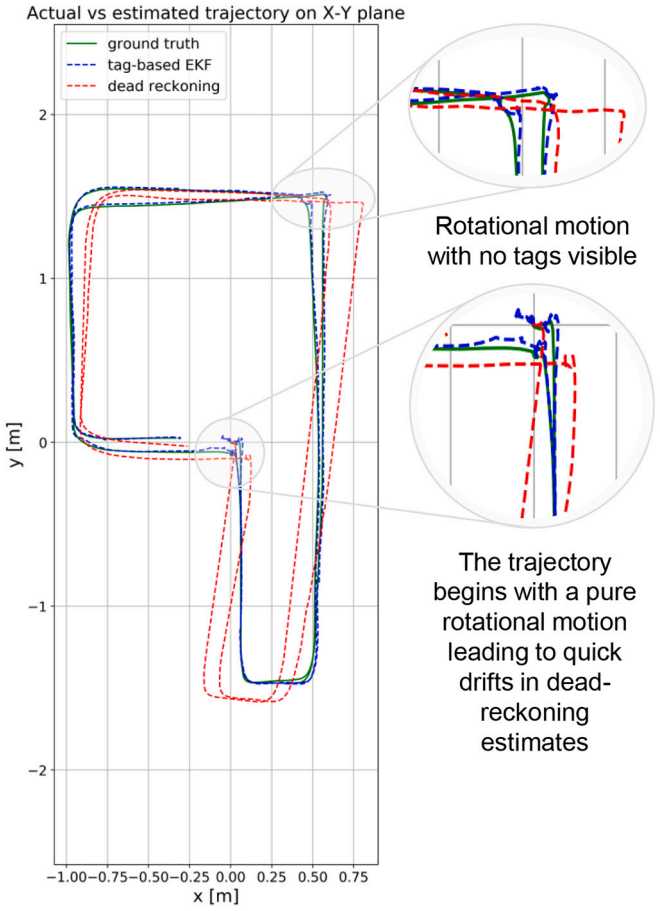



**Fig. 10.** Performance of the proposed tag-based localization method on the X − Y plane (planar trajectory). The maximum speed is 0.4 m/s, and the average speed is 0.1 m/s. Our estimates accurately track the ground truth, while dead-reckoning estimates drift quickly with the first rotational motion. The proposed tag-based localization method can handle local maneuvers in tag-blind zones and recover soon after detecting a tag.

method on the X − Y plane, which is also the plane of motion. The green, blue, and red lines correspond to ground truth, tag-based EKF, and dead-reckoning estimates, respectively. Dead reckoning is only based on odometry and velocity integrations and involves no corrections from tag measurements. It shows how the pose estimates would be if the tag measurements were discarded. As depicted in Fig. 10, dead reckoning (odometry-based estimates) drift quickly due to the UAV's initial rotational motion after take-off.

On the other hand, Fig. 11 shows that the proposed tag-based EKF accurately follows the ground truth in position and orientation estimates after convergence. It also illustrates how well our method could estimate the 6-DoF pose of the UAV in a planar motion where tags are not always visible. For this experiment, the root mean square error (RMSE) for the position estimates in 3D was as low as 0.0198 $m$. If the take-off and landing disruptions are excluded, RMSE is reduced to 0.0177 $m$. Since the UAV operates in a low-speed mode (average linear speed of 0.1 $m/s$), roll ($\theta_x$), pitch ($\theta_y$) estimates (in global (BIM) frame) are approximately zero and remain constant (Fig. 11).

Fig. 12 shows the estimation errors in position (Eq. 29) and orientation (Eq. 30), along with the corresponding $3\sigma$ uncertainty envelops. A larger envelope corresponds to more uncertain estimates. It also reports the minimum, maximum, and mean errors for each component. The mean values being close to zero experimentally confirm that our estimates are unbiased. The importance of tracking $3\sigma$ uncertainty envelops is threefold: (1) as the uncertainty on the position and orientation estimates remain bounded and the estimation errors remain within the estimated uncertainty envelops, our estimates are confirmed to be consistent. Otherwise, the uncertainties would have grown unboundedly, or the estimates would leave the envelope; (2) although our method remains consistent even when no tags are visible, it is observable that the uncertainty grows when the filter receives no tag measurements, happening twice in this experiment at $t = 88.8$ [$s$] and $t = 217.6$ [$s$] (highlighted intervals in Fig. 12); (3) tracking the uncertainties over time can be helpful in effectively planning tag locations and safe flight paths in a complex environment such as a construction job site.

### 6.2. Laboratory: the 3D circular trajectory

Simulation environments are, to some extent, ideal settings. To evaluate the performance in handling real-world data, being typically noisier, the proposed method is validated in a laboratory setting supported by ground truth data. The previous trajectory was almost planar, so the current case shows how tag-based EKF can handle 3D translational motions in real-world scenarios. The planar trajectory test also showed that having two sources of information can enable the estimator to handle situations where no tags are visible (at least for a short period). In this experiment, a 3D circular trajectory of radius one meter is designed such that the minimum and maximum camera-to-tag distances are 2.2 $m$ and 4.2 $m$, respectively (see Fig. 13). The tag properties remain the same as the previous experiment, while the camera faces the *2–5–1* tag set (along the $y-axis$) in a similar tag configuration shown in Fig. 9. The tags remain within the camera's field of view throughout the flight (see the video available at [72] for more visualizations).

Fig. 13 shows the localization results on the Y−Z and the X − Y planes, as well as in 3D. Again, odometry-based estimates (dead reckoning) quickly drift, whereas the tag-based estimates accurately follow the ground truth. A jump in estimates is observable during the UAV's take-off (center) and landing (right side) due to abrupt motions. Moreover, as expected [58], deviations from ground truth are observed when the camera-to-tag distance increases. The root mean square error (RMSE) in position for this experiment was 0.0348 $m$ and 0.2602 $m$, including and excluding take-off and landing disruptions respectively. Fig. 14 shows the position and orientation estimates against ground truth data and how well the proposed method could estimate the 6-DoF pose of the UAV in 3D. Slight biases can be seen in the $z$-component of the global position and orientation (i.e., yaw) due to imperfections in reference frame calibrations in practice.

As illustrated in Fig. 15, uncertainty generally grows as the UAV gets farther from and shrinks when it gets closer to the tags. This change is related to the reliability of the measurements. Larger tags provide more reliable measurements, which decreases the uncertainty in our estimates and vice versa. As highlighted, the $3\sigma$ bounds grow larger when the distance to the tags ($\sim y$) is maximum, except for $y$ estimates itself. It is also observable that the uncertainty in $y$, which is the normal direction to the tags' plane, is less sensitive to this relative distance. Again, significant errors are seen during landing and take-off due to swift motions.

Previously, we saw how the proposed method compares to odometry-based estimates. To further investigate the effectiveness of tag-based EKF, the vehicle's pose is estimated only based on the raw AprilTag package's [68] outputs and compared against the ground truth. As discussed earlier, although the solution is not stable and may suffer from orientation estimation ambiguity, the AprilTag package directly provides the relative transformation between tag $\tau$ and camera $c$ ($\mathbf{T}_{c\tau}$) using the corner point correspondences. Since the tag pose in the inertial (BIM) reference frame ($\mathbf{T}_{i\tau}$) and the camera to vehicle transformation ($\mathbf{T}_{vc}$) are assumed to be known in our problem, the vehicle's pose in the inertial frame ($\mathbf{T}_{vi}$) can be easily found:

$$\mathbf{T}_{vi} = \mathbf{T}_{vc}\mathbf{T}_{c\tau}(\mathbf{T}_{i\tau})^{-1} \tag{31}$$

Using the relative tag pose ($\mathbf{T}_{c\tau}$) provided by the AprilTag package,

