Received: 24 March 2018 | Revised: 30 April 2019 | Accepted: 2 May 2019

DOI: 10.1002/asjc.2162

SPECIAL ISSUE

WILEY

A lightweight autonomous MAV for indoor search and rescue

Yingcai Bi[1] | Menglu Lan[1] | Jiaxin Li[1] | Shupeng Lai[2] | Ben M. Chen[2]

[1]NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore

[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Correspondence
Yingcai Bi, NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore.
Email: yingcaibi@u.nus.edu

Abstract

Micro Aerial Vehicles (MAVs) have great potentials to be applied for indoor search and rescue missions. In this paper, we propose a modular lightweight design of an autonomous MAV with integrated hardware and software. The MAV is equipped with the 2D laser scanner, camera, mission computer and flight controller, running all the computation onboard in real time. The onboard perception system includes a laser-based SLAM module and a custom-designed visual detection module. A dual Kalman filter design provides robust state estimation by multiple sensor fusion. Specifically, the fusion module provides robust altitude measurement in the circumstance of surface changing. In addition, indoor-outdoor transition is explicitly handled by the fusion module. In order to efficiently navigate through obstacles and adapt to multiple tasks, a task tree-based mission planning method is seamlessly integrated with path planning and control modules. The MAV is capable of searching and rescuing victims from unknown indoor environments effectively. It was validated by our award-winning performance at the 2017 International Micro Air Vehicle Competition (IMAV 2017), held in Toulouse, France. The performance video is available on https://youtu.be/8H19ppS_VXM.

KEYWORDS
autonomous MAV, perception, planning, state estimation

1 | INTRODUCTION

Recently, MAVs are widely applied to aerial photography, infrastructures (e.g. bridge, power line) inspection, onsite mapping and others. In most cases, human pilots are required to control the drone manually. Some basic autonomous maneuvers may be applied with the presence of satisfactory GPS availability. However, a large variety of applications for MAVs would rely on the autonomous capabilities in GPS-denied environments to handle difficult, dirty, dangerous jobs. A typical scenario is a post-disaster search and rescue mission, in which MAVs must fly among buildings with GPS blocking or inside buildings without any GPS signal. To this end, a fully functional system incorporated with autonomous navigation capabilities is in high demand. This paper aims at developing a fully autonomous MAV working in typical GPS-denied environments, and applying it on indoor search and rescue.

Development of autonomous MAVs is one of the most active research areas in recent years. Early success on autonomous indoor flight based on 2D laser scanner are reported in [1] and [2]. They both emphasize the localization and planning components which are actually adapted from previous research on ground robots for autonomous indoor exploration. Later on, the accomplishment of autonomous flight with a camera as the primary sensor is shown in [3]. It is the very first systematic design to deal with vision-based autonomous navigation. A lot of recent attempts [4–6] focus on vision-based solutions. However, it is still a challenging problem in real applications considering computation and robustness constraints.

Algorithm development on fully autonomous MAVs are limited by the computational power of the platform. The software components consist of perception, planning, state estimation and control. In our designed system, the perception module includes laser-based localization and visual detection module. The planning module consists of mission planning and motion planning. The mission planning is necessary to juggle the demands of complex task lists in autonomous missions. It is a key component for fast deployment and can handle different situations in a well-organized manner. The motion planning provides collision-free and dynamically feasible tracking references for MAVs. For control purpose, the state estimation module provides high-frequency 6DOF estimation (3D position and attitude) by fusing multiple sensor measurements.

In GPS-denied environments, the solution for localization is a typical simultaneous localization and mapping (SLAM) problem. 2D laser-based SLAM [7,8] has been demonstrated to be a light-weight and high-accuracy solution. The performance is achieved under the 2.5D environment assumption, i.e, the environment in the vertical direction has no significant changes. This is often valid when MAVs fly with a fixed altitude. When the assumption is violated, a multi-level strategy [9] can be adapted. Vision-based SLAM [10,11] shows promising progress recently. However, it is still sensitive to light condition and tends to fail in featureless environments. Therefore, in a typical structured indoor environment which consists of texture-less walls, flying with a camera robustly is still a challenge. In this paper, We adopt one of the best laser SLAM algorithms [8] for localization purpose.

The state estimation can be solved with variants of Kalman filter conventionally [12], and also optimization-based methods thanks to the improved computation power recently [13]. Though optimization-based methods may lead to a more accurate solution, the computational resource limits their applicability to most of the embedded systems. In contrary, Kalman filter based methods, such as Extended Kalman Filter (EKF) [14] and Unscented Kalman Filter (UKF) [15], are widely applied for onboard sensor fusion and provide satisfactory results with better efficiency. To fulfill the state estimation requirements, we design dual Kalman filters for our system to handle complex scenarios, such as surface changing and indoor-outdoor transition.

Most of motion planning algorithms for MAVs adopt a two-layered approach [4,16], which decouples the vehicle dynamics and geometric constraints. Normally, a global planner is used to find a free path first and then a trajectory generation procedure is involved to handle the vehicle dynamics. In [16], a standard A* algorithm is performed in an octree-based map to generate a collision-free corridor. Then they formulate the corridor constraints into a set of convex position constraints and generate dynamically feasible trajectories within the corridor using numerical optimization techniques. The motion planning used in this paper is based on our previous work [17], which also use a two-layered approach and A* as the global planner. By comparison, to avoid numerical instabilities, we utilize an analytically two-point boundary value problem (TPBVP) solver to generate jerk-limited time-optimal trajectory.

This paper is an extension of the conference paper [18]. We introduce our light-weight MAV solution for autonomous flight in GPS-denied environments. By designing every component of the MAV, robust performance is achieved. In this paper, we explain the system in a more comprehensive manner, addressing the perception, state estimation and integrated planning methods for complex search and rescue missions. we discuss the improvement details to enhance the state estimator for more complex flight conditions. Furthermore, we present more experimental results to validate the MAV system.

This paper is organized as follows. Section 2 introduces the MAV platform, onboard hardware and software design. Section 3 describes the laser SLAM and visual target detection methods. Section 4 presents the multi-sensor state estimation by assembling dual Kalman filters. Section 5 explains the integration of our task planning, motion planning and flight control. Section 6 shows results from various experiments to validate the full system. Finally, we give the conclusion and future direction in Section 7.

2 | FLYING PLATFORM

2.1 | Hexarotor platform

We design a hexarotor MAV, shown in Figure 1, as a lightweight and versatile platform. It adopts a sandwich mechanical structure to minimize the weight of airframe and add the flexibility of changing onboard components. The avionics are customized to fit the structure and achieve an optimal center of gravity placement. The arm is mainly made of carbon fiber to reduce the moment of inertia. Also, Blade protector can be installed for indoor applications in case of collisions.

