An adaptive neural network approach to the tracking control of micro aerial vehicles in constrained space

Chao Zhang\textsuperscript{a,b}, Huosheng Hu\textsuperscript{b} and Jing Wang\textsuperscript{a}

\textsuperscript{a}Engineering Research Institute of USTB, University of Science and Technology Beijing, Beijing, China; \textsuperscript{b}Department of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

ABSTRACT
This paper presents an adaptive neural network approach to the trajectory tracking control of micro aerial vehicles especially when they are flying in a limited indoor area. Differing from conventional controllers, the proposed controller employs the outer position loop to directly generate angular velocity commands in the presence of unknown aerodynamics and disturbances and then the fast inner loop to handle the angular rate control. Adaptive neural networks are deployed to estimate all the uncertain factors with the adaptation law derived from the Lyapunov function. To achieve a real-time performance, a norm estimation approach of ideal weights is designed to achieve a high bandwidth and lighten the burden of computation burden. Meanwhile, a barrier Lyapunov function is introduced to guarantee the constraint of vehicle positions as well as the validity of the neural network estimation. Simulations and practical flight tests are conducted to verify the feasibility and effectiveness of the proposed control strategy.

1. Introduction
With the recent advancement of materials, micro electro mechanical sensors-based sensors and energy storage devices, micro aerial vehicles (MAVs) have achieved large-scale deployment (Budiyono et al., 2013; Howard & Kaminer, 1995), and been widely used in many civilian and military applications, e.g. wildfire monitoring, aerial filming, and pollution assessment, thanks to their easy construction and ability to take-off and land vertically (Cai, Dias, & Seneviratne, 2014). Meanwhile, these small flying machines, especially quadrotors have been seen in the research institutes worldwide because of their simple steering principle and low cost (How et al., 2009), such as the Autonomous Systems Lab at Swiss Federal Institute of Technology (Lupashin et al., 2014) and the General Robotics, Automation, Sensing and Perception Laboratory at University of Pennsylvania (Michael, Mellinger, Lindsey, & Kumar, 2010).

It is clear that MAVs have complex nonlinear behaviours, under-actuated property and strong system coupling, which makes the flight control very challenging. Some researchers tried to make some change in the MAV mechanical structure so that they can be controlled more accurately, such as reverse ducted fans units (Miwa & Marubashi, 2014) and thrust vectoring nozzles (Imamura, Uemura, Miwa, & Hino, 2014). These modifications will make the system dynamics more complicated and need more consideration of controller design. The position control system is an essential part of MAVs to stabilise their attitude and track the desired trajectories (Mahony, Kumar, & Corke, 2012). The classical PID (proportional–integral–differential) and PD feedback control methods have been used in many commercial MAVs and show acceptable performance (Efe, 2011; Erginer & Altug, 2007). However, their control accuracy decreases as the flying speed increases. At the same time, many model-based linear control approaches have been proposed as well, including optimal control, linear-quadratic regulation, and robust control (Kendoul, 2012).

Recently, many modern nonlinear control methods have been employed to improve the flight performance, such as adaptive control, backstepping control, feedback linearisation, and model predictive control. However, most of these control methods require accurate system models that are normally difficult to measure or obtain in a practical system. Achtelik, Bierling, Wang, Höcht, & Holzapfel (2011) proposed an adaptive control method for a quadrotor MAV in the presence of parameter uncertainties. The geometric method is also employed to a quadrotor MAV (Lee, Leok, & McClamroch, 2010), which is difficult to achieve real-time operation. Since all these approaches ignore the aerodynamic effects and external disturbances during the controller design, they are unable to be deployed in MAVs that operate outdoors at a high speed (Hoffmann, Huang, Waslander, & Tomlin, 2007). In order to estimate model parameters and compensate for system uncertainties in MAVs, system identification technology was deployed for control design in Abas, Legowo, and Akmeliawati (2011) and Samal, Anavatti, and Garratt (2009), and other methods include sliding mode control (Madani & Benallegue, 2006) and a disturbance observer for position control (Lee, Back, & Choy, 2012). A finite-time observer is proposed in Zavala-Rio, Fantoni, and Sanahuja (2016) to cope with the lack of velocity measurements. However, the sliding mode control suffers from the chattering problem, and may cause MAVs to crash under some extreme circumstances. In contrast, the backstepping method has a simple design process, but poor robustness to system changes (Bouabdallah & Siegwart, 2005). Robust control is deployed as an input compensator to improve the...
performance of a PD controller (Wang, Chen, Lu, & Zhong, 2013) and it can improve the system stability under parameter uncertainties and disturbances (Cai, Chen, Dong, & Lee, 2011), but there are only simulation results and cannot achieve real-time performance in many cases.

On the other hand, many intelligent control approaches have been introduced into flight control since they offer some advantages over conventional methods, especially in dealing with highly nonlinear systems and model uncertainties (Raza & D’Andrea, 2010). By using an iteration learning strategy, some precise aggressive maneuvers and acrobatic movements have been shown in the quadrotor platforms, such as balancing an inverted pendulum (Figueroa, Faust, Cruz, Tapia, & Fierro, 2014) and high-speed multi-flips (Lupashin, Schollig, Sherback, & D’Andrea, 2010). Excellent attitude tracking under near hover flight is realised via fuzzy logic control (FNU & Cohen, 2014). Other applications include an adaptive single hidden layer perceptron neural networks control of attitude and height (Johnson & Kannan, 2002), a neural network-based black-box identification for unknown dynamics of a helicopter (Samal et al., 2009), Lyapunov function-based weights updating (Diersk & Jaganathan, 2010), a recurrent neural network to handle the non-linear optimisation for model predictive control (Dalamakidis, Valavanis, & Piegl, 2011), an adaptive neural network to compensate external disturbances (Lei, Ge, & Fang, 2014), and the elimination of partial wing loss in flights (Kim et al., 2013). However, intelligent technologies normally involve a heavy burden of online computation burden, and may even fail and destabilise the whole system (Ren, Ge, Tée, & Lee, 2010).

