2014 11th IEEE International
Conference on Control & Automation (ICCA)
June 18-20, 2014. Taichung, Taiwan

An Efficient UAV Navigation Solution for Confined but Partially Known Indoor Environments

Fei Wang, Kangli Wang, Shupeng Lai, Swee King Phang, Ben M. Chen, Tong H. Lee

Abstract— **This paper presents a robust and efficient navigation solution for a quadrotor UAV to perform autonomous flight in a confined but partially known indoor environment. The main sensors used onboard of the UAV are two scanning laser range finders and an inertial measurement unit. When the indoor environment is structured and the coordinates of its key corner features are known, the UAV planer position can be efficiently calculated via the measurements from the first horizontally scanning laser range finder. The height of the UAV with respect to the ground can be robustly estimated by the second laser scanner which is mounted orthogonally to the first. Besides, this work also adopts a robust and perfect tracking control method with integral action to enable the UAV to track any smooth 3-D trajectories responsively and precisely. All computation is done onboard by an ARM-based embedded computer with limited processing power. The whole system was successfully tested in the 2013 Singapore Amazing Flying Machine Competition and helped the Unmanned Aircraft Systems Group from the National University of Singapore win the overall championship in the fully autonomous category.**

Keywords: UAV indoor navigation, lidar localization

I. INTRODUCTION

The research on the topic of UAV indoor navigation has been progressing fast in the last decade. There are two main challenges, namely the denied reception of GPS signal and the constraints of the indoor UAV platforms. Unlike the conventional GPS/INS based navigation in which the UAV global position and velocity can be directly obtained, indoor navigation needs to get these information by sophisticated algorithms based on relative sensing. It should be also noted that even if the GPS signal is available, its accuracy is not good enough for navigation in such confined space. To make things worse, indoor UAVs are usually designed to be small and having very limited payload. This results in limited onboard computational power which makes the aforementioned algorithms even harder to be implemented.

In formal terms, the method used by robots or autonomous vehicles to build up a map within an unknown environment, or to update a map within a known environment, while at the same time keeping track of their current location is called the simultaneous localization and mapping (SLAM) technique. Many theoretical works and practical implementations of SLAM on ground robots [1] [2], and on UAV platforms [3] [4] have been published in literature. However, few of these works have considered the computation limitation on miniature indoor UAVs and they usually exploit the unlimited payload on ground robots or rely on high-bandwidth communication to the ground control station (GCS) where a powerful computer is running the most computationally intensive algorithm. In consequence, some of them only work in controlled lab environments with short and line-of-sight communication. But for real-life applications in which ideal communications cannot be guaranteed, the performance is expected to be poor.

That being said, a more practical and robust navigation strategy should only rely on the UAV onboard computers for all necessary control and navigation functions. A few research groups are working towards this direction. In [5], an innovative laser-pointer-aided vision system is proposed to release the high computational load from dense image processing. [6] has demonstrated the possibility of real-time visual-inertial state estimation via a 1.6 GHz Atom computer onboard of the controlled UAV. In [7], hardware configuration has been optimized to achieve a highly efficient vision navigation system. The impressive work in [8] has pushed UAV onboard intelligence to the limit where a rather complicated indoor environment can be handled. Nevertheless, there must be a compromise between the complexity of the navigation algorithm and the complexity of the navigated environment under the current microprocessor technology.

In this work, we intend to solve the indoor navigation problem solely onboard of a miniature UAV flying in a structured indoor environment. The algorithm can be designed very efficient because three assumptions about the indoor environment are made:

1) The environment can be described by sparse features, which include corners and straight lines;
2) The line features are orthogonal to each other or off-set by multiples of a constant angle displacement, such as 30° or 45°.
3) The coordinates of the corner features are known.

