International Journal of Engineering & Technology, 7 (4.35) (2018) 904-911

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Intelligent Observer-Based Feedback Linearization for Autonomous Quadrotor Control

Izzuddin M. Lazim[1], Abdul Rashid Husain[1]*, Nurul Adilla Mohd Subha[1], Mohd Ariffanan Mohd Basri[1]

[1]*Department of Control and Mechatronics Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*

**Corresponding author E-mail: rashid@fke.utm.my*

Abstract

The presence of disturbances can cause instability to the quadrotor flight and can be dangerous especially when operating near obstacles or other aerial vehicles. In this paper, a hybrid controller called state feedback with intelligent disturbance observer-based control (SF-iDOBC) is developed for trajectory tracking of quadrotor in the presence of time-varying disturbances, e.g. wind. This is achieved by integrating artificial intelligence (AI) technique with disturbance observer-based feedback linearization to achieve a better disturbance rejection capability. Here, the observer estimates the disturbances acting on the quadrotor, while AI technique using the radial basis function neural network (RBFNN) compensates the disturbance estimation error. To improve the error compensation of RBFNN, the k-means clustering method is used to find the optimal centers of the Gaussian activation function. In addition, the weights of the RBFNN are tuned online using the derived adaptation law based on the Lyapunov method, which eliminates the offline training. In the simulation experiment conducted, a total of four input nodes and five hidden neurons are used to compensate for the error. The results obtained demonstrate the effectiveness and merits of the theoretical development.

Keywords: *Disturbance Observer; Feedback Linearization; K-means clustering; Neural Network; Quadcopter.*

1. Introduction

In recent decades, there has been an increase in the applications of autonomous flying robots or popularly known as drones. This includes military and civilian applications such as for inspection, photography, aerial surveillance, delivery, and reconnaissance in unsafe environments. A common type of drone is the quadrotor that basically is a helicopter with four rotors. It has a simple structure and great maneuverability with the ability to take off and land vertically [1]. Nevertheless, the quadrotor is inherently unstable and has a complex dynamic model which brings challenges in the controller design process. In addition, it is also very susceptible to disturbances such as wind gust. Hence, a decent control algorithm is needed for the quadrotor to fly and accomplish the designated missions effectively.

A considerable large number of studies have proposed linear control design for quadrotor control, including proportional-integral-derivative (PID) [2,3] and linear quadratic regulation (LQR) [4]. To design these controllers, a linearized quadrotor model is required. A common linearization technique to obtain the linear quadrotor model is by using Jacobian linearization about an operating point (i.e. hovering condition). Although several works have shown a satisfactory capability of the controllers that are based on the Jacobian linearized model, the performance is limited to the neighborhood of the equilibrium point. As a result, other operating points may cause the quadrotor to give unsatisfactory performance and robustness.

In contrast, nonlinear control based on feedback linearization technique produces a linear model representation of the nonlinear quadrotor model over a large set of operating points [5]. This is achieved by an appropriate coordinate transformation and nonlinear feedback law. Various linear control algorithms can then be implemented in the outer-loop to stabilize the transformed linear model. It has been presented in [6–9] that feedback linearization can be used to simplify the underactuated and nonlinear quadrotor model into decoupled linear systems with the assumption that the exact model is known. However, it is almost impossible in practice to obtain the precise system representation especially when disturbances (e.g. wind) acting on the system exist.

Adaptive and robust control are the common techniques used to improve the robustness of feedback linearization, e.g. [10,11]. However, these feedback-based techniques may not react directly and fast enough in the presence of strong disturbances. Furthermore, these approaches cause the closed-loop transient response to be shaped by the adaptive or robust control components instead of the nominal linear model [12]. To overcome these drawbacks, researchers have proposed a control technique so-called active anti-disturbance control (AADC).

Unlike the adaptive or robust approaches, AADC directly counteracts the disturbances via feedforward compensation based on the measured or estimated disturbance values by disturbance observer. This has led to the popularity of disturbance observer-based control (DOBC) for disturbance rejection in various systems. In [13], feedback linearization-based controller with a high order sliding mode observer is presented for trajectory tracking of a quadrotor in the presence of sinusoidal disturbances. In [14], a disturbance observer is implemented with feedback linearization for attenuating constant external disturbances. More recently, a simulation study in [15] proposed a linear disturbance observer to improve the robustness of feedback linearization on quadrotor control. The presented disturbance observer can asymptotically estimate constant disturbances but produces a bounded estimation error when estimating time-varying disturbances. Although higher bandwidth

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

can reduce the estimation error, it also makes the system susceptible to noise and has poor transient performance [16].

In recent years, there has been an increasing interest in the artificial neural network (ANN) for control applications due to the proven approximation capabilities, e.g. [17–19]. Radial basis function neural network (RBFNN) is a type of ANN that has simple network structure that avoids unnecessary and lengthy calculation of typical multilayer feed-forward network (MLP). In [20], adaptive RBFNN is implemented to estimate mismatched uncertainties. In [21,22], RBFNN is used to approximate the unknown perturbations to improve backstepping control of a quadrotor. In short, these works have demonstrated the good approximation ability of RBFNN. However, a systematic selection of Gaussian centers which has a great influence on RBFNN approximation ability is not described in those studies.

The main objective of this paper is to control the quadrotor in the presence of constant and time-varying disturbances. Motivated by the DOBC feedback linearization in [15], this paper proposes an improved disturbance rejection using intelligent disturbance observer based control (iDOBC). Here, the estimation error produced when estimating time-varying disturbances is compensated via RBFNN. The online weight adaptation of RBFNN is obtained using Lyapunov method. In addition, the systematic steps of finding the optimal centers for RBFNN using k-means clustering algorithm is also described. The proposed iDOBC is integrated with state feedback (SF) control forming so-called SF-iDOBC for trajectory tracking control of quadrotor. To the best of our knowledge, this study is the first to improve the disturbance rejection of DOBC using RBFNN compensator.

The rest of this paper is organized as follows. In section 2, feedback linearization technique is employed to linearize the nonlinear quadrotor dynamics. Section 3 described the proposed control system design. Firstly, a nominal state feedback controller for the quadrotor trajectory tracking is presented. Then, time-domain disturbance observer for estimating the disturbances affecting the quadrotor is described. RBFNN is then presented to compensate the estimation error. Here, selection of RBFNN centers using k-means clustering technique is described. In section 4, numerical simulation results and discussion are presented with comparison with the closely related work as a benchmark. Finally, the conclusion of this paper is given in the last section.

2. Feedback Linearization of Nonlinear Quadrotor Model

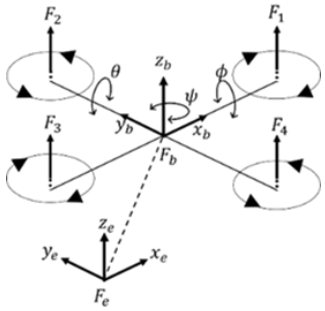


Fig. 1: Quadrotor configuration where F_1, F_2, F_3 and F_4 are the lift forces.

