Real-time Trajectory Tracking of a Quadrotor using Adaptive Backstepping Controller and RNN based Uncertainty Observer

Subhash Chand Yogi, Vibhu Kumar Tripathi, Archit Krishna Kamath and Laxmidhar Behera
Department of Electrical Engineering,
Indian Institute of Technology Kanpur, India
Email: subyogi@iitk.ac.in, vibhukt@iitk.ac.in, architkk@iitk.ac.in, lbehera@iitk.ac.in

Abstract—**This paper presents an approach for position and attitude control of a quadrotor using adaptive backstepping technique along with an uncertainty observer via Recurrent Neural Network (RNN). The quadrotor dynamics are expressed as two subsystems, namely translational and rotational, on which the backstepping control law has been developed. In comparison with feedforward neural networks, RNN has better dynamic characteristics and approximation capabilities. Therefore, an RNN based uncertainty observer has been employed to accommodate the system uncertainties as well as the unknown external disturbances. The proposed controller consists of two parts - an adaptive backstepping based controller that contains an RNN observer and a robust controller to deal with the approximation error induced by the RNN. The RNN parameters have been updated via an update law based on Lyapunov stability theory in an online manner where the overall system stability is also guaranteed. The proposed approach has been implemented in simulations for trajectory tracking of the quadrotor in the presence of parametric uncertainties and external disturbances. Also, the hardware results are presented to show the effectiveness of the proposed approach on DJI Matrice 100.**

Index Terms—**Quadrotor, Adaptive backstepping, Recurrent neural Network (RNN), Uncertainty Observer**

I. Introduction

Quadrotor UAVs have found great applications [1] [2] in last decade due to its abilities like aggressive maneuver, vertical take-off and landing and fixed point hovering. To make them suitable for industrial applications, it is essential to develop a robust control strategy that can handle all types of bounded external disturbances and tackle the problem of modeling inaccuracies and parametric uncertainties very precisely.

Several linear and nonlinear control strategies have been developed for the quadrotor in the past. Most of the nonlinear control design techniques such as feedback linearization [3] and dynamic inversion [4] are based on the linear input-output dynamics with cancellation of some useful nonlinearities where sometimes such cancellations are not desirable. In order to reduce the complexity in controller design while avoiding the cancellation of the useful nonlinearities, backstepping algorithms are gaining much attention for the flight controller design. The backstepping design is a recursive control algorithm that works by designing intermediate control laws for some of the state variables which are known as virtual or pseudo controls of the system. In [5], a robust control strategy using H_∞ controller for attitude stabilization and a backstepping controller for trajectory tracking of the quadrotor is developed. A backstepping controller is developed for attitude and position control of a quadrotor in [6]. Proposed approach is robust against disturbances and uncertainties and does not require the prior knowledge of the perturbations. Two nonlinear control strategies are developed for quadrotor motion control using sliding mode and backstepping approach in [7]. In [8], a composite nonlinear control scheme using sliding mode and backstepping approach for quadrotor is developed. Inner loop controller is established using sliding mode and backstepping controller is designed for outer loop control. However, sliding mode controller exhibits the chattering phenomenon. To limit the value of controller gain while maintaining the nominal performance, disturbance observer based control schemes have attracted the attention of the researchers nowadays. In [9], a nonlinear disturbance observer based backstepping controller is developed for the attitude control of a quadrotor. The performance and effectiveness of the proposed strategy are tested using numerical simulations. This work has been further extended for full control of the quadrotor in [10]. However, such disturbance observer techniques requires the complete information of model dynamics for realization effectively.

In most of the practical applications, the information of modelling and parametric uncertainties are not known in advance and very difficult to estimate using conventional model based observer techniques accurately. To tackle this estimation problem, disturbance observer design based on neural network and other learning approaches is being very popular recently. In [11], a neural network based backstepping controller is proposed for trajectory tracking of a quadrotor. Proposed controller does not need to have exact model information and physical parameters for implementation, Therefore this scheme can be applied to any quadrotor model irrespective of their mass, inertia and length. An adaptive control scheme for trajectory tracking using fuzzy logic based backstepping algorithm is developed in [12]. In this work, model based control law is approximated using fuzzy system and adaptive laws are derived using Lyapunov theory. In [13], a robust controller using backstepping and sliding mode algorithm with adaptive radial basis function neural network (RBFNN) is proposed for the attitude control of octorotor subjected

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

to modelling uncertainties and disturbances. The uncertainty observer based on adaptive RBFNN estimates the lumped uncertainties effectively without their bound information. In addition, the adaptive learning algorithm is derived using Lyapunov theory to learn the RBFNN parameters online and compensate the approximation error. However, RBFNN does not have feedback loop so there may not be possibility to capture the dynamic characteristics more accurately. To capture the dynamic characteristics more accurately and to enhance the approximation capability of a neural network, recurrent neural network (RNN) provides better solution. RNN based nonlinear control schemes for quadrotor are not very well presented in the literature and very few work has been done in this area [14]. To the best of our knowledge, RNN based backstepping controller for the real time trajectory tracking of a quadrotor is not developed so far. This motivates us to do this work. The main contribution are as follows:

- A robust controller for position as well as attitude subsystems using backstepping is developed and asymptotic stability is investigated using Lyapunov stability theory.
- Multi-Input-Multi-Output RNN based observer is developed for effective estimation of modelling uncertainty along with the external disturbances with the unknown upper bound. Only two RNN observer are required that reduces the computation complexity up to a large extent.
- The weights of the RNN are updated online where the update laws for weights have been estimated via Lyapunov stability theory by guarantee the overall system stability.
- Extensive numerical simulations are carried out in the presence of parametric uncertainty and external disturbances to check the effectiveness followed by real time validation using DJI Matrice 100.

The rest of this paper is organized as follows. The quadrotor model and problem formulation are presented in section II and III respectively. Proposed methodology is presented in section IV. Simulation and Hardware results are presented in V and VI respectively. Finally conclusion of the paper is given in section VII.