2.2 | Onboard avionics

A Hokuyo 2D laser scanner is mounted on top to provide range information in the horizontal direction. A downward-facing camera is used for vision-based detection

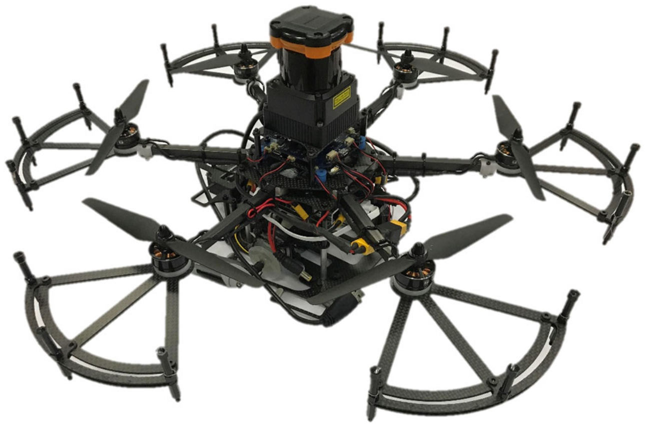


FIGURE 1 The hexarotor platform [Color figure can be viewed at wileyonlinelibrary.com]

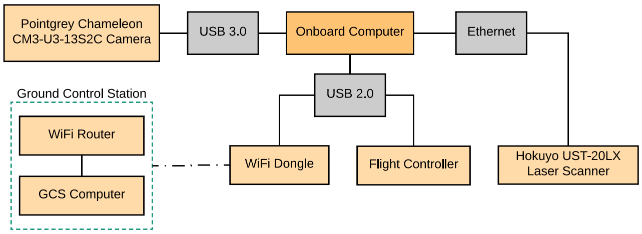


FIGURE 2 The hardware structure [Color figure can be viewed at wileyonlinelibrary.com]

and guidance. The flight control module includes various sensors such as gyroscopes, accelerometers, magnetometers, to measure the state of the MAV. Power distribution is implemented on a separate board to minimize electromagnetic interference. To balance between computation power and weight, we use a small-size (80g) computer with Intel Atom x5-Z8350 processor. With our optimized onboard software, it is capable of running perception, SLAM and planning simultaneously. The structure of hardware components and interfaces is illustrated in Figure 2. The Maximum take-off weight of the fully configured MAV can reach 1.8kg, with optional 200g payload.

2.3 | Onboard software

The software structure is modularized with a robust flight controller and self-designed UAVOS middleware based on the Robot Operating System (ROS). Figure 3 highlights the structure of our onboard software system. The main modules are perception, planning and flight control. Inside each module, there are multiple submodules which can be easily customized for different tasks.

The modular UAVOS architecture is necessary and efficient to enable fast development and deployment according to different mission elements. The ROS-based structure

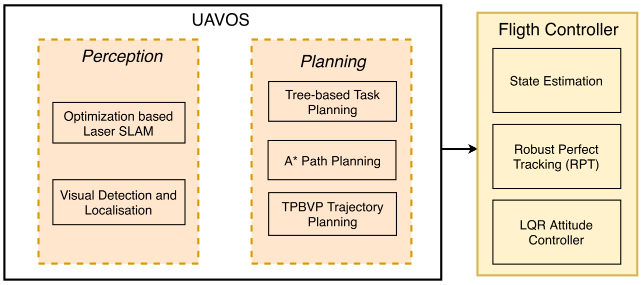


FIGURE 3 The software structure [Color figure can be viewed at wileyonlinelibrary.com]

can provide the capability thanks to the publish-subscribe messaging pattern. For example, when a new sensor is connected to the onboard computer, a driver module can be included and every other module can use the sensor data by only subscribing the topics accordingly. For applications of autonomous indoor navigation, the MAV can be configured with 2D laser scanner for 2.5D navigation (our case), or with cameras for 3D navigation. Although the two configurations differ a lot, only part of the perception module needs to be changed.

3 | PERCEPTION

3.1 | Laser SLAM

For structured 2.5D environments, the 2D laser scanner can provide accurate planar positioning information for MAVs. In this paper, we evaluate two state-of-the-art laser SLAM algorithms as the front-end of our state estimation module. They are Hector SLAM [7] and Cartographer SLAM [8]. To compare them, We briefly discuss and differentiate these two methods here.

Hector SLAM calculates the pose by aligning incoming laser scan with the map accumulated from all previous scans. Gauss-Newton optimization is utilized to find an optimal transformation. Moreover, a multi-resolution map representation inspired by image pyramids is applied to improve numerical stability. In comparison, Cartographer SLAM shares similar scan to map matching method as Hector SLAM. But it cooperates submaps and real-time loop closure algorithm to further reduce the position drift. The key idea is to build multiple submaps and generate constraints on a graph. The real-time loop closure is achieved by a branch-and-bound approach. Cartographer SLAM is a representative modern SLAM method, while Hector SLAM is regarded as accurate odometry that may drift without the help of loop closure.

3.2 | Visual target detection

We develop a visual detection module which supports both fiducial markers and color targets. It is an important supplement for the search and rescue mission. The relative position and orientation of targets can be calculated in real-time. We use the fiducial marker for precision landing on a moving platform, and color visual detection for searching the victims. The fiducial marker deployed in our system is Apriltag [19]. It is based on a near-optimal lexicographic coding system. Four corners on a marker can be detected with corresponding order under different lighting conditions and view angles. Utilizing the exact size of the marker, the camera pose and target position in body frame are completely determined with OpenCV solvePnP function. The Apriltag detection flowchart is shown in Figure 4.

For the color target detection, a color segmentation on HSV channel is first applied on the image. Thanks to the flexible task planning, we can restrict the searching area to improve detection efficiency. Therefore, a larger HSV threshold range is applied to increase the detection robustness. In our implementation with OpenCV, the HSV range for red color is from (165, 40, 50) to (179, 255, 255) and from (0, 40, 50) to (15, 255, 255). Contours are detected and analyzed in the binary image after segmentation. The detected contour which has the largest area and satisfies the minimum area requirement is regarded as the correct target. The target center is extracted and the target position can be computed considering the MAV's current state and assuming the target is on the ground plane. The color target detection algorithm is explained in Algorithm 1.

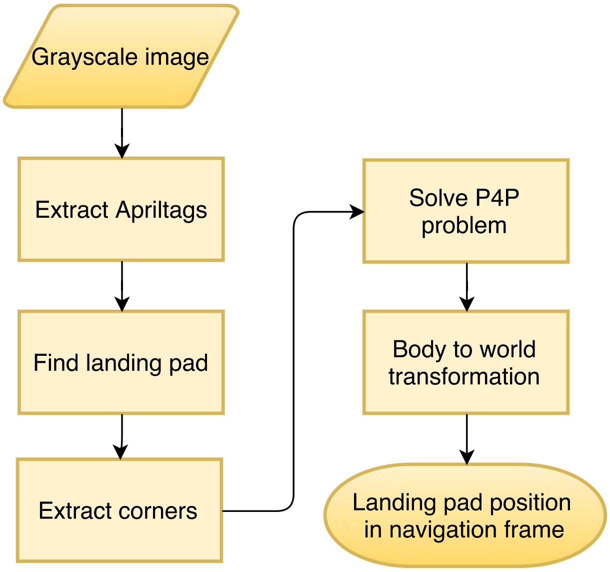


FIGURE 4 The fiducial target detection flowchart [Color figure can be viewed at wileyonlinelibrary.com]

Algorithm 1 Color target detection

```
segmentation in HSV color space
find all contours in the binary image
if any contour is found then
    approximate contours with polygons
    extract image centers from the minimal enclosing
    circles of polygons
    identify the contour with largest area
else
    return not found
end if
if largest area is greater than threshold then
    color target is detected
    calculate target position with line-of-sight angle
    and current MAV state
    return target position in navigation frame
else
    return not found
end if
```

4 | STATE ESTIMATION

4.1 | Sensor fusion structure

To get a consistent state estimation for the autonomous MAV, we design a two-stage multi-sensor fusion structure. The multi-sensor fusion structure is shown in Figure 5. The attitude and position are estimated with dual Kalman filters. The attitude filter takes angular rates, accelerations, heading from laser SLAM or magnetometer to estimate attitude. At the same time, the position filter utilizes position measurement from laser SLAM or GPS, height measurement from a distance sensor, altitude measurement from a barometer, accelerations and attitude estimation to provide a smooth position output. The design of dual filters decouples the attitude and position estimations, therefore is easy to tune in practical applications.