These assumptions appear to be strong but they still cover quite a lot of real-life conditions. First of all, Assumption 1 and 2 are usually met for indoor environments in modern man-made buildings. Moreover, the proposed algorithm will work as long as the majority of corner and line features in the target environment fulfills the assumptions. A few map

F. Wang is with the Temasek Laboratories NUS, Singapore. E-mail: tslwf@nus.edu.sg

K. Wang, B. M. Chen and T. H. Lee are with the Department of Electrical & Computer Engineering, NUS, Singapore. E-mail: {a0055862,bmchen,eleleeth}@nus.edu.sg

S. Lai and S. K. Phang are with the NUS Graduate School for Integrative Sciences & Engineering, NUS, Singapore. E-mail: {shupenglai,king@nus.edu.sg}

noises will not affect the performance too much. Although Assumption 3 makes the algorithm not suitable for advanced tasks such as exploring a completely unknown environment, missions like UAV autonomous surveillance and patrolling are still doable if minimal information about the indoor environment is known. Nevertheless, the main advantage of the proposed method lies in its efficiency. With the three assumptions met, the UAV localization algorithm can be designed in an innovative way so that an ARM-based embedded computer is more than enough to handle the computation.

Furthermore, to fly in a confined indoor space, the position of the UAV needs to be controlled extremely well so that windows, door ways or narrow corridors can be passed through safely. This requires a robust, responsive and zero steady-state error performance from the controlled platform, which leads to the application of robust and perfect tracking (RPT) method with integral action in controlling the outer loop of the UAV.

The content of this paper is organized as follows: Section II briefly talks about the hardware platform and the overall system structure. Section III contains the main contribution of this paper, where an efficient and robust localization algorithm based on measurements from two laser scanners is presented. Section IV explains how to apply the RPT control method to the outer loop of the UAV dynamic system so that the position control performance meets the stringent requirements from a confined environment. Section V will provide experimental results and flight test results to justify the performance of the overall system. Lastly, concluding remarks will be made in Section VI.

II. Hardware Platform and System Overview

Being mechanically simple and robust, quadrotor helicopters have been widely used as UAV platforms for research purposes these days. The platform used for the work described in this paper is also a quadrotor. Its dimensions are 35 cm in height and 86 cm in diagonal width. Its attitude angles are stabilized by an off-the-shelf control board called ‘Naza-M’ from DJI. This custom-made quadrotor has a maximum take-off weight of 2.9 kg and can fly in a near-hover condition for about 10 minutes.

For the onboard avionics, the IG-500N inertial measurement unit (IMU) from SBG Systems, is used to provide the UAV linear accelerations, angular rates and Euler angles. A URG-30LX scanning laser range finder from Hokuyo is used to measure the distance from the surrounding objects in a frontal 270° and 30 m’s range. Another URG-04LX 4 m scanning laser range finder is mounted orthogonally to the first and its readings can be used to calculate the altitude of the UAV with respect to the indoor floor. Only one onboard processor is used, namely the Gumstix Overo Fire. It is a 25-gram low-voltage ARM-based embedded computer. All algorithms, including both control law implementation and localization, are done on this tiny computer with computational power of merely 720 MHz. Fig. 1 shows the quadrotor platform with the three main sensors installed.

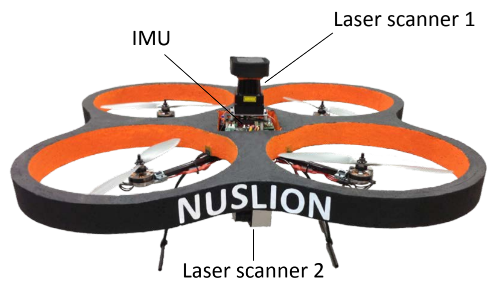


Fig. 1: The custom-made quadrotor platform

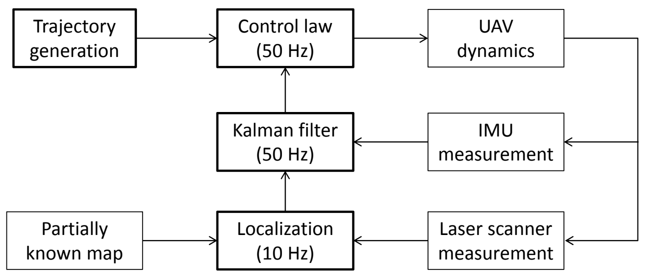


Fig. 2: Overall structure of the indoor navigation system

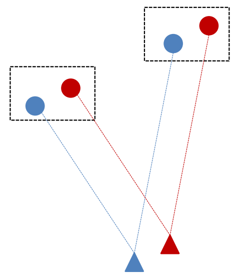
Fig. 2 shows the top-level structure of the overall control and navigation structure. As the UAV moves, a localization algorithm runs based on the measurements from two laser scanners to obtain the UAV 3-D position and heading in 10 Hz. Together with the acceleration and attitude angle information provided by the IMU sensor, a Kalman filter is designed to provide a smoother and higher frequency (50 Hz) pose estimation for the UAV so that the ‘Control law’ block has its needed measurements in high quality.