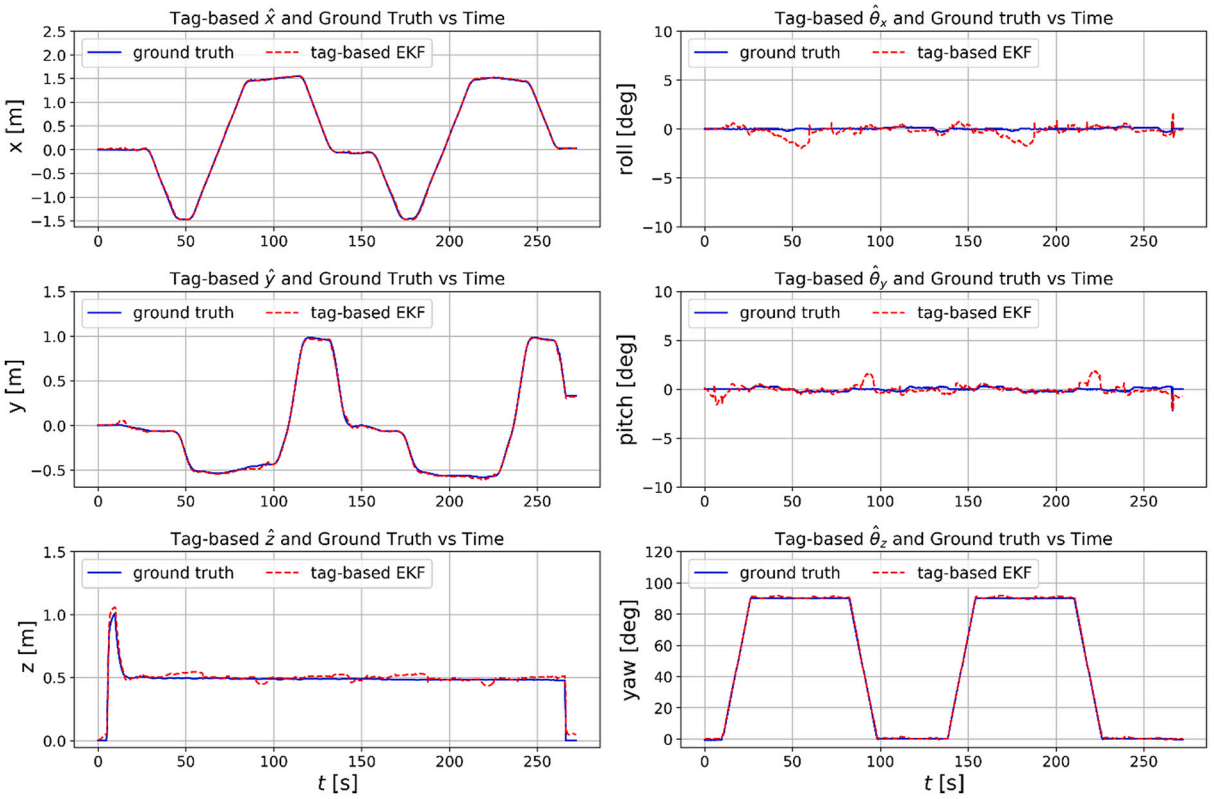



**Fig. 11.** - Performance of the proposed tag-based EKF method in position and orientation estimation (planar trajectory).

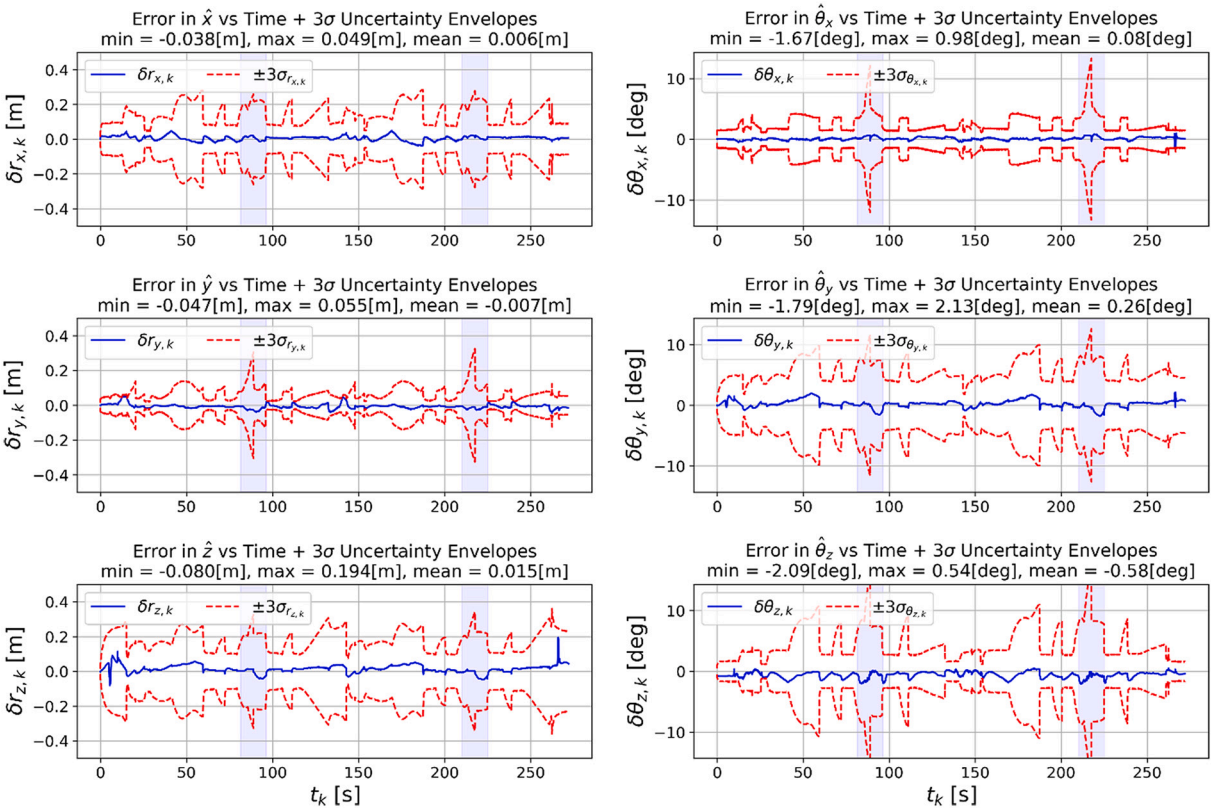



**Fig. 12.** – Position and orientation estimation errors with 3σ bounds (planar trajectory); time intervals with no tags visible (tag-blind zones) are highlighted.