MAVs have a great potential in indoor applications such as environment reconstruction and secure search. However, the indoor areas are always narrow and can only offer a small space for the vehicle to fly in. This will be a big challenge for the trajectory following control system since the vehicles cannot deviate from the reference very much; otherwise a crash to obstacles may occur. In other words, the output position of the MAV system has to be guaranteed in a limited range for unknown dynamic of a helicopter (Mahony et al., 2002), a neural network-based black-box identification for unknown dynamics of a helicopter (Samal et al., 2009), Lyapunov function-based weights updating (Diersk & Jaganathan, 2010), a recurrent neural network to handle the non-linear optimisation for model predictive control (Dalamakidis, Valavanis, & Piegl, 2011), an adaptive neural network to compensate external disturbances (Lei, Ge, & Fang, 2014), and the elimination of partial wing loss in flights (Kim et al., 2013). However, intelligent technologies normally involve a heavy burden of online computation burden, and may even fail and destabilise the whole system (Ren, Ge, Tée, & Lee, 2010).

The notation $\Omega$ denotes the skew-symmetric matrix of $\Omega$ with introduced so that the constraint of vehicle positions can be guaranteed.

The rest of this paper is organised as follows. Section 2 presents the mathematical models for the MAV used in this research. In Section 3, a novel two-loop control system is proposed for the MAV control, including the deployment of adaptive neural networks and Lyapunov function. Simulation and real flight tests are described in Section 4 to show the feasibility and effectiveness of the proposed control strategy. Finally, a brief conclusion and future work are given in Section 5.

## 2. Mathematical model

### 2.1. Dynamic model

A Hummingbird quadrotor made by Ascending Technology is used in this research. Figure 1 shows the two right-handed coordinate systems used in describing the MAV location and trajectory. The world frame $\mathcal{W}$, is defined by axes $\{X_W, Y_W, Z_W\}$ with $Z_W$ axis pointing upward. The body frame $B$ coincides with the centre of mass and is defined by axes $\{X_B, Y_B, Z_B\}$. $X_B$ is always aligned with the preferred forward direction and $Z_B$ perpendicular to the plane of four rotors. The transformation matrix $R_B$, between $\mathcal{W}$ and $B$ can be expressed by Euler angles or quaternion. Using the Newton’s second law, translational position $\mathbf{p}$ and rotational matrix $R$ equations of the MAV motion can be derived easily (Mahony et al., 2012)

$$m\ddot{\mathbf{p}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \mathbf{f}_a + \mathbf{d},$$  \hspace{1cm} (1)

$$\dot{R} = R\mathbf{\Omega},$$  \hspace{1cm} (2)

$$\dot{\mathbf{\Omega}} = -\mathbf{\Omega} \times R\mathbf{\Omega} + \mathbf{\tau} - \mathbf{g}_a + \mathbf{d}_r + \mathbf{d}_e,$$  \hspace{1cm} (3)

where $m$, $g$, and $J$ denote the total mass, the gravitational constant, and the inertia matrix w.r.t the frame $B$, respectively. $\mathbf{p}$ is the position vector w.r.t the frame $\mathcal{W}$. $\mathbf{f}_a$ and $\mathbf{d}$ are the aerodynamic effect and the unknown disturbances, respectively, in the translational loop. $\mathbf{g}_a$, $\mathbf{d}_r$, and $\mathbf{d}_e$ are the gyroscopic torque, the aerodynamic effect and unknown disturbances in the rotational loop. The control input $T$ and $\mathbf{\tau} = (\tau_1, \tau_2, \tau_3)^T$ represent the total thrust and the total torque produced by propellers. $\mathbf{\Omega} = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity vector w.r.t the frame $B$. The notation $\hat{\mathbf{\Omega}}$ denotes the skew-symmetric matrix of $\Omega$ with...
the following equation:
\[
\hat{\Omega} = \begin{bmatrix}
0 & -w_z & w_y \\
-w_z & 0 & -w_x \\
-w_y & w_x & 0 \\
\end{bmatrix}.
\] (4)

Taking the first derivative of (1) and using (2), the acceleration vector \( \mathbf{a} \) can be derived:
\[
\dot{\mathbf{a}} = R\hat{\Omega} \begin{bmatrix}
0 \\
\dot{T}/m \\
\end{bmatrix} + R \begin{bmatrix}
0 \\
0 \\
\end{bmatrix} + \dot{\mathbf{f}}_s/m + \ddot{\mathbf{d}}/m.
\] (5)

Substitute angular velocity vector \( \Omega \) into (5) and write in a compact form:
\[
\begin{bmatrix}
\dot{p} \\
\dot{v} \\
\dot{a} \\
\end{bmatrix} = \begin{bmatrix}
v \\
a \\
\mathbf{u} + \mathbf{f}_1 + \mathbf{f}_2 \\
\end{bmatrix},
\] (6)

It should be noticed that
\[
\mathbf{u} = [u_1 \ u_2 \ u_3]^T = R[Tw_3/m - Tw_x/m \ 0]^T,
\] (7)

where
- \( \mathbf{v} \) denotes the velocity vector w.r.t the frame \( W \);
- \( \mathbf{u} \) denotes the pseudo control input vector of the outer loop;
- the angular rate around the \( z \)-axis (\( w_z \)) will be controlled directly in the inner loop; and
- \( \mathbf{f}_1 \) and \( \mathbf{f}_2 \) represent the unknown aero-dynamic function and the whole disturbance, respectively.