III. Localization Algorithm

The UAV pose in the map frame can be represented by its 3-D coordinates x, y, z and heading angle ψ. To differentiate the localization results from their respective sensor sources, we divide the UAV pose into two parts, namely the planar pose (x, y, ψ), and the vertical height z. The first part can be estimated by the horizontal scanning laser range finder, similar to a 2D ground robot case, while the altitude of the UAV can be estimated by the second laser scanner.

A. Planar localization algorithm

The planar localization algorithm via the first laser scanner contains the fundamental ideas that make the whole navigation solution fast and efficient. With Assumption 1, the conventional point cloud matching algorithm can be avoided, leaving the number of point matching pairs single digits as compared to the original thousands. With Assumption 2, the estimation of rotational motion can be done by comparing the difference between line gradients instead of relying on point feature matching, thus making the estimation of rotational motion decoupled from translational motion. This decoupling feature is very beneficial because rotational motion usually


(a) Translation

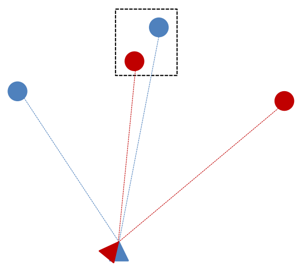

(b) Rotation

Fig. 3: Feature matching result after a small motion

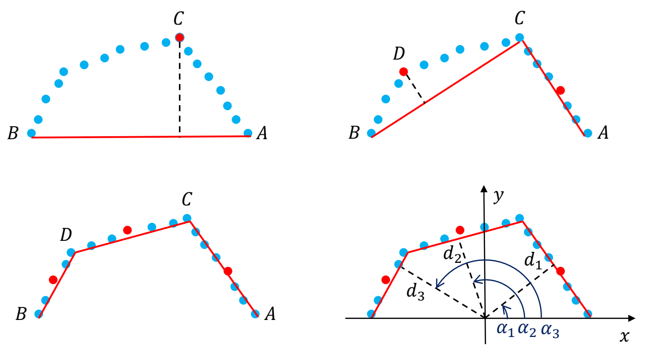


Fig. 4: The *split-and-merge* and line extraction algorithm

results in inconsistent point matching results, especially when the feature points are far away from the sensor source. From Fig. 3, one can see that the point matching result is correct in the first case which involves a small translation, but becomes totally wrong in the second case which involves a small rotation. As the method used in this paper estimates the rotational motion robustly and independently from the translational motion, the next stage point association and localization will have very stable performance.

The planar localization algorithm will be explained in four steps, which include feature extraction, rotation tracking, corner feature association and position tracking.

1) Feature extraction: The laser scanner used for this planar localization algorithm is a Hokuyo UTM-30LX sensor. For each frame of scanned data, the sensor will output 1081 integer numbers to represent the measured distances in millimeter from the rightmost angle to the leftmost angle sequentially. Each distance data is associated with its own angle direction. A simple transformation can be applied to the raw measurement data to convert it from polar coordinates (r_k, θ_k) to Cartesian coordinates (x_k, y_k):

$$\left\{ \begin{array}{lcl} x_k & = & r_k \cos\theta_k \\ y_k & = & r_k \sin\theta_k \end{array} \right. . \tag{1}$$

Then the *split-and-merge* algorithm [9] is applied to these array of 2D points so that they can be grouped into clusters with each cluster belonging to a straight line feature. Here, the main steps of *split-and-merge* algorithm is summarized below with Fig. 4 giving a graphical illustration:

1) Connect the first point A and the last point B of the input data by a straight line.
2) Find point C among all data points that has the longest perpendicular distance to the line AB.
3) If this longest distance is within a threshold, then a cluster is created with points in between A and B.
4) Else, the input points will be split into two subgroups, A-C and C-B. For each group, the *split-and-merge* algorithm will be applied recursively.

