World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:9, No:10, 2015

A Low-Cost Vision-Based Unmanned Aerial System for Extremely Low-Light GPS-Denied Navigation and Thermal Imaging

Chang Liu, John Nash, Stephen D. Prior

Abstract—This paper presents the design and implementation details of a complete unmanned aerial system (UAS) based on commercial-off-the-shelf (COTS) components, focusing on safety, security, search and rescue scenarios in GPS-denied environments. In particular, The aerial platform is capable of semi-autonomously navigating through extremely low-light, GPS-denied indoor environments based on onboard sensors only, including a downward-facing optical flow camera. Besides, an additional low-cost payload camera system is developed to stream both infra-red video and visible light video to a ground station in real-time, for the purpose of detecting sign of life and hidden humans. The total cost of the complete system is estimated to be $1150, and the effectiveness of the system has been tested and validated in practical scenarios.

Keywords—Unmanned aerial system, commercial-off-the-shelf, extremely low-light, GPS-denied, optical flow, infrared video.

I. Introduction

UNMANNED AERIAL VEHICLES (UAVs) are mainly used to replace manned aircraft or human operators, for the purpose of reducing downside risk and rising confidence in mission success. Small quadrotor is one of the most popular subset of UAVs. Because of its agile manoeuvrability, as well as its ability for vertical take-off and landing (VTOL) and stable hovering, it is commonly agreed to be an ideal candidate for search and rescue, surveillance, exploration, agriculture, monitoring and military applications in both indoor and outdoor environments. Being able to fly through space, naturally avoids dealing with rough terrain, when compared to ground vehicles.

Over the last decade, the Global Positioning System (GPS) has been the key to enabling the autonomy of UAVs. However, recently, due to the proven weakness of GPS signal in indoor and urban environments, and rapid development of onboard sensing and computation capability, there has been growing interest in developing and researching alternative navigation methods for UAVs in GPS denied environments [1]–[7]. Among them, computer vision is one of the most used approaches because of its low mass, low power consumption, low price, adjustable field of view (FOV), high accuracy, additional colour information and long range. In the past five years, the world's top research institutes paid high attention to developing advanced vision-based simultaneous localization and mapping (vSLAM) algorithms based on structure from motion (SFM) theory [8]–[15] aiming for consistent position estimation in complex computational fashion. On the contrary, optical flow and image moments based methods [16], [17] aiming for fast velocity estimation by reducing computation requirements. However, for most of the practical safety, security, search and rescue tasks, especially in indoor scenarios, sufficient lighting for either computer vision based autonomous control and vision based data gathering is never easily provided. The problem can be solved by using thermal imaging camera payload, while current off-the-shelf stabilised dual eye visible/thermal camera payloads range in cost from $4000 upwards. Therefore, being able to be realistically deployed in such hazard-rich environments requires the system to be cost effective to reduce the risk of any failure. Therefore, three challenges needs to be addressed in this work:

1) How to robustly and autonomously navigate in an extremely low-light, GPS-denied environment;
2) How to gather effective visual information for the search tasks in such environments;
3) Achieve the above two with relatively low cost.

Fig. 1 The prototype quadrotor UAV

II. Quadrotor Dynamics Modeling

This section presents the nonlinear dynamic model of the mini quadrotor, which forms the basis for the controller synthesis in Section III..

This work was supported by the Faculty of Engineering and the Environment, University of Southampton

Chang Liu, John Nash, and Stephen D. Prior are with the Faculty of Engineering and the Environment, University of Southampton, Southampton, SO16 7QF, UK (e-mail: cl21g11@soton.ac.uk, jn1g13@soton.ac.uk, S.D.Prior@soton.ac.uk, respectively).

The coordinate frames and system setup is indicated in Fig. 2. Quadrotor body frame is fixed to the quadrotor body following the right hand rule with X_b–axis pointing forward. Fig. 2 also shows that the quadrotor has the cross configuration with four motors numbered 1–4, with spinning directions as indicated.

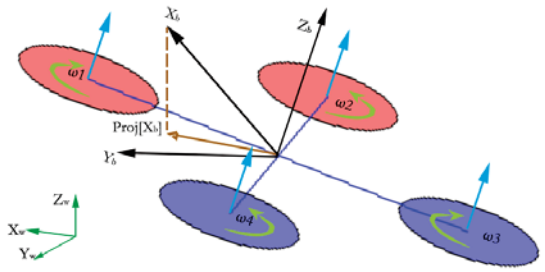


Fig. 2 Coordinate system and quadrotor setup

The rest of the paper uses $\mathbf{x_w}$, $\mathbf{y_w}$, $\mathbf{z_w} \in \mathbb{R}^3$ to denote unit vectors of the three world coordinates, thus $\mathbf{x_w} = (1,0,0)^\top, \mathbf{y_w} = (0,1,0)^\top, \mathbf{z_w} = (0,0,1)^\top$. And the unit vectors of the body frame are expressed in the world frame as $\mathbf{x_b}$, $\mathbf{y_b}$, $\mathbf{z_b} \in \mathbb{R}^3$. The 3D rotation of the quadrotor body is represented by rotation matrix $\mathbf{R} \in SO(3)$. Therefore, $\mathbf{R}$ can also be expressed as:

$$\mathbf{R} = \begin{bmatrix} \mathbf{x_b} & \mathbf{y_b} & \mathbf{z_b} \end{bmatrix}. \tag{1}$$

Note that we express the heading of the quadrotor as the unit vector parallel to the projection of X_b–axis onto the X_w–Y_w plane in world frame, denoted as $proj[\mathbf{x_b}] \in \mathbb{R}^3$, in Fig. 2.