II. Quadrotor Dynamics

The dynamical equations governing the motion of the 6DoF, under-actuated quadrotor is presented using a Newton-Euler equations [15] as shown below:

$$\begin{cases} \ddot{x} = \frac{u_1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - \frac{\rho_x\dot{x}}{m} \\ \ddot{y} = \frac{u_1}{m}(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) - \frac{\rho_y\dot{y}}{m} \\ \ddot{z} = \frac{u_1}{m}(\cos\phi\cos\theta) - G_a - \frac{\rho_z\dot{z}}{m} \\ \ddot{\phi} = (\frac{J_y-J_z}{J_x})\dot{\theta}\dot{\psi} - \frac{I_r}{J_x}\dot{\theta}\bar{\omega} - \frac{\rho_\phi}{J_x}\dot{\phi} + \frac{u_2}{J_x} \\ \ddot{\theta} = (\frac{J_z-J_x}{J_y})\dot{\phi}\dot{\psi} - \frac{I_r}{J_y}\dot{\phi}\bar{\omega} - \frac{\rho_\theta}{J_y}\dot{\theta} + \frac{u_3}{J_y} \\ \ddot{\psi} = (\frac{J_x-J_y}{J_z})\dot{\phi}\dot{\theta} - \frac{\rho_\psi}{J_z}\dot{\psi} + \frac{u_4}{J_z} \end{cases} \tag{1}$$

where:

- $[x\ y\ z]$ represents the position and altitude of the quadrotor in meters.
- $[\phi\ \theta\ \psi]$ represents the roll, pitch and yaw of the quadrotor in rads.
- m, J_x, J_y and J_z represents the mass of the quadrotor and the moment of inertia in the x, y and z directions respectively.
- ρ_x, ρ_y, ρ_z, ρ_ϕ, ρ_θ and ρ_ψ represents the aerodynamic damping coefficients.
- I_r represents the moment of inertia of the rotor.
- $\bar{\omega}$ is given in terms of angular speeds of the rotors as $\omega_4 + \omega_3 - \omega_1 - \omega_2$ and G_a represents the acceleration due to gravity.
- $[u_1\ u_2\ u_3\ u_4]$ represents the control inputs to the quadrotor and is given as:

$$\begin{cases} u_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 = bl(\omega_1^2 + \omega_4^2 - \omega_2^2 - \omega_3^2) \\ u_3 = bl(\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2) \\ u_4 = d(\omega_4^2 + \omega_2^2 - \omega_1^2 - \omega_3^2) \end{cases} \tag{2}$$

where, ω_i is the angular speed of the i^{th} rotor; l is the distance between a rotor and the center of mass of the quadrotor; b and d are thrust and drag factor respectively.

III. Problem formulation

The quadrotor dynamics (1) can be composed into two subsystems namely translational subsystem and rotational subsystem. If $\Gamma = [x\ y\ z]^T$, $\Sigma = [\phi\ \theta\ \psi]^T$ and the external disturbances in both the subsystems $d_p = [d_x\ d_y\ d_z]^T$, $d_a = [d_\phi\ d_\theta\ d_\psi]^T$ then both the subsystems are written as:

$$\ddot{\Gamma} = f(\Gamma, \dot{\Gamma}) + g(\Gamma, \dot{\Gamma})u + d_p \tag{3}$$

$$\ddot{\zeta} = f_a(\zeta, \dot{\zeta}) + g_a(\zeta, \dot{\zeta})u + d_a \tag{4}$$

let us consider virtual control inputs form (1) as:

$$u_x = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)u_1 \tag{5}$$

$$u_y = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)u_1 \tag{6}$$

$$u_z = (\cos\phi\cos\theta)u_1 \tag{7}$$

For brevity, write $f(\Gamma, \dot{\Gamma})$ as f, $f_a(\zeta, \dot{\zeta})$ as f_a, $g(\Gamma, \dot{\Gamma})$ as g and $g_a(\zeta, \dot{\zeta})$ as g_a. When there exists modelling uncertainties, one can write

$$\ddot{\Gamma} = f + \delta f + (g + \delta g)u_p + d_p \tag{8}$$

$$\ddot{\zeta} = f_a + \delta f_a + (g_a + \delta g_a)u_a + d_a \tag{9}$$

where

$$f = \begin{bmatrix} -\frac{\rho_x\dot{x}}{m} \\ -\frac{\rho_x\dot{y}}{m} \\ -G_a - \frac{\rho_x\dot{z}}{m} \end{bmatrix},\ g = \frac{1}{m}I_{3\times3} \text{ and } u_p = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{10}$$

$$f_a = \begin{bmatrix} (\frac{J_y-J_z}{J_x})\dot{\theta}\dot{\psi} - \frac{I_r}{J_x}\dot{\theta}\bar{\omega} - \frac{\rho_\phi}{J_x}\dot{\phi} \\ (\frac{J_z-J_x}{J_y})\dot{\phi}\dot{\psi} - \frac{I_r}{J_y}\dot{\phi}\bar{\omega} - \frac{\rho_\theta}{J_y}\dot{\theta} \\ (\frac{J_x-J_y}{J_z})\dot{\phi}\dot{\theta} - \frac{\rho_\psi}{J_z}\dot{\psi} \end{bmatrix},\ u_a = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{11}$$

$$g_a = \begin{bmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_x} & 0 \\ 0 & 0 & \frac{1}{J_x} \end{bmatrix} \tag{12}$$

The lumped uncertainties can be written as $D_p = \delta f + \delta g u_p + d_p$ and $D_a = \delta f_a + \delta g_a u_a + d_a$ where $D_p =$

$[D_x\ D_y\ D_z]^T$, $D_a = [D_\phi\ D_\theta\ D_\psi]^T$ needs to be estimated. We can write (8) and (9) as:

$$\ddot{\Gamma} = f + gu_p + D_p \tag{13}$$

$$\ddot{\zeta} = f_a + g_a u_a + D_a \tag{14}$$

Finally, the objective can be summarized as: To design the control laws u_p and u_a for translational subsystem (13) and rotational subsystems (14) respectively such that the generated control force drives the quadrotor to track the reference trajectory despite of modelling uncertainty and external disturbances while ensuring the stability of the closed loop system. The lumped uncertainties are assumed to be upper bounded by some unknown constant and are estimated with the help of RNN. Thus, one can write mathematically as:

$$\lim_{t\to\infty} x \to x_d,\ \lim_{t\to\infty} y \to y_d,\ \lim_{t\to\infty} z \to z_d \text{ and } \lim_{t\to\infty} \psi \to \psi_d$$

IV. Adaptive Backstepping Controller and Uncertainty Observer

This section is divided into two parts to design the control law for each of the subsystems.

