International Journal of Control, Automation and Systems 16(5) (2018) 2458-2472
http://dx.doi.org/10.1007/s12555-017-0099-x
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Decentralized Backstepping Control of a Quadrotor with Tilted-rotor under Wind Gusts

Abdul-Wahid A. Saif*, Abdulrahman Aliyu, Mujahed Al Dhaifallah, and Moustafa Elshafei

Abstract: Conventional unmanned aerial vehicles, quadrotor have a plethora of applications for civilian and military purposes. Quadrotors as the name implies usually have four input variables (fixed rotors) which are used to drive six outputs (i.e., 3 translational and 3 rotational motions), and this leads to coupling between motions. Tilt- rotor quadrotors are more versatile because they have more input variables to independently control its orientation and position without coupling. In this paper, a decentralized backstepping control approach is used to generate a new set of inputs capable of independently and simultaneously achieve decoupling of motions while rejecting wind disturbances. The tiltrotor quadrotor dynamic is first decentralized to achieve six subsystems, then controller inputs for each subsystem are generated via Lyapunov based backstepping method whereby the controller parameters are optimized by Differential Evolution (DE) technique. This system exhibits robustness capability because it is able to reject external disturbances.

Keywords: Backstepping control, decentralization, robotics technology, UAVs, wind gusts.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have been extensively used for various civilian and military purposes. Examples of such applications are pick and place objects, delivery, surveillances, traffic monitoring, rescue missions, patrolling forests in case of fire outbreak, warfare, and other risky missions.

A comprehensive history on the evolution of UAVs is presented in [1]. Quadrotor UAVs, have attracted more attention because of its compact nature and versatility. Modeling of conventional quadrotors exists in numerous literature mainly based on Euler-Lagrange or Newton-Euler formalism. These modeling techniques presented in [1–4], have different control schemes to achieve various flight modes. For instance, in [1], a PID, LQR, SMC and backstepping methods are used to control the quadrotor orientations. In [2], a quadrotor, carrying payloads was controlled by PID controllers. In [3], feedback linearization technique was used to decompose the quadrotor dynamics into nested loops where, the inner loop was used for attitude control while the outer loop for position control. In [4], a hybrid backstepping and feedback linearization control via visual feedback are used.

In spite of the milestone recorded with the conventional quadrotor, its model structure still possesses some limitations regarding the orientation angles (roll, pitch and yaw) and position (x, y, z) coupling because of insufficient inputs. Typically, the outputs are either the position together with yaw angle for tracking purposes or orientation angle together with altitude for vertical takeoff and landing (VTOL), hence making it incongruous for some particular tasks. In light of this, different modeling and control schemes have been employed to its structure so as to complement such deficiencies. In order to address the flaws associated with the conventional quadrotors, tilt-wing and tilt-rotor quadrotors evolved as in [5–8] thereby improving the actuation capacity of the conventional quadrotor.

In [5], a tilt-rotor quadrotor was used to perform a "tilting on the spot" flight mode by introducing four additional inputs in a bid to decouple motion, but the quadrotor body is tilted as well. This was achieved using an output feedback linearization control technique. In [6], the rotors of a quadrotor are placed on a tilt-wing with the aim of achieving a hovering and airplane flight mode. Where, an LQR controller was utilized to achieve this object. In [7], twelve inputs model was introduced to achieve different flight modes and to showcase fault tolerance capability using PID controllers. In [8], an adaptive control technique was applied to a tilt-roll quadrotor structure. A cascade

Manuscript received February 24, 2017; revised November 25, 2017 and April 5, 2018; accepted May 14, 2018. Recommended by Associate Editor Yang Tang under the direction of Editor Hyun-Seok Yang. The authors would like to thank DSR at KFUPM for the support of this work under the project No. IN141048.

Abdul-Wahid A. Saif, Abdulrahman Aliyu, Mujahed Al Dhaifallah, and Moustafa Elshafei are with the Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, 31261 KSA (e-mails: awsaif@kfupm.edu.sa, g201306010@kfupm.edu.sa, mujahed@kfupm.edu.sa, elshafei@kfupm.edu.sa). Mujahed Al Dhaifallah is on loan with College of Engineering at Wadi Aldawaser, Prince Sattam bin Abdulaziz University, KSA.
* Corresponding author.

©ICROS, KIEE and Springer 2018

PID was first used to test the quadrotor performance. In [9], a thrust tilting approach was also introduced. Hitherto, additional inputs have only been successfully aimed at decoupling the quadrotor orientation either to achieve forward flight mode or hovering as the case may be. This is particularly useful in applications requiring high maneuverability.