The overview of the nonlinear model is shown in Fig. 3, where inputs of the model, δ_n, are the normalized pulse width modulation (PWM) command signal to the electronic speed controller (ESC) of motors, and outputs of the model are 3 dimensional (3D) position vector $\mathbf{p_w}$ $(= (x,y,z)^\top)$ in the world frame and body rotation matrix $\mathbf{R}$. The following subsections will explain the included components individually.

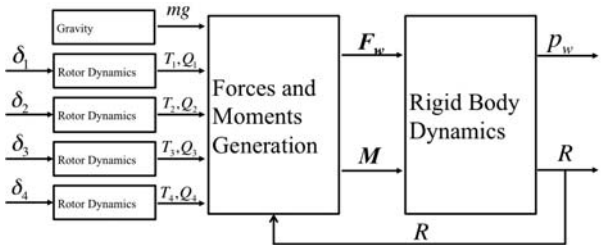


Fig. 3 Overview of the quadrotor dynamics model

A. Rotor Dynamics

The propulsion system of the quadrotor includes two pairs of counter-rotating ESC-motor-propeller systems. The dynamics of the four systems are identical and are approximated by the rotor dynamics model. The model receives the normalized PWM command, δ, and outputs thrust, T, in g and torque, Q, in Nm. Given that the ESC controls the angular speed of the motors linear to the PWM command it receives, based on the aerodynamics of the propeller, each propulsion system can be modelled as:

$$T = c_T(\delta - c_o)^2, \tag{2}$$

$$Q = c_Q T, \tag{3}$$

where c_T, c_Q and c_o are non-dimensional coefficients, and c_T and c_o can be easily obtained from bench static thrust tests.

B. Force and Moment Generation

This module converts all the forces and moments into a force vector, $\mathbf{F_w} \in \mathbb{R}^3$ in the world frame, and a moment vector, $\mathbf{M} \in \mathbb{R}^3$ about each body axis. The gravitational force, the rotor thrusts and rotor moments are considered as three main factors for this module.

The total vector $\mathbf{F_w}$ is derived in the world frame as the sum of gravitational force and thrust force. The gravitational force in the world frame only applies to negative Z_w axis, and the force generated by the four rotors applies to the positive Z_b axis in the body frame. Therefore:

$$\mathbf{F_w} = \mathbf{F_{gravity}} + \mathbf{F_{thrust}}. \tag{4}$$

where:

$$\mathbf{F_{thrust}} = \mathbf{R}\begin{pmatrix} 0 \\ 0 \\ T_{total} \end{pmatrix} = \mathbf{z_b} T_{total}, \tag{5}$$

$$\mathbf{F_{gravity}} = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix}, \tag{6}$$

where m is quadrotor mass and g is standard gravitational acceleration. And T_{total} is the total thrust provided by the four rotors. $\mathbf{R}$ is the rotation matrix of the body frame. And $\mathbf{z_b}$ is the unit vector of Z_b–axis, as defined in (1).

The other output from the module is the moment vector in the body frame, which is approximated in this paper to be generate by the thrusts and torques of the four rotors. Roll moment is contributed by the thrust difference between rotors 1, 4 and 2, 3. Pitch moment is contributed by the thrust difference between rotors 1, 2 and 3, 4. Yaw moment is contributed by the torque difference between rotors 1, 3 and 2, 4. Thus, by also substituting (3), it then can be formulated as:

$$\mathbf{M} = \mathbf{BCT}, \tag{7}$$

where:

$$\mathbf{B} = \begin{bmatrix} \frac{\sqrt{2}}{2}l & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2}l & 0 \\ 0 & 0 & c_Q \end{bmatrix}, \tag{8}$$

$$\mathbf{C} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \tag{9}$$

$$\mathbf{T} = (T_1, T_2, T_3, T_4)^\top. \tag{10}$$

The thrust vector, $\mathbf{T}$, represents the thrusts generated by the four rotors, and l is the distance between rotors and the quadrotor centre of mass.

C. Rigid-Body Dynamics

Rigid-body dynamics formularises the translational and rotational dynamics of the quadrotor, by utilising the simplified Newton-Euler formalism. Therefore, the resulting position vector, $\mathbf{p_w} \in \mathbb{R}^3$ in world frame, and angular speed vector $\mathbf{\Omega} \in \mathbb{R}^3$ about each body axis, can be obtained as:

$$m\ddot{\mathbf{p}}_\mathbf{w} = \mathbf{F_w}, \tag{11}$$

$$\mathbf{J}\dot{\mathbf{\Omega}} = \mathbf{M}, \tag{12}$$

where $\mathbf{J}$ is the inertia matrix of the quadrotor, and since The proposed quadrotor is approximately four way symmetrical, $\mathbf{J}$ is assumed to be a diagonal matrix.