For the inner loop, the control input is a three-dimensional (3D) vector \( \tau \) and control state variable is angular speed \( \Omega \). Rewrite (3):
\[
\dot{\Omega} = b\tau - b\Omega \times J\Omega + \mathbf{f}_\text{in},
\] (8)

where \( b = J^{-1} \) and \( \mathbf{f}_\text{in} \) denotes all the model uncertainties and external disturbances.

### 2.2. Motor model

As seen in Figure 1, motor 1 is the motor on the \(+X_B\) arm with a yellow label and the other three motors are allocated to \(+Y_B, -X_B, -Y_B\) arms, respectively. For a typical multi-rotor flying vehicle, each rotor produces a thrust force in its \( Z_B\)-axis and a torque around its \( Z_B\)-axis. A basic relationship between the two factors and their rotation speed \( n_i \) is \( \mathbf{f}_i = k_j n_i^2, M_i = k_m n_i^2 \). Then, we can write the general relationship in the matrix form for a quadrotor used in this paper
\[
\begin{bmatrix}
\tau_1 \\
\tau_2 \\
\tau_3 \\
\tau_4 \\
\end{bmatrix} = \begin{bmatrix}
lk_j & 0 & -lk_j & 0 \\
-lk_j & 0 & lk_j & 0 \\
k_m & -k_m & k_m & -k_m \\
k_j & k_f & k_f & k_f \\
\end{bmatrix} \begin{bmatrix}
n_1^2 \\
n_2^2 \\
n_3^2 \\
n_4^2 \\
\end{bmatrix}.
\] (9)

where \( l \) denotes the distance from the rotor to the centre of quadrotor, and the unit of motor speed is revolutions per minute (rpm). The parameters \( k_j \) and \( k_m \) can be regarded as constants and be determined from static thrust tests. Hence, we can achieve the required motor speeds by inverse operation of (9).

### 3. Controller design

Figure 2 shows the block diagram of the proposed two-loop cascade controller used in this research, which is inspired by the work of Wang et al. (2011) and Achtelik, Lynen, Chli, and Siegwart (2013). More specifically, the outer loop is deployed for controlling angular velocities instead of the MAV attitude and the inner loop is used for the body-fixed angular rate control. This cascade control structure can maximise the system transmission bandwidth and avoid the singularity problem. Note that the outputs of roll and pitch angles are not in the controlled state anymore, the only implicitly of which appears in the rotation matrix. We can estimate them using the Kalman filter or through an external motion capture system (Achtelik, 2014).

Multi-loop control systems have been widely used in under-actuated quadrotors systems with six degree of freedoms (DOFs) and four control inputs (Fritsch, Tromba, & Lohmann, 2014). The outer loop is for relatively slow position control (10–100 Hz) and the inner loop is for fast attitude control (500–1000 Hz). Different from these existing systems, the neural networks is introduced in the outer loop to handle system uncertainties. Moreover, a norm estimation of ideal weights vector approach is deployed to realise real-time calculation (Zuo & Wang, 2014). A BLF is used to validate the NN approximation during flights. A simple linear active disturbance rejection control (ADRC) control approach is adopted in the inner loop to enable the control loop to achieve good robustness and fast convergence.

#### 3.1. Outer control loop

Here we focus on the design of a nonlinear controller for a quadrotor system in order to smoothly track a given trajectory \( \mathbf{p}_d \) and the practical outputs can always be constrained. Assume that the MAV trajectories are generated with sufficient smoothness and bounded time derivatives, e.g. a set of 3rd order polynomials (Mellinger, Michael, & Kumar, 2012). A BLF approach is used to guarantee that the output positions remain in a determined set about \( \mathbf{p}_d \) and the radial basis function (RBF) neural networks are employed to compensate for the unknown aero-dynamics and disturbances.

Let \( \mathbf{z}_1 = \mathbf{p} - \mathbf{p}_d, \mathbf{z}_2 = \mathbf{v} - \mathbf{v}_d - \alpha_1 \) and \( \mathbf{z}_3 = \mathbf{a} - \mathbf{a}_d - \alpha_2 \), where \( \mathbf{v}_d \) and \( \mathbf{a}_d \) are the first and second derivative of the given trajectory, respectively, \( \alpha_1 \) and \( \alpha_2 \) are stabilising functions to be determined later.

**Assumption 3.1:** Suppose the aero-dynamic function and the unknown disturbance can be estimated by neural networks, i.e. \( \mathbf{f}_1 = (W_{1i}^*)^T \phi_1(x_1) + \xi_1, \mathbf{f}_2 = (W_{2i}^*)^T \phi_2(x_2) + \xi_2 \), where \( W_{1i}^*, W_{2i}^* \) are the optimal weight vectors, \( \phi_1(\cdot), \phi_2(\cdot) \) the basis functions of neural networks, \( x_1, x_2 \) the NN input and \( \xi_1, \xi_2 \) denote the approximation errors.