External disturbances are inevitable and are sometimes referred to as constants in some research papers, for instance in [10], however, the increase in wind speed is usually neglected. This may lead to misleading conclusions about the flight accuracy. However, atmospheric disturbances, particularly wind gust models, incorporated in the modelling dynamics reflect a more realistic approach. For conventional quadrotors, various wind modelling techniques have been investigated in [11–13]. In [11] and [12], a PID controller and decentralized PID neural network controller respectively, were used to reject wind disturbance derived from the Dryden wind modelling technique. In [13], a macro approach to wind modeling was introduced to study the effect of wind disturbance on a quadrotor. In [14] ducted fans were used to generate wind gust in an experimental setup and a switching MPC controller was utilized to achieve some robustness against such disturbance.

In [15], backstepping control was compared to sliding mode control technique on a conventional quadrotor, which shows that the former is able to better control orientation angles in the presence of high perturbations. Backstepping control has also been used in some other research works whereby the quadrotor model equations are broken into subsystems as in [16–18]. This is intuitive because the conventional quadrotor can now be directly driven by four available inputs. However, selecting the gains for backstepping controller gives commendable results, but using optimization techniques are better. For instance, in [19] for conventional quadrotor whereby the controller parameters gotten from Particle Swarm Optimization (PSO) were proven to produce better results when compared to GA. In [20], the backstepping controller was used to control the quadrotor under external disturbance which was considered as a constant in this case. However, in [21], a robust optimal controller was introduced to control a hexarotor structure in which a time varying wind disturbance was considered. In [22] a robust control technique based on disturbance observer was used where the disturbance included some nonlinearities and Dryden wind model was considered an external disturbance.

The effect of wind disturbance on a tilt-rotor quadrotor under various orientation flight modes is an important mission but yet to be investigated. However, the conventional quadrotor is incapable of completely achieving this objective because, for instance, it cannot hover at a tilt angle on the spot. In this paper, a decentralized backstepping controller is used to precisely control the tilt-rotor quadrotor orientation and position individually under wind disturbance whereby a more robust optimization technique is used to derive the controller gains. This decentralization method was applied in [23] for a class of nonlinear MIMO systems, but the controller parameters are empirically selected. In [24], the same approach was only used to control conventional quadrotors but the wind disturbance was not considered. In [25, 26], feedback linearization control of quadrotor with tiltable rotors and wind disturbance was presented.

In [27], optimization techniques, Differential Evolution (DE), PSO and Evolutionary Algorithms (i.e., GA), underwent a comparative study, over 34 widely used benchmark problems. Also, in [28], DE, PSO, and GA were compared for hard clustering problems. In all cases, the results show that Differential Evolution (DE) consistently outperforms PSO and GA. In [29], DE was also proven to surpass PSO when compared over twelve constraint nonlinear test functions. DE sets a more outstanding results in addition to its simplicity, robustness, convergence time and finds optimum values in almost every run. Recently, PSO was used in [30] for PID tuning of conventional quadrotor but in [31], DE, PSO and Artificial Bee Colony (ABC) algorithm were compared for PID tuning. Although DE surpassed PSO, ABC exhibited some unique characteristics.

This is the motivation behind choosing the DE as the optimization technique to derive our controller parameters where the wind disturbance model used in this work exhibits some advantages over the regularly used Drysden wind model as would be discussed later.

The paper is organized as follows: Section 2 discusses the dynamic model of tilt rotor quadrotor. Section 3 presents the decentralization of the tiltrotor quadrotor while Section 4 gives the wind gust modelling technique based on macro approach. In Section 5, the backstepping control methodology applied is given. Section 6 gives the simulation for different test cases and advantages of the quadrotor design. Section 7 summarizes the conclusion of this work.

2. DYNAMIC MODELING

For a realistic simulation, mathematical model of a quadrotor is essential for good controller design. The mathematical model considered in this work is a modified version of that represented in [7], whereby the motion have been decoupled. The following assumptions applies;

- Translation dynamics is expressed with respect to a fixed world coordinate frame, while the rotational dynamic is expressed with respect to body fixed frame for the purpose of simplicity.

- Body frame origin and center of gravity are assumed to be coincident.

- Air friction and drag moment together with external wind forces are considered.

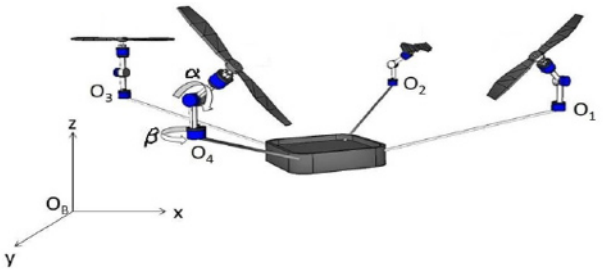

Fig. 1. Quadrotor structure.