III. Controller Design

Based on the dynamics model developed in the previous section, a nonlinear robust controller is designed to ultimately control the quadrotor 3D position, $\mathbf{p_w}$, and heading, $proj[\mathbf{x_b}]$, to match user input command $\mathbf{p}^\star_\mathbf{w}$ and $proj[\mathbf{x}^\star_\mathbf{b}]$. As shown in Fig. 4, three nested sub-controllers are developed. The quadrotor is controlled accordingly by taking the feedback measurement from inertial measurement unit (IMU) and vision based position sensor. Here we assume the position is obtained from the position sensor.

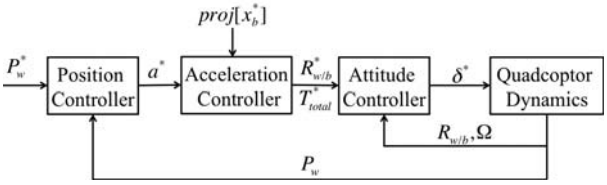


Fig. 4 Overview of the nonlinear controller

A. Attitude Controller

The attitude controller receives the desired body orientation represented as a rotation matrix $\mathbf{R}^\star$, and the desired total thrust provided by the four rotors, $T^\star_{total}$, from acceleration controller output. With the help of attitude feedback measurement, $\mathbf{R}$, from IMU attitude fusion, and angular velocity, $\mathbf{\Omega}$, from gyroscope, then it commands the normalised PWM signals, $\delta^\star_n$, to the four ESCs of motors. It is designed to minimise both the attitude tracking error, $\mathbf{e_R} \in \mathbb{R}^3$, and angular velocity error, $\mathbf{e_\Omega} \in \mathbb{R}^3$, while maintaining the total thrust, T_{total}, as commanded.

Similar with [18], given the desired orientation, $\mathbf{R}^\star$, the attitude error, $\mathbf{e_R}$, is defined to be the Sine of the angle of rotation about each body axis to go from $\mathbf{R}$ to $\mathbf{R}^\star$. It can be formulated as:

$$\mathbf{e}_{\mathbf{R}\times} = \frac{1}{2}(\mathbf{R}^{\star\top}\mathbf{R} - \mathbf{R}^\top\mathbf{R}^\star), \tag{13}$$

which yields a skew-symmetric matrix. The cross map $\times : \mathbb{R}^3 \to \mathfrak{so}(3)$, thus we get $\mathbf{e_R} = (e_{Rx}, e_{Ry}, e_{Rz})^\top$. Moreover, the angular velocity error, $\mathbf{e_\Omega}$, is defined as:

$$\mathbf{e_\Omega} = \mathbf{\Omega} - \mathbf{R}^\top\mathbf{R}^\star\mathbf{\Omega}^\star, \tag{14}$$

where $\mathbf{\Omega} \in \mathbb{R}^3$ is the angular velocity about each axis of the desired body frame, measured directly from the gyroscope.

Then, we can apply the proportional–derivative (PD) control law to compute the desired angular acceleration vector, $\boldsymbol{\alpha}^\star \in \mathbb{R}^3$, about each body axis to be applied to the quadrotor body in order to minimise the difference between $\mathbf{R}$ and $\mathbf{R}^\star$, thus:

$$\boldsymbol{\alpha}^\star = -\mathbf{k_p}\mathbf{e_R} - \mathbf{k_d}\mathbf{e_\Omega}, \tag{15}$$

where $\mathbf{k_p}, \mathbf{k_d} \in \mathbb{R}^3$ are non-negative gain vectors, which can be tuned depending on the aggressiveness of the required manoeuvre. Therefore, based on (12), the desired moment, $\mathbf{M}^\star \in \mathbb{R}^3$, to be generated onto the quadrotor body can be computed by:

$$\mathbf{M}^\star = \mathbf{J}^{-1}\boldsymbol{\alpha}^\star, \tag{16}$$

And then we add total thrust control. Thus, based on (7) we can say:

$$\begin{bmatrix}\mathbf{M}^\star \\ \frac{1}{4}T^\star_{total}\end{bmatrix} = \begin{bmatrix}\mathbf{B} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1\end{bmatrix}\begin{bmatrix}\mathbf{C} \\ \mathbf{1}_{1\times4}\end{bmatrix}\mathbf{T}^\star, \tag{17}$$

where matrix $\mathbf{B}$ and $\mathbf{C}$ are defined in (8) and (9) respectively, and $\mathbf{T}^\star = (T^\star_1, T^\star_2, T^\star_3, T^\star_4)^\top$ is the desired thrust vector, which represents the desired thrust command to each rotor. Therefore, the thrust command for individual rotors can be computed by reversing (17):

$$\mathbf{T}^\star = \begin{bmatrix}\mathbf{C} \\ \mathbf{1}_{1\times4}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{B} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1\end{bmatrix}^{-1}\begin{bmatrix}\frac{1}{4}T^\star_{total} \\ \mathbf{M}^\star\end{bmatrix}. \tag{18}$$

This expression of thrust commands not only applies the desired moment to the quadrotor, but also ensures the total thrust provided by the four rotors is equal to $T^\star_{total}$.

Finally, to generate the normalised PWM signal command, $\delta^\star_n$, for individual rotor, simply apply inverted (2) on $T^\star_n$. Then we get:

$$\delta^\star_n = \sqrt{\frac{T^\star_n}{c_T}} + c_o. \tag{19}$$

This nonlinear attitude tracking controller is demonstrated to recover from any initial orientation in the simulation in [18], which yields almost global exponentially attractiveness when the initial attitude error is less than 180°.