After obtaining the clusters of points, two choices of line extraction methods can be used. The first is to use least square line fitting by considering all points in the cluster, while the second is to simply connect the first point and the the last point. Although the second method looks a bit harsh, these two methods surprisingly result in more or less the same quality of line features in a clean and structured indoor environment, thanks to the laser scanner's superior range accuracy and angular resolution. The second method actually triumphs in computational time and it is finally chosen as the way to get the line features. By convention, each line can be represented by two parameters, namely the line's normal direction α_k and its perpendicular distance to the center of laser scanner d_k. In the last sub-figure of Fig. 4, xy axes represent the laser scanner frame.

2) Rotation tracking: In this step, Assumption 2 will be utilized in an innovative way to keep track of the robot's heading direction ψ. Without loss of generality, let the map frame x-axis align with one of the walls in the indoor environment. Then all the walls will have their directions at $n\alpha$, where α is the constant angle displacement and n can be any integers. Choose one of the walls currently observable and let its direction be β_l in the laser scanner frame. Then we have this wall's map frame direction β_m as:

$$\begin{aligned} \beta_m &= \psi_t + \beta_l \\ &= \psi_{t-1} + \Delta\psi_t + \beta_l \\ &= n_i \alpha. \end{aligned}$$

where ψ_t and ψ_{t-1} are the UAV headings in the current frame and previous frame respectively and $\Delta\psi_t$ is the inter-frame heading movement. Obviously, $(\psi_{t-1} + \Delta\psi_t + \beta_l)$ is divisible by α, which leads to

$$\Delta\psi_t = -[(\psi_{m,t-1} + \beta_l)\,\%\,\alpha], \tag{2}$$

where the operator % is defined in this paper as:

$$a\%b = \begin{cases} (a \mod b) & , \text{ if } (a \mod b) \le b/2 \\ (a \mod b) - b & , \text{ otherwise} \end{cases} \tag{3}$$

After obtaining $\Delta\psi_t$, the UAV heading can be updated as

$$\psi_{m,t} = \psi_{m,t-1} + \Delta\psi_t. \tag{4}$$

Using the above method, the UAV heading is tractable frame by frame provided that the initial heading $\psi_{m,0}$ is known. However, it should be noted that this heading tracking algorithm only works when the UAV inter-frame rotational motion is less than $\alpha/2$. Fortunately, a 10 Hz laser scanner is fast enough to handle the non-aggressive flight cases. In actual implementation, the longest line extracted for the current frame can be used for the heading alignment.

3) Point feature association: The end points of the line clusters can be treated as local point features, in which some of them should physically associate with the known map corners. The next step is to associate these local point features to the globally known map features. This can be done by transforming the locally observed point features to the global map frame based on the information of previous-frame UAV position $[x_{t-1}, y_{t-1}]$ and the current-frame UAV heading ψ_t. As the UAV rotational motion has been resolved, the difference between the obtained feature points and the known map feature points should be caused by translational motion only. By considering the fact that this translational motion between frames of 10 Hz is very small, the nearest neighbor searching is more than enough to associate them well. The 2D transformation from the laser scanner local frame to the global map frame can be calculated as,

$$q_{j,m} = [x_{m,t-1}, y_{m,t-1}]^{\mathrm{T}} + R_t \times q_{j,l}, \tag{5}$$

where $q_{j,l}$ is the local feature point, and R_t is the rotation matrix from local frame to global frame calculated based on ψ_t,

$$R_t = \begin{bmatrix} \cos\psi_t & \sin\psi_t \\ -\sin\psi_t & \cos\psi_t \end{bmatrix}. \tag{6}$$