- Structure is assumed inflexible and symmetric.
- Propellers are assumed to be rigid (no blade flapping).
- The rotors are located at points O1, O2, O3, and O4, tilted with respect to the fixed rotor points as shown in Fig. 1.
- These rotor frames are taken parallel to the body fixed reference frame at the center of gravity.
- The translational and rotational dynamic equations are established according to Newton-Euler formation.

Let $R^B_{r_i}$ represent the orientation rotor axis Oi with respect to the fixed rotor body frame. By denoting α_i, the rotational angle about y_i and β_i, the rotational axes about z_i as shown in above Fig. 1. Then the rotational matrix from the rotors-rotating frame to the fixed rotor frame is given by:

$$R^B_{r_i} = \begin{bmatrix} c\beta_i c\alpha_i & -s\beta_i & c\beta_i s\alpha_i \\ s\beta c\alpha_i & c\beta_i & s\beta_i s\alpha_i \\ -s\alpha_i & 0 & c\alpha_i \end{bmatrix}, \tag{1}$$

where c denotes cosine and s denotes sine. The rotor thrust (L_i) and moments (T_d) at the center of gravity is proportional to square of the rotor speed (w). Typically,

$$L_i = bw_i^2, \quad T_{di} = dw_i^2,$$

where b and d are the lift and drag constants respectively. For each rotor thrust, F_i is therefore given by:

$$F_i = \begin{bmatrix} c\beta_i c\alpha_i & -s\beta_i & c\beta_i s\alpha_i \\ s\beta c\alpha_i & c\beta_i & s\beta_i s\alpha_i \\ -s\alpha_i & 0 & c\alpha_i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ bw_i^2 \end{bmatrix}. \tag{2}$$

The moments M_i, consists of the drag moment M_{iA} and moments generated by the thrust component M_{iB} given by:

$$\begin{aligned} M_i =& M_{iA} + M_{iB} \\ =& \begin{bmatrix} c\beta_i c\alpha_i & -s\beta_i & c\beta_i s\alpha_i \\ s\beta c\alpha_i & c\beta_i & s\beta_i s\alpha_i \\ -s\alpha_i & 0 & c\alpha_i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ d\omega_i^2.\delta(i) \end{bmatrix} + r_i x F_i. \end{aligned} \tag{3}$$

$\delta(i) = [1,1,-1,-1]$ accounts for the direction of rotation to each rotor (i.e., rotors 1 and 2 rotates counter clockwise, and rotors 3 and 4 rotates clockwise) and r_i represents the vector from center of gravity (CoG) to the reference point of the rotors given by:

$$\begin{aligned} r_1 &= [l,0,-h], \quad r_2 = [0,l,-h], \\ r_3 &= [-l,0,-h], \quad r_4 = [0,-l,-h]. \end{aligned} \tag{4}$$

l and h respectively represents the horizontal and vertical displacements from the rotors to CoG.

The translational dynamic equation established in the reference earth frame is given by:

$$m\ddot{r} = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T - F_a + F + D, \tag{5}$$

where F_a represents the aerodynamic drag, F represents the thrust force and D may represent any disturbance and $[0 \quad 0 \quad -mg]T$ represents acceleration due to gravity:

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} - \begin{bmatrix} K_1\dot{x} \\ K_2\dot{y} \\ K_3\dot{z} \end{bmatrix} + R\sum_{i=1}^{4} F_i + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}. \tag{6}$$

For small objects, air is approximately proportional to velocity and K_1, K_2 and K_3 [27], are the drag coefficients. d_1, d_2, and d_3 are the disturbances components of wind gust that will be discussed later. R is the body Euler transformation matrix with respect to the earth inertia frame given by:

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi s\theta & c\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix}, \tag{7}$$

where the roll angle, φ is rotation about x axis, pitch angle, θ is rotation about y axis and yaw angle [3], ψ is rotation about z axis.

The rotational dynamic established in the reference body frame is given by:

$$\ddot{\Omega} = \mathrm{I}^{-1}[(-\dot{\Omega} \times \mathrm{I}\dot{\Omega}) - M_g - M_f + M + M_d], \tag{8}$$

where $\Omega^T = [\varphi \quad \theta \quad \psi]$, M_d represents a random disturbance moment, M_f is the drag/friction moments with K_4, K_5 and K_6representing the drag coefficients [27], and is given by:

$$M_d = \begin{bmatrix} m_{dp} & m_{dq} & m_{dr} \end{bmatrix}^T, \tag{9}$$

$$M_f = \begin{bmatrix} K_4\dot{\varphi} & K_5\dot{\theta} & K_6\dot{\psi} \end{bmatrix}^T. \tag{10}$$