B. Acceleration Controller

The Acceleration controller receives desired acceleration command vector, $\mathbf{a}^\star = (a^\star_x, a^\star_y, a^\star_z)^\top \in \mathbb{R}^3$, from the position controller output, and the quadrotor heading command, $proj[\mathbf{x_b}]$, from user. It then converts the commands into the desired orientation, $\mathbf{R}^\star$, and desired total thrust, $T^\star_{total}$, commands for attitude controller.

For a given acceleration command, $\mathbf{a}^\star$, based on (11) and (4), the desired thrust force acts on the quadrotor in world frame, $\mathbf{F^\star_{thrust}}$, can be computed as:

$$\mathbf{F^\star_{thrust}} = m\mathbf{a}^\star - \mathbf{F_{gravity}}. \tag{20}$$

Based on (5), $\mathbf{F^\star_{thrust}}$ is shown to have the same direction with the desire body Z_b–axis, $\mathbf{z^\star_b}$, with the magnitude equals to desired total thrust, $T^\star_{total}$. Thus:

$$T^{\star}_{total} = \|\mathbf{F}^{\star}_{\mathbf{thrust}}\|, \tag{21}$$

$$\mathbf{z}^{\star}_{\mathbf{b}} = \frac{\mathbf{F}^{\star}_{\mathbf{thrust}}}{T^{\star}_{total}}. \tag{22}$$

Then by assuming the desire heading command, $proj[\mathbf{x_b}]$, is not parallel to $\mathbf{z}^{\star}_{\mathbf{b}}$, we can obtain the unit axis vectors $\mathbf{y}^{\star}_{\mathbf{b}}$ and $\mathbf{x}^{\star}_{\mathbf{b}}$ by:

$$\mathbf{y}^{\star}_{\mathbf{b}} = \frac{\mathbf{z}^{\star}_{\mathbf{b}} \times proj[\mathbf{x_b}]}{\|\mathbf{z}^{\star}_{\mathbf{b}} \times proj[\mathbf{x_b}]\|}, \tag{23}$$

$$\mathbf{x}^{\star}_{\mathbf{b}} = \mathbf{y}^{\star}_{\mathbf{b}} \times \mathbf{z}^{\star}_{\mathbf{b}}. \tag{24}$$

Therefore, the desired quadrotor output rotation matrix will be:

$$\mathbf{R}^{\star} = \begin{bmatrix} \mathbf{x}^{\star}_{\mathbf{b}} & \mathbf{y}^{\star}_{\mathbf{b}} & \mathbf{z}^{\star}_{\mathbf{b}} \end{bmatrix}. \tag{25}$$

C. Velocity Controller

The velocity controller outputs the desired acceleration vector, $\mathbf{a}^{\star}$, to the acceleration controller, in order to minimises the error between desired velocity, $\dot{\mathbf{p}}^{\star}_{\mathbf{w}}$, commanded from the user, and quadrotor velocity measurement, $\dot{\mathbf{p}}_{\mathbf{w}}$, in the world frame.

The position error vector, $\mathbf{e_p}$, and velocity error vector, $\mathbf{e_v}$, are defined as:

$$\mathbf{e_p} = \mathbf{p_w} - \mathbf{p}^{\star}_{\mathbf{w}}, \tag{26}$$

$$\mathbf{e_v} = \dot{\mathbf{p}}_{\mathbf{w}} - \dot{\mathbf{p}}^{\star}_{\mathbf{w}}. \tag{27}$$

Then proportional-integral-acceleration (PI-A) control law is applied, which yields an expression as:

$$\mathbf{a}^{\star} = -\mathbf{k}'_{\mathbf{i}}\mathbf{e_p} - \mathbf{k}'_{\mathbf{p}}\mathbf{e_v} - \mathbf{k}'_{\mathbf{a}}\ddot{\mathbf{p}}_{\mathbf{w}}, \tag{28}$$

where $\mathbf{k}'_{\mathbf{i}}$, $\mathbf{k}'_{\mathbf{p}}$ and $\mathbf{k}'_{\mathbf{a}}$ are non-negative controller gain vectors, which can be tuned according to the require aggressiveness of the position tracking performance. $\dot{\mathbf{p}}^{\star}_{\mathbf{w}}$ is from user command and $\dot{\mathbf{p}}_{\mathbf{w}}$ is measured from velocity estimator described in next section. $\mathbf{p}^{\star}_{\mathbf{w}}$ and $\mathbf{p_w}$ are obtained by integrating $\dot{\mathbf{p}}^{\star}_{\mathbf{w}}$ and $\dot{\mathbf{p}}_{\mathbf{w}}$. And $\ddot{\mathbf{p}}_{\mathbf{w}}$ is obtained directly from accelerometer measurement.

Additionally we add a position-error-integral term with gain value k'_h for z-axis position controller to remove altitude control steady-state error.