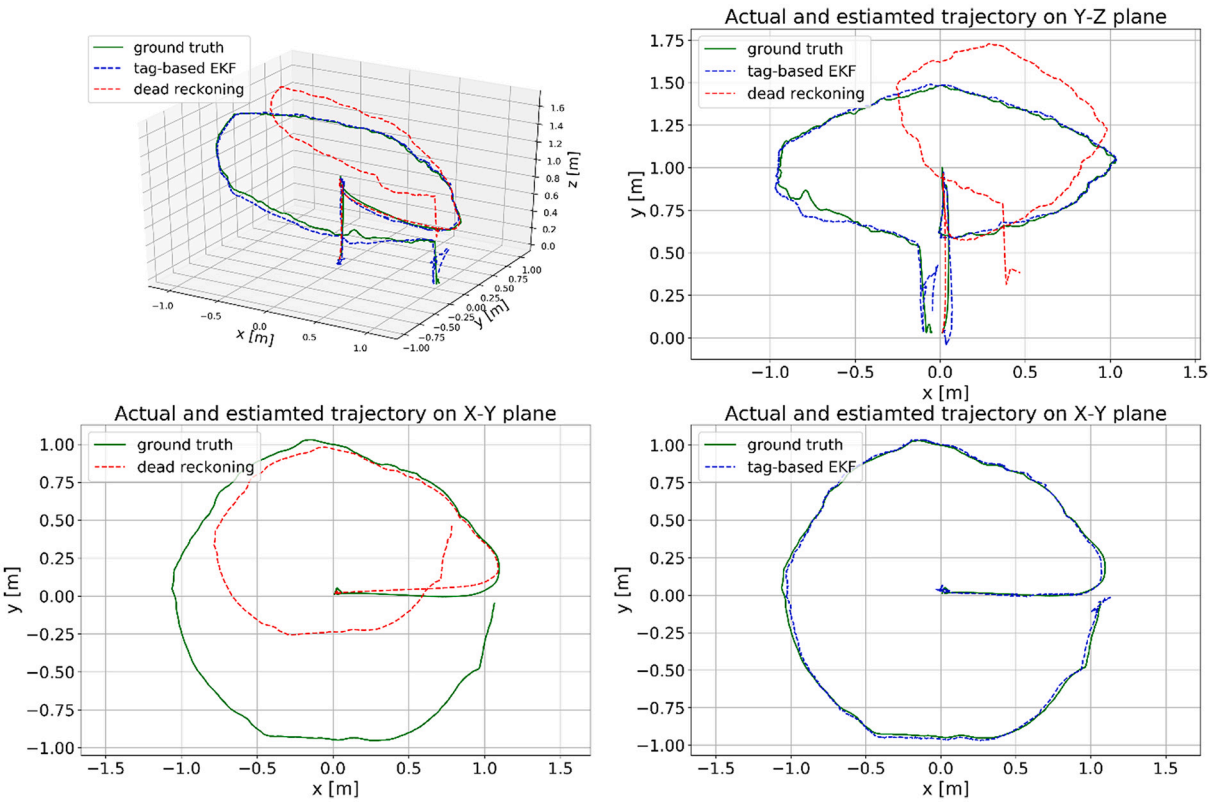



**Fig. 13.** - Performance of the proposed tag-based localization method in 3D (3D circular trajectory). The maximum speed is 0.5 m/s during take-off, and the average speed is 0.08 m/s. Our estimates accurately track the ground truth, while dead reckoning drifts almost immediately. Some disruptions can be observed in position estimates during landing and take-off due to agile motions.

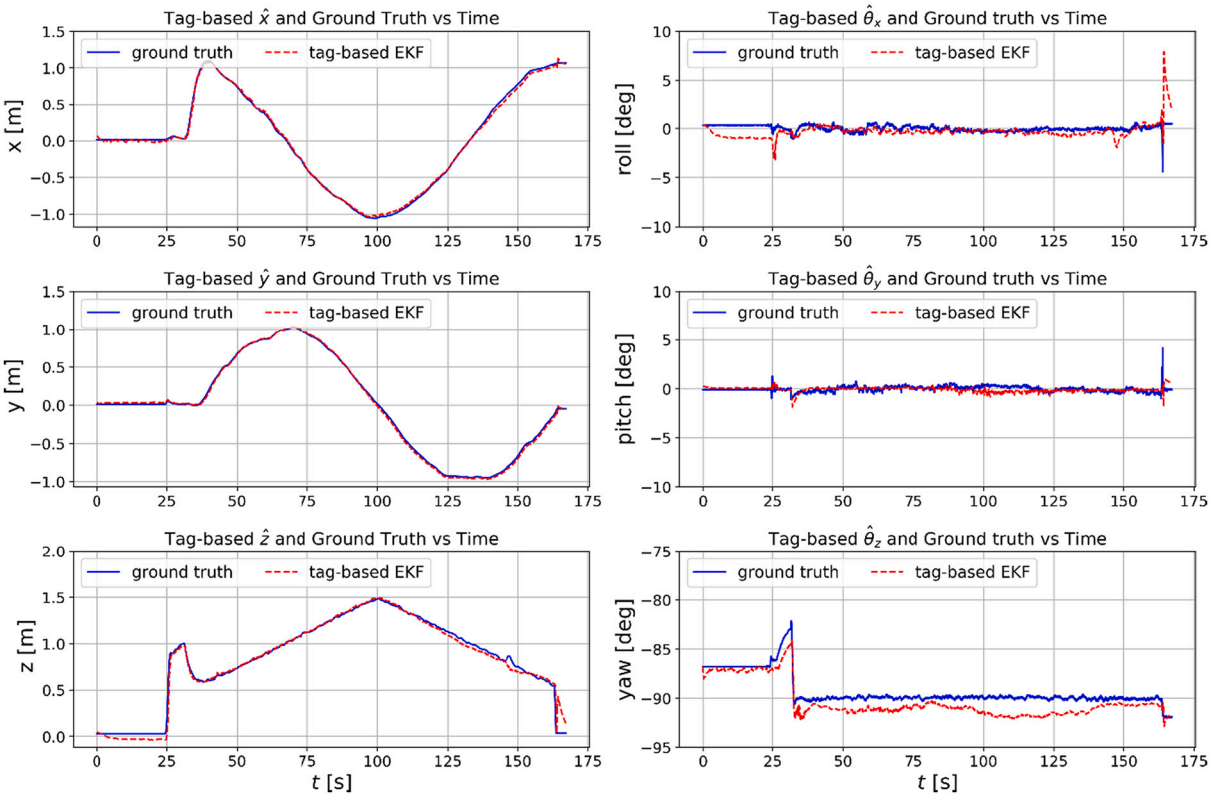



**Fig. 14.** - Performance of the proposed method in position and orientation (3D circular trajectory).

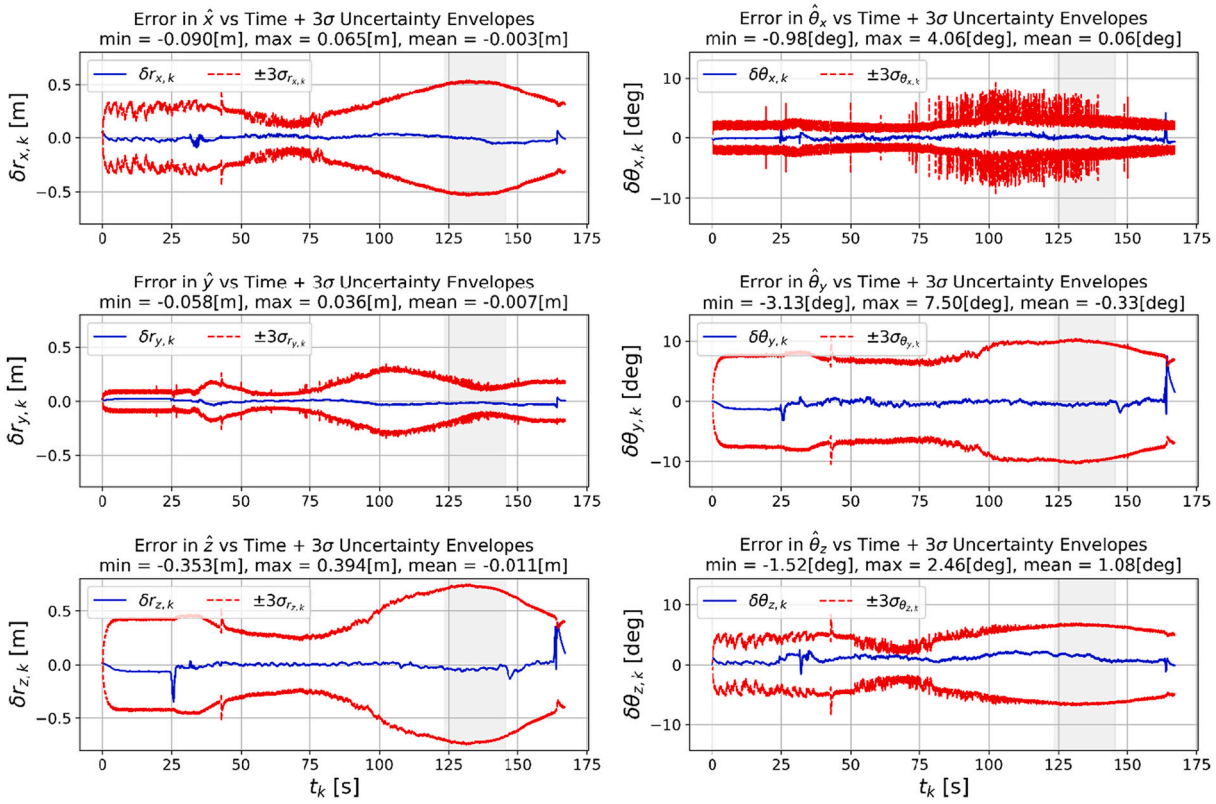



**Fig. 15.** – Position and orientation estimation errors with 3σ bounds (3D circular trajectory): the time interval with the maximum relative camera-tag distance is highlighted.