For the quadrotor configuration shown in Fig. 1, symbols (x, y, z) and (ϕ, θ, ψ) denote the absolute position with respect to the earth frame F_e and orientation (roll, pitch, yaw) of the quadrotor, respectively. The 6-DOF nonlinear quadrotor dynamics are given as follows [6]:

$$\ddot{x} = \alpha\, u^{[1]}/m + T_x/m \tag{1}$$

$$\ddot{y} = \beta u^{[1]}/\mathrm{m} + \mathrm{T_y}/\mathrm{m} \tag{2}$$

$$\ddot{z} = -a_g + \gamma\, u^{[1]}/m + T_z/m \tag{3}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\left(I_y - I_z\right)/I_x + u^{[2]}/I_x \tag{4}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\,(I_z - I_x)/I_y + u^{[3]}/I_y \tag{5}$$

$$\ddot{\psi} = \dot{\psi}\dot{\theta}\left(I_x - I_y\right)/I_z + u^{[4]}/I_z \tag{6}$$

Here, $\boldsymbol{u} = \left[u^{[1]}, u^{[2]}, u^{[3]}, u^{[4]}\right]^t$, m and a_g are the control inputs of the system, the mass of quadrotor, and the gravitational acceleration, respectively. Symbols I_q and T_q $(q = x, y, z)$ denote, respectively the moment of inertia along each axis and the disturbing forces on the quadrotor due to the wind, and

$$\begin{aligned} \alpha &= \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \beta &= \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ \gamma &= \cos\phi \cos\theta \end{aligned} \tag{7}$$

From Eqs. (1) – (6), it can be seen that the quadrotor is an underactuated and nonlinear system. Feedback linearization is one of the common approaches used to simplify the quadrotor dynamics. It involves the transformation of the nonlinear dynamics into the equivalent linear system via nonlinear state feedback and coordinate transformation [5]. Fig. 2 illustrates the linearization of quadrotor dynamics using feedback linearization approach. Several studies have discussed the feedback linearization of quadrotor model (e.g. [7][15]). Therefore, the linearization technique is discussed briefly in this paper.

For feedback linearizing the quadrotor dynamics, the absolute position (x, y, z) and the yaw angle (ψ) are chosen as the outputs. Then, the real control signal $\boldsymbol{u}$ is replaced by $\bar{\boldsymbol{u}} = \left[\bar{u}^{[1]}, \bar{u}^{[2]}, \bar{u}^{[3]}, \bar{u}^{[4]}\right]^t$. In this case, $u^{[1]}$ has been delayed by the double integrator, while other control signals remain unchanged [7].

$$\begin{aligned} \bar{u}^{[1]} &= \ddot{\xi}, \quad \ddot{\zeta} = \xi, \quad \zeta = u^{[1]} \\ \bar{u}^{[2]} &= u^{[2]} \\ \bar{u}^{[3]} &= u^{[3]} \\ \bar{u}^{[4]} &= u^{[4]} \end{aligned} \tag{8}$$

By using the control input $\bar{\boldsymbol{u}}$, dynamics in Eqs. (1) – (6) can be written as

$$\ddot{\mathbf{x}} = \mathbf{f} + \sum_{j=1}^{4} \mathbf{g}_j \bar{u}^{[j]} \tag{9}$$

where $\mathbf{x} = [p_x, \dot{p}_x, p_y, \dot{p}_y, p_z, \dot{p}_z, \zeta, \xi, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T \in \mathbb{R}^b$, $\mathbf{y} = [h_1, h_2, h_3, h_4]^T = [x, y, z, \psi]^T$, and

$$\mathbf{f}(\mathbf{x}) = \left[\dot{p}_x, \frac{(\alpha\zeta + T_x)}{m}, \dot{p}_y, \frac{(\beta\zeta + T_y)}{m}, \dot{p}_z, -a_g + \frac{(\gamma\zeta + T_z)}{m}, \xi, 0, \dot{\phi}, \frac{\dot{\theta}\dot{\psi}(I_y - I_z)}{I_x}, \dot{\theta}, \frac{\dot{\phi}\dot{\psi}(I_z - I_x)}{I_y}, \dot{\psi}, \frac{\dot{\psi}\dot{\theta}(I_x - I_y)}{I_z}\right]^T,$$

$$\mathbf{g}_1 = [0,0,0,0,0,0,1,0,0,0,0,0,0,0]^T,$$

$$\mathbf{g}_2 = \left[0,0,0,0,0,0,0,0,0,\frac{1}{I_x},0,0,0,0\right]^T,$$

$$\mathbf{g}_3 = \left[0,0,0,0,0,0,0,0,0,0,0,\frac{1}{I_y},0,0\right]^T,$$

$$\mathbf{g}_4 = \left[0,0,0,0,0,0,0,0,0,0,0,0,0,\frac{1}{I_z}\right]^T$$

For the nonlinear system (9), the relative degree $\{r_1, r_2, r_3, r_4\}$ is given by $r_1 = r_2 = r_3 = 4$ and $r_4 = 2$ and dimension $b = 14$. Thus, the input-output decoupling problem is solvable for the quadrotor dynamics using the following control input [7]:

$$\bar{u} = \mathbf{\Delta}^{-1}(\mathbf{x})(-\mathbf{b}(\mathbf{x}) + \mathbf{v}) \tag{10}$$

with $\mathbf{v} = [v_x, v_y, v_z, v_\psi]^T$ is the vector of control inputs for the linearized model, and

$$\mathbf{\Delta}(\mathbf{x}) = \begin{bmatrix} L_{\mathbf{g}_1} L_f^{r_1-1} h_1(\bar{x}) & \cdots & L_{\mathbf{g}_4} L_f^{r_1-1} h_1(\bar{x}) \\ \vdots & \ddots & \vdots \\ L_{\mathbf{g}_1} L_f^{r_4-1} h_4(\bar{x}) & \cdots & L_{\mathbf{g}_4} L_f^{r_4-1} h_4(\bar{x}) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\alpha}{m} & \frac{\zeta\Gamma}{mI_x} & \frac{\zeta\gamma c\psi}{mI_y} & -\frac{\zeta\beta}{mI_z} \\ \frac{\beta}{m} & -\frac{\zeta\Lambda}{mI_x} & \frac{\zeta\gamma s\psi}{mI_y} & \frac{\zeta\alpha}{mI_z} \\ \frac{\gamma}{m} & -\frac{\zeta s\phi c\theta}{mI_x} & -\frac{\zeta c\phi s\theta}{mI_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{11}$$

$$\mathbf{b}(\bar{x}) = \begin{bmatrix} L_f^{r_1} h_1(\mathbf{x}) \\ \vdots \\ L_f^{r_4} h_4(\mathbf{x}) \end{bmatrix} \tag{12}$$

where $\Gamma = \cos\phi \sin\psi - \sin\phi \sin\theta \cos\psi$ and $\Lambda = \cos\phi \cos\psi + \sin\phi \sin\theta \sin\psi$. $L_{\mathbf{g}}^{\mathbf{k}} h_i$ denotes the $\mathbf{k}$-th Lie derivative of h_i along g.