To be consistent with the frame definition of the control system, we define all the related frames in North-East-Down (NED) order. Figure 6 depicts our frame definition. O_G is the global navigation frame fixed on the takeoff position, and O_L is the local frame fixed on the position when a positioning sensor is available. In an indoor flight without GPS, O_G and O_L are coincident since only SLAM is working all the time. In an indoor-outdoor transition flight, the local frame is decided according to the starting position of each positioning sensor. O_l, O_c, O_d represents onboard laser frame, camera frame and distance sensor frame accordingly. They are attached on respective sensors rigidly. The body frame O_b is fixed on the position of inertial management unit (IMU). The relative transfor-

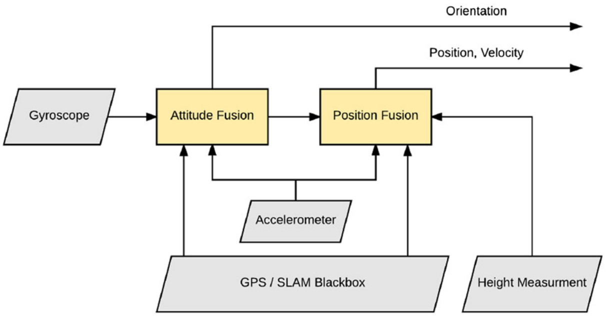


FIGURE 5 The sensor fusion structure [Color figure can be viewed at wileyonlinelibrary.com]

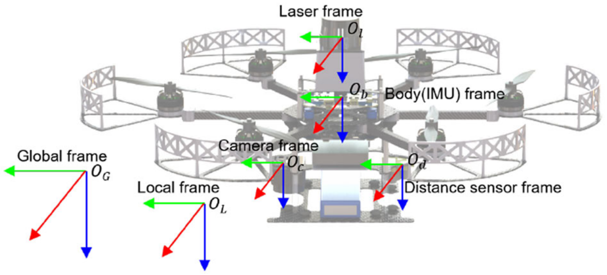


FIGURE 6 The sensor frame definition [Color figure can be viewed at wileyonlinelibrary.com]

mation T between body frame and each sensor frame is assumed to be calibrated offline. All sensor measurements are transformed into body frame before fusion.

4.2 | Sensor models

The IMU provides 3-axis accelerations and 3-axis angular rates with respect to body frame. The measurement model contains the Gaussian white noise term $n(t)$ and a slowly varying sensor bias $b(t)$. The acceleration measurement $a_m(t)$ and angular rate measurement $\omega_m(t)$ on each axis are therefore written as:

$$\omega_m(t) = \omega(t) + b_\omega(t) + n_\omega(t), \tag{1}$$

$$a_m(t) = a(t) + b_a(t) + n_a(t). \tag{2}$$

We use a barometer as a sensor that measures altitude change. The altitude change in meters can be converted from the pressure measured in Pascal [20] by

$$\Delta h = 44330 \cdot \left(1 - (\frac{P}{P_0})^{0.19}\right) - h_{init}, \tag{3}$$

where $P_0 = 101.325kPa$ is the standard condition for pressure, and h_{init} is the initial altitude calculated at the take-off position. The barometer measurement is sensitive to atmospheric pressure change and contains thermal noise, which makes it unsuitable to track highly dynamic movements due to high noise level and low update rate. The solution in [20] models the altitude measurement from barometer as a zero-mean Gaussian white noise (standard deviation of $1m$) and use a reference barometer to eliminate the atmospheric pressure change. They achieved sufficient accuracy, but the reference barometer is not applicable on MAVs which is common to encounter air disturbance while flying. Therefore, we model the altitude measurements with one more term of time-varying drift which is the barometer altitude bias b_{baro}:

$$h_{baro} = h(t) + b_{baro}(t) + n_{baro}(t). \tag{4}$$

The barometer bias can be observed when distance sensor is available. For the distance sensor, we model it with a zero-mean additive Gaussian white noise by considering the height of underlying surface:

$$h_{dist} = h(t) - h_{sur}(t) + n_{dist}(t), \tag{5}$$

where $h_{sur}(t)$ is the ground surface height which is updated when a valid surface change is detected.

4.3 | Attitude estimation

The orientation estimation is based on a Kalman filter represented with quaternion [21]. Based on gravity measurements, only roll and pitch angle are observable but the heading angle is not. The heading angle is observed by fusing the heading measurement from laser SLAM. This intrinsically solves the issue of severe magnetic interference in indoor environments and provides an accurate and smooth orientation for flight control.

The state vector includes angular velocities $\boldsymbol{\omega}_b$, angular accelerations $\dot{\boldsymbol{\omega}}_b$, vector of gravity $\mathbf{r}_g$, vector of laser heading $\mathbf{r}_l$ in body frame. The measurement vector contains angular velocities, the vector of gravity, the vector of heading measured directly from laser SLAM:

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\omega}_b \\ \dot{\boldsymbol{\omega}}_b \\ \mathbf{r}_g \\ \mathbf{r}_l \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \bar{\boldsymbol{\omega}}_b \\ \bar{\mathbf{r}}_g \\ \bar{\mathbf{r}}_l \end{bmatrix}. \tag{6}$$

A constant angular acceleration is assumed in the process model. The velocity $\dot{\mathbf{r}}$ of any vector in body frame is related with the rotation of body frame $\boldsymbol{\omega}_b$. The relationship is

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}_b + \boldsymbol{\omega}_b^{\wedge}\mathbf{r}_b, \tag{7}$$

in which $[\cdot]^{\wedge}$ represents the skew symmetric matrix. We have $\dot{\mathbf{r}}_b = 0$ since there is no time dependent motion for gravity and heading vector. The non-linear dynamic model is

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}) = \begin{pmatrix} \boldsymbol{\omega}_k + \dot{\boldsymbol{\omega}}_k \Delta t + \mathbf{w}_{\boldsymbol{\omega}} \\ \dot{\boldsymbol{\omega}}_k + \mathbf{w}_{\dot{\boldsymbol{\omega}}} \\ \mathbf{r}_{g,k} + \boldsymbol{\omega}_k^{\wedge}\mathbf{r}_{g,k}\Delta t + \mathbf{w}_{\mathbf{r}_g} \\ \mathbf{r}_{l,k} + \boldsymbol{\omega}_k^{\wedge}\mathbf{r}_{l,k}\Delta t + \mathbf{w}_{\mathbf{r}_l} \end{pmatrix}, \tag{8}$$

where $\mathbf{w}_{[\cdot]}$ represents the relevant Gaussian white noise. For a standard EKF algorithm, the linearized Jacobian matrices are

$$\mathbf{A}_k = \frac{\partial \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k)}{\partial \mathbf{x}_k} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\tilde{\mathbf{r}}_{g,k}\Delta t & \mathbf{0} & \mathbf{I} + \boldsymbol{\omega}_k^{\wedge}\Delta t & \mathbf{0} \\ -\tilde{\mathbf{r}}_{m,k}\Delta t & \mathbf{0} & \mathbf{0} & \mathbf{I} + \boldsymbol{\omega}_k^{\wedge}\Delta t \end{bmatrix}, \tag{9}$$

$$\mathbf{W}_k = \frac{\partial \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k)}{\partial \mathbf{w}_k} = \mathbf{I}. \tag{10}$$

The observation model is represented as a linear function

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x}_k. \tag{11}$$

The orientation is represented as the rotation matrix $\mathbf{R}_{bn}$ and extracted after every iteration step of the Kalman filter:

$$\mathbf{R}_{bn} = \begin{bmatrix} \mathbf{e}_b^{n_x} & \mathbf{e}_b^{n_y} & \mathbf{e}_b^{n_z} \end{bmatrix}. \tag{12}$$

The base vector in z-direction can be calculated using the vector of gravity:

$$\mathbf{e}_b^{n_z} = -\frac{\mathbf{r}_b^g}{\left\|\mathbf{r}_b^g\right\|}. \tag{13}$$

The base vector in y-direction is calculated according to the orthogonality:

$$\mathbf{e}_b^{n_y} = \frac{\mathbf{e}_b^{n_z} \times \mathbf{r}_b^g}{\left\|\mathbf{e}_b^{n_z} \times \mathbf{r}_b^g\right\|}. \tag{14}$$

Consequently, the x-direction base vector can be determined as:

$$\mathbf{e}_b^{n_x} = \frac{\mathbf{e}_b^{n_z} \times \mathbf{e}_b^{n_y}}{\left\|\mathbf{e}_b^{n_z} \times \mathbf{e}_b^{n_y}\right\|}. \tag{15}$$

The assumption for the presented attitude filter relies on the gravity vector to provide a valid angle estimation. This assumption is violated during aggressive flight. To alleviate this effect, we compensate for the acceleration measurements when a valid position sensor is available.