A. Position subsystem controller design

1) Backstepping Controller: The position subsystem (13) can be written as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f + gu_p + D_p \end{cases} \tag{15}$$

where $x_1 = [x\ y\ z]^T$, $x_2 = [\dot{x}\ \dot{y}\ \dot{z}]^T$ If the desired position is $x_{1d} = [x_d\ y_d\ z_d]^T$ then the tracking error can be written as:

$$e_1 = x_{1d} - x_1 \tag{16}$$

$$\dot{e_1} = \dot{x}_{1d} - x_2 \tag{17}$$

Let a stabilizing control signal (Ω) for $\kappa_1 > 0$ where $\kappa_1 \in \mathbb{R}^{3\times 3}$ and the velocity error (e_2) are given as:

$$\Omega = -\kappa_1 e_1 + \dot{x}_{1d} \tag{18}$$

$$e_2 = x_2 - \Omega \tag{19}$$

Let us consider the Lyapunov function as:

$$V_1 = \frac{1}{2} e_1^T e_1 \tag{20}$$

The time derivative of V_1 can be written as:

$$\dot{V_1} = e_1^T \dot{e_1} \tag{21}$$

$$= e_1^T (x_2 - \dot{x_{1d}}) = e_1(x_2 - \Omega - \kappa_1 e_1) \tag{22}$$

$$= e_1^T e_2 - e_1^T \kappa_1 e_1 \tag{23}$$

To compute the control law, let us consider the augmented Lyapunov function as

$$V_2 = V_1 + \frac{1}{2} e_2^T e_2 \tag{24}$$

By taking the time derivative of V_2

$$\dot{V_2} = \dot{V_1} + e_2^T \dot{e_2} \tag{25}$$

$$= e_1^T e_2 - e_1^T \kappa_1 e_1 + e_2^T (f + gu_p + D_p - \dot{x_{1d}} + \kappa_1 \dot{e_1}) \tag{26}$$

The following controller is chosen to make system asymptotic stable for some $\kappa_2 > 0$ where $\kappa_2 \in \mathbb{R}^{3\times 3}$ as:

$$u_p = \frac{1}{g}[-f - D_p + \ddot{x_{1d}} - e_1 - \kappa_2 e_2 - \kappa_1 \dot{e_1}] \tag{27}$$

$$= \frac{1}{g}[-\kappa_2 e_2 - e_1 - f - D_p - \kappa_1 \dot{e_1} + \ddot{x_{1d}}] \tag{28}$$

By combining (26) and (27), (26) becomes

$$\begin{aligned} \dot{V_2} &= \dot{V_1} + e_2^T \dot{e_2} \\ &= e_1^T e_2 - e_1^T \kappa_1 e_1 + e_2^T(-e_1 - \kappa_2 e_2) \\ &= -e_1^T \kappa_1 e_1 - e_2^T \kappa_2 e_2 \\ \dot{V_2} &\leq 0 \end{aligned} \tag{29}$$

Thus the system stability is guaranteed in the sense of Lyapunov.

However, the lumped uncertainty D_p is required to implement the control law (28) which is generally not known in real-time. Since RNN provides better approximation capabilities and can capture the dynamic characteristics well than the other feed-forward techniques. Moreover, RNN doesn't require the system dynamics knowledge unlike the disturbance observer techniques [9], [10]. Thus an RNN based uncertainty observer has been proposed.

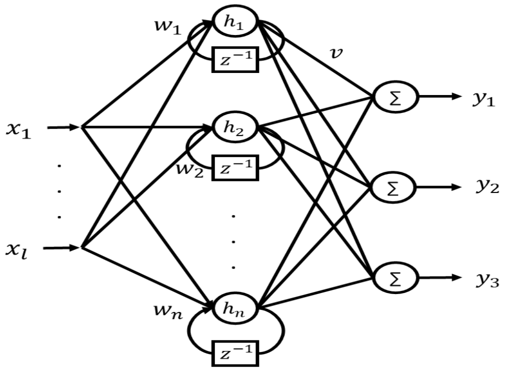


Fig. 1: RNN Architecture

2) RNN based Uncertainty Observer: The architecture of RNN is shown in Figure 1 which has three layers having l inputs and three outputs. Hidden layer has self feedback connection with the weight vector w and has n neurons.

Here the lumped uncertainties D_p are being estimated which can be approximated via RNN as

$$D_p = v^T H(e_2, b, c, w) + \epsilon \tag{30}$$

where $H = [h_1 \quad h_2 \quad ..... \quad h_n]^T, \quad h_i = e^{-net_i}$ and $net_i = \sum_{j=1}^{m} \frac{(x_j + w_j^D ex_{hi} - c_{ij})^2}{b_{ij}^2}$ and ex_{hi} is the value of h_i before one time instant. ϵ is the approximation error which is bounded by some unknown constant as $\epsilon \leq |\epsilon_m|$. Since the ideal weights v, b, c, w are not known thus the estimation of lumped uncertainties can be given as:

$$\hat{D}_p = \hat{v}^T \hat{H}(e_2, \hat{b}, \hat{c}, \hat{w}) \tag{31}$$

The estimation error is given as $\tilde{D}_p = D - \hat{D}_p$ and can be simplified as:

$$\begin{aligned}\tilde{D}_p &= v^T H + \epsilon - \hat{v}^T \hat{H} \\ &= v^T H + \epsilon - (v - \tilde{v})^T (H - \tilde{H}) \\ &= \tilde{v}^T H + v^T \tilde{H} - \tilde{v}^T \tilde{H} + \epsilon \\ &= \tilde{v}^T \hat{H} + \hat{v}^T \tilde{H} + \tilde{v}^T \tilde{H} + \epsilon \end{aligned} \tag{32}$$

To obtain $\tilde{H}$, the Taylor series expansion has been employed and thus one can expand $\tilde{H}$ as:

$$\tilde{H} = \frac{\partial \tilde{H}}{\partial b}(b - \hat{b}) + \frac{\partial \tilde{H}}{\partial c}(c - \hat{c}) + \frac{\partial \tilde{H}}{\partial w}(w - \hat{w}) + O \tag{33}$$

$$\tilde{H} = \begin{bmatrix} \tilde{h}_1 \\ \tilde{h}_2 \\ \vdots \\ \tilde{h}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial \tilde{h}_1}{\partial b} \\ \frac{\partial \tilde{h}_2}{\partial b} \\ \vdots \\ \frac{\partial \tilde{h}_n}{\partial b} \end{bmatrix} \tilde{b} + \begin{bmatrix} \frac{\partial \tilde{h}_1}{\partial c} \\ \frac{\partial \tilde{h}_2}{\partial c} \\ \vdots \\ \frac{\partial \tilde{h}_n}{\partial c} \end{bmatrix} \tilde{c} + \begin{bmatrix} \frac{\partial \tilde{h}_1}{\partial w} \\ \frac{\partial \tilde{h}_2}{\partial w} \\ \vdots \\ \frac{\partial \tilde{h}_n}{\partial w} \end{bmatrix} \tilde{w} + O \tag{34}$$

where,

$$\begin{aligned} w &= \begin{bmatrix} w_1 & w_2 & \dots & w_m \end{bmatrix}^T \\ c &= \begin{bmatrix} c_{11} & \dots & c_{1m}, & c_{21} & \dots & c_{2m}, & \dots & c_{n1} & \dots & c_{nm} \end{bmatrix}^T \\ b &= \begin{bmatrix} b_{11} & \dots & b_{1m}, & b_{21} & \dots & b_{2m}, & \dots & b_{n1} & \dots & b_{nm} \end{bmatrix}^T \end{aligned}$$