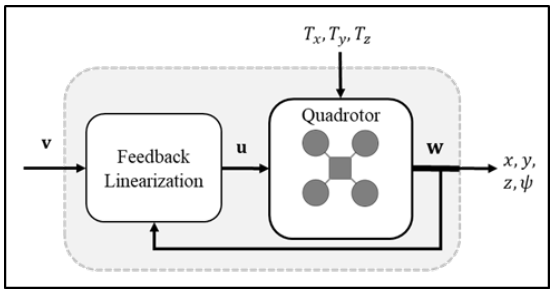


Fig. 2: Feedback linearization of the nonlinear quadrotor system.

In the absence of external disturbances (i.e. $T_g = 0$), Eq. (10) transforms Eqs. (1) – (6) into four decoupled dynamics given as

$$\frac{d(\bar{x})}{dt} = v_x, \quad \frac{d(\bar{y})}{dt} = v_y, \quad \frac{d(\bar{z})}{dt} = v_z, \quad \frac{d(\bar{\psi})}{dt} = v_\psi \tag{13}$$

On the other hand, the presence of disturbances (i.e. $T_g \neq 0$) cause the linearization to be inexact which produce the nominal part and the unknown disturbance part, $\mathbf{d} = [d_x, d_y, d_z, d_\psi]^T$ given as

$$\frac{d(\bar{x})}{dt} = v_x + d_x, \quad \frac{d(\bar{y})}{dt} = v_y + d_y, \quad \frac{d(\bar{z})}{dt} = v_z + d_z, \quad \frac{d(\bar{\psi})}{dt} = v_\psi + d_\psi \tag{14}$$

Eq. (14) can be written in the state space equation given as

$$\dot{\mathbf{w}}_i = \mathbf{A}_i \mathbf{w}_i + \mathbf{B}_i v_i + \mathbf{B}_i d_i \quad j \in \{x, y, z, \psi\} \tag{15}$$

where $\mathbf{w}_a = [q, \dot{q}, \ddot{q}, \dddot{q}]^T$ and $\mathbf{w}_\psi = [\psi, \dot{\psi}]^*$, while

$$\mathbf{A}_x = \mathbf{A}_y = \mathbf{A}_z = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{B}_x = \mathbf{B}_y = \mathbf{B}_z = [0\ 0\ 0\ 1]^T,$$

$$\mathbf{B}_\psi = [0\ 1]^T$$

It is shown in this section that the quadrotor dynamics can be transformed using feedback linearization approach into linear dynamics with unknown disturbance part. A control algorithm v_i that makes the quadrotor track the desired reference trajectory while rejecting external disturbance shall be presented in the next section.

3. Control System Design

This section presents the nominal control algorithm using state-feedback system that allows the quadrotor to track a given trajectory. Then, disturbance observer is derived to estimate the disturb-

ances affecting the quadrotor flight. Finally, RBFNN is proposed in this paper to compensate for the estimation error produced by the disturbance observer. K-means clustering method is presented to find the optimal centers of the RBFNN.

3.1. State Feedback Control for Nominal Plant

State-feedback controller is implemented to enable the quadrotor's trajectory tracking given as

$$v_{o_j} = -\mathbf{K}_j \mathbf{w}_j + G_j R_j \tag{16}$$

Here, $\mathbf{K}_j$ is the feedback gain matrix, G_j is the feedforward gain, and R_j is the desired output trajectory. The nominal controller in Eq. (16) is derived by considering ideal plant (i.e. $d_j = 0$). By substituting Eq. (16) as the input of Eq. (15), the closed-loop dynamics is given as

$$\begin{aligned}\dot{\mathbf{w}}_j &= \mathbf{A}_j \mathbf{w}_j + \mathbf{B}_j\left(-\mathbf{K}_j \mathbf{w}_j + G_j R_j + d_j\right) \\ &= \left(\mathbf{A}_j - \mathbf{B}_j \mathbf{K}_j\right)\mathbf{w}_j + G_j \mathbf{B}_j R_j + \mathbf{B}_j d_j\end{aligned} \tag{17}$$

Therefore, it can be seen that the closed-loop dynamics using nominal control in Eq. (16) is influenced by the external disturbances d_j. To remove the disturbances, this paper proposed an improved disturbance observer as presented in the next subsections.

3.2. Disturbance observer

The following time-domain disturbance observer from [15] is implemented to estimate the disturbances, d_j in Eq. (15):

$$\dot{z}_j = -\mathbf{L}_j \mathbf{B}_j\left(z_j + \mathbf{L}_j \mathbf{w}_j\right) - \mathbf{L}_j\left(\mathbf{A}_j \mathbf{w}_j + \mathbf{B}_j v_j\right), \tag{18}$$

$$\hat{d}_j = \mathbf{L}_j \mathbf{w}_j + z_j \tag{19}$$

where z_j is the internal variable of the observer, $\hat{d}_j$ is the estimated disturbance, and $\mathbf{L}_j$ is the observer gain matrix to be designed. Define the disturbance estimation error as follows:

$$\varepsilon_j = d_j - \hat{d}_j \tag{20}$$

By differentiating ε_j w.r.t. time and substituting with (15) and (18) - (20) yields

$$\begin{aligned}\dot{\varepsilon}_j &= \dot{d}_j - \dot{\hat{d}}_j \\ &= -\mathbf{L}_j \mathbf{B}_j \varepsilon_j + \dot{d}_j\end{aligned} \tag{21}$$

The observer gain matrix, $\mathbf{L}_j$ is designed as

$$\mathbf{L}_j = L_{d_j} \mathbf{B}_j^{+} \tag{22}$$

with $L_{d_j} > 0$, and $\mathbf{B}_j^{+}$ is the pseudo-inverse of the matrix $\mathbf{B}_j$. Then, (21) becomes

$$\dot{\varepsilon}_j = -L_{d_j} \varepsilon_j + \dot{d}_j \tag{23}$$

Hence, the disturbance estimation error in (23) is BIBO stable if the observer gain $L_{d_j} > 0$. The solution of the error dynamics is given as

$$\varepsilon_j = \exp^{-L_{d_j} t} \varepsilon_j(0) + \int_0^t \mathrm{e}^{-L_{d_j}(t-\tau)} \dot{d}_j(\tau) d\tau \tag{24}$$

where symbol e is the exponential function and t is the time variable. For constant disturbances (i.e. $\dot{d}_j = 0$), the estimation error ε_j converges to zero asymptotically. Meanwhile for bounded disturbance derivative $\left(\dot{d}_j(t) < \rho_j\right)$ and bounded initial estimation error $\left(\varepsilon_j(0) < \chi_j\right)$, the disturbance estimation error is also bounded $\left(\varepsilon_j < \varpi_j\right)$ for $t \geq 0$. In this paper, RBFNN compensator is proposed for compensating the bounded estimation error ε_j.