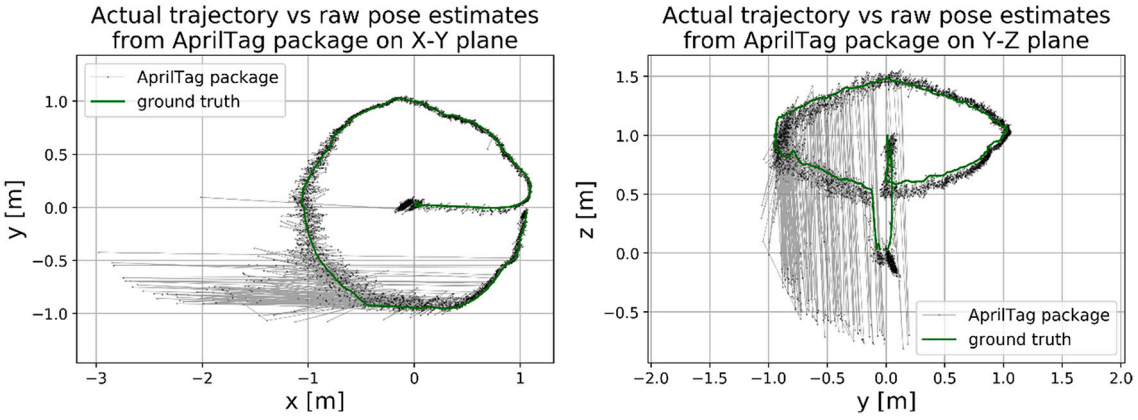



**Fig. 16.** - Raw AprilTag measurements are noisy and suffer from ambiguity in their orientation estimates, resulting in jumps in the vehicle's position estimates (3D circular trajectory).

the vehicle pose is calculated applying Eq. 31. Fig. 16 compares the position measurements from a single tag, i.e., *Tag2* (see Fig. 9), with the ground truth trajectory. The raw estimates based on tags alone are noisy. The estimates become unreliable when the relative camera to tag distance passes a threshold. This threshold depends on camera resolution, focal length, tag size, and relative orientation. For instance, in this experiment where the camera remains almost normal to the tags' plane, given the tag size (16.5 *cm*), image resolution (856 × 480 $px^2$), camera's focal length (~520 *px*), the distance threshold is less than 4 m. Noisy estimates alone make safe autonomous flight challenging, while orientation ambiguity in greater distances results in instability and noticeable jumps in position estimates.

Fig. 17 compares the absolute error in 3D position estimates for the dead reckoning (using IMU only), raw pose estimates based on the AprilTag package (using tag measurements only), and the proposed tag-based EKF (fusion of IMU and tag measurements). Error accumulates in dead reckoning resulting in drifts in the odometry-based estimates; raw pose estimates based on tags are noisy and unstable, whereas tag-based EKF is more stable and has the minimum error (except for take-off and landing).

### 6.3. *Quantitative evaluation*

The performance of different approaches of dead reckoning, tag-only, and tag-based EKF were evaluated quantitatively in different scenarios. The quantitative evaluation results for our custom-designed

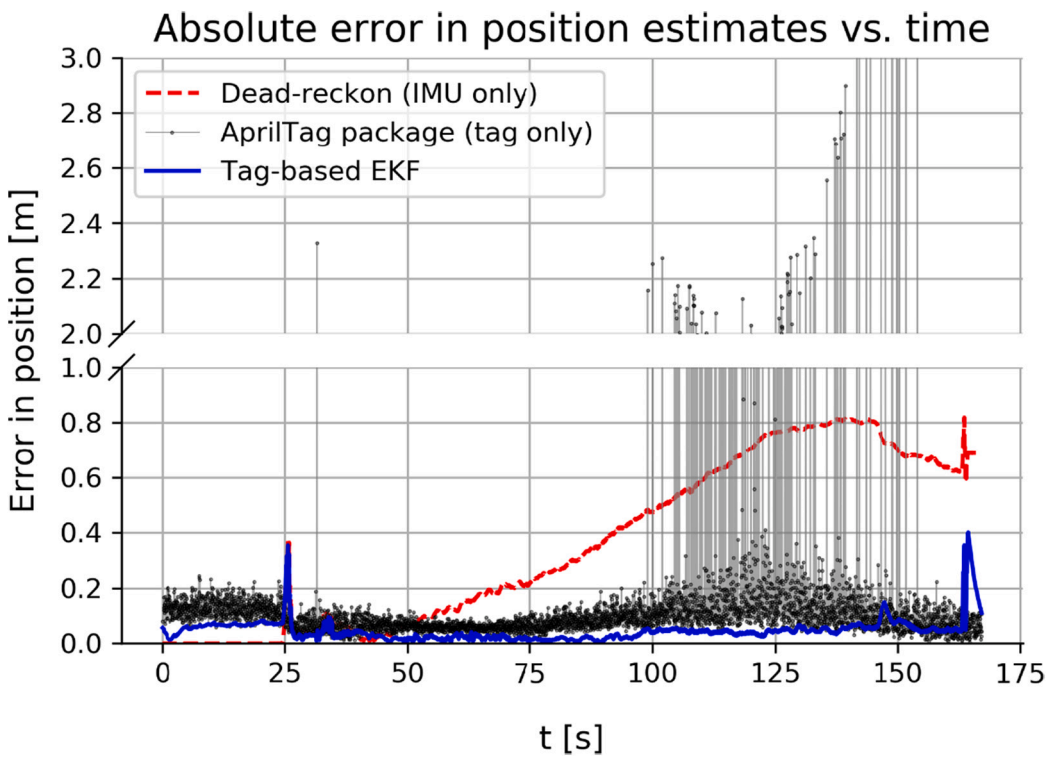
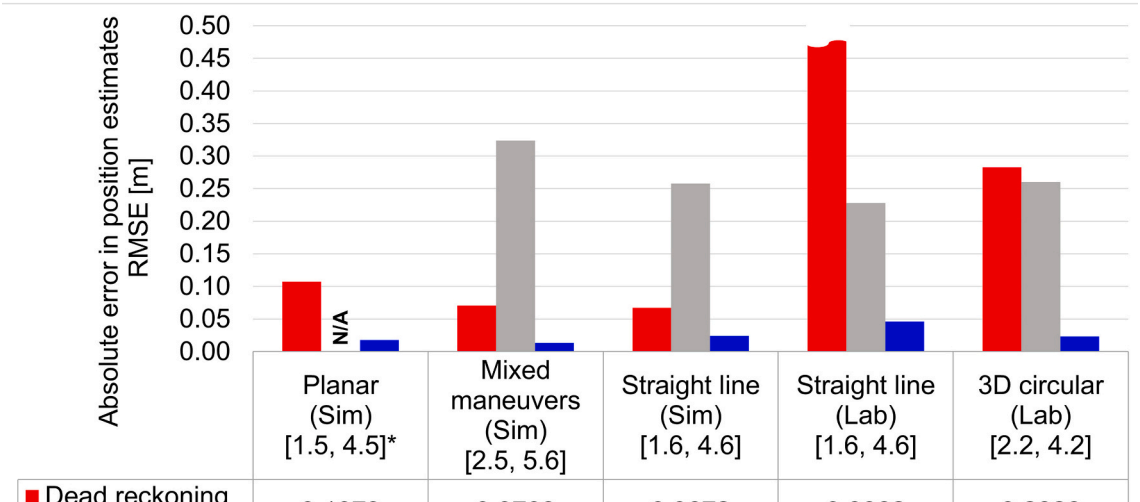