TABLE I
PLATFORM DETAILS

Quantity name	Value	Unit
Total mass with thermal camera (m)	1108	g
Quadrotor mass	868	g
Length from motor to centre of mass (l)	255	mm
Propeller size	9.4×5.0	$inch$
Motor Kv	960	RPM/V
Dual-eye camera payload mass including gimbal	240	g
Thermal camera resolution	80×60	W×H pixels

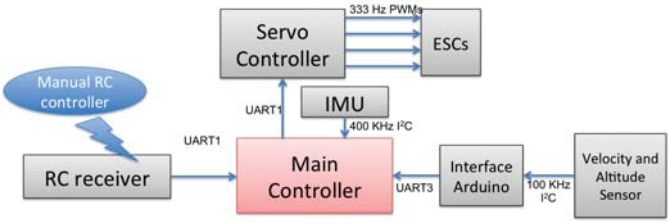
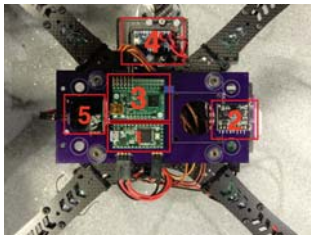
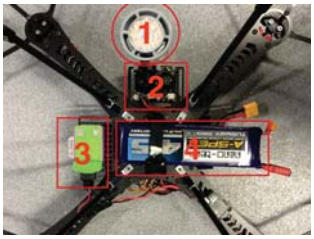


Fig. 5 Block diagram



(a) Top view.
1. Main Teensy controller;
2. FreeIMU;
3. Servo controller;
4. Interface Arduino;
5. Voltage regulator



(b) Bottom view.
1. Spotlight;
2. PX4Flow camera;
3. Magnetic mount for camera;
4. Li-Po battery

Fig. 6 Mechanical system layout

IV. Implementation

This section summarises the implementation details for the working UAS system, including mechanical setup, and autopilot electronics and software description. The quadrotor basic details are summarised in Table. I.

A. Chassis and Propulsion System

We selected GF360 carbon fibre quadrotor frame for industrial standard design suitable to fast prototyping applications. The 360 mm motor span optimised for 9.4 $inch$ propeller, here we use DJI E310 propulsion system for robustness and simplicity. This results in a 600 mm tip-to-tip span, which is almost the maximum safe size to manoeuvre through standard UK doorways (762 mm width).

B. Flight Controller Implementation

Thanks to the high speed Teensy 3.1 processor and a dedicated servo controller, the control loop implementing Section III executes within 3 ms. The physical layout is shown in Fig. 6a and Fig. 6b, and the block diagram in Fig. 5 shows the interactions between components.

- **Main Controller Board** is based on Teensy 3.1 MCU board. It is an ARM based Arduino compatible development board, which features very small form factor (35×18 mm) and fast processor (ARM Cortex–M4 with up to 96 MHz clock speed). It is ideal for a flight controller.
- **IMU** is based on FreeIMU sensor suite [19], including a MPU6050 gyroscope-accelerometer combo-chip, a HMC5883L magnetometer and MS5611-01BA high resolution pressure sensor. However, only the MPU6050

chip is used in this implementation. The orientation fusion estimation uses the library provided with the sensor.

- **Servo Controller** is based on Pololu Mini Maestro Servo Controller board. It is a dedicated servo controller board, which features high resolution (0.25 μs) servo PWM output to 12 channels, with update rate up to 333 kHz, and the fast UART Serial protocol makes it easy to receive command from the main controller board.

C. Velocity and Altitude Estimator Implementation

The horizontal position is obtained by integrating the horizontal velocity. The horizontal velocity of the vehicle is obtained by fusing the measurements from PX4Flow camera [16] and IMU. The vertical position is directly measured by the ultrasonic sensor on PX4Flow camera. On the PX4Flow, the CMOS high speed vision sensor with 21 degree field of view, measures the optical flow at 100 Hz. Then it obtains the ground velocity relative to quadrotor by scaling the average optical flow by the ground distance. Moreover, by subtracting the scaled gyroscope rate, thus it compensates for the optical flow caused by roll and pitch rotation.

In particular, the PX4Flow camera measures the horizontal velocity, $\mathbf{v_c} = [v_{cx}, v_{cy}, 0]^\top$ and vertical position, h. Note that $\mathbf{v_c}$ is always measured with respect to vehicle heading, $proj[\mathbf{x_b}]$, thus the measured vehicle horizontal velocity $\mathbf{v'_w}$ is:

$$\mathbf{v'_w} = proj[\mathbf{x_b}] \odot \mathbf{v_c}, \tag{29}$$

where $\odot$ represents vector element-wise multiplication. Then, given the IMU sampling time, ΔT, and measured body acceleration, $\mathbf{a}$, in body frame from accelerometer, we apply multiple rate complementary filter to compute the estimated vehicle horizontal velocity $\mathbf{v_w}^t = [v_{wx}, v_{wy}, v_{wr}]^\top$ at time t

$$\mathbf{v_w}^t = n(\mathbf{v_w}^{t-1} + \Delta T \mathbf{Ra}) + (1-n)\mathbf{v'_w}, \tag{30}$$

where the coefficient n can be computed by

$$n = \frac{\tau}{\tau + T}. \tag{31}$$

Thus, τ is the time constant for the complementary filter. Therefore the vehicle position $\mathbf{p_w}$ can be computed by

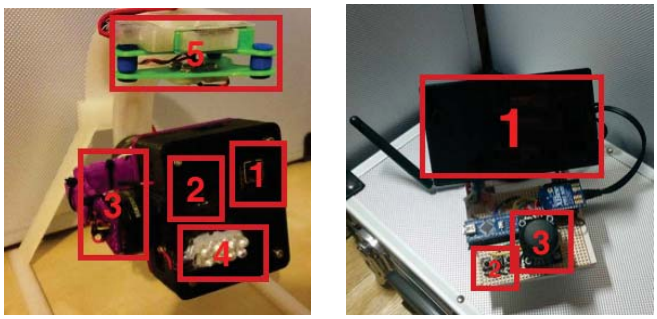
$$\dot{\mathbf{p_w}} = [v_{wx}, v_{wy}, \dot{h}]^\top. \tag{32}$$

The filter executes at IMU sampling frequency, and $\mathbf{v'_w}$ remains the latest measurement value before new velocity is updated from PX4Flow camera.