3.3. RBFNN Compensator

Because of the inability of the nominal controller in Eq. (16) to attenuate disturbances, this paper proposed a new control algorithm for disturbance compensation using disturbance observer and RBFNN described as follows:

$$v_j = v_{o_j} - \hat{d}_j - v_{nn_j} \tag{25}$$

Here, v_{o_j} is the nominal state feedback controller in Eq. (16), $\hat{d}_j$ is the estimated disturbance in Eq. (19), and v_{nn_j} is the compensation of the bounded disturbance estimation error using RBFNN which is derived in this section.
By substituting the proposed controller in Eq. (25) as the input of the linearized quadrotor model in Eq. (15), the closed-loop dynamics is given as

$$\dot{\mathbf{w}}_j = \left(\mathbf{A}_j - \mathbf{B}_j \mathbf{K}_j\right)\mathbf{w}_j + G_j \mathbf{B}_j R_j + \mathbf{B}_j\left(\varepsilon_j - v_{nn_j}\right) \tag{26}$$

A reference model representing the ideal closed-loop dynamics is introduced as follows:

$$\begin{aligned}\dot{\mathbf{w}}_{m_j} &= \mathbf{A}_j \mathbf{w}_{m_j} + \mathbf{B}_j v_{o_j} \\ &= \left(\mathbf{A}_j - \mathbf{B}_j \mathbf{K}_j\right)\mathbf{w}_{m_j} + G_j \mathbf{B}_j R_j\end{aligned} \tag{27}$$

where $\mathbf{w}_{m_j}$ is the state of the reference model. An error between the closed-loop plant and the reference model is formed when both dynamics are fed with the same reference signal, R_j. This error is defined as

$$e_j = \mathbf{w}_j - \mathbf{w}_{m_j} \in \mathbb{R}^n \tag{28}$$

By differentiating Eq. (28) and substituting with Eqs. (26) - (27), the error dynamics is given as follows:

$$\dot{e}_j = \left(\mathbf{A}_j - \mathbf{B}_j \mathbf{K}_j\right) e_j + \mathbf{B}_j\left(\varepsilon_j - v_{nn_j}\right) \tag{29}$$

Since $\mathbf{K}_j$ is chosen such that matrix $\mathbf{A}_j - \mathbf{B}_j \mathbf{K}_j$ is Hurwitz, there exists a real symmetric positive definite matrix $\mathbf{P}_j$ satisfying

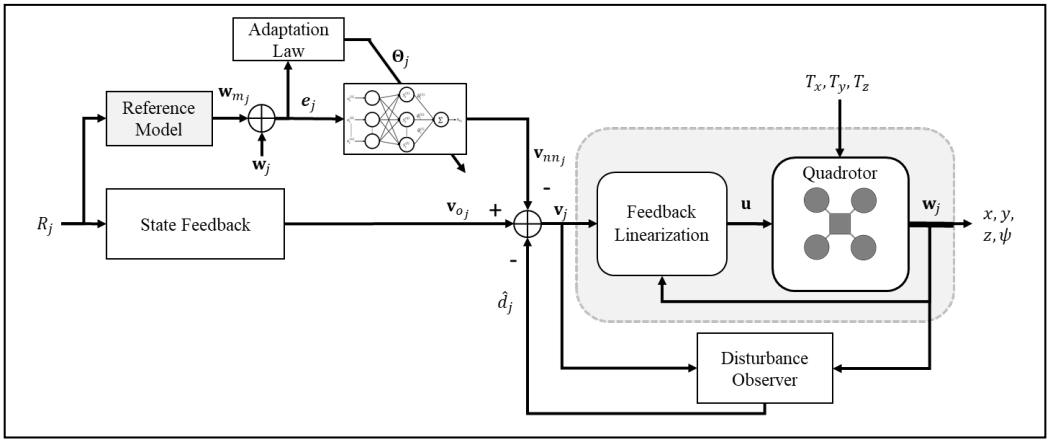


Fig. 3: Block diagram of the proposed control scheme.

$$\left(\mathbf{A}_{\bar{j}} - \mathbf{B}_{\bar{j}}\mathbf{K}_{\bar{j}}\right)^T \mathbf{P}_{\bar{j}} + \mathbf{P}_{\bar{j}}\left(\mathbf{A}_{\bar{j}} - \mathbf{B}_{\bar{j}}\mathbf{K}_{\bar{j}}\right) = -\mathbf{Q}_{\bar{j}} \tag{30}$$

where $\mathbf{Q}_{\bar{j}}$ is a positive definite matrix chosen by the designer.
In this paper, RBFNN is implemented to compensate for the unknown bounded disturbance estimation error, $\mathcal{E}_{\bar{j}} < \varpi_{\bar{j}}$. The structure of a typical RBFNN is shown in Fig. 3 with three layers given as follows [21]:

1. *Input layer*: In this layer, $\boldsymbol{e}_{\bar{j}}$ which is the error between the closed-loop plant and the reference model, is selected as the input.
2. *Hidden neuron*: The hidden layer consists of $\bar{l}$ number of neurons with the nonlinear Gaussian activation function given as

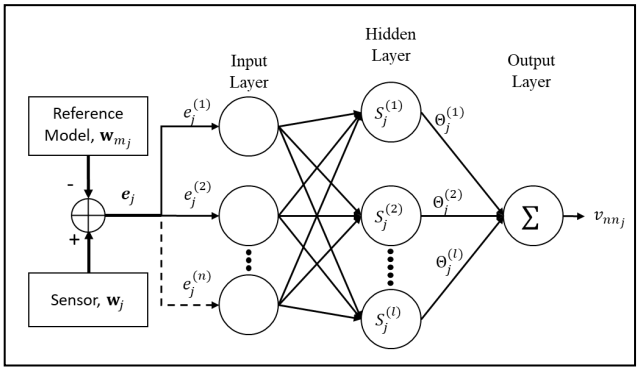


Fig. 4: Structure of the implemented radial basis function neural network (RBFNN).

$$S_{\bar{j}}^{(\bar{i})} = \exp\left(-\frac{\left\|\boldsymbol{e}_{\bar{j}} - \mu_{\bar{j}}^{(\bar{i})}\right\|}{\eta_{\bar{j}}^{(\bar{i})}}\right), \bar{i} = 1, \cdots, \bar{l} \tag{31}$$

where $\boldsymbol{\eta}_{\bar{j}} = \left[\eta_{\bar{j}}^{(1)}, \cdots, \eta_{\bar{j}}^{(\bar{l})}\right]^*$ and $\boldsymbol{\mu}_{\bar{j}} = \left[\mu_{\bar{j}}^{(1)}, \cdots, \mu_{\bar{j}}^{(\bar{l})}\right]^*$ are the width and centers of Gaussian functions, respectively.