The derivative terms can be obtained as follows:

$$\frac{\partial h_1}{\partial c} = \begin{bmatrix} \frac{\partial h_1}{\partial c_{11}} & \frac{\partial h_1}{\partial c_{12}} & \dots & \frac{\partial h_1}{\partial c_{1m}}, & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$\frac{\partial h_2}{\partial c} = \begin{bmatrix} 0 & \dots & 0, & \frac{\partial h_2}{\partial c_{21}} & \frac{\partial h_2}{\partial c_{22}} & \dots & \frac{\partial h_2}{\partial c_{2m}}, & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$\frac{\partial h_n}{\partial c} = \begin{bmatrix} 0 & \dots & 0 & \dots & 0, & \frac{\partial h_n}{\partial c_{n1}} & \frac{\partial h_n}{\partial c_{n2}} & \dots & \frac{\partial h_n}{\partial c_{nm}} \end{bmatrix}$$

Similarly $\frac{\partial H}{\partial b}$, $\frac{\partial H}{\partial w}$ can also be estimated. From (34), one can write

$$\tilde{H} = Dh_b \tilde{b} + Dh_c \tilde{c} + Dh_w \tilde{w} + O \tag{35}$$

By substituting (35) into (32), the following expression is obtained as:

$$\begin{aligned} \tilde{D}_p &= \tilde{v}^T \hat{H} + \hat{v}^T (Dh_b \tilde{b} + Dh_c \tilde{c} + Dh_w \tilde{w} + O) + \tilde{v}^T \tilde{H} + \epsilon \\ &= \tilde{v}^T \hat{H} + \hat{v}^T Dh_b \tilde{b} + \hat{v}^T Dh_c \tilde{c} + \hat{v}^T Dh_w \tilde{w} + \Delta \end{aligned} \tag{36}$$

where $\Delta = \hat{v}^T O + \tilde{v}^T \tilde{H} + \epsilon$ is considered as total approximation error which has an unknown bound as $\Delta \leq \Delta_m$.

3) Stability Analysis: In the presence of parametric uncertainties and external disturbances, the following controller can be implemented as:

$$u_p = g^{-1}[-\kappa_2 e_2 - e_1 - f - \hat{D}_p - \kappa_1 \dot{e_1} + \ddot{x_{1d}}] + u_p^{Robust} \tag{37}$$

where u_p^{Robust} is the robust controller which is taken into account to cope with the approximation error induced by RNN and is explained in this section. The proposed strategy has been explained in the Figure 2.

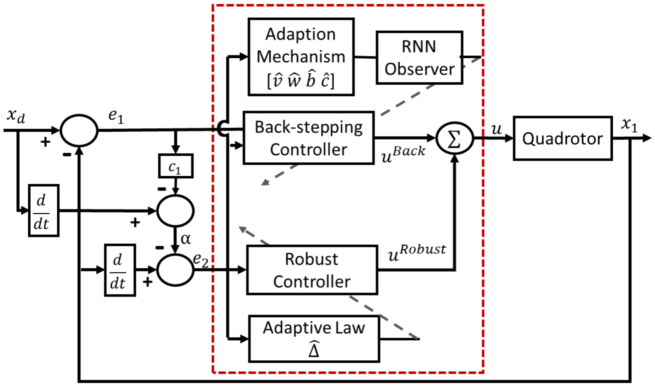


Fig. 2: Proposed control strategy

Theorem 1: For the given position subsystem (15), when the control signal (37) is applied along with the robust controller (38) and RNN observer (31) where the RNN parameters are updated online using (40) then the overall system stability can be guaranteed and thus the tracking error converges to zero asymptotically.

$$u_p^{Robust} = -g^{-1}(\hat{\Delta} + \chi e_2) \tag{38}$$

$$\text{where } \dot{\hat{\Delta}} = \eta_4 e_2 \tag{39}$$

$$\begin{cases} \dot{\hat{v}} = \eta_v^{-1} \hat{H} e_2^T \\ \dot{\hat{b}} = \eta_1 Dh_b \hat{v} e_2 \\ \dot{\hat{c}} = \eta_2 Dh_c \hat{v} e_2 \\ \dot{\hat{w}}^D = \eta_3 Dh_w \hat{v} e_2 \end{cases} \tag{40}$$

Proof: Consider the Lyapunov's function as:

$$\begin{aligned} V_3 = &\frac{1}{2} e_1^T e_1 + \frac{1}{2} e_2^T e_2 + \frac{1}{2} tr(\tilde{v}^T \eta_v^{-1} \tilde{v}) + \frac{1}{2\eta_1} \tilde{b}^T \tilde{b} + \frac{1}{2\eta_2} \tilde{c}^T \tilde{c} \\ &+ \frac{1}{2\eta_3} \tilde{w}^T \tilde{w} + \frac{1}{2\eta_4} \tilde{\Delta}^T \tilde{\Delta} \end{aligned} \tag{41}$$

where $\tilde{\Delta} = \Delta - \hat{\Delta}$ and η_i $i = 1, 2, 3, 4$ are positive coefficients.

The time derivative of V_3 can be written as:

$$\begin{aligned} \dot{V_3} = &e_1^T \dot{e_1} + e_2^T \dot{e_2} - \frac{1}{2} tr(\tilde{v}^T \eta_v^{-1} \dot{\hat{v}}) - \frac{1}{2\eta_1} \tilde{b}^T \dot{\hat{b}} - \frac{1}{2\eta_2} \tilde{c}^T \dot{\hat{c}} \\ &- \frac{1}{2\eta_3} \tilde{w}^T \dot{\hat{w}} - \frac{1}{2\eta_1} \tilde{\Delta} \dot{\hat{\Delta}} \end{aligned} \tag{42}$$

Now

$$\dot{e_2} = \dot{x_2} - \ddot{x_{1d}} + \kappa_1 \dot{e_1} \tag{43}$$

$$= f + g u_p + D_p - \ddot{x_{1d}} + \kappa_1 \dot{e_1} \tag{44}$$

Substituting the control law (37) in the above equation, one can get

$$\begin{aligned} \dot{e_2} &= -\kappa_2 e_2 - e_1 + \tilde{D}_p + g u_p^{Robust} \\ &= -\kappa_2 e_2 - e_1 + \tilde{v}^T \hat{H} + \hat{v}^T Dh_b \tilde{b} + \hat{v}^T Dh_c \tilde{c} \\ &\quad + \hat{v}^T Dh_w \tilde{w} + \Delta + g u_p^{Robust} \end{aligned} \tag{45}$$