4.4 | Position estimation

The general position estimation is fused using a linear kalman filter formulation [22]. For each axis, the state vector includes position p, velocity v and acceleration bias b_a:

$$\mathbf{x} = \begin{bmatrix} p \\ v \\ b_a \end{bmatrix}, \mathbf{z} = \begin{bmatrix} \bar{p} \\ \bar{v} \end{bmatrix}. \tag{16}$$

The system input is linear accelerations measured by the onboard accelerometer. The system model is:

$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{b}_a \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \\ b_a \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} a + \begin{bmatrix} 0 \\ \mathrm{w}_a \\ \mathrm{w}_{b_a} \end{bmatrix}, \tag{17}$$

where $\mathrm{w}_{[\cdot]}$ represents the noise. The discrete-time system matrices are

$$\mathbf{A}_k = \begin{bmatrix} 1 & t_s & -\frac{t_s^2}{2} \\ 0 & 1 & -t_s \\ 0 & 0 & 1 \end{bmatrix}, \tag{18}$$

$$\mathbf{B}_k = \begin{bmatrix} \frac{t_s^2}{2} & t_s & 0 \end{bmatrix}, \tag{19}$$

in which t_s is the sampling time. The discrete-time covariance matrix for process model is

$$\mathbf{Q}_k = \begin{bmatrix} \left(\frac{t_s^3}{3}\sigma_a^2 + \frac{t_s^5}{20}\sigma_{b_a}^2\right) & (\frac{t_s^2}{2}\sigma_a^2 + \frac{t_s^4}{8}\sigma_{b_a}^2) & -\frac{t_s^3}{6}\sigma_{b_a}^2 \\ (\frac{t_s^2}{2}\sigma_a^2 + \frac{t_s^4}{8}\sigma_{b_a}^2) & (t_s\sigma_a^2 + \frac{t_s^3}{3}\sigma_{b_a}^2) & -\frac{t_s^2}{2}\sigma_{b_a}^2 \\ -\frac{t_s^3}{6}\sigma_{b_a}^2 & -\frac{t_s^2}{2}\sigma_{b_a}^2 & t_s\sigma_{b_a}^2 \end{bmatrix}, \tag{20}$$

where $\sigma_{[\cdot]}^2$ represents the power spectral density of noise. The observation can be position measurements from SLAM or GPS. Since the position measurement is usually delayed comparing with IMU data, we use a buffer to save all the states in a time window. We assume the delay time is constant and can be acquired by experiments. When new measurement arrives, it is first matched to the respective timestamp and then used to calculate state correction. The observation model is a trivial linear function

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k. \tag{21}$$

As each sensor data are not necessarily updated at the same frequency, the observation model can be easily separated to handle different sensors when data arrive. The discrete-time system covariance is approximated according to sampling time t_s or frequency f_s:

$$R_k \approx \frac{R(t)}{t_s} = R(t) \cdot f_s. \tag{22}$$

Altitude estimation model is an extension on the z-axis position filter with one more barometer altitude bias state b_{baro}. We denote the basic position state vector with altitude h, velocity v_h, acceleration bias b_a. The measurement state includes altitude from barometer h_{baro}, and height from distance sensor $h_{dist} - h_{sur}$:

$$\tilde{\mathbf{x}} = \begin{bmatrix} h \\ v_h \\ b_a \\ b_{baro} \end{bmatrix}, \quad \tilde{\mathbf{z}} = \begin{bmatrix} h_{baro} \\ h_{dist} - h_{sur} \end{bmatrix}. \tag{23}$$

The system input is z-direction acceleration, and the discrete-time process model is augmented from $\mathbf{A}_k$ in position filter formulation:

$$\tilde{\mathbf{A}}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{24}$$

$$\tilde{\mathbf{B}}_k = \begin{bmatrix} \frac{t_s^2}{2} & t_s & 0 & 0 \end{bmatrix}. \tag{25}$$

The discrete-time covariance matrix for process model is

$$\tilde{\mathbf{Q}}_k = \begin{bmatrix} \mathbf{Q}_k & 0 \\ \mathbf{0} & t_s\sigma_{b_{baro}}^2 \end{bmatrix}. \tag{26}$$

The observation model can be derived as

$$\tilde{\mathbf{z}}_k = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tilde{\mathbf{x}}_k. \tag{27}$$

4.5 | Indoor-outdoor transition

For the indoor-outdoor transition problem, we adopt the stochastic cloning method [23]. The idea is to add past states in the filter, and update the filter with relative measurements from SLAM. The advantage of doing this is to maintain a correct covariance estimation and output a consistent position. When the drone operate in both indoor and outdoor environments, we treat GPS as global measurement and SLAM output as relative measurement. The state vector includes past position p_l, position p, velocity v, acceleration bias b_a:

$$\check{\mathbf{x}} = \begin{bmatrix} p_l \\ p \\ v \\ b_a \end{bmatrix}, \check{\mathbf{z}} = \begin{bmatrix} \Delta p \\ p \\ v \end{bmatrix}. \tag{28}$$

The process model is augmented respectively:

$$\check{\mathbf{A}}_k = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{A}_k \end{bmatrix}, \tag{29}$$

$$\check{\mathbf{B}}_k = \begin{bmatrix} 0 & \frac{t_s^2}{2} & t_s & 0 \end{bmatrix}. \tag{30}$$

The discrete time covariance matrix for process noise is

$$\check{\mathbf{Q}}_k = \begin{bmatrix} 0 & \mathbf{0} \\ 0 & \mathbf{Q}_k \end{bmatrix}. \tag{31}$$

By applying stochastic cloning technique on the filter, the covariance matrix need to be augmented with the position variance P_{p_l} when past state p_l is initialized:

$$\check{\mathbf{P}}_k = \begin{bmatrix} P_{p_l} & P_{p_l} & 0 & 0 \\ P_{p_l} & & & \\ 0 & & \mathbf{P}_k & \\ 0 & & & \end{bmatrix} \tag{32}$$

The respective observation model is

$$\check{\mathbf{z}}_k = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \check{\mathbf{x}}_k. \tag{33}$$

5 | PLANNING AND CONTROL

5.1 | Task planning

We use a tree-based framework [24] to manage multiple tasks of different types. The tasks are organized into a tree structure and executed in a manner of depth-first traversal. Figure 7 shows a typical tree structure. Each leaf node task contains only one single action. In other words, executing that leaf node task is equivalent to executing the corresponding action. Some flight action examples are ‘take-off’, ‘fly to a global coordinate’, ‘fly to a local coordinate’, ‘drop a payload’, ‘land’.