3. *Output layer*: Output of RBFNN is the approximated disturbance estimation error, $\mathcal{E}_{\bar{j}}$ calculated using the weighted sum method given as follows:

$$v_{nn_{\bar{j}}} = \boldsymbol{\Theta}_{\bar{j}}^T \boldsymbol{S}_{\bar{j}} \tag{32}$$

with $\boldsymbol{\Theta}_{\bar{j}} = \left[\Theta_{\bar{j}}^{(1)}, \cdots, \Theta_{\bar{j}}^{(\bar{l})}\right]^* \in \mathbb{R}^{\bar{l}}$ is the weight vector.
As other AI techniques, RBFNN requires learning algorithm to achieve the designated tasks which is to find the optimal selection of the center $\mu_{\bar{i}}^{(\bar{i})}$, and weights $\Theta_{\bar{i}}^{(\bar{1})}$, $\bar{i} = 1, \ldots, \bar{l}$. The k-means clustering algorithm was used in this paper to find the optimal center $\mu_{\bar{i}}^{(\bar{i})}$. Meanwhile, Lyapunov based approach was proposed for the online weight update. These algorithms are presented in the next subsections.
Based on the universal approximation ability of RBFNN [23], there exists an optimal RBFNN to learn the unknown disturbance estimation error, $\mathcal{E}_{\bar{i}}$ over a compact region $\varpi_{\bar{i}} \in \mathbb{R}^n$ such that

$$\mathcal{E}_{\bar{i}} = \boldsymbol{\Theta}_{\bar{i}}^{*T} \boldsymbol{S}_{\bar{i}} + \epsilon_{\bar{i}} \tag{33}$$

where $\epsilon_{\bar{i}}$ is the RBFNN approximation error. Here, the optimal weight vector $\boldsymbol{\Theta}_{\bar{i}}^*$ is defined as

$$\boldsymbol{\Theta}_{\bar{j}}^* = \arg\min_{\boldsymbol{\Theta} \in \mathbb{R}^{\bar{l}}} \left\{ \sup_{\boldsymbol{e}_{\bar{j}} \in \varpi_{\bar{j}}} \left| \mathcal{E}_{\bar{j}} - v_{nn_{\bar{j}}} \right| \right\} \tag{34}$$

Assumption 1: RBFNN approximation error $\epsilon_{\bar{i}}$ is zero. This assumption will hold for comparatively less complex functions with a sufficiently large number of adjustable weight [24].

By substituting (32) and (33) into (29) yields

$$\dot{\boldsymbol{e}}_{\bar{j}} = \left(\mathbf{A}_{\bar{j}} - \mathbf{B}_{\bar{j}}\mathbf{K}_{\bar{j}}\right)\boldsymbol{e}_{\bar{j}} + \mathbf{B}_{\bar{j}}\left(\tilde{\boldsymbol{\Theta}}_{\bar{j}}^T \boldsymbol{S}_{\bar{j}} + \epsilon_{\bar{j}}\right) \tag{35}$$

where $\tilde{\boldsymbol{\Theta}}_{\bar{i}} = \boldsymbol{\Theta}_{\bar{i}}^* - \boldsymbol{\Theta}_{\bar{i}}$. The adaptation law for updating the RBFNN estimation weight vector $\boldsymbol{\Theta}_{\bar{i}}$ is given as

$$\dot{\boldsymbol{\Theta}}_{\bar{j}} = \lambda_{\bar{j}} \boldsymbol{S}_{\bar{j}} \boldsymbol{e}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \mathbf{B}_{\bar{j}} \tag{36}$$

with $\lambda_{\bar{i}} > 0$ is an adaptation gain to be designed. Block diagram of the proposed control scheme is shown in Fig. 4.

3.3.1. Stability Analysis

Theorem 1: For the nonlinear quadrotor system (1) - (6) which is linearized using (10) and yields (15), consider control law (25) with (19) and (32), and adaptation law (36). If Assumption 1 is satisfied, then, the proposed controller SF-iDOBC guarantees the disturbance rejection, or

$$\lim_{t \to \infty} \boldsymbol{e}_{\bar{j}}(t) = 0 \tag{37}$$

Proof:
Consider the following Lyapunov candidate:

$$V_{\bar{j}} = \frac{1}{2}\boldsymbol{e}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} + \frac{1}{2\lambda_{\bar{j}}} \mathrm{tr}\{\tilde{\boldsymbol{\Theta}}_{\bar{j}}^T \tilde{\boldsymbol{\Theta}}_{\bar{j}}\} \quad (38)$$

where $\mathrm{tr}\{\cdot\}$ is the trace operator of matrix algebra. Differentiating (38) w.r.t. time and substituting with (30) and (35) yields:

$$\dot{V}_{\bar{j}} = -\frac{1}{2}\boldsymbol{e}_{\bar{j}}^T \mathbf{Q}_{\bar{j}} \boldsymbol{e}_{\bar{j}} + \mathbf{S}_{\bar{j}}^T \tilde{\boldsymbol{\Theta}}_{\bar{j}} \mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} + \boldsymbol{\epsilon}_{\bar{j}}^T \mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} + \frac{1}{\lambda_{\bar{j}}} \mathrm{tr}\left\{\dot{\tilde{\boldsymbol{\Theta}}}_{\bar{j}}^T \tilde{\boldsymbol{\Theta}}_{\bar{j}}\right\} \quad (39)$$

By noting that $\mathbf{S}_{\bar{j}}^T \tilde{\boldsymbol{\Theta}}_{\bar{j}} \mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} = \mathrm{tr}\{\mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} \mathbf{S}_{\bar{j}}^T \tilde{\boldsymbol{\Theta}}_{\bar{j}}\}$, and $\dot{\tilde{\boldsymbol{\Theta}}}_{\bar{j}} = -\dot{\boldsymbol{\Theta}}_{\bar{j}}$, then Eq. (39) becomes

$$\dot{V}_{\bar{j}} = -\frac{1}{2}\boldsymbol{e}_{\bar{j}}^T \mathbf{Q}_{\bar{j}} \boldsymbol{e}_{\bar{j}} + \frac{1}{\lambda_{\bar{j}}} \mathrm{tr}\left(\lambda_{\bar{j}} \mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} \mathbf{S}_{\bar{j}}^T - \dot{\boldsymbol{\Theta}}_{\bar{j}}^T\right) \tilde{\boldsymbol{\Theta}}_{\bar{j}} + \boldsymbol{\epsilon}_{\bar{j}}^T \mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} \quad (40)$$

By substituting adaptation law in Eq. (36), Eq. (40) becomes

$$\dot{V}_{\bar{j}} = -\frac{1}{2}\boldsymbol{e}_{\bar{j}}^T \mathbf{Q}_{\bar{j}} \boldsymbol{e}_{\bar{j}} + \boldsymbol{\epsilon}_{\bar{j}}^T \mathbf{B}_{\bar{j}}^T \mathbf{P}_{\bar{j}} \boldsymbol{e}_{\bar{j}} \quad (41)$$

Therefore, based on Assumption 1, $\dot{V}_{\bar{j}} < 0$. This completes the proof.