**Fig. 17.** 3D trajectory: the comparison of the absolute error in position estimates for the dead reckoning (only IMU), estimates based on the AprilTag package (only tag measurements), and the proposed tag-based EKF (fusion of IMU and tag measurements). Dead reckoning accumulates error and drifts, raw pose estimates based on tags are noisy and unstable, while our tag-based EKF is smooth, reliable, and more accurate.




| | Planar (Sim) [1.5, 4.5]* | Mixed maneuvers (Sim) [2.5, 5.6] | Straight line (Sim) [1.6, 4.6] | Straight line (Lab) [1.6, 4.6] | 3D circular (Lab) [2.2, 4.2] |
|---|---|---|---|---|---|
| Dead reckoning (IMU only) | 0.1073 | 0.0708 | 0.0672 | 0.9382 | 0.2826 |
| AprilTag package (tag only) | N/A** | 0.3235 | 0.25781 | 0.22781 | 0.2602 |
| Tag-based EKF | 0.0177 | 0.0135 | 0.0241 | 0.04635 | 0.02315 |

** [Min camera-to-tag distance in meters, Max camera-to-tag distance in meters]*
*** Discontinuous position estimates based on only tags (tags are not always visible).*

**Fig. 18.** - Performance evaluation of dead reckoning (IMU only), AprilTag package (tag only), and the proposed tag-based EKF on five custom-designed experiments in simulation and laboratory settings.

experiments introduced in Table 2 are reported in Fig. 18. In this evaluation, the disruption caused by sudden motions during the platform take-off and landing were left out. It can be observed that dead reckoning in simulation is much more reliable than the laboratory results, as expected. The difference in dead reckoning results is due to the noisiness of the odometry data in the real world. Our observations show

that drifts in the altitude estimates in the laboratory are the leading cause for this significant difference. The laboratory and simulation results for *Straight line* experiment, shown in Fig. 18, are excellent examples of how accumulative errors can lead to substantial errors in dead reckoning estimates, although the simulation could not capture that.

On the other hand, the estimates solely based on raw tag measurements are relatively the same among the tests we conducted, regardless of the experiment environment. This observation proves that the simulated vision pipeline is closely mimicking its actual counterpart. However, as discussed earlier, tags alone are not reliable enough for autonomous navigation. As shown in Fig. 18, in our experiments, the corresponding RMSE for tag-only was in the order of tens of centimeters. This quantity is not reported for the *Planar trajectory* test due to discontinuity of position estimates when tags were out of sight. The correlation of the camera-to-tag distance and the quality of tag readings is also observable in both laboratory and simulation results. Therefore, tag measurements suffer from noisiness and instability while requiring tags to be always in line of sight and within a certain distance.

The fusion of these two sources of information in the proposed tag-based EKF, however, allows for smooth, continuous, and accurate estimates. In our experiments, the achieved RMSE for tag-based EKF was as low as a few centimeters, even when tags were not always visible, or RMSE in dead reckoning got close to one meter (Fig. 18).

## 7. Discussion

The provided quantitative and qualitative evaluations showed that the proposed tag-based EKF could enable an off-the-shelf compact UAV with a minimum suite of sensors (an RGB camera and an IMU) to be accurately and robustly localized in a GPS-denied indoor environment. The proposed method is lightweight, inexpensive, and designed to address the technical and practical localization challenges in indoor construction environments. Construction sites alter rapidly and include many low-texture/repetitive areas that our method can handle. Lightweight localization allows for efficient usage of UAVs' limited computational and power resources, resulting in longer flight times and operations. The hardware cost is another barrier to the scalability of robotic data capture solutions in construction. The presented localization system is inexpensive and can enable a variety of low-cost robotic platforms to autonomously perform in indoor construction environments, although handheld devices such as smartphones can easily benefit from the same system.

Our experiments showed that dead reckoning drifts over time, and tags alone are insufficient for robust and accurate localization. Tags might not always be visible in an ever-changing construction setting, and the direct pose estimates from tags alone are noisy and suffer from ambiguity. On the other hand, our proposed method properly fuses these two sources of information to yield consistent global pose estimates in real-time.

### 7.1. Practical considerations

From the conducted experiments, we collectively identified some factors that our method is sensitive to, which should be considered in practice. This section enumerates and discusses these considerations.

#### 7.1.1. Tag size

Once a tag is detected, one of the main factors affecting the tag measurement reliability is the tag's size in the image. A larger tag in the image produces more reliable corner point measurements. For fixed tag size and camera lens parameters, the relative camera-tag distance and orientation play a critical role in tag detection rate and corner point measurements. Therefore, using actual larger tags can help increase their effective range. The effective range of a tag is defined as the range within which the corresponding tag measurement is considered reliable.

The AprilTag algorithm [52] denies any 4-pixel or smaller line segments during the tag detection process. Thus, any tag with smaller projected side lengths of four pixels is undetectable. We found that in practice, 10–15 pixels is a minimum projected tag side length for consistent detections and reliable localization measurements. In an ideal situation, when the camera optical axis is perpendicular to a visible tag, the camera-to-tag distance can be roughly approximated as $d = \frac{f}{i} \times s$, where $d$ is camera-to-tag distance, $f$ is camera focal length, $i$ is minimum side length of the tag in the image, and $s$ is the tag size. Therefore, in a $l \times w$ image ($w < l$) and for a particular tag size and focal length, the upper limit for the effective range ($d_{max}$) is when $i \cong$ 15 pixels and the lower limit ($d_{\min}$) obtains when $i = w$.

Although larger tags will increase the effective range, deploying larger tags may not always be piratical on a construction site. Thus, we chose to use ubiquitous letter-size paper sheets in our experiments. This choice limits the tag size to $0.16 - 0.17\ m$ in length.

Moreover, given the assumption of planar tags in our formulation, any curl or bend can affect the measurements. The use of thicker paper may last longer in a construction setting and remain straight during deployment.

#### 7.1.2. Calibration

The constant rigid-body transformations in our formulation directly impact the estimation results. Therefore, an accurate estimation of transformations between the involved reference frames (e.g., camera, IMU, and ground truth) through calibration is crucial in getting reliable results. More importantly, as tag locations are assumed known, it is vital to pay particular attention while surveying them. Sensor calibration, including camera and IMU calibration, is also essential.

#### 7.1.3. Camera resolution and intrinsics

As discussed, camera-to-tag distance is a relative quantity that depends on tag size, camera resolution, and focal length. The effective range can be adjusted by tuning these camera properties (e.g., using a higher resolution camera or changing the focal length).

#### 7.1.4. Occlusions, motion blur, and illumination conditions

Occlusions, motion blur, and poorly lit environments degrade the performance of any vision-based technique, including ours. It impacts the detection rate and adds noise to our measurements. However, as our method uses IMU alongside, these conditions can be tolerated to some extent, relying on odometry data for short-term estimates. In general, the combination of visual and inertial information provides robustness to poor texture, motion blur, and occlusions.

#### 7.1.5. Tag-blind zones

The first experiment demonstrated that the proposed method is sufficiently robust to handle zones with no tags visible. However, since our system is tightly coupled in data fusion, it relies on odometry-based predictions when no tags are detected. Hence, reliable short-term estimates in these zones depend heavily on the prediction quality. When IMU data is used for prediction solely, the estimates are expected to drift relatively quickly. To improve that, given enough features in the scene, one can take advantage of natural features to remain more robust in areas without visible tags. Visual-inertial odometry using a frontal camera and IMU (e.g., VINS-mono [34]) is expected to perform more reliably in tag-blind zones. However, it comes with the cost of more computation, memory, and power demand.

### 7.2. Theoretical considerations

This section discusses the advantages of the proposed on-manifold formulation and the presented tag measurement model.