An additional downward-facing 6 W LED spotlight with 38 degree beam angle, as shown in Fig. 6b, is used to provide sufficient lighting for the PX4Flow camera.

D. Thermal Camera Payload and Ground Station

Thermal camera payload, as shown in Fig. 7a, is a fully custom designed dual eye visible/thermal camera, featuring 2-axis brushless gimbal stabilisation and onboard processing. For thermal imaging the FLIR Lepton thermal imaging core is used to detect heat signatures with 80×60 resolution. In addition, the Raspberry NoIR HD camera is used for visible light imaging with 2592×1944 pixels resolution, and the removal of the IR filter allows for dark areas to be lit using IR light.



(a) Dual-eye camera payload.
1. Lepton thermal camera;
2. Normal camera;
3. Brushless gimbal;
4. IR LED;
5. Magnetic mounting

(b) Ground station.
1. Colour monitor;
2. Push button;
3. Joystick

Fig. 7 Camera payload and ground station

The processing of video and action of commands from the ground station, as shown in Fig. 7b, is performed by a Raspberry Pi Model A+ stored within the payload. User commands (switch between visible and thermal camera, and gimbal tilt) are transmitted via an Xbee pair from the ground station to the camera payload, while the output live video from camera is transmitted back via 25 mW 5.8 GHz transmitter.

The housing and mounting of the payload camera was produced using rapid prototyping methods using an FDM 3d printer. The magnetic conductive coupling mechanism was also design for fast installation, which is shown in both Fig. 6b and Fig. 7a.

E. Components Cost Summary

The following table summarises the price of individual onboard components. The total cost of the entire system is indicated in the last row. Note that the cost will be significantly reduced with higher quantity production.

V. Test Results

An indoor flight test was conducted in an almost completely dark building as indicated in Fig. 8. The manual tuning was conducted in advance of this trial and the following tuning parameters in Table. III were used in this trial.

Flight data was recorded to demonstrate the control performance and validate the theory. Fig. 9 and Fig. 10 show the command-response graphs of terms directly controlled by the user, including horizontal velocity, altitude and heading angle, over 50 seconds duration. Fig. 9a and Fig. 9b shows the horizontal velocity can be effectively controlled within ± 0.2 m/s accuracy and 0.5 s response time. Fig. 10a shows that the altitude is controlled within ± 0.15 m accuracy. Fig. 10b shows the heading angle is controlled within ± 0.1 rad with a small steady-state error as a result of the PD attitude controller. Moreover, to verify the cascaded control architecture, Two

TABLE II
COMPONENTS COST SUMMARY WITH RETAIL PRICES

	Item	Price ($)
Quadrotor	Teensy 3.1 Main Controller	19.8
	GF360 Frame	85.80
	Turnigy 4500mah 3S Battery	49.86
	DJI E310 Propulsion System	223
	FreeIMU 4.0.3	59.6
	Pololu Maestro Servo Controller	51.8
	Optical Flow Camera (PX4FLOW)	159.7
	Subtotal	**650**
Camera Payload	FLIR Lepton Thermal Camera	175
	Raspberry Pi NoIR Camera	27.6
	Raspberry Pi Model A+	26.4
	2× DYS Hollow Shaft 2606 Brushless Gimbal Motor	26.1
	Quanum Micro Alexmos Brushless Gimbal Controller	79.4
	2× Zigbee Module	47.1
	TX5810-100 5.8 GHz Transmitter	39
	ImmersionRC Uno5800 Receiver	51.2
	4.3 Colour TFT Monitor	22.8
	ATmega328p Arduino Nano	4.22
	Subtotal	**499**
	Total Cost	**1149**

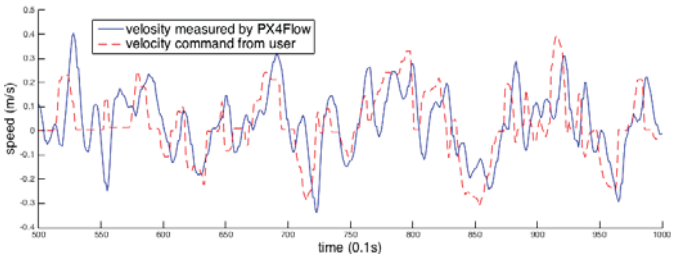
TABLE III
PARAMETERS

Controller	x-axis	y-axis	z-axis
$\mathbf{k_p}$	2200	2200	15
$\mathbf{k_d}$	460	460	10
$\mathbf{k_i'}$	0.7	0.7	1.4
$\mathbf{k_p'}$	3.2	3.2	2.3
$\mathbf{k_a'}$	0.45	0.45	0.5
k_h'	x	x	0.7