3.3.2. K-Means Clustering for Centers Selection

To use RBFNN as a compensator, the required parameters are the weight vector, and also the centers and width of the Gaussian function in Eq. (31). In this paper, the weight vector is updated online via adaptation law in Eq. (36).
The Gaussian activation function in Eq. (31) depends on the center vector $\boldsymbol{\mu}_{\bar{j}}$ and the width value $\boldsymbol{\eta}_{\bar{j}}$. These two parameters are important in determining the effectiveness of the neural network. Generally, the Gaussian function is more sensitive when the center $\boldsymbol{\mu}_{\bar{j}}$ is close to the RBFNN input $\boldsymbol{e}_{\bar{j}}$. Meanwhile, the width of the Gaussian function $\boldsymbol{\eta}_{\bar{j}}$ represents the covering scope for the input [25].
In this paper, k-means clustering algorithm is used to find the optimal centers and width of the Gaussian which more accurately reflect the distribution of the dataset (input $\boldsymbol{e}_{\bar{j}}$).
Proposed by MacQueen [26], k-means clustering algorithm is a technique commonly used to automatically partition a dataset into k groups. The following steps describe the algorithm to find the RBFNN centers and widths offline:

Step 1) Error between the closed-loop plant and the reference model, $\boldsymbol{e}_{\bar{j}}$ is selected as the input data (referred to as the dataset) for clustering. The dataset is obtained by simulating the plant in Fig. 4 using the nominal controller $v_{\bar{j}} = v_{0_{\bar{j}}}$ for a period of time.
Step 2) Initialize k cluster centers from the randomly selected dataset obtained in step 1.
Step 3) Calculate the Euclidean distances between each of the datasets with the centers. Assign each of the datasets to the nearest cluster center.
Step 4) Once all of the datasets are assigned, calculate the mean (average) position of the datasets that are assigned to the cluster. These are the new cluster centers.
Step 5) Repeat step 3 and step 4. Stop when none of the cluster center assignment in step 3 change.
Step 6) The width of each center is obtained by averaging the distance between all dataset in the cluster and the cluster center.

4. Simulation Results and Discussion

In this section, the performance analysis of the proposed SF-iDOBC is presented. Parameters from [6] are adopted for the quadrotor model where $m = 0.6\,\mathrm{kg}$, $I_x = I_y = 0.005796\,\mathrm{kg \cdot m^2}$, $I_z = 0.010296\,\mathrm{kg \cdot m^2}$ and $a_g = 9.81\,\mathrm{ms^{-2}}$. The gains for the state feedback control law and disturbance observer are $\mathbf{K}_x = \mathbf{K}_y = \mathbf{K}_z = [16, 32, 24, 8]$, $\mathbf{K}_\psi = [1, 2]$, $G_x = G_y = G_z = 16$, $G_\psi = 1$, and $L_{d_{\bar{j}}} = 10$. To simulate the effect of disturbances on the quadrotor, Dryden wind gust model [27] is used in the simulation environment. The generated disturbance forces are shown in Fig. 5.

To compensate for the disturbance estimation error $\mathcal{E}_x$, $\mathcal{E}_y$ and $\mathcal{E}_z$, RBFNN with four inputs, five hidden neurons, and one output (4-5-1) are used respectively. Meanwhile, RBFNN with two inputs, five hidden neurons, and one output (2-5-1) is used for compensating the disturbance estimation error $\mathcal{E}_\psi$. Adaptation gain of the RBFNN is $\lambda_{\bar{j}} = 1000$, and matrix $\mathbf{P}_{\bar{j}}$ is obtained using (30) by setting $\mathbf{Q}_x = \mathbf{Q}_y = \mathbf{Q}_z = [1,1,1,1]$ and $\mathbf{Q}_\psi = [1,1]$. The centers of the Gaussian functions obtained using the k-means clustering method are given as follows:

$$\boldsymbol{\mu}_q = \begin{bmatrix} -0.2115 & 0.0143 & 0.0977 & 0.0059 & -0.3862 \\ -0.3833 & 0.0073 & -0.1273 & 0.2753 & 0.0596 \\ 0/2060 & -0.0023 & -0.1785 & -0.0818 & 0.6080 \\ 1.0322 & -0.0234 & 0.3232 & -0.7271 & -0.1306 \end{bmatrix}$$

$$\boldsymbol{\mu}_\psi = \begin{bmatrix} 0.5219 & 0.2907 & 0.1157 & 0.0026 & 0.7348 \\ -0.2773 & -0.1962 & -0.0886 & -0.0022 & -0.1791 \end{bmatrix}$$

and the corresponding widths are given as

$$\boldsymbol{\eta}_q = [0.5471 \quad 0.1043 \quad 0.2791 \quad 0.4588 \quad 0.3742]$$

$$\boldsymbol{\eta}_\psi = [0.0644 \quad 0.0566 \quad 0.0427 \quad 0.0055 \quad 0.0806]$$

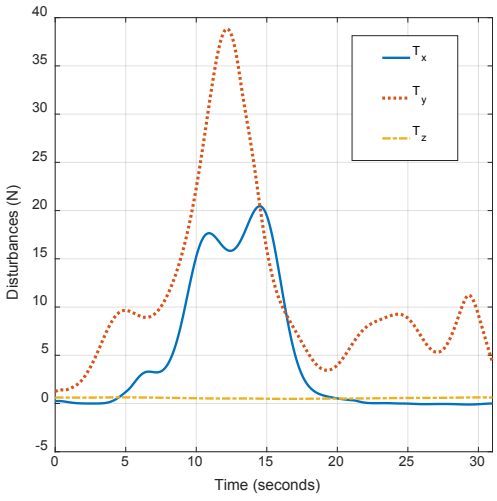


Fig. 5: Wind disturbances.

Two simulation experiments have been carried out to verify the effectiveness of the proposed controller. The performance of the controller is compared with the controller proposed in [15] using state feedback with disturbance observer (SF-DOBC). Quantitative performances are measured using integral absolute error (*IAE*) given as

$$IAE = \int_0^T |E(t)|\, dt \tag{42}$$

where T is the period of the simulation, and E is the error between the desired and actual value.

4.1 Simulation Experiment 1: Quadrotor Hovering

Hover is one of the basic maneuvers for a quadrotor. In this simulation experiment, the quadrotor is required to maintain a fixed position in the presence of time-varying wind disturbances generated by using Dryden wind gust model as shown in Fig. 5. To achieve this objective, the proposed controller in Eq. (25) is implemented on the quadrotor model. As a benchmark, performance of the quadrotor using the controller is compared with the results obtained by using SF-DOBC. The desired hovering position is set at $[R_x, R_y, R_z] = [1, -1, 2]m$ and $R_\psi = 0rad$. Results of the simulation experiments are shown in Fig. 6 – Fig. 7.