From (42) and (45), the following expression is obtained as:

$$
\begin{aligned}
\dot{V_3} &= e_1^T e_2 - e_1^T \kappa_1 e_1 + e_2^T(-\kappa_2 e_2 - e_1 + \tilde{v}^T \hat{H} + \hat{v}^T Dh_b \tilde{b} \\
&+ \hat{v}^T Dh_c \tilde{c} + \hat{v}^T Dh_w \tilde{w} + \Delta + g u_p^{Robust}) - \frac{1}{2\eta_4} \tilde{\Delta} \dot{\hat{\Delta}} \\
&- \frac{1}{2} tr(\tilde{v}^T \eta_v^{-1} \dot{\hat{v}}) - \frac{1}{2\eta_1} \tilde{b}^T \dot{\hat{b}} - \frac{1}{2\eta_2} \tilde{c}^T \dot{\hat{c}} - \frac{1}{2\eta_3} \tilde{w}^T \dot{\hat{w}} \\
&= -e_1^T \kappa_1 e_1 - e_2^T \kappa_2 e_2 + e_2^T(\tilde{v}^T \hat{H} + \hat{v}^T Dh_b \tilde{b} + \hat{v}^T Dh_c \tilde{c} \\
&+ \hat{v}^T Dh_w \tilde{w} + \Delta + g u_p^{Robust}) - \frac{1}{2} tr(\tilde{v}^T \eta_v^{-1} \dot{\hat{v}}) \\
&- \frac{1}{\eta_1} \tilde{b}^T \dot{\hat{b}} - \frac{1}{\eta_2} \tilde{c}^T \dot{\hat{c}} - \frac{1}{\eta_3} \tilde{w}^T \dot{\hat{w}} - \frac{1}{\eta_1} \tilde{\Delta} \dot{\hat{\Delta}} \\
&= -e_1^T \kappa_1 e_1 - e_2^T \kappa_2 e_2 + tr\left(\tilde{v}^T \hat{H} e_2 - \tilde{v}^T \eta_v^{-1} \dot{\hat{v}}\right) + e_2^T \Delta \\
&+ tr\left(\tilde{v}^T \hat{H} e_2 - \tilde{v}^T \eta_v^{-1} \dot{\hat{v}}\right) + \left(e_2^T \hat{v}^T Dh_b \tilde{b} - \frac{1}{2\eta_1} \tilde{b}^T \dot{\hat{b}}\right) + \\
&\left(e_2^T \hat{v}^T Dh_c \tilde{c} - \frac{1}{\eta_2} \tilde{c}^T \dot{\hat{c}}\right) + \left(e_2^T \hat{v}^T Dh_{w^D} \tilde{w}^D - \frac{1}{\eta_3} \tilde{w}^T \dot{\hat{w}}\right) \\
&+ e_2^T g u_p^{Robust} - \frac{1}{\eta_1} \tilde{\Delta} \dot{\hat{\Delta}} \qquad (46)
\end{aligned}
$$

When the robust controller (38) and update rules (40) are employed in the above equation, one can get

$$
\begin{aligned}
\dot{V_3} &= -e_1^T \kappa_1 e_1 - e_2^T \kappa_2 e_2 + e_2^T \Delta - e_2^T(\hat{\Delta} + \chi e_2) - \frac{1}{\eta_1} \tilde{\Delta} \dot{\hat{\Delta}} \\
&= -e_1^T \kappa_1 e_1 - e_2^T(\kappa_2 + \chi) e_2 \\
&= -e_1^T \kappa_1 e_1 - e_2^T \kappa_3 e_2 \\
&\leq -\lambda_{min}(\kappa_1)||e_1||^2 - \lambda_{min}(\kappa_3)||e_2||^2 \\
\dot{V_3} &\leq 0 \qquad (47)
\end{aligned}
$$

By employing the Lyapunov stability theory, one can conclude that all the parameters are bounded and the tracking errors e_1 and e_2 approaches to zero asymptotically. This completes the proof.

The controller obtained in (37) and (38) can be applied into position subsystem (15) to track the desired point. Since u_x, u_y and u_z are the auxiliary input signals, one can easily obtain the expression for actual control input u_1 as:

$$u_1 = ||u_p|| = \sqrt{u_x^2 + u_y^2 + u_z^2} \qquad (48)$$

To navigate the quadrotor from given initial position to some desired position, some roll and pitch movement will take place. The expression of desired roll angle ϕ_d and pitch angle θ_d has been obtained in terms of u_x, u_y and u_z as follows:

$$\phi_d = \sin^{-1}\left(\frac{u_x \sin\psi_d - u_y \cos\psi_d}{u_1}\right) \qquad (49)$$

$$\theta_d = \tan^{-1}\left(\frac{u_x \cos\psi_d + u_y \sin\psi_d}{u_z}\right) \qquad (50)$$

B. Attitude subsystem controller design

The attitude subsystem (14) can be written as follows if $\zeta_1 = [\phi\ \theta\ \psi]^T$, $\zeta_2 = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$:

$$\begin{cases} \dot{\zeta_1} = \zeta_2 \\ \dot{\zeta_2} = f_a + g_a u_a + D_a \end{cases} \qquad (51)$$

If the desired Euler angles are $\zeta_{1d} = [\phi_d\ \theta_d\ \psi_d]^T$ then the tracking error and angular velocity error are defined respectively as:

$$e_3 = \zeta_{1d} - \zeta_1 \qquad (52)$$

$$e_4 = \zeta_2 - \xi \qquad (53)$$

where ξ is the stabilizing signal and given as

$$\xi = -\mu_1 e_3 + \dot{\zeta}_{1d} \qquad (54)$$

for $\mu_1 > 0$ where $\mu \in \mathbb{R}^{3\times3}$ is the diagonal gain matrix

By following the same procedure as given in section IV-A, the controller u_a for attitude subsystem can be given as:

$$u_a = g_a^{-1}[-\mu_2 e_4 - e_2 - f_a - \hat{D}_a - \mu_1 \dot{e}_1 + \ddot{\zeta}_{1d}] + u_a^{Robust} \qquad (55)$$

where $\mu_2 \in \mathbb{R}^{3\times3} > 0$ is the diagonal gain matrix, u_a^{Robust} is the robust controller and is obtained as:

$$u_a^{Robust} = -g_a^{-1}(\hat{\Delta}_a + \beta e_4) \qquad (56)$$

$$\text{where } \beta \in \mathbb{R}^{3\times3} > 0 \text{ and } \dot{\hat{\Delta}}_a = \eta_{a4} e_4 \qquad (57)$$

The lumped uncertainties are estimated using RNN observer as:

$$\hat{D}_a = \hat{v}_a^T \hat{H}_a(e_4, b_a, c_a, w_a) \qquad (58)$$

If the estimation error $\tilde{D}_a = D_a - \hat{D}_a = v_a^T H_a + \epsilon_a - \hat{D}_a$ where ϵ_a is the approximation error due to RNN observer then with the application of Taylor series expansion similarly as done for position subsystem, one can obtain

$$\tilde{D}_a = \tilde{v}_a^T \hat{H}_a + \hat{v}_a^T(DA_b \tilde{b}_a + DA_{c_a} \tilde{c}_a + DA_{w_a} \tilde{w}_a) + \Delta_a \qquad (59)$$

Each term in the above equation can be estimated by following the same as in (34) and (36). The overall control architecture of quadrotor is explained in Figure 3.