In this paper, the NN activation functions are chosen as the commonly used Gaussian functions, i.e. \( \phi_i = \exp[-(x_i - c_i_j)/(\eta_i j)] \), \( j = [1, 2], i = [1, 2, ..., i] \), where \( \eta_i j \) is the total number of nodes, \( c_i_j \) the centres vector of the corresponding node, the width of the Gaussian function. The common
method is to find an update law for the estimated weight vector, but we choose to update its norm, which avoids too much computation burden. For \(i = 1, 2, 3\), let \(\theta_{1i} = \|W_{1i}^s\|\), \(\theta_{2i} = \|W_{2i}^s\|\), and \(\hat{\theta}_{1i}, \hat{\theta}_{2i}\) denote their estimation, where \(\|\cdot\|\) is the standard 2-norm operation, i.e. for a given vector \(x\), \(\|x\| = \sqrt{x^T x}\).

Through an analysis of aerodynamic and blade flapping effects (Mahony et al., 2012) and (Huang, Hoffmann, Waslender, & Tomlin, 2009), the input of the first neural network of each axis loop is simply \(v\) and \(a\). To compensate for the unknown disturbances, Euler angles \(\Theta, \Omega, a, p\) are defined as input to the second neural network.

Then, consider the following Lyapunov function,

\[
V_a = \frac{1}{2} \sum_{i=1}^{3} \ln \left(\frac{k_{bi}^2}{k_{bi}^2 - z_{i1}^2}\right) + \frac{3}{2} \sum_{i=1}^{3} z_{i2}^2 + \frac{3}{2} \sum_{i=1}^{3} z_{i3}^2.
\]

(10)

where \(k_b = [k_{b1}, k_{b2}, k_{b3}]^T\) is a positive constant vector that denotes the constraint bound of output, i.e. \(|z_{i1}| < k_{bi}(i = 1, 2, 3)\). The first term is a BLF taken from Ren et al. (2010).

Since the general Lyapunov function has the same structure in three axes, we will only consider for one axis from now on and neglect the axis subscript for simplification.

\[
V_1 = \frac{1}{2} \ln \left(\frac{k_{b1}^2}{k_{b1}^2 - z_{11}^2}\right) + \frac{3}{2} z_{12}^2 + \frac{3}{2} z_{13}^2.
\]

(11)

Differentiating \(V_1\) against time yields

\[
\dot{V}_1 = \frac{z_{11} \alpha_{11}}{k_{b1}^2 - z_{11}^2} + z_2 \left(\frac{z_{11}}{k_{b1}^2 - z_{11}^2} + z_3 + \alpha_2 - \alpha_1\right) + z_3 (\dot{\alpha}_2 - \dot{\alpha}_d).
\]

(12)

Let \(\alpha_1 = -k_1 z_{11}, \alpha_2 = -k_2 z_2 - \frac{z_3}{k_{b1}^2 - z_{11}^2} + \alpha_1 (k_1, k_2 > 0)\), yields

\[
\dot{V}_1 = \frac{z_{11} \alpha_{11}}{k_{b1}^2 - z_{11}^2} + z_2 (\dot{k}_2 z_2 + z_3) + z_3 (\dot{\alpha}_2 - \dot{\alpha}_d) \\
= \frac{z_{11} \alpha_{11}}{k_{b1}^2 - z_{11}^2} - k_2 z_2^2 + z_3 (\dot{f}_1 + f_2 - \alpha_2 - \alpha_d + z_2)
\]

(13)

and substitute \(f_1 = (W_{1s}^T \phi_1 + \xi_1), f_2 = (W_{2s}^T \phi_2 + \xi_2)\) into (13) with using Young’s inequality, yields

\[
\dot{V}_1 = -k_1 \frac{z_{11}^2}{k_{b1}^2 - z_{11}^2} - k_2 z_2^2 + z_3 (\dot{\alpha}_2 - \dot{\alpha}_d + z_2) \\
+ z_3 ((W_{1s}^T \phi_1 + (W_{2s}^T \phi_2 + \xi_1 + \xi_2) \\
\leq -k_1 \ln \left(\frac{k_{b1}^2}{k_{b1}^2 - z_{11}^2}\right) - k_2 z_2^2 + z_3 (\dot{\alpha}_2 - \dot{\alpha}_d + z_2) \\
+ \frac{1}{2} z_3^2 (\dot{\theta}_1 \|\phi_1\|^2 + \dot{\theta}_2 \|\phi_2\|^2) + z_3^2 + \frac{1}{2} \xi_1^2 + \frac{1}{2} \xi_2^2 + 1.
\]

(14)

Based on the above results, we are now in a position to establish the following theorem.

**Theorem 3.1:** Consider the system in (6) under Assumption 3.1. If the initial error \(|z_{i1}(0)| < k_{bi}\) and the control law is chosen as

\[
u_i = -k_3 z_{3i} + \alpha_{2i} + \alpha_{di} - z_{2i} - z_{bi}\left(1 + \frac{1}{2} \dot{\theta}_{1i} \|\phi_{1i}\|^2 + \frac{1}{2} \dot{\theta}_{2i} \|\phi_{2i}\|^2\right)
\]

(15)

with the adaptive law

\[
\dot{\theta}_{1i} = \frac{1}{2} \lambda_{1i} z_{3i}^2 \|\phi_{1i}\|^2 - \lambda_{1i} \lambda_{2i} \theta_{1i}, \\
\dot{\theta}_{2i} = \frac{1}{2} \mu_{1i} z_{3i}^2 \|\phi_{2i}\|^2 - \mu_{1i} \mu_{2i} \theta_{2i}
\]

(16)

where \(i = 1, 2, 3\) denote three axes respectively; \(k_{3i}, \lambda_{1i}, \lambda_{2i}, \mu_{1i}, \mu_{2i}\) are positive design parameters. The controller can guarantee that signals of the closed-loop system are uniformly bounded, the tracking errors converge to a neighborhood of zero and the position uniformly remains in a determined set \(|z_{i1}| \leq k_{bi}\).