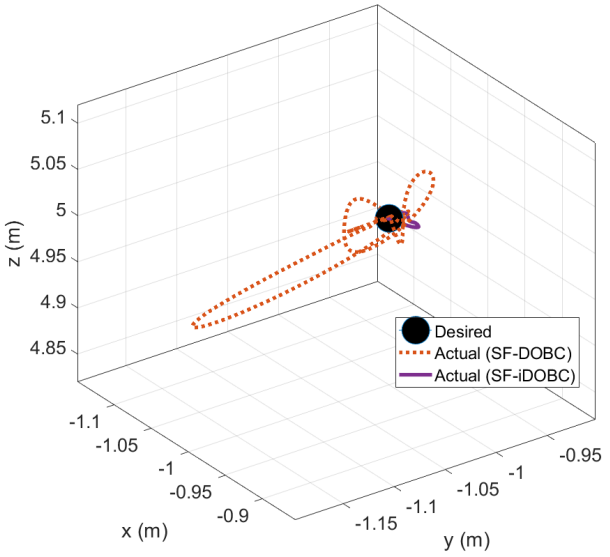


Fig. 7: Quadrotor hovering at the desired position.

In Fig. 6, the trajectories of the quadrotor using SF-DOBC and SF-iDOBC are shown. Clearly, the quadrotor can maintain the desired position using SF-DOBC and SF-iDOBC. However, quadrotor with SF-DOBC shows a larger deviation from the hovering position as it cannot reject the wind disturbances quite well. On the other hand, quadrotor with SF-iDOBC shows only a slight deviation from the desired position. This indicates the capability of the proposed controller to reject the disturbances as verified by quantitative analysis using *IAE* tabulated in Table 1. Notice that the performance of both controllers for heading (ψ) control was identical as shown in Fig. 7. This is because of no aerodynamic moment disturbance is considered.

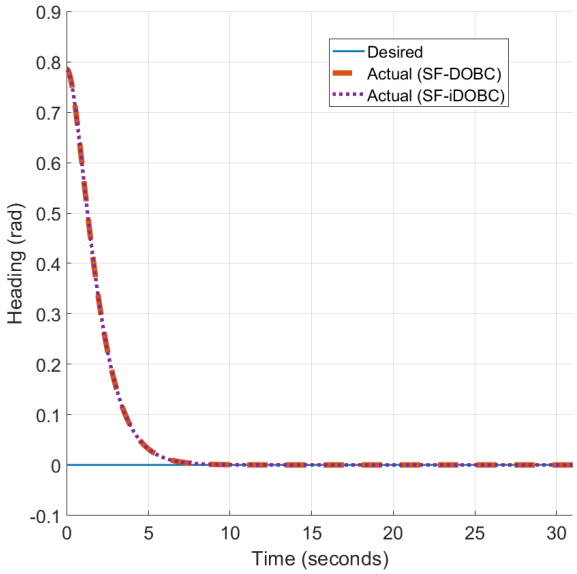


Fig. 8: Time response of quadrotor heading.

Table 1: Quantitative comparison between SF-DOBC and SF-iDOBC for quadrotor hovering problem.

Controller	*IAE*			
	x	y	z	ψ
SF-DOBC	0.374	0.794	0.177	1.571
SF-iDOBC	0.038	0.058	0.027	1.571
Reduction (%)	89.839	92.695	84.746	0.000

4.2 Simulation Experiment 2: Quadrotor Trajectory Tracking

To further verify the proposed controller capability, a time-varying trajectory tracking of the quadrotor is presented. Here, the practical ability of the quadrotor to track the desired time-varying trajectory while rejecting wind disturbances is emphasized. The

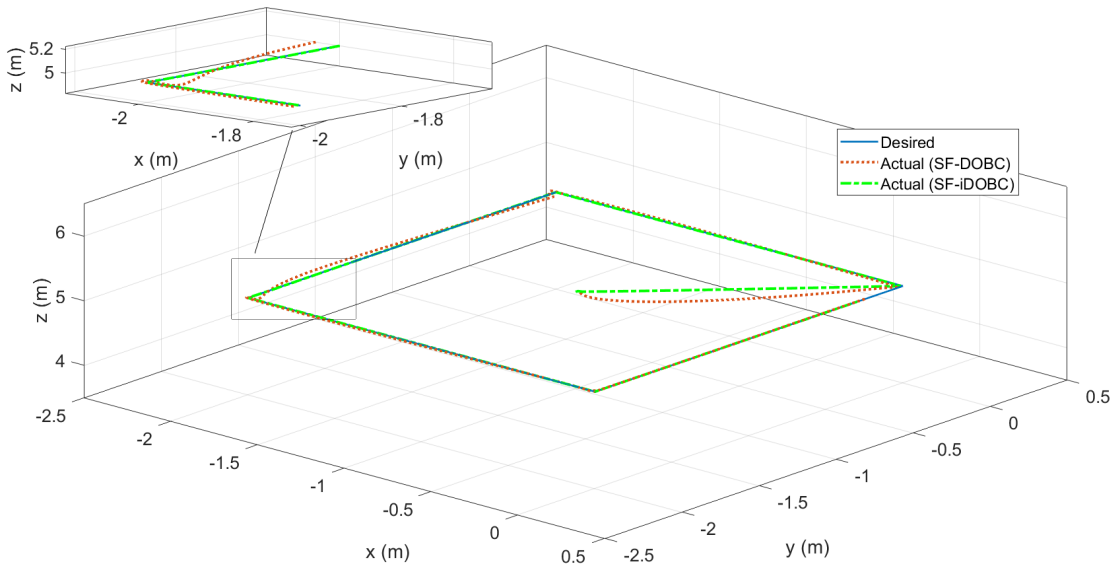


Fig. 6: Position trajectory of the quadrotor using SF-DOBC and the proposed SF-iDOBC.

wind disturbances are generated using Dryden wind model in Fig. 5. From the initial position at $[x(0), y(0), z(0)] = [-1, -1, 5]$, the quadrotor needs to track a trajectory given as follows:

$$[x(t), y(t), z(t)] = \begin{cases} [0,0,5] & \text{for } 0 < t \leq 5 \\ [-3,0,5] & \text{for } 5 < t \leq 13 \\ [-3,-3,5] & \text{for } 13 < t \leq 21 \\ [0,-3,5] & \text{for } 21 < t \leq 29 \\ [0,0,5] & \text{for } 29 < t \leq 31 \end{cases}$$

Fig. 8 - Fig. 9 show the simulation results of the quadrotor trajectory tracking in the presence of wind disturbances using SF-DOBC and SF-iDOBC, respectively.
In Fig. 8, the quadrotor trajectory using SF-DOBC has some deviation from the desired trajectory caused by the wind disturbances. In contrast, quadrotor trajectory using SF-iDOBC is smoother and has less deviation from the desired trajectory as shown in Fig. 9. This result further verifies the capability of the proposed controller over the closely related controller (SF-DOBC) as supported by quantitative performance analysis using *IAE* index tabulated in Table 2.