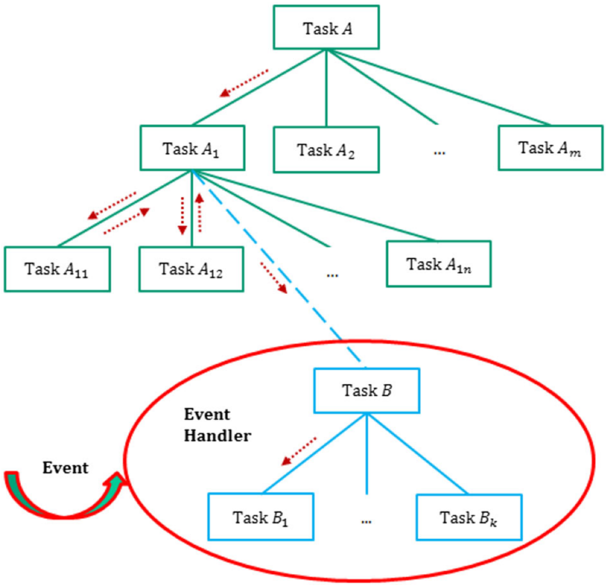


FIGURE 7 The task planning structure [Color figure can be viewed at wileyonlinelibrary.com]

The task planning is further enhanced with a reactive planning scheme triggered by events. An event is an occurrence of the asynchronous external activity that requires an action to take. As a result, the current action is terminated and a special action is inserted and executed to handle the event. For example, when a visual target is detected during a target searching mission, a ‘target found’ event will be triggered and a ‘navigate to found target’ action takes over. If the event is not triggered, the default action will be completed.

5.2 | Path planning

The MAV adopts a two-layered planning structure [17] to achieve online path planning and obstacle avoidance. First of all, a search-based global planner, specifically, A* is used to find the geometric collision-free path on the updated map without considering any vehicle dynamics. Then, the dynamically-feasible trajectory is generated along the collision-free path by solving a series of two-point boundary value problems. Figure 8 illustrates the procedure of our path planning. The local target will be switched if the trajectory of current path enters the deceleration phase.

5.3 | Cascaded control

A cascaded control structure [25] is applied for precise and robust maneuvering. The controller structure is displayed in Figure 9. Outer loop controller, which adopts robust perfect tracking (RPT) algorithm, transfers the user command into acceleration reference. Whereas the LQR based inner loop controller takes in acceleration reference and generates motor control signals.

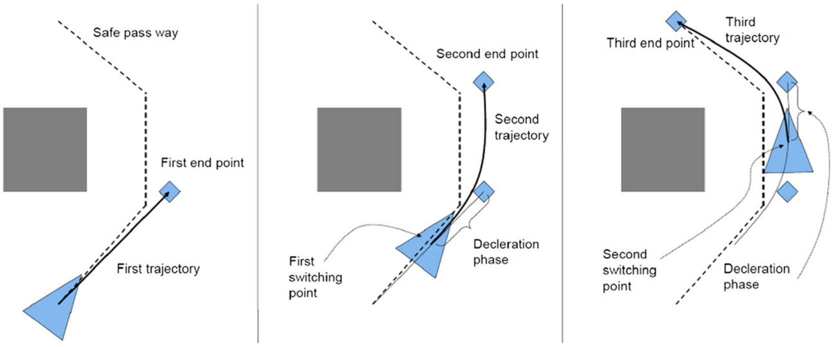


FIGURE 8 The path planning structure [Color figure can be viewed at wileyonlinelibrary.com]

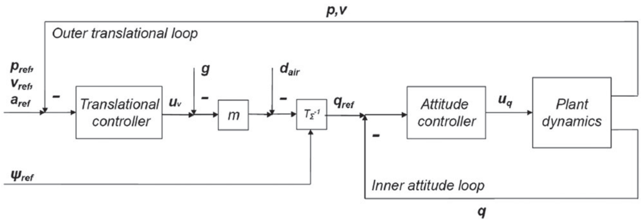


FIGURE 9 The flight controller structure

TABLE 1 Comparison of SLAM methods

SLAM methods	Accuracy (ATE)	CPU usage
Cartographer	0.059m	60%
Hector	0.071m	40%

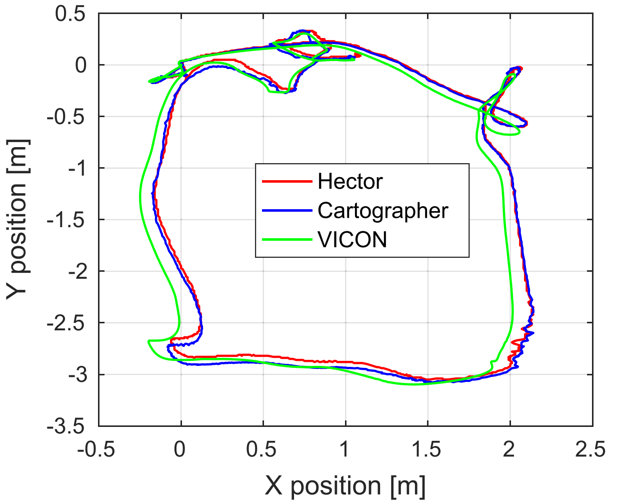


FIGURE 10 The comparison of SLAM accuracy in VICON room [Color figure can be viewed at wileyonlinelibrary.com]

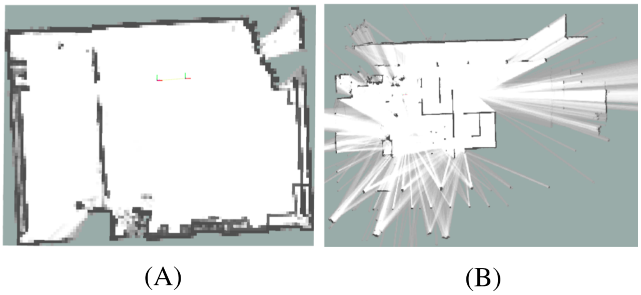


FIGURE 11 Occupancy maps built with Cartographer SLAM. (A) shows map of Vicon room (around 8x6m, 0.1m resolution), (B) shows map of a larger test field (around 20x15m, 0.1m resolution) [Color figure can be viewed at wileyonlinelibrary.com]

6 | EXPERIMENTAL RESULTS

6.1 | Onboard perception

6.1.1 | SLAM comparison

We carefully evaluate two SLAM methods, which are Cartographer [8] and Hector [7]. Figure 10 shows the position outputs from both methods. Also, ground truth data from VICON motion capture system is plotted for comparison. The result indicates that both methods can provide competitive accuracy compared with ground truth. It is notable that the result of Cartographer is closer to ground truth in the area near the starting point. This should be the contribution of loop closure, which is not implemented in Hector.

We also examine the real-time performance of each method by checking the approximate CPU usage of the onboard computer. The quantitative results for accuracy and computation efficiency are shown in Table 1.

The average translation error (ATE) of Hector SLAM is 0.071m, whereas ATE of Cartographer SLAM is 0.059m. The CPU usage of Hector SLAM is around 40% of one CPU core while that of Cartographer SLAM is 60%. In summary, Cartographer SLAM maintains comparable efficiency and outperforms Hector SLAM in terms of accuracy. Therefore, we deploy Cartographer SLAM onboard for a variety of applications. Examples of built occupancy grid map by Cartographer SLAM are shown in Figure 11.

6.1.2 | Visual detection

The color target detection can handle partial occlusion as long as the target occupies a large enough region in the image. The detection is further confined with a mission-level searching area to improve the robustness. The Apriltag detection is robustness to scale, rotation

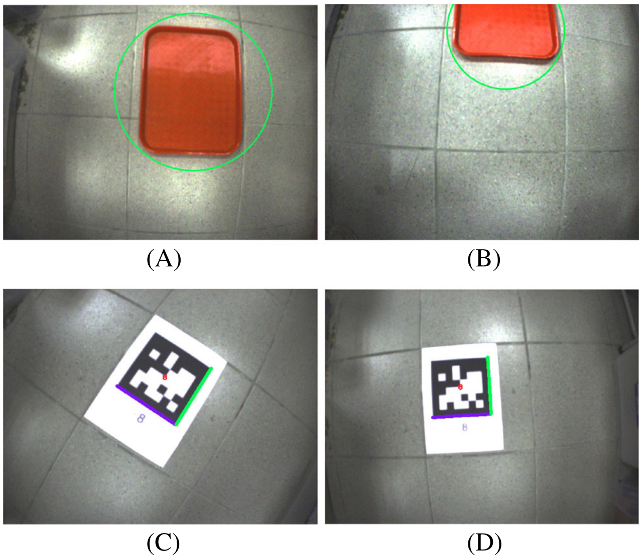


FIGURE 12 Visual target detection. The first row (A,B) shows the detection of red target, green line shows the bounding circle of target. The second row (C,D) shows the detection of Apriltag target, red dot shows the center and green and blue lines show the bounding box [Color figure can be viewed at wileyonlinelibrary.com]

and light change. The number of false positives is negligible to guarantee the stability of guidance. We use an A4-size Apriltag as our landing pad marker. To reduce the computational consumption, an image resolution of 320x240 is applied and the detection rate is up to 10Hz. The detection is valid between 0.4m and 3m with this setup. Figure 12 demonstrates the effectiveness of our detection algorithms.