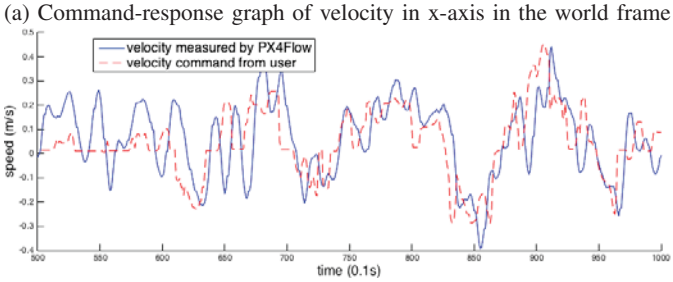
28 second command-response graphs are shown in Fig. 11, indicating the velocity controller performance and the intermediate control signal between acceleration controller and attitude controller at the corresponding time. Note that we have rotated the velocity in the world frame to match the quadrotor heading so that the pitch angle of the quadrotor will result in the change in x-axis velocity change in match-heading frame. It is clearly shown that the output from acceleration controller (red in Fig. 11b reacts to the velocity error (indicated as the difference between red/dashed and blue/solid in Fig. 11a), and acts as the command input to the attitude controller, although the output from acceleration controller also reacts to the acceleration measured directly by the accelerometer, which is not shown in the graph. Moreover, Fig. 11b also shows that there is significant offset (steady-state error) as expected from the PD attitude controller design with an imperfect mass balance of quadrotor body, which has been sufficiently compensated by the higher level velocity controller. It is indicated by the resulting pitch angle (blue/solid in Fig. 11b centred at $0\ rad$.

Finally, included in Fig. 12 is the captured images from the visible and thermal camera of the same scene. External lighting is provided in this case for the sake of indicating the scene in normal image, although the lighting condition will have negligible effect on the thermal image. As shown, the thermal camera equipped is very sensitive to heat signature, which is the direct sign of life and hidden humans.

Fig. 8 Indoor flight test scene (extremely low light)



(a) Command-response graph of velocity in x-axis in the world frame



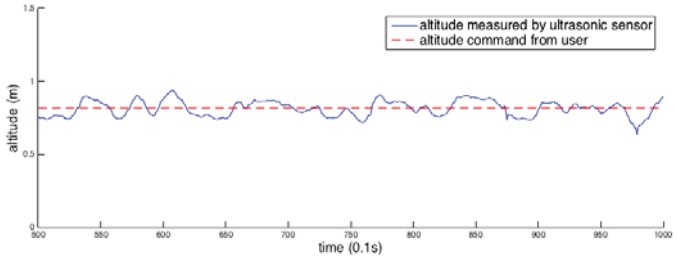
(b) Command-response graph of velocity in y-axis in the world frame

Fig. 9 Velocity control performance evaluation graphs

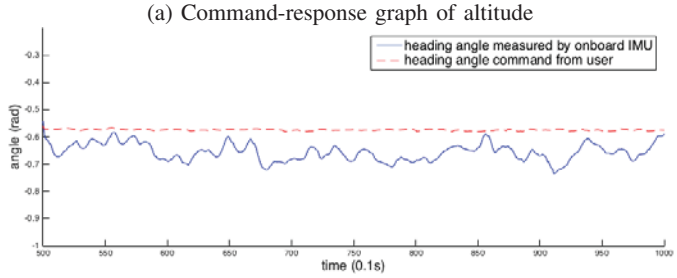
VI. CONCLUSION AND FUTURE WORK

This paper has shown the design and implementation details of a low-cost UAS which is capable of semi-autonomous manoeuvre in extremely low light indoor environment without the need for GPS signal. It is also able to stream both visible and thermal video to ground station in real time. The test data has shown that with the present control and sensor implementation, the aerial platform is able to reliably manoeuvre in the testing environment with user only sending the velocity, altitude and heading command, which greatly simplifies the skill requirement for a human pilot. Moreover, the result has also shown the effectiveness of the customised low cost thermal camera on heat signature gathering in such environment, which shows the great potential of deploying such system in safety, security, search and rescue scenarios.

The future work includes: reducing the size of both quadrotor platform and the payload camera to give more

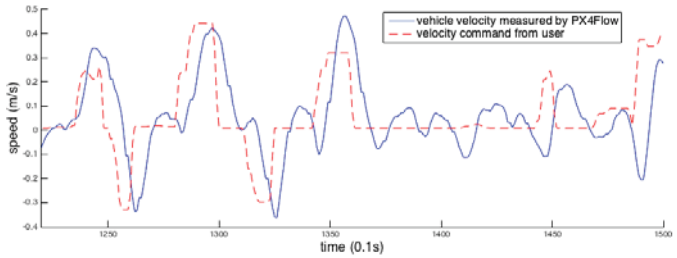


(a) Command-response graph of altitude

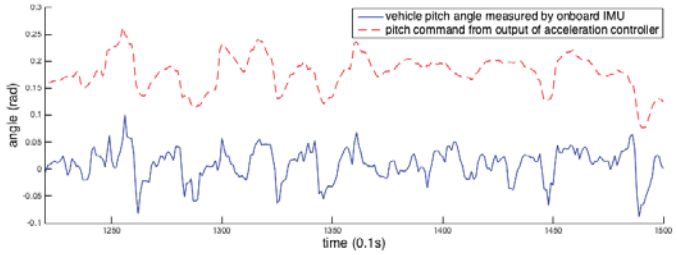


(b) Command-response graph of the heading angle

Fig. 10 Altitude and heading control performance evaluation graphs



(a) Short-term command-response graph of velocity in x-axis in match-heading frame



(b) Short-term command-response graph of pitch angle

Fig. 11 Cascaded control validation graphs

freedom for indoor manoeuvre; improving communication capability to enable operation beyond line-of-sight; develop sense and avoid capability to increase confidence of flight in very constraint spaces.