Table 2: Quantitative comparison between SF-DOBC and SF-iDOBC for quadrotor trajectory tracking problem.

Controller	*IAE*		
	x	y	z
SF-DOBC	14.261	13.501	0.179
SF-iDOBC	13.985	12.727	0.039
Reduction (%)	1.935	5.733	78.212

5. Conclusion

The aim of this paper is to control the quadrotor in the presence of time-varying disturbances, e.g. wind. An improved iDOBC is developed, where the RBFNN is proposed to compensate the bounded estimation error produced by the standard disturbance observer. The optimal centers for RBFNN's Gaussian function are found offline using the k-means clustering algorithm. Meanwhile, the weight vector of the RBFNN is tuned online using the derived adaptation law. It has been proven via a Lyapunov function that the developed control strategy guarantees the stability of the system. Finally, the effectiveness of the proposed SF-iDOBC is verified by some simulation results. It is found that the quadrotor with SF-iDOBC performs better in the presence of wind disturbances as compared with the benchmark. The possibility of a real-time implementation of the designed controller will be further investigated in the future.

Acknowledgment

The first author would like to thank the Ministry of Education (MOE) and Universiti Sains Islam Malaysia (USIM) for their financial support under SLAB scheme, and all the authors acknowledge the support by Universiti Teknologi Malaysia (UTM) in term of lab facilities. The project is supported under the Research University Grant No. Q.J130000.2608.14J62, Universiti Teknologi Malaysia.

References

[1] I.M. Lazim, A.R. Husain, N.A.M. Subha, Z. Mohamed, M.A. Mohd Basri, Optimal Formation Control of Multiple Quadrotors Based on Particle Swarm Optimization, in: Asian Simul. Conf., Springer, Singapore, 2017: pp. 121–135.
[2] A.L. Salih, M. Moghavvemi, H.A.F. Mohamed, K.S. Gaeid, Modelling and PID controller design for a quadrotor unmanned air vehicle, in: 2010 IEEE Int. Conf. Autom. Qual. Test. Robot., 2010: pp. 1–5.
[3] T. Sangyam, P. Laohapiengsak, W. Chongcharoen, I. Nilkhamhang, Path tracking of UAV using self-tuning PID controller based on fuzzy logic, in: Proc. SICE Annu. Conf. 2010, 2010: pp. 1265–1269.
[4] A. Dharmawan, T.K. Priyambodo, others, Model of Linear Quadratic Regulator (LQR) Control Method in Hovering State of Quadrotor, J. Telecommun. Electron. Comput. Eng., 9 (2017) 135–143.
[5] M.A. Henson, D.E. Seborg, Feedback Linearizing Control, in: Nonlinear Process Control, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997: pp. 149–232.
[6] A. Mahmood, Y. Kim, Leader-following formation control of quadcopters with heading synchronization, Aerosp. Sci. Technol., 47 (2015) 68–74.
[7] V. Mistler, A. Benallegue, N.K. M'Sirdi, Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback, in: Proc. 10th IEEE Int. Work. Robot Hum. Interact. Commun., 2001: pp. 586–593.
[8] F. Sabatino, Quadrotor control : modeling, nonlinear control design, and simulation, KTH Royal Institute of Technology, 2015.
[9] A. Mahmood, Y. Kim, Decentralized formation control of quadcopters using feedback linearization, in: 2015 6th Int. Conf. Autom. Robot. Appl., 2015: pp. 537–541.
[10] I.H. Choi, H.C. Bang, Quadrotor-tracking controller design using adaptive dynamic feedback-linearization method, Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng., 228 (2014) 2329–2342.
[11] W. Zhao, T.H. Go, Quadcopter formation flight control combining MPC and robust feedback linearization, J. Franklin Inst., 351 (2014) 1335–1355.
[12] L.B. Freidovich, H.K. Khalil, Performance Recovery of Feedback-Linearization-Based Designs, IEEE Trans. Automat. Contr., 53 (2008) 2324–2334.
[13] A. Benallegue, A. Mokhtari, L. Fridman, Feedback linearization and high order sliding mode observer for a quadrotor UAV, in: Int. Work. Var. Struct. Syst. 2006. VSS'06., 2006: pp. 365–372.
[14] A. Mokhtari, N.K. M'Sirdi, K. Meghriche, A. Belaidi, Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle, Adv. Robot., 20 (2006) 71–91.
[15] A. Aboudonia, A. El-Badawy, R. Rashad, Disturbance observer-based feedback linearization control of an unmanned quadrotor helicopter, Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng., 230 (2016) 877–891.
[16] J.S. Bay, Fundamentals of Linear State Space Systems, WCB/McGraw-Hill, 1999.
[17] M. Sharma, A.J. Calise, Neural-Network Augmentation of Existing Linear Controllers, J. Guid. Control. Dyn., 28 (2005) 12–19.
[18] S.S.A. Ali, M. Moinuddin, K. Raza, S.H. Adil, An Adaptive Learning Rate for RBFNN Using Time-Domain Feedback Analysis, Sci. World J., 2014 (2014).
[19] H. Sun, L. Hou, Y. Li, Disturbance observer based dynamic surface tracking control for a class of uncertain nonlinear systems with mismatched disturbances, in: 2016 12th World Congr. Intell. Control Autom., 2016: pp. 605–610.
[20] M.M. Arefi, M.R. Jahed-Motlagh, H.R. Karimi, Adaptive Neural Stabilizing Controller for a Class of Mismatched Uncertain Nonlinear Systems by State and Output Feedback, IEEE Trans. Cybern., 45 (2015) 1587–1596.
[21] M.A.M. Basri, K.A. Danapalasingam, A.R. Husain, Intelligent adaptive backstepping control for MIMO uncertain non-linear quadrotor helicopter systems, Trans. Inst. Meas. Control, 37 (2015) 345–361.
[22] M.A.B.M. Basri, Intelligent Backstepping Control of Quadrotor Unmanned Aerial Vehicle, Universiti Teknologi Malaysia, 2015.
[23] T. Chen, H. Chen, Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks, IEEE Trans. Neural Networks, 6 (1995) 904–910.
[24] L. Behera, I. Kar, Intelligent Systems and Control: Principles and Applications, Oxford University Press, New Delhi, India, 2010.
[25] J. Liu, Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation, Springer Science & Business Media, 2013.
[26] J. MacQueen, others, Some methods for classification and analysis of multivariate observations, in: Proc. Fifth Berkeley Symp. Math. Stat. Probab., 1967: pp. 281–297.
[27] S. Waslander, C. Wang, Wind Disturbance Estimation and Rejection for Quadrotor Position Control, in: AIAA Infotech@aerosp. Conf., Seattle, WA, 2009.