**Proof:** Denote the norm estimation errors of ideal weight vectors as \(\hat{\theta}_1 = \theta_1 - \hat{\theta}_1, \hat{\theta}_2 = \theta_2 - \hat{\theta}_2\) and consider the following Lyapunov function from (11)

\[
V = V_1 + \frac{1}{2\lambda_1} \hat{\theta}_1^2 + \frac{1}{2\mu_1} \hat{\theta}_2^2.
\]

(17)
Differentiating $V$ against time yields

$$
\dot{V} = \dot{V}_1 + \frac{1}{\lambda_1} \dot{\theta}_1 \dot{\theta}_1 + \frac{1}{\mu_1} \dot{\theta}_1 \dot{\theta}_2
$$

(18)

and substituting (14),(15) and (16) into (18)

$$
\dot{V} \leq -k_1 \ln \left( \frac{k_b^2}{k_0^2} - k_2 z_1^2 - k_3 z_3^2 - \lambda_2 \dot{\theta}_1^2 - \mu_2 \dot{\theta}_2^2 + 1 \right) + \frac{1}{2} \xi_1^2 + \frac{1}{2} \xi_2^2 + \lambda_2 \dot{\theta}_1^2 + \mu_2 \dot{\theta}_2^2.
$$

(19)

Write (19) in a compact form

$$
\dot{V} \leq -k V + \epsilon,
$$

(20)

where $k = \min \{2k_1, 2k_2, 2k_3, \lambda_1 \lambda_2, 2\mu_1 \mu_2\}$ and $\epsilon = 1 + \frac{1}{2} \xi_1^2 + \frac{1}{2} \xi_2^2 + \lambda_2 \dot{\theta}_1^2 + \mu_2 \dot{\theta}_2^2$. The following results can be achieved.

1. $k$ and $\epsilon$ are both positive constants, and $|z_1(0)| < k_0$. By using the Lemma 1 proposed in Ren et al. (2010), we can obtain $|z_1(t)| < k_0, \forall t > 0$.

2. Let $\eta = \frac{\epsilon}{\tau}$, then (20) satisfies

$$
0 \leq V(t) \leq \eta + (V(0) - \eta)e^{-\epsilon t} \leq \eta + V(0)
$$

(21)

Therefore, from (17), we infer that $z_1, z_2, z_3, \tilde{\theta}_1, \tilde{\theta}_2$ are bounded. Noting that $z_1$ and $p_d$ are bounded, $p$ is also bounded. Then it is easy to verify $\alpha_1$ and $\alpha_2$ are bounded since $|z_1(0)| \leq k_p$. Finally, from the boundedness of $\alpha_1, \alpha_2, z_1, z_2, z_3$, we conclude that $p, v, a$ and $u$ are bounded.

3. From (21), we obtain

$$
\frac{1}{2} \ln \left( \frac{k_b^2}{k_0^2} - z_1^2 \right) \leq \eta + (V(0) - \eta)e^{-\epsilon t}.
$$

(22)

Taking exponentials on both sides of (22) yields

$$
|z_1(t)| \leq k_b \sqrt{1 - e^{-2(\eta + (V(0) - \eta)e^{-\epsilon t})}}.
$$

(23)

It shows that when $t \to \infty$, $|z_1(t)|$ will converge to $k_b \sqrt{1 - e^{-2\eta}}$. Therefore, by appropriate selection of design parameters, the error bound can be made arbitrarily small.

For simplification, the proof was done for one axis in the system. However, it can be easily expanded to three axes since they have the same Lyapunov functions. Therefore, the control approach can guarantee that all signals in three axes will satisfy Theorem 3.1.

**3.2. Inner control loop**

For the inner loop on $\Omega$, it is a relative one degree system and thus a single proportional control can stabilise it. Many trials based on an initial value from simulations have been done to find appropriate parameters which can provide an acceptable performance. However, to improve the robustness under unknown external disturbances, a simple ADRC approach based on P-controller is adopted to command needed torque for each axis considering our limit onboard computing capability. ADRC can achieve a high-speed convergence with excellent robustness, which has been shown in many previous works (Gao, 2013). Furthermore, it can be employed on our 60 MHz onboard micro-processors. We can design a similar neural network-based approach as we have done in the outer loop, but the calculation burden will be heavy for an ARM7 processor and it is unnecessary because we only need the inner loop to be fast and reliable. Actually, we will tune the 1 kHz inner-loop controller parameters first and make it a stable understructure for all kinds of high-level applications during practical experiments. Rewrite (8) in an extended form for each axis

$$
\begin{align*}
\dot{x}_1 &= f_1 + d_1, \\
\dot{x}_2 &= f_2 + d_2, \\
\dot{x}_3 &= f_3 + d_3,
\end{align*}
$$

(24)

where $f_1, f_2, f_3$ are the model uncertainties caused by model errors and $d_1, d_2, d_3$ denote the external disturbances. We will design an independent controller for each axis.