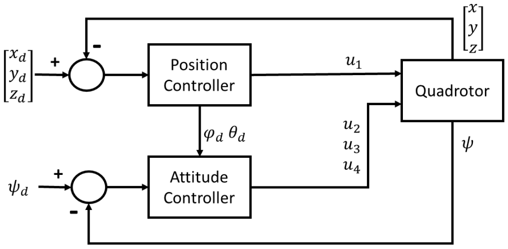


Fig. 3: Overall control of quadrotor

Theorem 2: If controller (55) along with the robust controller (56) and the RNN based uncertainty observer (58) are applied to the attitude subsystem (51) then the tracking error converges to zero asymptotically where the parameters of RNN observer are updated online as:

$$\begin{cases} \dot{\hat{v}}_a = \eta_v a^{-1} \hat{H}_a e_4^T \\ \dot{\hat{b}}_a = \eta_{a1} DA_b \hat{v} e_4 \\ \dot{\hat{c}}_a = \eta_{a2} DA_c \hat{v} e_4 \\ \dot{\hat{w}}_a = \eta_{a3} DA_w \hat{v} e_4 \end{cases} \qquad (60)$$

where $\eta_{a1}, \eta_{a2}, \eta_{a3} \in \mathbb{R}$ and $\eta_{va} \in \mathbb{R}^{3\times 1}$ are the learning rates to update the parameters of RNN.

Proof: Consider the Lyapunov function as

$$V_4 = \frac{1}{2}e_3^T e_3 + \frac{1}{2}e_4^T e_4 + \frac{1}{2}tr(\tilde{v}_a^T \eta_{v_a}^{-1} \tilde{v}_a) + \frac{1}{2\eta_{a1}}\tilde{b}_a^T \tilde{b}_a + \frac{1}{2\eta_{a2}}\tilde{c}_a^T \tilde{c}_a + \frac{1}{2\eta_{a3}}\tilde{w}_a^T \tilde{w}_a + \frac{1}{2\eta_{a4}}\tilde{\Delta}_a^T \tilde{\Delta}_a \quad (61)$$

By following the same procedure as given in Theorem 1 and with the help of (52), (53), (55), (56), (57) and (60), we get

$$\begin{aligned} \dot{V}_3 &= -e_3^T \mu_1 e_3 - e_4^T(\mu_2 + \beta)e_4 \\ &\leq -\lambda_{min}(\mu_1)||e_3||^2 - \lambda_{min}(\mu_2 + \beta)||e_4||^2 \\ \dot{V}_4 &\leq 0 \end{aligned} \quad (62)$$

Thus the overall system stability cab be guaranteed using Lyapunov stability theory and one can conclude that the tracking error converges to zero asymptotically where all the parameters of RNN observer remains bounded.

V. Simulation Results

To validate the effectiveness of the proposed controller, a regulation task is performed in simulation where the parameters of quadrotor have been taken from [15]. Figures 4 - 9 represents the position, altitude and the attitude of the quadrotor, when performing the regulation task, under the influence of the proposed controller against the classical backstepping controller. It must be noted that in both the cases, a disturbance of $0.2\sin 0.6t$ is applied to the quadrotor along with the parametric uncertainty of -20% and the control law parameters are chosen to be same. Figures 4 - 9 shows that the proposed controller performs better at tracking with smaller margins of errors as compared to the classical backstepping control. In addition, the proposed controller takes lesser effort as compared to that in the case of classical control strategy.

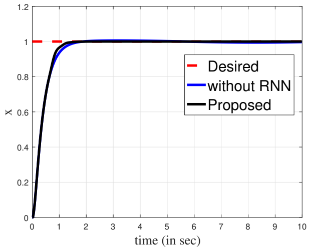


Fig. 4: x-position

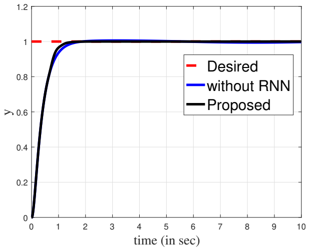


Fig. 5: y-position

Figures 10-13 represents the control effort that is required in performing the regulation task under the influence of the proposed controller as well as the classical backstepping controller. The control effort u_1, shown in Figure 10, has a larger magnitude in the classical backstepping controller as compared to that in the proposed approach. In addition, the control efforts u_2, u_3 and u_4 are almost similar in magnitude to that of the classical backstepping controller. This proves that the proposed RNN-based control strategy does not increase the burden on the actuators.