6.2 | State estimation

6.2.1 | Indoor flight

We fuse sensor data from IMU, barometer and distance sensor to estimate flight altitude. This configuration introduces sensor redundancy, which can handle the surface change problem and give robust altitude estimation. The raw measurements from barometer are depicted in Figure 13. The measurements are captured by flying the MAV with changing altitudes. It can be observed that there is a static bias on barometer altitude. This indicates that we can trust barometer in the long-term by considering the bias. On the other hand, short-term measurement is problematic in some cases. Since the air pressure is susceptible to air disturbance, we can observe significant large errors during the altitude changing. This is because of the changing air distance when the motor speed is changing. Also, if the MAV flies near surfaces, the air disturbance would change due to ground effect. Based on this observation, we do not trust the barometer altitude in a very short time when the distance sensor is regarded as failures.

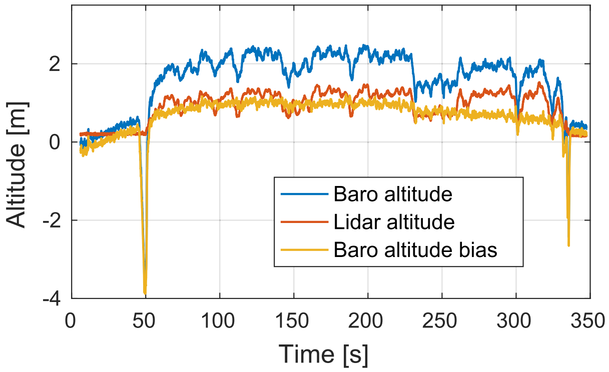


FIGURE 13 Different altitude measurments [Color figure can be viewed at wileyonlinelibrary.com]

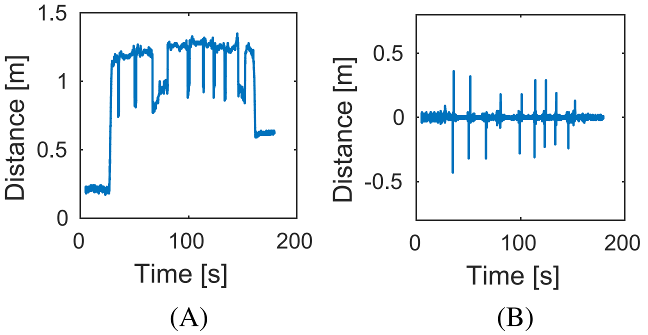


FIGURE 14 Lidar distance measurements with surface changing [Color figure can be viewed at wileyonlinelibrary.com]

Figure 14A shows raw measurements from the distance sensor under the circumstance of different surfaces. Obvious data jumps can be seen when the surface changes. Moreover, there are some unexpected errors which give extreme large values. The results suggest that surface difference must be detected and errors must be rejected. Intuitively, we can decide the sudden jump by differentiating the distance value. However, it is hard to detect it with a fixed threshold due to the progressive variance of distance measurements. Figure 14B shows the differential result on lidar distance to illustrate this problem.

Considering all the discussed cases, the overall altitude sensor fusion strategy is shown in Figure 15. The system continuously monitors the health status of each sensor. If none of the sensors fails, we use all the measurements. Otherwise, we detect sensor failure and reject unexpected measurements. When the surface is changing, the distance sensor is not trustworthy, and the barometer in a short time is suspicious with air disturbance. The IMU info is passed through directly to predict the altitude. When the distance sensor gives stable output, we can subtract it with the current altitude to determine the surface height. On the other hand, if the distance sensor fails for a long time, the barometer will help to bound the IMU drift.

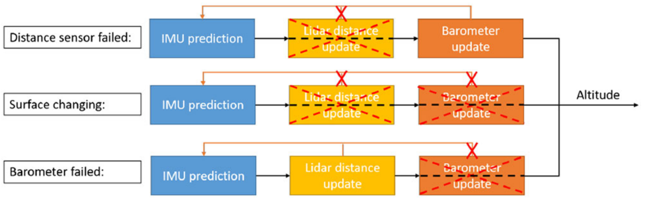


FIGURE 15 Altitude fusion strategy [Color figure can be viewed at wileyonlinelibrary.com]

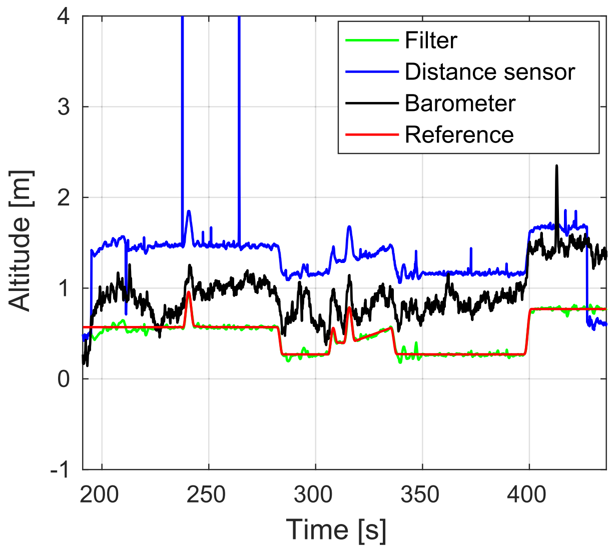


FIGURE 16 Altitude estimation performance [Color figure can be viewed at wileyonlinelibrary.com]

By complementing different characteristics of the distance sensor and barometer, our method provides a robust altitude estimation, which is shown in Figure 16. In our implementation, variance of acceleration σ_a^2, variance of acceleration bias $\sigma_{b_a}^2$, variance of baro bias $\sigma_{b_{baro}}^2$, variance of barometer altitude $\sigma_{h_{baro}}^2$, variance of distance sensor altitude $\sigma_{h_{dist}}^2$ are as follows:

$$\sigma_a^2 = 0.09, \quad \sigma_{b_a}^2 = 0.01,$$
$$\sigma_{b_{baro}}^2 = 1, \quad \sigma_{h_{baro}}^2 = 0.09,$$
$$\sigma_{h_{dist}}^2 = 0.01.$$

The error between distance sensor altitude and current altitude estimation $e_{h_{dist}}$ is utilized to detect suspicious and surface changing:

$$h_{dist} = \begin{cases} suspicious, & if \quad e_{h_{dist}} > \sigma_{h_{dist}}. \\ changing, & if \quad e_{h_{dist}} > 3\sigma_{h_{dist}} \quad in \quad 0.2s. \end{cases}$$

As can be seen from the result in Figure 16, surface differences are successfully detected all the time, and barometer bias is correctly estimated. As a result, the altitude output is very stable and matched with the reference.

The whole estimated states include 3D position, velocity and attitude. In addition, the system is designed to be robust by utilizing sensor redundancy. Figure 17 shows the complete 6D state estimation result with laser SLAM and robust altitude estimation. Since laser SLAM is accurate, the filtered position follows SLAM result very well. In the zoom-in view, It can be seen that the filtered position is smoother than SLAM data, which is good for control purpose. Moreover, smooth velocity estimation on each axis is also presented to demonstrate the filter performance.