4) Position tracking: Similar to the method in rotation tracking, the current position can be calculated based on the previous-frame position $[x_{m,t-1}, y_{m,t-1}]$ and an incremental change $[\Delta x_t, \Delta y_t]$:

$$[x_{m,t}, y_{m,t}] = [x_{m,t-1}, y_{m,t-1}] + [\Delta x_t, \Delta y_t], \tag{7}$$

where

$$\begin{bmatrix} \Delta x_t \\ \Delta y_t \end{bmatrix} = \frac{\sum\limits_{n_{i,j}} w_j (p_i - q_{j,m})}{\sum w_j}. \tag{8}$$

This incremental change can be calculated as an average displacement of all the associated features. By considering the laser scanner noise model, i.e. points further away are more noisy, the matched point features are given different weights w_j in calculating the average displacement. The closer the feature points, the larger the weight.

B. Height measurement

In an indoor environment with completely flat ground, UAV height measurement can be simply obtained via a sonar or a one-point laser range finder. However, for the cases when the UAV needs to fly over tables, chairs and windowsills, these sensors will fail badly as the UAV cannot distinguish between the actual floor surface and the surfaces of other objects. Barometer may be a candidate, but its accuracy does not meet the requirement for a UAV to fly in confined indoor environments. To solve this problem, a second laser scanner is mounted orthogonally to the first and a height calculation algorithm with robust floor identification is developed and integrated into the navigation system.

Similar to the line extraction algorithm mentioned above, the same *split-and-merge* method can be applied also. After filtering out those line segments with dissimilar gradients to the ground plane, the rest are sorted by their perpendicular distances to the laser scanner center. The furthest line segments are kept, and among them the longest one is believed to be the true ground. Finally, the UAV height can be calculated as the perpendicular distance of this line to the laser scanner center, compensated by the offset between the laser scanner and the UAV center of gravity (CG) as well as the UAV attitude angles. Using this method, an accurate height measurement can be obtained as long as the laser scanner projects a portion of its laser beams onto the true ground. It even works for the case when the UAV flies over protruding objects on the ground.

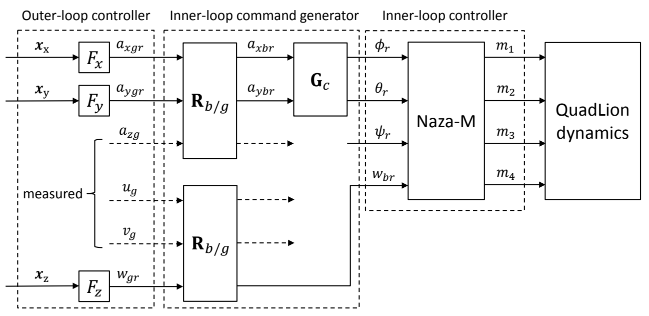


Fig. 5: Control structure of QuadLion

IV. Control

As the platform is already stabilized in the attitude dynamics by the Naza-M controller, only an outer-loop controller for position tracking needs to be designed (see Fig. 5). Here, we adopt a RPT control concept from [10] and apply it to the case of QuadLion trajectory tracking. Theoretically, a system controlled by this method is able to track any given reference with arbitrarily fast settling time subjected to disturbances and initial conditions. The basic idea is as follows. For a linear time invariant system

$$\Sigma = \begin{cases} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + E\boldsymbol{w} \\ \boldsymbol{y} &= C_1\boldsymbol{x} + D_1\boldsymbol{w} \\ \boldsymbol{h} &= C_2\boldsymbol{x} + D_2\boldsymbol{u} + D_{22}\boldsymbol{w} \end{cases}, \tag{9}$$

with $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{y}, \boldsymbol{h}$ being the state, control input, disturbance, measurement and controlled output respectively, the task of RPT controller is to formulate a dynamic measurement control law of the form

$$\begin{aligned} \dot{\boldsymbol{v}} &= A_c(\varepsilon)\boldsymbol{v} + B_c(\varepsilon)\boldsymbol{y} + G_0(\varepsilon)r + \ldots + G_{\kappa-1}(\varepsilon)r^{\kappa-1}, \\ \boldsymbol{u} &= C_c(\varepsilon)\boldsymbol{v} + D_c(\varepsilon)\boldsymbol{y} + H_0(\varepsilon)r + \ldots + H_{\kappa-1}(\varepsilon)r^{\kappa-1}, \end{aligned}$$

so that when an proper $\varepsilon^* > 0$ is chosen,

1) The resulted closed-loop system is asymptotically stable subjected to zero reference.
2) If $e(t,\varepsilon)$ is the tracking error, then for any initial condition $\boldsymbol{x}_0$, there exists:

$$\|e\|_p = \left(\int_0^\infty |e(t)^p| dt\right)^{1/p} \to 0, \quad as \quad \varepsilon \to 0. \tag{10}$$