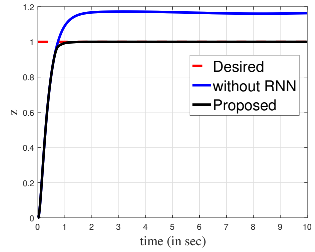


Fig. 6: z-position

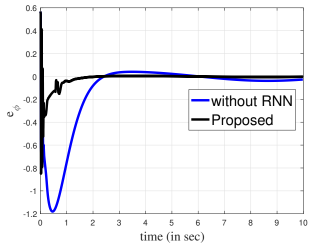


Fig. 7: Error plot of ϕ

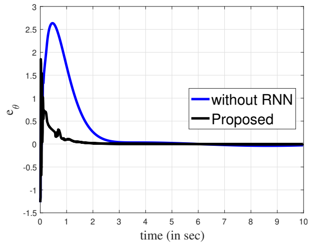


Fig. 8: Error plot of θ

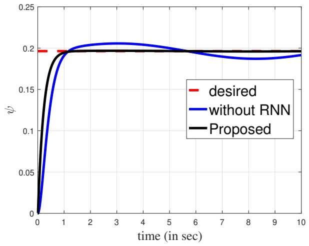


Fig. 9: Yaw angle ψ

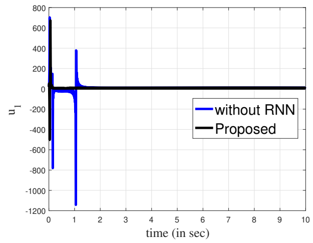


Fig. 10: Control Input u_1

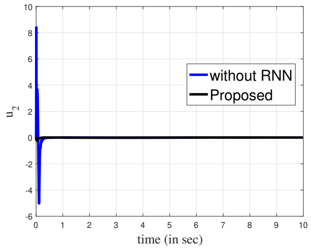


Fig. 11: Control Input u_2

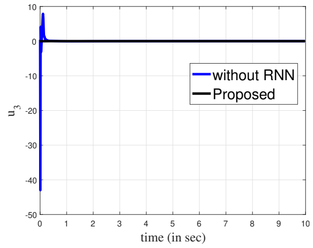


Fig. 12: Control Input u_3

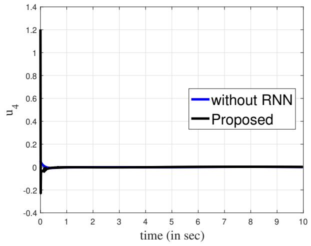
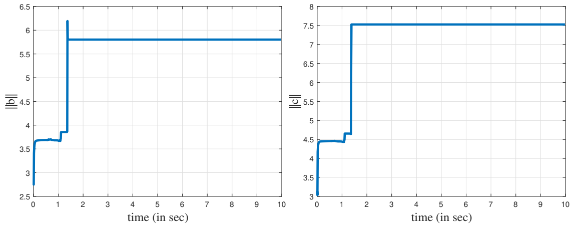


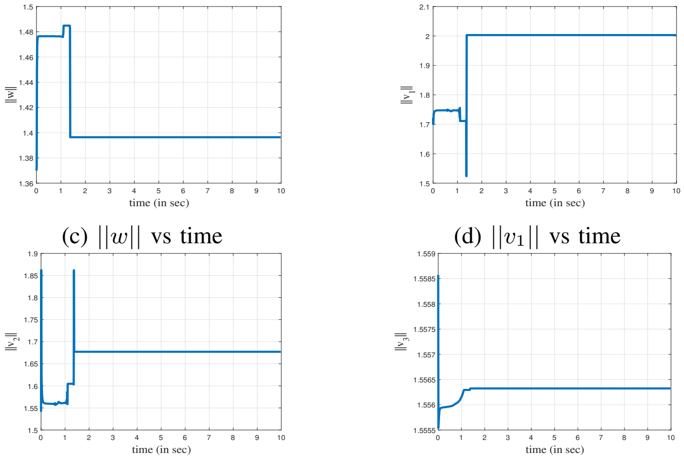
Fig. 13: Control Input u_4

Figures 14a-14f represent the evolution of the RNN weights of the attitude subsystems where $v = [v_1\ v_2\ v_3]$, the weights of position subsystem are updated in same manner as that of attitude subsystem. one can see that all the weights are bounded which ensures stability of the proposed control strategy.



(a) $||b||$ vs time

(b) $||c||$ vs time



(c) $||w||$ vs time

(d) $||v_1||$ vs time

(e) $||v_2||$ vs time

(f) $||v_3||$ vs time

Fig. 14: Weight Evolution in Attitude Subsystem

VI. Hardware Results

This section presents the results obtained with the real-time implementation of the proposed controller on the DJI Matrice 100. The DJI Matrice 100, shown in Figure 15, alongside the DJI Manifold, is capable of implementing user defined control strategies. A detailed description of the DJI Matrice 100 and the associated ROS DJI-OSDK can be found in our previous work [16]. Since the numerical simulations have already covered the regulation task, the hardware is used to validate the efficacy of the proposed strategy in a tracking task. Hence the goal of this section is to used the DJI Matrice 100 to perform a lemniscate tracking task. To establish robustness of the proposed controller against external disturbances, the tracking task is performed in an outdoor environment as shown in Figure 16. Weights of the RNN are initialized randomly between -1 and 1. Of the several trials performed, results from one such is presented in this section.

Fig. 15: DJI Matrice 100

Fig. 16: Test Environment

Figure 17 represents the 3D plot of the trajectory tracked by the quadrotor against the desired trajectory. The quadrotor was subjected to a wind disturbance of 10.26 km/hr blowing south-west. It can be noted from the Figure 17 that the proposed strategy effectively tracks the desired trajectory in the x, y and z directions. From the several trials conducted, the maximum error in tracking is found to be 18.32 cm, 23.22 cm and 17.14 cm in the x, y and z directions respectively.

Figure 18 represents the error in the euler angle tracking. Since the desired values of the euler angles are not given by the user (except yaw, which is considered to be fixed at zero here), it only makes sense to represent the error in tracking the euler angles rather than the actual euler angles in the case of a tracking task. From the Figure 18 it can be concluded that the proposed controller effectively tracks the inner loop euler angles. This feature of the controller can prove to be important in cases where the quadrotor is expected to perform acrobatic maneuvers as well.

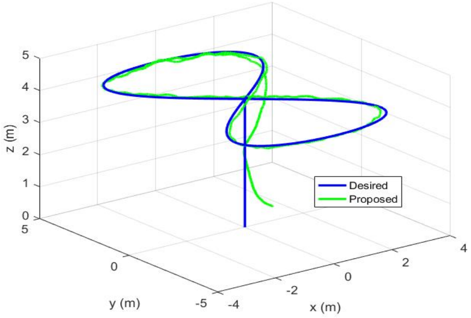


Fig. 17: 3D Trajectory

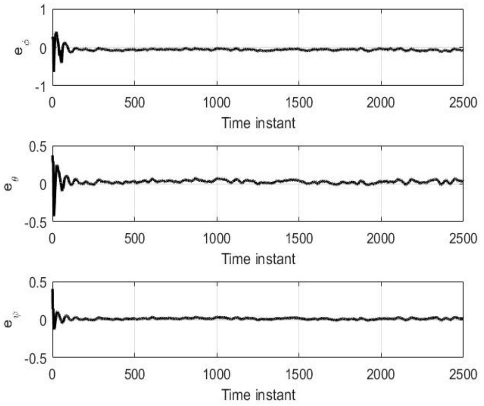


Fig. 18: Error in tracking euler angles

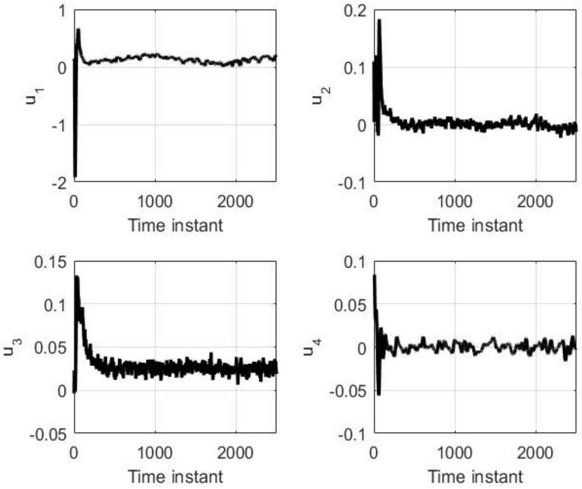


Fig. 19: Control Effort during tracking

Figure 19 represents the control effort required by the proposed control strategy to perform the tracking task. The control effort in the proposed work is almost equal in magnitude to that in our previous work [16]. This is key observation to be noted as the effect of adding the RNN observer does not significantly increase the stress on the actuators even in the presence of external disturbances.