The advantages of the proposed on-manifold formulation over more

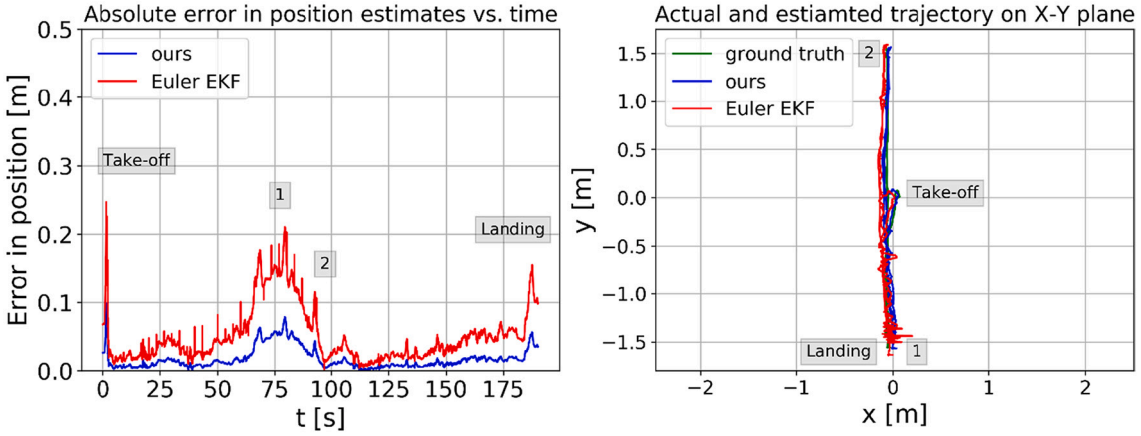



**Fig. 19.** - The comparison of the absolute error in position estimates (left) and the estimated and actual trajectory on X-Y plane (right) for Euler EKF (using Euler-angle-based parametrization and relative camera-tag transformations as measurements) and ours (on-manifold formulation and corner point correspondences as measurements) in "Straight line" experiment in laboratory. The deployed tag configuration is shown in Fig. 9.

traditional ones such as Euler-angle-based are threefold: (1) The Euler angle parametrization suffers from the singularity problem, also known as *gimbal lock*. For instance, for the *zyx* Euler sequence, a singularity exists at $\theta_y = \frac{\pi}{2} + k\pi\ (k \in \mathbb{Z})$. However, the pose and rotation in our formulation are stored in a singularity-free format. This is a critical property that allows for versatile applications of our method from the localization of autonomous UAVs to handheld devices such as smartphones; (2) The derivatives of scalar trigonometric functions in Euler-angle-based EKF can lead to cumulative numerical errors [65]. However, on-manifold manipulations occur at the matrix level, handling derivatives with precision and ease. For more information and quantitative performance comparisons between these two approaches, please refer to [73]; and (3) On-site tag placement may be subject to installation errors in practice. Although modeling the installation error remains out of the scope of this paper, the proposed formulation facilitates stochastic error modeling. This is not a straight-froward task in the case of more traditional approaches such as Euler-angle-based formulation.

The proposed tag measurement model is based on corner point correspondences rather than directly incorporating the relative transformation provided by the AprilTag package. Our experiments already showed that these relative transformations suffer from noisiness and instability, depending on the camera-tag relative distance.

To further investigate the impact of opting the proposed on-manifold formulation and tag measurement model, we estimate the same trajectory using our method and an Euler-angle-based EKF with direct tag measurements, denoted as *Euler EKF* in short. To this end, we compare ours with the EKF node of the open-source ROS localization package [74], *ekf_localization_node*. In this EKF implementation, Euler-angles are used. Moreover, this node gets camera-tag relative transformations as measurements directly. The results are shown in Fig. 19, where it is observable that the farther the UAV gets from the tags, the higher the errors in position are using *Euler EKF*. On the other hand, our approach more accurately and smoothly estimates the UAV's 3D position.

## 8. Conclusions and future work

Automated indoor data collection using autonomous mobile robots, including UAVs, can potentially increase the collection speed and accuracy of the frequent data required for construction inspections and tracking. One of the critical enablers for autonomous navigation is robust global localization. However, localization is challenging in low-texture, continually changing, and GPS-denied indoor construction environments. Additionally, most of the existing data capture solutions in the construction literature and industry are still costly. To address these challenges, this work proposed a low-cost, lightweight tag-based visual-inertial localization method to enable autonomous navigation of inexpensive, off-the-shelf UAVs, with a camera and an IMU, in indoor construction environments. The proposed formulation is based on an on-manifold EKF, suitably addressing the rotation and pose topological structure. In this implementation, we used AprilTags and a compact UAV, *Parrot Bebop2*. Our method was validated via case studies in both laboratory and a photo-realistic, BIM-enabled simulation environment. The performance was verified through quantitative and qualitative analyses. Our results showed our method could reach an RMSE of $2-5$ *cm* in position.

Having tags placed in known locations in the workspace, the proposed method can be instantly adapted and deployed for a wide range of indoor localization applications while overcoming many limitations that vision-based techniques may face in indoor construction environments (e.g., perceptual aliasing and feature scarcity). However, manual tag placement/replacement can be tedious and should be optimized using a tag placement planner. The manual process of tag placement/replacement may also be subject to installation errors, affecting the performance of tag-based localization. Paper-printable tags may be subject to damage in indoor construction sites, which can be improved by using thicker paper sheets or spraying the tags instead of printing them on paper, where applicable. Although studying safety impacts is beyond the scope of this research, any safety issues that mobile robots, including UAVs, may bring about in indoor construction are worth investigating. Quantitative performance analysis of the state-of-the-art localization methods in indoor construction environments is another interesting topic to be investigated.

In our future work, we are interested in feeding the tag-based estimates in a feedback control loop to autonomously fly, perform automatic data collection missions, and extend our validation in an actual construction setting. Another extension to this work is to consider the installation errors in the placement-replacement process using a stochastic approach. Another direction we plan to investigate is planning tag placement/replacements to reduce the manual work. We are also interested in using our method for localizing multiple platforms in a construction setting. Further studies may focus on bringing the computations on board or mapping the tags in the environment.

Interested readers are referred to the supplementary videos for more visualizations and information regarding the experiments conducted in the laboratory [72] and the simulation [71] environments.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.autcon.2021.104112.

## Declaration of Competing Interest

✓ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

Financial support from the Natural Science and Engineering Research Council (NSERC) grant number RGPIN-2017-06792 is greatly appreciated. The first author is grateful to Professors Tim Barfoot and Kamran Esmaeili for their constructive advice and thankful to Abhishek Goudar, Adam Heins, and Lukas Brunke from the Dynamics System Lab at the University of Toronto Institute for Aerospace Studies (UTIAS) for their invaluable inputs and collaborations. The presented opinions, findings, conclusions in this work are those of authors and do not necessarily reflect the views of the entities mentioned above.