Similar to the case introduced in [11], the outer dynamics of QuadLion is differentially flat. That means all its state variables and inputs can be expressed in terms of algebraic functions of flat outputs and their derivatives. A proper choice of flat outputs could be

$$\sigma = [x, y, z, \psi]^{\mathrm{T}}. \tag{11}$$

It can be observed that the first three outputs, x, y, z, are totally independent. In other words, we can consider the UAV as a mass point with constrained velocity, acceleration, jerk and etc. in the individual axis of the 3-D global frame when designing its outer-loop control law. Hence, a stand-alone RPT controller based on multiple-layer integrator model in each axis can be designed to track the corresponding reference in that axis. For the x-axis or the y-axis, the nominal system can be written as

$$\begin{cases} \dot{\boldsymbol{x}}_n = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_n + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_n \\ \\ \boldsymbol{y}_n = \boldsymbol{x}_n \end{cases}, \tag{12}$$

where $\boldsymbol{x}_n$ contains the position and velocity state variables and u_n is the desired acceleration.

To achieve better tracking performance, it is common to include an error integral to ensure zero steady-state error subjected to step inputs. This requires an augmented system to be formulated as

$$\begin{cases} \dot{\boldsymbol{x}}_{xy} = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_{xy} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_{xy} \\ \\ \boldsymbol{y}_{xy} = \boldsymbol{x}_{xy} \\ \\ h_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_{xy} \end{cases}, \tag{13}$$

where $\boldsymbol{x}_{xy} = \begin{bmatrix} \int (p_e) & p_r & v_r & a_r & p & v \end{bmatrix}^{\mathrm{T}}$ with p_r, v_r, a_r as the position, velocity and acceleration references in the controlled axis, p, v as the actual position and velocity and $p_e = r_p - p$ as the tracking error of position. In Fig. 5, $\boldsymbol{x}_x$ and $\boldsymbol{x}_y$ are the respective representation of $\boldsymbol{x}_{xy}$ in the x- and y-axis. By following the procedures in [12], an linear feed back control law of the form below can be acquired,

$$u_{xy} = F_{xy} \boldsymbol{x}_{xy}, \tag{14}$$

where

$$F_{xy} = \Bigg[\begin{matrix} \frac{k_i \omega_n^2}{\varepsilon^3} & \frac{\omega_n^2 + 2\zeta\omega_n k_i}{\varepsilon^2} & \frac{2\zeta\omega_n + k_i}{\varepsilon} \\ 1 & -\frac{\omega_n^2 + 2\zeta\omega_n k_i}{\varepsilon^2} & -\frac{2\zeta\omega_n + k_i}{\varepsilon} \end{matrix} \Bigg]. \tag{15}$$

Here, ε is a design parameter to adjust the settling time of the closed-loop system. ω_n, ζ, k_i are the parameters that determines the desired pole locations of the infinite zero structure of (13) through

$$p_i(s) = (s + k_i)(s^2 + 2\zeta\omega_n s + \omega_n^2) \tag{16}$$

The z-axis control is similar but in a lower-order form. As the inner-loop is directly looking for velocity reference in this axis, it is straight forward to model the outer loop as a single integrator from velocity to position, and it leads to the augmented system as

$$\begin{cases} \dot{\boldsymbol{x}}_z = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_z \\ \\ \boldsymbol{y}_z = \boldsymbol{x}_z \\ \\ h_z = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_z \end{cases}. \tag{17}$$

where $\boldsymbol{x}_z = \begin{bmatrix} \int (p_e) & p_r & v_r & p \end{bmatrix}^{\mathrm{T}}$. This leads to a linear feedback control law of

$$u_z = F_z \boldsymbol{x}_z, \tag{18}$$

where

$$F_z = \left[\begin{matrix} -\frac{\omega_n^2}{\varepsilon} & \frac{2\omega_n\zeta}{\varepsilon^2} & 1 & -\frac{2\omega_n\zeta}{\varepsilon^2} \end{matrix} \right].$$