Figures 20 represent the evolution of the RNN weights for the attitude and position subsystems when the quadrotor performs the trajectory tracking task. It should be noted that the weights are bounded which ensures efficient tracking of the quadrotor. The video of the experimental results of quadrotor tracking the lemniscate trajectory can be found in [17].

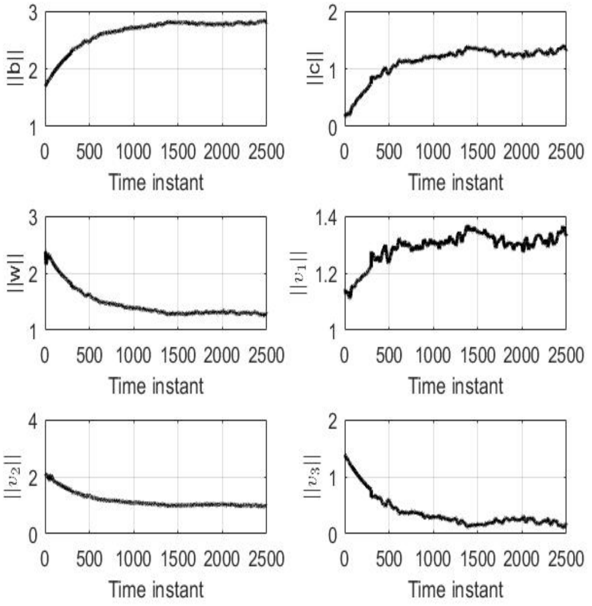


Fig. 20: Weight Evolution in Position and Altitude Subsystem

VII. Conclusion

This paper presents an adaptive backstepping control strategy with a RNN-based disturbance observer for the position and altitude tracking of a quadrotor UAV. Quadrotor dynamics are subjected to parametric uncertainties, modeling inaccuracies and bounded external disturbances. In order to tackle this problem, the paper presents an RNN based observer which helps in approximating the unknown system dynamics. In addition to this, the paper also presents a technique of making the overall control law robust to external disturbance. The parameters of the RNN are updates online using Lyapunov stability theorem which also ensures overall system stability.

To assess the efficacy of the proposed control strategy, numerical simulations are conducted using MATLAB. The numerical simulations presented are for that of a regulation task. From the graphs obtained in the associated section, one can note that the proposed controller has a superior tracking performance as compared to that of the classical backstepping control law. To validate the design in real-time, the proposed control strategy is implemented on the DJI Matrice 100 to track a lemniscate trajectory. It can be noted that the maximum error in tracking is found to be 18.32 cm, 23.22 cm and 17.14 cm in the x, y and z directions respectively. In addition a key observation that can be noted is that the proposed strategy performs tracking effectively with control efforts that are comparable to that of a FTSMSTC proposed in our previous work [16]. The video results [17] reveals that the lag caused by the online tuning of the RNN is insignificant and hence the proposed algorithm can be used for several industrial applications that involve the use of vision algorithms.

References

[1] Giusti, Alessandro, et al. ”A machine learning approach to visual perception of forest trails for mobile robots.” IEEE Robotics and Automation Letters 1.2 (2015): 661-667.

[2] S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, G. Wang, Real-time obstacle detection and tracking for sense-and-avoid mechanism in uavs, IEEE Transactions on Intelligent Vehicles 3 (2) (2018) 185-197.

[3] Lee, Daewon & Sastry, Shankar. (2009). Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. International Journal of Control, Automation and Systems, 7(3), 419-428. International Journal of Control, Automation and Systems. 7. 419-428. 10.1007/s12555-009-0311-8.

[4] Das, Abhijit, Kamesh Subbarao, and Frank Lewis. ”Dynamic inversion with zero-dynamics stabilisation for quadrotor control.” IET control theory & applications 3.3 (2009): 303-314.

[5] Raffo, Guilherme V., Manuel G. Ortega, and Francisco R. Rubio. ”Backstepping/nonlinear H_∞ control for path tracking of a quadrotor unmanned aerial vehicle.” 2008 American Control Conference. IEEE, 2008.

[6] Das, Abhijit, Frank Lewis, and Kamesh Subbarao. ”Backstepping approach for controlling a quadrotor using lagrange form dynamics.” Journal of Intelligent and Robotic Systems 56.1-2 (2009): 127-151.

[7] Tripathi, Vibhu Kumar, Laxmidher Behera, and Nishchal Verma. ”Design of sliding mode and backstepping controllers for a quadcopter.” 2015 39th National Systems Conference (NSC). IEEE, 2015.

[8] Chen, Fuyang, et al. ”Robust backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV.” IEEE Transactions on Industrial Electronics 63.8 (2016): 5044-5056.

[9] Qingtong, Wei, et al. ”Backstepping-based attitude control for a quadrotor UAV using nonlinear disturbance observer.” 2015 34th Chinese Control Conference (CCC). IEEE, 2015.

[10] Tripathi, Vibhu Kumar, Laxmidhar Behera, and Nishchal Verma. ”Disturbance observer based backstepping controller for a quadcopter.” IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2016.

[11] Madani, Tarek, and Abdelaziz Benallegue. ”Adaptive control via backstepping technique and neural networks of a quadrotor helicopter.” IFAC Proceedings Volumes 41.2 (2008): 6513-6518.

[12] Yacef, Fouad, Omar Bouhali, and Mustapha Hamerlain. ”Adaptive fuzzy backstepping control for trajectory tracking of unmanned aerial quadrotor.” 2014 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2014.

[13] Peng, Cheng, et al. ”Modeling and robust backstepping sliding mode control with Adaptive RBFNN for a novel coaxial eight-rotor UAV.” IEEE/CAA Journal of Automatica Sinica 2.1 (2015): 56-64.

[14] Mohajerin, Nima, and Steven L. Waslander. ”Modelling a quadrotor vehicle using a modular deep recurrent neural network.” 2015 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2015.

[15] F. Chen, R. Jiang, K. Zhang, B. Jiang, and G. Tao, “Robust backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV,” IEEE Trans. Ind. Electron., vol. 63, no. 8, pp. 5044–5056, Aug. 2016.

[16] Tripathi, Vibhu Kumar, et al. ”Fast Terminal Sliding Mode Super Twisting Controller For Position And Altitude Tracking of the Quadrotor.” 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

[17] Video Result https://youtu.be/xqx9hDve1uA

978-1-7281-6926-2/20/$31.00 ©2020 IEEE