6.2.2 | Indoor-outdoor transition

To demonstrate the filter performance for indoor-outdoor transition flight, we fly the MAV in an outdoor basketball court where both GPS and laser SLAM is available. GPS

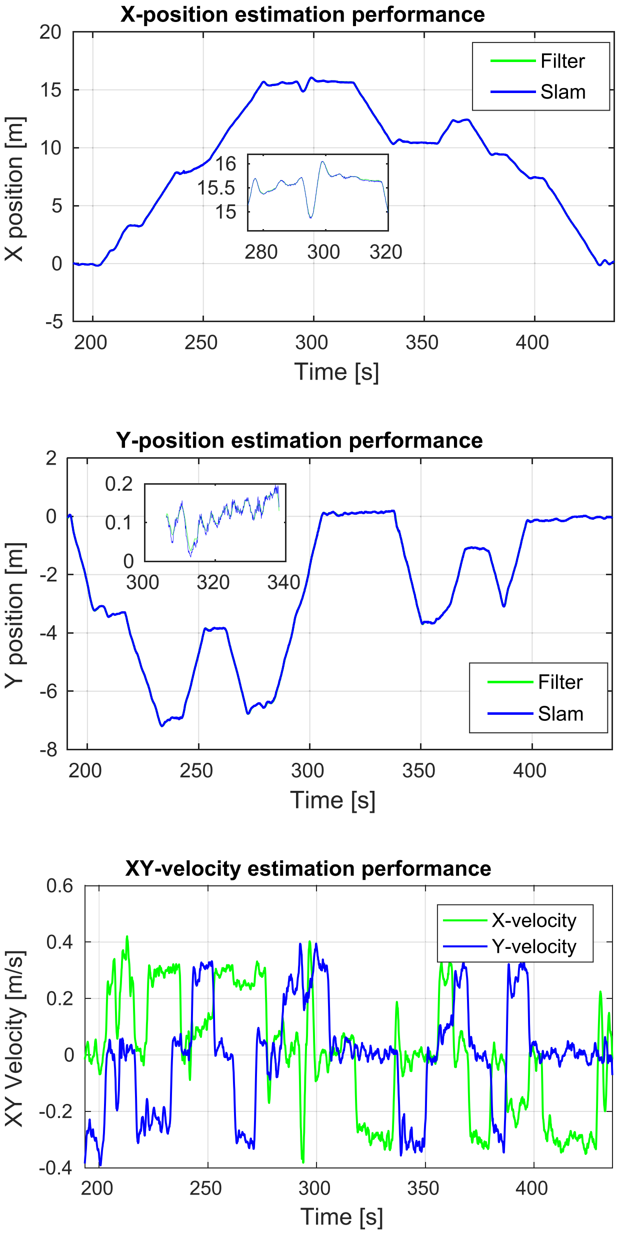


FIGURE 17 Kalman filter based state estimation result [Color figure can be viewed at wileyonlinelibrary.com]

can be disabled or enabled at any time using a switch from a transmitter. GPS is updated at 5Hz, and laser SLAM is updated at 20Hz. We use parameters of $\sigma^2_{p_{gps}} = 4$, $\sigma^2_{p_{slam}} = 0.25$. The state estimation result for indoor-outdoor transition is shown in Figure 18. Laser SLAM result is regarded as a reference due to its high accuracy. The data is separated into three parts in the figure. The first part uses GPS as a position source, the filter follows GPS measurements but smoother by compensating GPS errors with IMU. The second part disables GPS and switches to laser SLAM, the filter follows the relative measurements from SLAM with high confidence. The third part disables laser SLAM and resumes to use GPS. We can see that the filter track GPS output as expected. The result demonstrates the effectiveness of our filter by switching between global measurements (GPS) and relative measurements (SLAM). The position estimation variance on one axis validates a reasonable change with stochastic cloning method. When SLAM is used as relative measurements, variance increases based on previous converged variance. When GPS is used again, variance decreases since it provides global measurements.

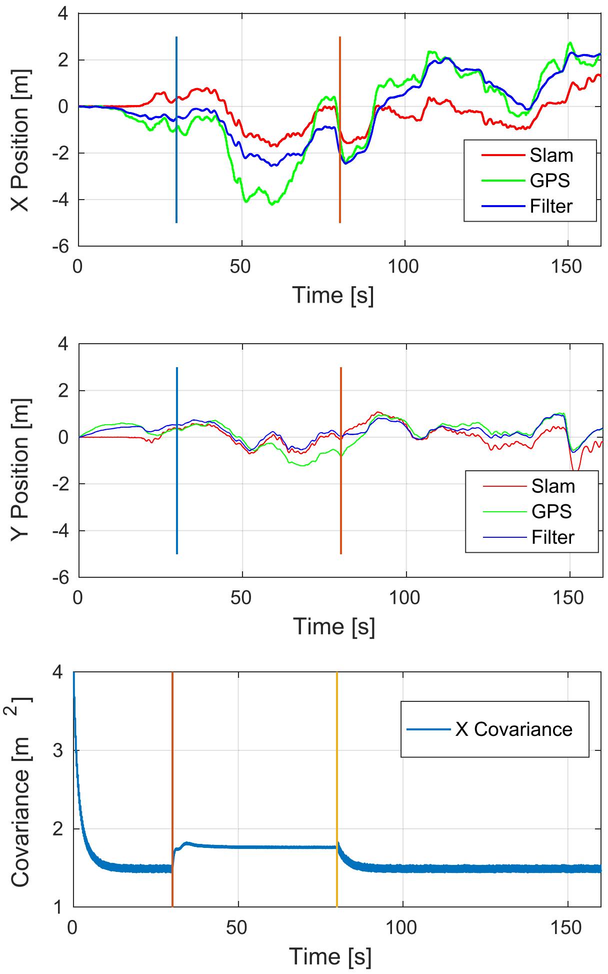


FIGURE 18 State estimation result for indoor-outdoor transition (GPS-SLAM-GPS) [Color figure can be viewed at wileyonlinelibrary.com]

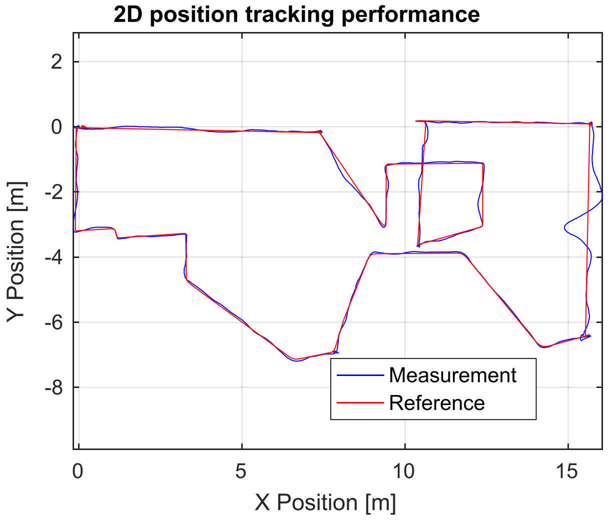


FIGURE 19 The red curve shows the reference, and the blue one gives the measurement [Color figure can be viewed at wileyonlinelibrary.com]

6.3 | Flight performance

Our system has been exhaustively tested in different indoor environments. As a case study, we demonstrate the proposed system in the 2017 International Micro Air Vehicles Competition (IMAV 2017) held in Toulouse, France, and won the championship*. The competition simulates a rescue mission in a damaged building. The mission elements include takeoff and landing on a moving platform, navigating through obstacles, searching and identifying casualties inside, also resisting wind disturbance.