Acknowledgment

The author would like to thank Mantas Brazinskas and Mehmet Ali Erbil for their continued technical support and encouragement. The author offers sincere appreciation for the learning opportunities provided by them.

References

[1] A. Bachrach and S. Prentice, "RANGE Robust autonomous navigation in GPS denied environments," *Journal of Field Robotics*, 2011.

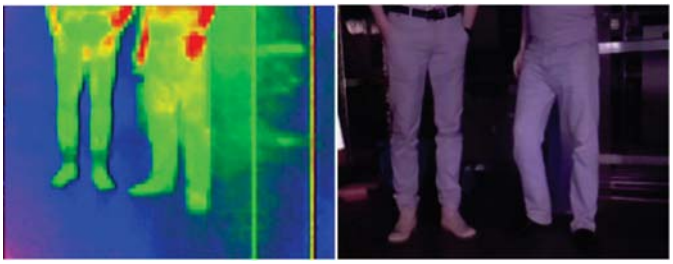
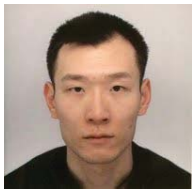


Fig. 12 Comparison of IR and normal camera images

[2] A. Bry, "State estimation for aggressive flight in GPS-denied environments using onboard sensing," *International Conference on Robotics and Automation*, no. Icra, pp. 1–8, May 2012.
[3] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," *Intelligent Robots and Systems*, pp. 2815–2821, Oct. 2012.
[4] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor." *Robotics: Science and Systems*, 2013.
[5] E. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics*, pp. 1–38, 2011.
[6] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular SLAM based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.
[7] M. W. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, and R. Siegwart, "Visual-inertial SLAM for a small helicopter in large outdoor environments," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2651–2652, Oct. 2012.
[8] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1–10, Nov. 2007.
[9] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," *Computer VisionECCV 2014*, pp. 1–16, 2014.
[10] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," *Proc. IEEE Intl. Conf. on Robotics and Automation*, 2014.
[11] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE : Probabilistic , Monocular Dense Reconstruction in Real Time," *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
[12] G. Vogiatzis and C. Hernández, "Video-based, real-time multi-view stereo," *Image and Vision Computing*, vol. 29, no. 7, pp. 434–441, June 2011.
[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," *2011 International Conference on Computer Vision*, pp. 2320–2327, Nov. 2011.
[14] C. Roussillon, A. Gonzalez, and J. Solà, "RT-SLAM: a generic and real-time visual SLAM implementation," *Computer Vision Systems*, 2011.
[15] J. Engel and D. Cremers, "Semi-Dense Visual Odometry for a Monocular Camera," *IEEE International Conference on Computer Vision (ICCV)*, 2013.
[16] D. Honegger and L. Meier, "An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications," *Intl. Conf. Robotics and Automation (ICRA 2013)*, 2013.
[17] R. Mebarki and V. Lippiello, "Image moments-based velocity estimation of UAVs in GPS denied environments," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, 2014, pp. 1–6.
[18] T. Lee, M. Leok, and N. McClamroch, "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE (3)," *arXiv preprint arXiv:1003.2005*, no. i, p. 8, Mar. 2010.
[19] F. Varesano, "FreeIMU: An Open Hardware Framework for Orientation and Motion Sensing," *arXiv preprint*, 2013.



Chang Liu started his PhD investigation into 'Visual control strategies for small remotely piloted aircraft (RPA) operations in GPS denied environments' at the University of Southampton in October 2013. Before that, he obtained a Masters Degree in Artificial Intelligence from the same University in 2012, and a Bachelor's Degree in Electronic Engineering from Newcastle University in 2011.

In the one-year gap between his Masters and the PhD, he undertook a 6-month placement at Kite Power Solutions as an embedded software engineer, helping with the research on autonomous flight of the surf kite to generate electric power.

His research interests are monocular vision, optical flow, simultaneous localisation and mapping, visual inertial navigation system, system modelling and control, embedded intelligence, and powered tether system.

Dr Stephen D. Prior has been working in the area of Field Robotics for the past 25 years. His research interest in autonomous systems relates to a shortlisted entry to the MoD Grand Challenge event in August 2008, where he led a team to design, make and test a novel unmanned aerial vehicle, which consisted of a patented Y6 arrangement. On the basis of this, he founded the Autonomous Systems Lab and has been researching with a small team of staff/students working on defence-related robotic technologies. He is on the editorial board for the International Journal of Micro Air Vehicles and has published widely on the subject. Recent work involved the design and development of a series of Nanotechnology platforms, which were demonstrated and flown at the DSEI exhibition at the Excel Centre in London (September 2011), as well as developing the winning entry to the DARPA UAVForge challenge 2012. During the last two years he has been developing a novel tethered UAS solution for persistent stare capability.