Theoretically, when the design parameter ε is small enough, the RPT controller can give arbitrarily fast responses. However, due to the constraints of the UAV physical dynamics and its inner-loop bandwidth, it is safer to limit the bandwidth of the outer loop to be much smaller than that of the inner-loop dynamics. For the case of QuadLion, the following design parameters are used:

$$x, y \text{ axis}: \begin{cases} \varepsilon & = & 1 \\ \omega_n & = & 0.99 \\ \zeta & = & 0.707 \\ k_i & = & 0.25 \end{cases} \qquad z \text{ axis}: \begin{cases} \varepsilon & = & 1 \\ \omega_n & = & 0.559 \\ \zeta & = & 2 \end{cases}$$

There is still one problem unsolved. From Fig. 5, it can be seen that the output from the outer-loop controller in physical meaning is the desired accelerations in xy-axis and the desired velocity in z-axis, both in global frame. However, the inner-loop controller is looking for attitude references (ϕ_{r}, θ_{r}) and the body-frame z-axis velocity reference. A conversion is needed to link the two control layers together. This leads to another functional block called the *Inner-loop command generator*, in which a rotational conversion from the global frame to the body frame $\mathbf{R}_{b/g}$ is needed and another matrix $\mathbf{G}_c$ is used to convert the desired acceleration references to the desired attitude angles. For all quadrotor UAVs,

$$\mathbf{G}_{\mathrm{c}} \approx \left[\begin{matrix} 0 & 1/g \\ -1/g & 0 \end{matrix} \right]. \tag{19}$$

where g is the gravitational constant.

V. The Implementation Results

The proposed control and navigation algorithms are implemented in C++ language and executed by the UAV onboard computer. For the planar localization algorithm, the average computation time is about 12 ms for a single frame of laser scanner data. Table V shows the position error

TABLE I: Performance of the planar localization algorithm

Position (m)	RMS error (m)	Execution time (s)
$(-2,-4)$	$(0.06,0.05)$	0.011
$(2,-5)$	$(0.07,0.09)$	0.017
$(11,-3)$	$(0.13,0.09)$	0.011
$(1,6)$	$(0.04,0.05)$	0.009

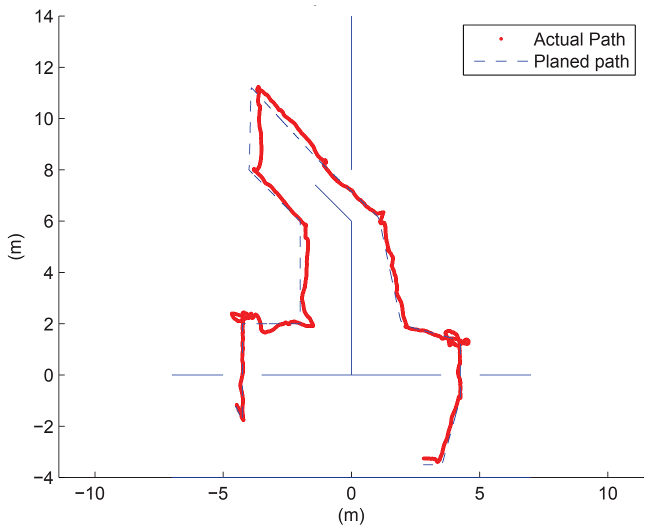
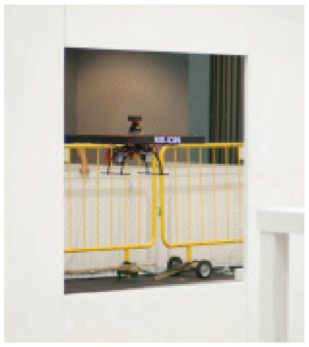