According to the ADRC theory (Han, 2009), each controller consists of TD (tracking differentiator), ESO (extended state observer) and state error feedback law. Taking $x$-axis angular speed loop for example, Figure 3 depicts the structure of the inner controller used in this paper.

```
TD v
P-controller r?
Plant t
```

Figure 3. ADRC control diagram for the inner control loop.

TD is employed to generate the transient profile:

$$
\begin{align*}
\dot{v}_1 &= f_{\text{han}}(v_1 - v, v_2, r_0, \theta) \\
v_1 &= v_1 + h v_2 \\
v_2 &= v_2 + h f_{\text{han}},
\end{align*}
$$

(25)

where $v$ is the reference signal, $v_1$ the optimised signal input and $v_2$ the differential estimation of the reference signal. $f_{\text{han}}(x_1,
\(x_3, r, h\) is a special nonlinear function with the following form which can track the signal differential rapidly.

\[
d = rh, \quad d_0 = hd, \quad y = x_1 + hx_3
\]

\[
a_0 = \sqrt{d^2 + 8|r|y}
\]

\[
a = \begin{cases} 
  x_3 + \frac{a_0 - d}{y} \text{sign}(y), & |y| > d_0 \\
  x_2 + \frac{r}{h}, & |y| \leq d_0 
\end{cases}
\]

\[
\text{tan} = -\begin{cases} 
  \text{sign}(a), & |a| > d \\
  r^2, & |a| \leq d,
\end{cases}
\]

where \(r\) and \(h\) are positive parameters to be designed.

The ESO mainly aims to use an augmented state-space model that includes unknown uncertainties, as an additional state. Its original form employs nonlinear observer gains. In this paper, the linear gains are adopted with parameters \(\beta_1\) and \(\beta_2\) considering the practical onboard computational capability.

\[
\begin{aligned}
e &= o_1 - w \\
o_1 &= o_1 + h(o_2 - \beta_1 e + br) \\
o_2 &= o_2 + h(-\beta_2 e),
\end{aligned}
\]

where \(w\) is the angular velocity output, \(o_1\) and \(o_2\) denote the observer outputs of the angular speed and the unknown extended state, respectively.

For the control law design, instead of the original nonlinear feedback, the linear form is adopted here. It consists of a basic P (proportion) controller and a compensation for uncertainties via ESO output \(o_2\).

\[
\begin{aligned}
e_1 &= v_1 - o_1 \\
\tau &= \frac{1}{k_p}(k_p e_1 - o_2),
\end{aligned}
\]

where \(k_p\) is the feedback proportion coefficient and \(b\) is the control input gain. The stability of the closed-loop system has been studied (Xue & Huang, 2011) and we will validate its fast respond performance later to guarantee the effectiveness and stability of the outer-loop design.

### 3.3. Control allocation

In the outer loop, the pseudo control input \(u\) is generated by (15), and therefore the desired commands for the inner loop can be obtained by

\[
\begin{aligned}
w_{x,d} &= -\frac{m}{T}(R_{1,2}u_1 + R_{2,2}u_2 + R_{3,2}u_3) \\
w_{y,d} &= \frac{m}{T}(R_{1,1}u_1 + R_{2,1}u_2 + R_{3,1}u_3) \\
\dot{T}_d &= m(R_{1,3}u_1 + R_{2,3}u_2 + R_{3,3}u_3).
\end{aligned}
\]

For \(w_{x,d}\), we can free its control and only need the measured information of it, i.e. \(w_{x,d} = 0\). Otherwise, the desired angular speed can also be produced simply by a P(I) controller from yaw reference signals in some applications. The outer loop takes charge of NN estimation and position control based on the Vicon system, and communicates with the quadrotors via a wireless serial link, i.e. XBee ZigBee modules, for information exchange. The outer loop and the communication both work at 50 Hz.

The inner-loop ADRC controller commands the appropriate torque \(\tau\) and then uses it to calculate the desired speed of four rotors. Taking inverse operation of (9), we have

\[
\begin{bmatrix} n_1^2 \\
n_2^2 \\
n_3^2 \\
n_4^2 \\
\end{bmatrix} =
\begin{bmatrix}
0 & -\frac{1}{2k_1} & -\frac{1}{4k_1} & \frac{1}{4k_1} \\
\frac{1}{2k_1} & 0 & -\frac{1}{4k_2} & \frac{1}{4k_2} \\
\frac{1}{2k_2} & \frac{1}{2k_1} & 0 & -\frac{1}{4k_3} \\
\frac{1}{2k_3} & \frac{1}{2k_2} & \frac{1}{2k_1} & 0 \\
\end{bmatrix}
\begin{bmatrix} \tau_1 \\
\tau_2 \\
\tau_3 \\
\tau_4 \\
\end{bmatrix}.
\]

The rotor speed, i.e. speed of brushless motors, is controlled by four independent micro-controllers which are pre-adjusted in the AscTec Autopilot unit. Therefore, we need to transfer the desired speeds to the respective motor controller. The onboard unit also takes charge of collecting sensor data, computing angular velocities and angles in all axes and runs three inner-loop controllers. All these processes are executed with a control loop frequency of 1 kHz.