The flight performance evaluation with 3D reference and measurements is depicted in Figure 19. The tracking error is relatively small given the trajectory that is generated and always changing in real-time. It can be observed that our system handles multiple types of tasks and maneuvers very well in one flight. The tracking accuracy is very impressive with an average tracking error of 0.095m. A short period of obvious large tracking error can be observed from the result. This is corresponding to the flight area simulating wind disturbance with industry fans. The MAV can recover to the reference trajectory with a bounded drifting error.

7 | CONCLUSIONS

In this paper, we have presented our autonomous MAV solution for completing search and rescue in GPS-denied

*http://www.imavs.org/2017/

environments. Comprehensive hierarchical structures for both hardware and software make our system robust enough to accomplish versatile tasks. We have developed the perception, planning and state estimator with constrained onboard computational power. The system performance is validated in a variety of real flights. Currently, we are working on a vision-based navigation solution for our system. In the near future, We will expand the autonomous capabilities of the system to general 3D environments by incorporating vision-based navigation.

ORCID

Yingcai Bi https://orcid.org/0000-0002-5956-1326

REFERENCES

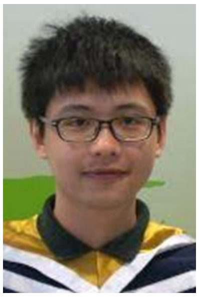
1. A. Bachrach et al., *RANGE–robust autonomous navigation in GPS-denied environments*, J. Field Rob. **28** (2011), no. 5, 644–666.
2. S. Grzonka, G. Grisetti, and W. Burgard, *A fully autonomous indoor quadrotor*, IEEE Trans. Robot. **28** (2012), no. 1, 90–100.
3. D. Scaramuzza et al., *Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments*, IEEE Robot. Autom. Mag. **21** (2014), no. 3, 26–40.
4. M. Burri et al., *Real-time visual-inertial mapping, re-localization and planning onboard MAVs in unknown environments*, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 1872–1878.
5. D. Droeschel et al., *Multilayered mapping and navigation for autonomous micro aerial vehicles*, J. Field Rob. **33** (2016), no. 4, 451–475.
6. Z. Fang et al., *Robust autonomous flight in constrained and visually degraded shipboard environments*, J. Field Rob. **34** (2017), no. 1, 25–52.
7. S. Kohlbrecher et al., *A flexible and scalable SLAM system with full 3D motion estimation*, 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Kyoto, Japan, 2011, pp. 155–160.
8. W. Hess et al., *Real-time loop closure in 2D LIDAR SLAM*, 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271–1278.
9. S. Shen, N. Michael, and V. Kumar, *Autonomous multi-floor indoor navigation with a computationally constrained MAV*, 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011, pp. 20–25.
10. R. Mur-Artal and J. D. Tardós, *ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras*, IEEE Trans. Robot. **33** (2017), no. 5, 1255–1262.
11. J. Engel, V. Koltun, and D. Cremers, *Direct sparse odometry*, IEEE Trans. Pattern Anal. Mach. Intell. **40** (2018), no. 3, 611–625.
12. A. Shaghaghian and P. Karimaghaee, *Improving GPS/INS integration using FIKF-filtered innovation Kalman filter*, Asian J. Control **21** (2018), no. 2, 1–10.
13. V. Indelman et al., *Information fusion in navigation systems via factor graph based incremental smoothing*, Robot. Auton. Syst. **61** (2013), no. 8, 721–738.
14. S. Lynen et al., *A robust and modular multi-sensor fusion approach applied to MAV navigation*, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013, pp. 3923–3929.
15. A. Chambers et al., *Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments*, American Control Conference (ACC), 2014, Portland, OR, USA, 2014, pp. 1892–1899.
16. J. Chen, T. Liu, and S. Shen, *Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments*, 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1476–1483.
17. S. Lai et al., *A robust online path planning approach in cluttered environments for micro rotorcraft drones*, Control Theory Technol. **14** (2016), no. 1, 83–96.
18. Y. Bi et al., *Robust autonomous flight and mission management for mavs in gps-denied environments*, 2017 11th Asian Control Conference (ASCC), Gold Coast, Australia, 2017, pp. 67–72.
19. E. Olson, *AprilTag: A robust and flexible visual fiducial system*, 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011, pp. 3400–3407.
20. M. Tanigawa et al., *Drift-free dynamic height sensor using MEMS IMU aided by MEMS pressure sensor*, 2008 5th Workshop on Positioning, Navigation and Communication, Hannover, Germany, 2008, pp. 191–196.
21. L. Meier et al., *Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision*, Auton. Robot. **33** (2012), no. 1-2, 21–39.
22. F. L. Lewis, L. Xie, and D. Popa, *Optimal and robust estimation: With an introduction to stochastic control theory*, CRC press, 2007.
23. A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick, *SC-KF mobile robot localization: A stochastic cloning kalman filter for processing relative-state measurements*, IEEE Trans. Rob. **23** (2007), no. 4, 717–730.
24. M. Molina et al., *Specifying complex missions for aerial robotics in dynamic environments*, International Micro Air Vehicle Competition and Conference, Beijing, China, (2016), 270–277.
25. K. Wang, Y. Ke, and B. M. Chen, *Autonomous reconfigurable hybrid tail-sitter UAV U-Lion*, Sci. China Info. Sci. **60** (2017), no. 3, 33201.

AUTHOR BIOGRAPHIES

Yingcai Bi received his Ph.D degree from Graduate School for Integrative Sciences and Engineering, National University of Singapore (NUS) in 2018. He has a M.Eng degree in Control Science and Engineering from Beihang University (BUAA) and a B.Eng degree in Automation Engineering from Nanjing University of Aeronautics and Astronautics (NUAA). His research interests lie on localization, mapping and autonomous navigation for unmanned systems.

Menglu Lan received her B.Eng (1st) degree in Electrical and Computer Engineering from National University of Singapore in 2015. She is currently pursuing her PhD degree in NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, with research interest in task and motion planning for micro sized aerial vehicles (MAVs).

Jiaxin Li is currently a Research Scientist at nuTonomy Singapore (an Aptiv company). He was a Postdoc Researcher at National University of Singapore (NUS) in 2018. He received his B.Eng degree from Mechanical Engineering of Tsinghua University in 2014, and Ph.D degree from the School of Integrated Science and Technology of NUS in 2018. His research interests include computer vision, point cloud recognition, SLAM, self-driving car and unmanned aerial vehicle.



Shupeng Lai received his B.Eng. degree from the Department of Electronic and Electrical Engineering, Nanyang Technological University in 2012 and his Ph.D from the school of integrated science and technology from the National University of Singapore. He is currently a research fellow at the National University of Singapore. His research interests lie in motion planning, game theory and navigation of multiple unmanned systems. E-mail: elelais@nus.edu.sg.

Ben M. Chen is currently a Professor in the Department of Mechanical and Automation Engineering at the Chinese University of Hong Kong. He was a Provost's Chair Professor in the Department of Electrical and Computer Engineering, the National University of Singapore (NUS), where he was also serving as the Director of Control, Intelligent Systems and Robotics Area, and Head of Control Science Group, NUS Temasek Laboratories. His current research interests are in unmanned systems, robust control and control applications.

Dr. Chen has published more than 400 journal and conference articles, and a dozen research monographs in control theory and applications, unmanned systems and financial market modeling. He had served on the editorial boards of several international journals including IEEE Transactions on Automatic Control and Automatica. He currently serves as an Editor-in-Chief of Unmanned Systems. Dr. Chen has received a number of research awards nationally and internationally. His research team has actively participated in international UAV competitions, and won many championships in the contests. He is an IEEE Fellow.

How to cite this article: Bi Y, Lan M, Li J, Lai S, Chen BM. A lightweight autonomous MAV for indoor search and rescue. *Asian J Control*. 2019;21:1732–1744. https://doi.org/10.1002/asjc.2162