Fig. 6: Localization result after one complete flight

and computation time of the localization algorithm when the UAV is hand held at four stationary locations in the indoor environment. The Root-Mean-Square (RMS) error of the estimated position is very small. To show its dynamic performance, the localization trajectory after one complete flight is logged and shown in Fig. 6. It is actually the full flight path in accomplishing the missions of the SAFMC 2013. Fig. 7(a) and Fig. 7(b) show two photo snaps in the competition, in which the quadrotor UAV flies through a window and drops a payload to the target location. These two tasks require very high localization accuracy as well as precise position control performance from the UAV system. With this kind of robust performance, the Unmanned Aircraft Systems Group from the National University of Singapore have won the top three awards of the competition, namely the “Overall Championship”, the “Best Performance Award”, and the “Most Creative Award”.


(a) Fly through a window

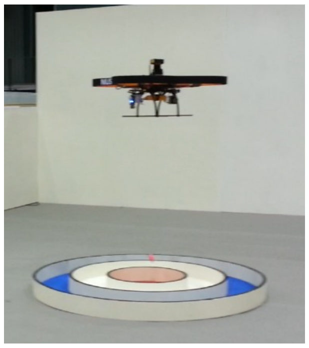

(b) Drop a payload precisely

Fig. 7: Fly-off in the SAFMC competition

VI. CONCLUSIONS

In conclusion, this paper has proposed a robust and efficient navigation solution for a quadrotor UAV to perform autonomous flight in a confined but partially known indoor environment. By assuming structured indoor features and pre-known key corner coordinates, the 3-D pose of the UAV can be accurately estimated by a 2-laser-scanner setup. By utilizing this measurement and applying the RPT control method with integral action, the controlled UAV is able to navigate in the indoor environment fully autonomously. All navigation computation is done by an ARM-based embedded computer onboard of the UAV. The whole UAV system was verified in the SAFMC 2013 and acquired the top awards. Extension of the work can be done to tackle more complicated indoor environments, and autonomous path planning algorithm can be integrated into the system for more demanding applications.

REFERENCES

[1] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, “6D SLAM with an application in autonomous mine mapping,” in *IEEE International Conference on Robotics and Automation*, 2004.

[2] Y. Zhuang, K. Wang, W. Wang, and H. Hu, “A hybrid sensing approach to mobile robot localization in complex indoor environments,” *International Journal of Robotics and Automation*, vol. 27, no. 2, pp. 198–205, 2012.

[3] S. Grzonka, G. Grisetti, and W. Burgard, “Towards a navigation system for autonomous indoor flying,” in *IEEE International Conference on Robotics and Automation*, pp. 2878–2883, 2009.

[4] B. B. Moshe, N. Shvalb, J. Baadani, I. Nagar, and H. Levy, “Indoor positioning and navigaton for micro UAV drones - work in progress,” in *IEEE 27th Convention of Electrical & Electronics Engineerings in Israel*, 2012.

[5] M. K. Mohamed, S. Patra, and A. Lanzon, “Designing simple indoor navigation system for UAVs,” in *19th Mediterranean Conference on Control & Automation*, 2011.

[6] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments,” in *IEEE International Conference on Robotics and Automation*, pp. 957–964, 2012.

[7] S. Ehsan and K. D. McDonald-Maier, “On-board vision processing for small UAVs: Time to rethink strategy,” in *NASA/ESA Conference on Adaptive Hardware and Systems*, 2009.

[8] S. Shen, M. Nathan, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained MAV,” in *IEEE International Conference on Robotics and Automation*, 2011.

[9] G. Borges and M. J. Aldon, “A split-and-merge segmentation algorithm for line extraction in 2D range images,” in *15th International Conference on Pattern Recognition*, 2000.

[10] B. M. Chen, T. H. Lee, and V. Venkataramanan, *Hard Disk Drive Servo Systems*. Advances in Industrial Control Series, New York: Springer, 2002.

[11] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *IEEE International Conference on Robotics and Automation*, 2011.

[12] B. M. Chen, *Robust and H_∞ Control*. Communications and Control Engineering Series, Springer, 2000.