### 4. Experimental tests

In our robotics laboratory, there are several types of MAVs, such as the famous Parrot AR.Drone quadrotors, AscTec Pelican drones and Hummingbird quadrotors. Figure 4 shows the Hummingbird that is agile for flight in constrained spaces and durable enough to survive most crashes. It should be noticed that the Hummingbird offers a high-level onboard microcontroller for directly controlling the speed of each rotor.

On the other hand, the Vicon Motion Capture System is fixed in the lab to provide a position estimate via tracking special markers, at a very fast speed (50–100 Hz) and high accuracy (mean error is several millimetres). The model of a quadrotor with four markers used in the Vicon system is shown in Figure 5. The specific parameters for the Hummingbird quadrotor used in this paper are listed in Table 1.

#### 4.1. Simulation results

First, we did the inner-loop simulation on the Matlab platform. The sampling time is fixed to 0.001 s as the same in our real applications. The total unknown uncertainties are defined as
a nonlinear function, i.e. $0.1 \sin(10t) \Omega$. Input signal is a time-varying square wave. The controller parameters is chosen as $r_0 = 100$, $h = 0.001$, $\beta_1 = 1000$, $\beta_2 = 33, 333$, $r = 500$.

As shown in Figure 6, the inner-loop controller can converge in 0.02 s for each step input, which means the convergence of outer-loop control inputs can be guaranteed. The maximum torque value is around 0.5 Nm in Figure 7, which can be achieved in the practical system.

Then, for whole system simulation, the reference signals in horizontal directions are $\rho_{dx} = 2 \sin(t)$, $\rho_{dy} = 2 \cos(t)$, and $\rho_{dz}$ is generated by a second-order reference model. The aerodynamic function is $0.07v_i^2 + 0.1a_i^2$; $(i = 1, 2, 3)$ and the error bound for position $k_b = [0.5 \ 0.5 \ 0.5]^T$. The external disturbances function is defined as $0.1 \sin(2t)$. To determine the control parameters, $k_{i1}$, $k_{i2}$, $k_{i3}$ are errors gains of position, velocity and acceleration which are similar to common feedforward PD control. Therefore, these parameters are modified from a PD controller well-tuned by Matlab Simulink toolbox and fixed at $k_{i1} = k_{i2} = 1$, $k_{i3} = 2$.

For neural network details, since only the norm estimation would be updated, more neural nodes can be adopted to enhance the estimation performance. So, $\phi_{i1}$ contains 300 nodes with centres evenly spaced in $[-5, 5] \times [-5, 5]$ and widths $0.1(1 \leq k \leq 300)$; $\phi_{i2}$ also contains 300 nodes with centres evenly spaced in $[-1, 1] \times [-5, 5] \times [-5, 5] \times [0, 6]$ and widths $0.1(1 \leq k \leq 300)$. $\lambda_{i1}$, $\lambda_{i2}$, $\mu_{i1}$, $\mu_{i2}$ determine the updating speed of neural networks. A balance has to be made between the response speed and system convergence. After trials from small values, we found the system shows a good performance around 1 so they are fixed at $\lambda_{i1} = \lambda_{i2} = \mu_{i1} = \mu_{i2} = 1$.

Therefore, we construct two neural networks with two and four dimension inputs, respectively, for each axis controller. With the proposed BLF method, the output position can always stay within the specified compact $[|p| \leq |p_d + k_b|]$ and the output velocity and acceleration are limited by the system actuator saturations all the time, i.e. they cannot exceed the maximal system limit, which ensures that neural networks approximation is valid. The neural network update frequency is the same as the outer-loop controller.

The initial position is $[0, 1.5, 0]$ which is a relatively extreme initial position compared with the given bound $k_b$. The sampling time of the outer position loop is fixed to 0.02 s. Our control target is that the quadrotor can track the given reference path accurately and the output position will not exceed the given bound at any time. The simulations are running on our Matlab simulator, and we can easily get the needed information and flight pose in this interface as shown in Figure 8 and 3D tracking path result is shown in Figure 9. Figure 10 shows output tracking results. It can be seen that the tracking performance achieved our control goals. The norm of neural networks weights estimation results can be seen in Figure 11, with only 6
update parameters instead of 600 in the conventional weights estimation method. To show the validity of the proposed output constraint design, we compare the error outputs in the X-axis derived from our controller and a standard adaptive backstepping law with the same parameters, i.e. \( u_i \) is the same structure in (15) but \( \alpha_2 \) change to \( \alpha_2 = -k_2 z_2 - z_1 + \dot{\alpha}_1 \). From the results in Figure 12, we can see the position output by the proposed controller always resides in the given bound (0.5), while the standard law cannot limit the output in the desired set. It indicates the term \( z_1/(k_2^2 - z_1^2) \) in \( \alpha_2 \) is the key to limiting the output, i.e. when the output is close to the bound value, it will generate a large control input to prevent the quadrotor from going farther.

### 4.2. Real experimental results

The Hummingbird quadrotor and the Vicon system are deployed to validate the proposed control scheme that is described above. The outer-loop controller is running in a high-level PC at 50 Hz, communicating with the Vicon server and onboard embedded system via TCP/IP and wireless serial port, respectively. We add a payload with an elastic cable to the quadrotor, as seen in Figure 13. Our reference trajectory is taking-off to 1.2 m high, moving radius 1 m circles and landing in the end, which is planned in advance. Four 25 cm-diameter, 1.5 m high cylinder obstacles are placed near the flight path as
Figure 13. No. 1 obstacle is placed at (0, −1.6), No. 2 at (1.6, 0), No. 3 at (0, 0.5), and No. 4 at (0, 1.6). The position outputs are measured by the Vicon motion system at 50 Hz.