## References

[1] T. Czerniawski, F. Leite, Automated digital modeling of existing buildings: a review of visual object recognition methods, Autom. Constr. 113 (2020), https://doi.org/10.1016/j.autcon.2020.103131.
[2] S. Cai, Z. Ma, M.J. Skibniewski, S. Bao, Construction automation and robotics for high-rise buildings over the past decades: a comprehensive review, Adv. Eng. Inform. 42 (2019), 100989, https://doi.org/10.1016/j.aei.2019.100989.
[3] J. Teizer, Status quo and open challenges in vision-based sensing and tracking of temporary resources on infrastructure construction sites, Adv. Eng. Inform. (2015), https://doi.org/10.1016/j.aei.2015.03.006.
[4] J.S. Bohn, J. Teizer, Benefits and barriers of construction project monitoring using high-resolution automated cameras, J. Constr. Eng. Manag. (2010), https://doi.org/10.1061/(ASCE)CO.1943-7862.0000164.
[5] K. Asadi, A. Kalkunte Suresh, A. Ender, S. Gotad, S. Maniyar, S. Anand, M. Noghabaei, K. Han, E. Lobaton, T. Wu, An integrated UGV-UAV system for construction site data collection, Autom. Constr. 112 (2020), 103068, https://doi.org/10.1016/j.autcon.2019.103068.
[6] K. Asadi, H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, T. Wu, Vision-based integrated mobile robotic system for real-time applications in construction, Autom. Constr. 96 (2018) 470–482, https://doi.org/10.1016/j.autcon.2018.10.009.
[7] Y. Ham, K.K. Han, J.J. Lin, M. Golparvar-Fard, Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (UAVs): a review of related works, Visualization Eng. 4 (2016) 1, https://doi.org/10.1186/s40327-015-0029-z.
[8] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, G. Reitmayr, Augmented reality for construction site monitoring and documentation, Proc. IEEE 102 (2014) 137–154, https://doi.org/10.1109/JPROC.2013.2294314.
[9] S. Siebert, J. Teizer, Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system, Autom. Constr. (2014), https://doi.org/10.1016/j.autcon.2014.01.004.
[10] L. Xu, C. Feng, V.R. Kamat, C.C. Menassa, An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments, Autom. Constr. 104 (2019) 230–245, https://doi.org/10.1016/j.autcon.2019.04.011.
[11] H. Hamledari, B. McCabe, S. Davari, Automated computer vision-based detection of components of under-construction indoor partitions, Autom. Constr. 74 (2017) 78–94, https://doi.org/10.1016/j.autcon.2016.11.009.
[12] B.Y. McCabe, H. Hamledari, A. Shahi, P. Zangeneh, E.R. Azar, Roles, Benefits, and challenges of using UAVs for indoor smart construction applications, Congress on Computing in Civil Engineering, Proceedings. 2017 (June 2017) 349–357, https://doi.org/10.1061/9780784480830.043.
[13] M. Blösch, S. Weiss, D. Scaramuzza, R. Siegwart, Vision based MAV navigation in unknown and unstructured environments, Proceedings - IEEE Int. Conf. on Robotics and Automation. (2010) 21–28, https://doi.org/10.1109/ROBOT.2010.5509920.
[14] R. Muñoz-Salinas, M.J. Marín-Jimenez, R. Medina-Carnicer, SPM-SLAM: simultaneous localization and mapping with squared planar markers, Pattern Recogn. 86 (2019) 156–171, https://doi.org/10.1016/j.patcog.2018.09.003.
[15] A. Ibrahim, A. Sabet, M. Golparvar-Fard, BIM-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform, in: Proceedings of the 2019 European Conference for Computing in Construction 1, 2019, pp. 182–189, https://doi.org/10.35490/ec3.2019.195.
[16] D. Scaramuzza, Z. Zhang, Aerial robots, visual-inertial Odometry of, encyclopedia of, Robotics. (2020) 1–9, https://doi.org/10.1007/978-3-642-41610-1_71-1.
[17] S. Shen, N. Michael, V. Kumar, Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs, Proceedings - IEEE Int. Conf. on Robotics and Automation. (June 2015) 5303–5310, https://doi.org/10.1109/ICRA.2015.7139939.
[18] L. Xu, C. Feng, V.R. Kamat, C.C. Menassa, A scene-adaptive descriptor for visual SLAM-based locating applications in built environments, Autom. Constr. 112 (2020), 103067, https://doi.org/10.1016/j.autcon.2019.103067.
[19] P. Kim, J. Chen, Y.K. Cho, SLAM-driven robotic mapping and registration of 3D point clouds, Autom. Constr. 89 (2018) 38–48, https://doi.org/10.1016/j.autcon.2018.01.009.
[20] Spot | Boston Dynamics. https://www.bostondynamics.com/spot, 2021 (accessed May 17, 2021).
[21] N. Kayhani, B. McCabe, A. Abdelaal, A. Heins, A.P. Schoellig, Tag-Based Indoor Localization of UAVs in Construction Environments: Opportunities and Challenges in Practice, in: Constr. Res. Congr. 2020, American Society of Civil Engineers, Reston, VA, 2020, pp. 226–235, https://doi.org/10.1061/9780784482865.025.
[22] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, ROS: An Open-Source Robot Operating System. http://stair.stanford.edu, 2009.
[23] T.P. Deasy, W.G. Scanlon, Stepwise algorithms for improving the accuracy of both deterministic and probabilistic methods in WLAN-based indoor user localisation, Int. J. Wireless Inf. Networks 11 (2004) 207–215, https://doi.org/10.1007/s10776-004-1234-1.
[24] T. Liu, L. Yang, Q. Lin, Y. Guo, Y. Liu, Anchor-free backscatter positioning for RFID tags with high accuracy, in, Proc. - IEEE INFOCOM (2014) 379–387, https://doi.org/10.1109/INFOCOM.2014.6847960.
[25] S.N. Razavi, O. Moselhi, GPS-less indoor construction location sensing, Autom. Constr. 28 (2012) 128–136, https://doi.org/10.1016/j.autcon.2012.05.015.
[26] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent advances in indoor localization: a survey on theoretical approaches and applications, IEEE Commun. Surv. and Tutorials. 19 (2017) 1327–1346, https://doi.org/10.1109/COMST.2016.2632427.
[27] A. Shahi, A. Aryan, J.S. West, C.T. Haas, R.C.G. Haas, Deterioration of UWB positioning during construction, Autom. Constr. (2012), https://doi.org/10.1016/j.autcon.2012.02.009.
[28] L. Mainetti, L. Patrono, I. Sergi, A survey on indoor positioning systems, 2014 22nd international conference on software, telecommunications and computer networks, SoftCOM 2014 (2014) 111–120, https://doi.org/10.1109/SOFTCOM.2014.7039067.
[29] J. Xiao, Z. Zhou, Y. Yi, L.M. Ni, A survey on wireless indoor localization from the device perspective, ACM Comput. Surv. 49 (2016) 1–31, https://doi.org/10.1145/2933232.
[30] K. Witrisal, P. Meissner, Performance bounds for multipath-assisted indoor navigation and tracking (MINT), IEEE Int. Conf. Commun. (2012) 4321–4325, https://doi.org/10.1109/ICC.2012.6363827.
[31] M. Ibrahim, O. Moselhi, Inertial measurement unit based indoor localization for construction applications, Autom. Constr. 71 (2016) 13–20, https://doi.org/10.1016/j.autcon.2016.05.006.
[32] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza, SVO: Semidirect visual Odometry for monocular and multicamera systems, IEEE Trans. Robot. (2017), https://doi.org/10.1109/TRO.2016.2623335.
[33] C. Campos Martínez, R. Elvira, J.J. Gómez Rodríguez, J.M.M. Montiel, J.D. Tardós, ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-inertial and Multi-map SLAM, ArXiv, 2020, https://doi.org/10.1109/TRO.2021.3075644.
[34] T. Qin, P. Li, S. Shen, VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator, IEEE Trans. Robot. 34 (2018) 1004–1020, https://doi.org/10.1109/TRO.2018.2853729.
[35] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): part II, IEEE Robot. Autom. Mag. 13 (2006) 108–117, https://doi.org/10.1109/MRA.2006.1678144.
[36] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1052–1067, https://doi.org/10.1109/TPAMI.2007.1049.
[37] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Trans. Robot. 31 (2015) 1147–1163, https://doi.org/10.1109/TRO.2015.2463671.
[38] H. Strasdat, J.M.M. Montiel, A.J. Davison, WITHDRAWN: visual SLAM: why filter? Image Vis. Comput. (2012) https://doi.org/10.1016/j.imavis.2012.08.007.
[39] F. Boniardi, A. Valada, R. Mohan, T. Caselitz, W. Burgard, Robot localization in floor plans using a room layout edge extraction network, IEEE International Conference on Intelligent Robots and Systems. (2019) 5291–5297, https://doi.org/10.1109/IROS40897.2019.8967847.
[40] H. Peel, S. Luo, A.G. Cohn, R. Fuentes, Localisation of a mobile robot for bridge bearing inspection, Autom. Constr. 94 (2018) 244–256, https://doi.org/10.1016/j.autcon.2018.07.003.
[41] H. Freimuth, M. König, Planning and executing construction inspections with unmanned aerial vehicles, Autom. Constr. 96 (2018) 540–553, https://doi.org/10.1016/j.autcon.2018.10.016.
[42] J.J. Lin, A. Ibrahim, S. Sarwade, M. Golparvar-Fard, Bridge inspection with aerial robots: automating the entire pipeline of visual data capture, 3D mapping, Defect Detection, Analysis, and Reporting, J. Comp. Civil Eng. 35 (2021) 04020064, https://doi.org/10.1061/(asce)cp.1943-5487.0000954.
[43] F. Wang, J.-Q. Cui, B.-M. Chen, T.H. Lee, A comprehensive UAV indoor navigation system based on vision optical flow and laser FastSLAM, Acta Automat. Sin. 39 (2013) 1889–1899, https://doi.org/10.3724/SP.J.1004.2013.01889.
[44] M. Gheisari, J. Irizarry, B.N. Walker, UAS4SAFETY: the potential of unmanned aerial systems for construction safety applications, in: Constr. Res. Congr. 2014, American Society of Civil Engineers, Reston, VA, 2014, pp. 1801–1810, https://doi.org/10.1061/9780784413517.184.