The mass of the payload is 65 g about 32% of the maximal taking off weight (200 g) and the length of the elastic cable is 50 cm. The inertia of the payload would introduce large unknown uncertainties and there are obstacles in the environment. If we cannot limit our practical position, the quadrotor will certainly collide with them. However, we clearly observed that the control performance remains acceptable in experiments. Figures 14 and 15 show the actual trajectory and details in three axes of our control scheme without considering the payload. Tracking errors can be seen in Figure 16 and the neural network norm estimation results is shown in Figure 17. They indicate that the proposed control method can compensate for internal and external disturbances well. We can see the norm estimation performance works relatively fast from the result on the Z-axis, which is mainly disturbed by additional mass of the payload. The control input and the output attitude are shown in Figure 18. From the above results, we can see the system output can track the given trajectory in real time and the tracking errors remain smaller than k_s and the vehicle never runs into obstacles during whole flight tests. The detailed results can also be seen in the online videos. Note that a short delay between the MAV trajectory and the given signal is caused by the transmission delay, i.e. wireless Zigbee communication (around 40–60 ms).

Remark 4.1: When calculating the thrust command \( T_d \) using (6), \( \dot{T}_d \) obtained from pseudo control input \( u \) becomes a desired command but it cannot be controlled directly actually. In order to acquire \( T_d \), we have to integrate \( \dot{T}_d \). However, in practical controller realisation, it is very difficult to get \( T_d \) from \( \dot{T}_d \) since the integral process introduces large errors and the external environment is noisy. The height control performance is much worse than the simulation results and may be unstable sometimes during practical tests. Here, we employ \( \alpha_2 \) to generate \( T \) directly, i.e. \( T = m(R_{1,3}\alpha_{2x} + R_{2,3}\alpha_{2y} + R_{3,3}\alpha_{2z} + g), R_{i,3} (i = 1, 2, 3) \) are elements in the third column of the rotational matrix \( R \). This operation is natural because the relative degree for the altitude loop is two and that for the horizontal loop is three. The thrust remains at that instant when the x, y position starts to change. More details can also be seen in the work presented by Achtelik et al. (2013).

5. Conclusion
This paper addresses the issue of constrained position control of a quadrotor. The proposed approach is able to track dynamic trajectory and to limit the real output trajectory in narrow indoor applications. A two-loop cascade control was developed for improving the system bandwidth and speeding up the inner-loop respond. The control algorithm in the outer position

Figure 14. 3D trajectory in real flight test.

Figure 15. Position tracking outputs in flight tests.

Figure 16. Practical path tracking errors on three axes.

Figure 17. Practical norm estimation results on three axes.
loop is derived from the BLF method and adaptive neural network. The norm adaptation instead of weights adaptation was deployed to greatly reduce the required number of updated variables in the control process and achieve high loop frequency. The inner loop is a simple linear ADRC control of angular rates. We can see its fastness and robustness in the simulation results. Finally, the proposed control approach was successfully implemented on a Hummingbird quadrotor in both simulation and real experiments. On the other hand, the MAV always suffers from unexpected fast disturbance such as strong wind in practical applications. The proposed approach may be imperfect and need to be tested. Therefore, our future research includes handling fast changing uncertainties, and generating smooth trajectories as well as some precise motion control of the MAV platform.

Acknowledgements
The first author has been financially supported by scholarship from China Scholarship Council. The authors would like to thank Mr Robin Dowling at the University of Essex for his technical support and Mr Andre Ryll at the Ascending Technology company for his help and enlightening discussions.

Disclosure statement
No potential conflict of interest was reported by the authors.

Funding
The first author has been financially supported by scholarship from China Scholarship Council.

Notes on contributors

Chao Zhang received his BSc degree in automation from the University of Science and Technology Beijing, Beijing, China in 2010 and continued his MSc degree in control science and engineering until 2012. He is currently pursuing his Ph.D. degree in control science and engineering at the University of Science and Technology Beijing. He was a visiting researcher at the University of Essex, Colchester, UK from 2014 to 2015. His research interests include neural network-based motion control, flying robots control and advanced control application in engineering systems.

Huosheng Hu is a professor in the School of Computer Science & Electronic Engineering, University of Essex, United Kingdom, leading the robotics research. His research interests include autonomous robots, human-robot interaction, embedded systems, multi-robot collaboration, pervasive computing, sensor integration, intelligent control, rehabilitation and networked robotics. He has published around 500 papers in journals, books and conferences, and received a number of best paper awards. He is a fellow of IET, and a fellow of Institute of Measurement & Control, UK. He has been a program chair or a committee member for many international conferences such as IEEE, ICRA, IROS, ICMA, ROBIO conferences. He currently serves as editor-in-chief for *International Journal of Automation and Computing* (Springer), editor-in-chief for Robotics (MDPI), and editor-in-chief for *Digital Communications and Networks* (Elsevier).

Jing Wang received his BSc and MSc degrees in automation from the University of Science and Technology Beijing, Beijing, China. In 1986, he joined the University of Science and Technology Beijing as a Lecturer. Currently, he is a professor in the engineering research institute of USTB, University of Science and Technology Beijing. His current research interests include advanced control application in engineering systems, metallurgical industry automation, advanced electric drives and industrial robotics.

References

![Figure 18. Control inputs and attitude information results in flight tests.](image-url)