[45] J.G. Martinez, G. Albeaino, M. Gheisari, R.R.A. Issa, L.F. Alarcón, iSafeUAS: an unmanned aerial system for construction safety inspection, Autom. Constr. 125 (2021), https://doi.org/10.1016/j.autcon.2021.103595.
[46] M. Jin, S. Liu, S. Schiavon, C. Spanos, Automated mobile sensing: towards high-granularity agile indoor environmental quality monitoring, Build. Environ. 127 (2018) 268–276, https://doi.org/10.1016/j.buildenv.2017.11.003.
[47] A. Adán, B. Quintana, S.A. Prieto, F. Bosché, An autonomous robotic platform for automatic extraction of detailed semantic models of buildings, Autom. Constr. 109 (2020), 102963, https://doi.org/10.1016/j.autcon.2019.102963.
[48] B.R.K. Mantha, C.C. Menassa, V.R. Kamat, Robotic data collection and simulation for evaluation of building retrofit performance, Autom. Constr. 92 (2018) 88–102, https://doi.org/10.1016/j.autcon.2018.03.026.
[49] K. Asadi, H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, T. Wu, Vision-based integrated mobile robotic system for real-time applications in construction, Autom. Constr. 96 (2018) 470–482, https://doi.org/10.1016/j.autcon.2018.10.009.
[50] M. Fiala, ARTag, a fiducial marker system using digital techniques, in: Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR, 2005, pp. 590–596, https://doi.org/10.1109/CVPR.2005.74.
[51] E. Olson, A robust and flexible visual fiducial system, Proceedings - IEEE Int. Conf. on Robotics and Automation. (2011) 3400–3407, https://doi.org/10.1109/ICRA.2011.5979561.
[52] J. Wang, E. Olson, AprilTag 2: efficient and robust fiducial detection, IEEE International Conference on Intelligent Robots and Systems. (Novem 2016) 4193–4198, https://doi.org/10.1109/IROS.2016.7759617.
[53] B. Atcheson, F. Heide, W. Heidrich, CALTag: High precision fiducial markers for camera calibration, in: VMV 2010 - Vision, Model. Vis., 2010, pp. 41–48, https://doi.org/10.2312/PE/VMV/VMV10/041-048.
[54] G. Schweighofer, A. Pinz, Robust pose estimation from a planar target, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 2024–2030, https://doi.org/10.1109/TPAMI.2006.252.
[55] R. Muñoz-Salinas, M.J. Marín-Jimenez, E. Yeguas-Bolivar, R. Medina-Carnicer, Mapping and localization from planar markers, Pattern Recogn. 73 (2018) 158–171, https://doi.org/10.1016/j.patcog.2017.08.010.
[56] C. Brommer, D. Malyuta, D. Hentzen, R. Brockers, Long-duration autonomy for Small Rotorcraft UAS Including Recharging, in: 2018 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2018, pp. 7252–7258, https://doi.org/10.1109/IROS.2018.8594111.
[57] K. Shaya, A. Mavrinac, J.L.A. Herrera, X. Chen, A self-localization system with global error reduction and online map-building capabilities, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2012, pp. 13–22, https://doi.org/10.1007/978-3-642-33503-7_2.
[58] N. Kayhani, A. Heins, W.D. Zhao, M. Nahangi, B. McCabe, A.P. Schoellig, Improved tag-based indoor localization of UAVs using extended Kalman Filter, in: Proc. 36th Int. Symp. Autom. Robot. Constr. ISARC 2019, 2019, pp. 624–631, https://doi.org/10.22260/isarc2019/0083.
[59] R. Muñoz-salinas, M.J. Marín-jimenez, R. Medina-carnicer, SPM-SLAM : simultaneous localization and mapping with squared planar markers, Pattern Recogn. 86 (2019) 156–171, https://doi.org/10.1016/j.patcog.2018.09.003.
[60] R. Muñoz-Salinas, R. Medina-Carnicer, UcoSLAM: simultaneous localization and mapping by fusion of keypoints and squared planar markers, Pattern Recogn. 101 (2020), https://doi.org/10.1016/j.patcog.2019.107193.
[61] J.L. Sanchez-Lopez, V. Arellano-Quintana, M. Tognon, P. Campoy, A. Franchi, Visual marker based multi-sensor fusion state estimation * *during this work Jose Luis Sanchez-Lopez has been funded by the Eiffel excellence scholarship program of the French Ministry of Foreign Affairs and international development and victor Arellano-Q, IFAC-PapersOnLine. 50 (2017) 16003–16008, https://doi.org/10.1016/j.ifacol.2017.08.1911.
[62] M. Neunert, M. Bloesch, J. Buchli, An open source, fiducial based, visual-inertial motion capture, System (2015), https://doi.org/10.1080/03610926.2010.521279.
[63] H.E. Nyqvist, M.A. Skoglund, G. Hendeby, F. Gustafsson, Pose estimation using monocular vision and inertial sensors aided with ultra wide band, in: 2015 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015, 2015, pp. 13–16, https://doi.org/10.1109/IPIN.2015.7346940.
[64] P. Jin, P. Matikainen, S.S. Srinivasa, Sensor fusion for fiducial tags: highly robust pose estimation from single frame RGBD, IEEE International Conference on Intelligent Robots and Systems. (Septe 2017) 5770–5776, https://doi.org/10.1109/IROS.2017.8206468.
[65] T.D. Barfoot, State Estimation for Robotics, Cambridge University Press, Cambridge, 2017, https://doi.org/10.1017/9781316671528.
[66] Parrot Drones - Discover our range of professional drones. https://www.parrot.com/en/drones, 2021 (accessed May 17, 2021).
[67] GitHub - AutonomyLab/bebop_autonomy, ROS driver for Parrot Bebop Drones 1.0 & 2.0, 2021. https://github.com/AutonomyLab/bebop_autonomy (accessed May 7, 2021).
[68] AprilRobotics/apriltag. https://github.com/AprilRobotics/apriltag, 2021 (accessed May 7, 2021).
[69] M. Krogius, A. Haggenmiller, E. Olson, Flexible Layouts for Fiducial Tags, 2020, pp. 1898–1903, https://doi.org/10.1109/iros40897.2019.8967787.
[70] Parrot-Sphinx 1.2.1 documentation. https://developer.parrot.com/docs/sphinx/whatissphinx.html, 2021 (accessed June 21, 2020).
[71] N. Kayhani, Simulation Tests (Mixed Motion) - YouTube. https://youtu.be/91pLvsEtxCk, 2021 (accessed May 17, 2021).
[72] N. Kayhani, Lab Tests (3D Circular Trajectory) - YouTube. https://youtu.be/76PFOSlYQGs, 2021 (accessed May 18, 2021).
[73] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza, On-manifold Preintegration for real-time visual-inertial Odometry, IEEE Trans. Robot. 33 (2017) 1–21, https://doi.org/10.1109/TRO.2016.2597321.
[74] T. Moore, D. Stouch, A generalized extended kalman filter implementation for the robot operating system, in: Adv. Intell. Syst. Comput., 2016, pp. 335–348, https://doi.org/10.1007/978-3-319-08338-4_25.