I, the body inertia matrix and M_g the gyroscopic forces are respectively given by:

$$\mathrm{M}_g = \mathrm{I}_R \sum_{i=1}^{4} (\Omega x \overline{\omega_i})\delta(i);$$

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}, \tag{11}$$

I_R; rotor inertia and

$$\overline{\omega_i} = \begin{bmatrix} c\beta_i\alpha_i & -s\beta_i & c\beta_i s\beta_i \\ s\beta c\alpha_{i_i} & c\beta_i & s\beta_i s\beta_i \\ -s\alpha_i & 0 & c\beta_i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_i \end{bmatrix}, \tag{12}$$

$$M = \sum_{i=1}^{4} \mathrm{M}_i. \tag{13}$$

The equations of motion can be represented as:

$$\dot{X} = f(X,U), \tag{14}$$

where

$$\begin{aligned} X &= \left[x, \dot{x}, y, \dot{y}, z, \dot{z}, \varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}\right]; \\ U &= [w_1, \alpha_1, \beta_1, w_2, \alpha_2, \beta_2, w_3, \alpha_3, \beta_3, w_4, \alpha_4, \beta_4]. \end{aligned} \tag{15}$$

Therefore,

$$\dot{X} = \begin{pmatrix} x_2 \\ \begin{pmatrix} 1/m((cx_{11}cx_9)u_1 \\ +(-sx_{11}cx_7 + cx_{11}sx_9sx_7)u_2 \\ +(sx_{11}sx_7 + cx_{11}sx_9cx_7)u_3 \\ -K_1x_2 + d_1) \end{pmatrix} \\ x_4 \\ \begin{pmatrix} 1/m((sx_{11}cx_9)u_1 \\ +(cx_{11}cx_7 + sx_{11}sx_9sx_7)u_2 \\ +(-cx_{11}sx_7 + sx_{11}sx_9cx_7)u_3 \\ -K_2x_4 + d_2) \end{pmatrix} \\ x_6 \\ \begin{pmatrix} -g + 1/m((-sx_9)u_1 \\ +(cx_9sx_7)u_2 + (cx_9cx_7)u_3 \\ -K_3x_6 + d_3) \end{pmatrix} \\ x_8 \\ \begin{pmatrix} x_{10}x_{12}I_1 + 1/I_{uu}(u_4 - I_R(x_{12}S_2 + x_{10}S_3) \\ -K_4x_8 + m_{dp}) \end{pmatrix} \\ x_{10} \\ \begin{pmatrix} x_8x_{12}I_2 + 1/I_{vv}(u_5 - I_R(x_{12}S_1 - x_8S_3) \\ -K_5x_{10} + m_{dq}) \end{pmatrix} \\ x_{12} \\ \begin{pmatrix} x_8x_{10}I_3 + 1/I_{ww}(u_6 - I_R(-x_{10}S_1 + x_8S_2) \\ -K_6x_{12} + m_{dr}) \end{pmatrix} \end{pmatrix}, \tag{16}$$

where

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{i=4} c\beta_i s\alpha_i b w_i^2 & \sum_{i=1}^{i=4} s\beta_i s\alpha_i b w_i^2 & \sum_{i=1}^{i=4} c\alpha_i b w_i^2 \end{bmatrix}^T;$$

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{i=4} c\beta_i s\alpha_i \omega_i \delta(i) & \sum_{i=1}^{i=4} s\beta_i s\alpha_i \omega_i \delta(i) \\ \sum_{i=1}^{i=4} c\alpha_i \omega_i \delta(i) \end{bmatrix}^T,$$

$$\begin{bmatrix} u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} \left(\begin{array}{l} \sum_{i=1}^{4} c\beta_i s\alpha_i d\omega_i{}^2 \delta(i) - b\omega_1{}^2(-s\beta_1 s\alpha_1 h) \\ \quad - b\omega_2{}^2(-c\alpha_2 l - s\beta_2 s\alpha_2 h) \\ \quad - b\omega_3{}^2(-s\beta_3 s\alpha_3 h) \\ \quad - b\omega_4{}^2(-c\alpha_4 l - s\beta_4 s\alpha_4 h) \end{array} \right) \\ \left(\begin{array}{l} \sum_{i=1}^{4} s\beta_i s\alpha_i d\omega_i{}^2 \delta(i) \\ \quad - b\omega_1{}^2(c\alpha_1 l + c\beta_1 s\alpha_1 h) \\ \quad - b\omega_2{}^2(c\beta_2 s\alpha_2 h) \\ \quad - b\omega_3{}^2(-c\alpha_3 l - c\beta_3 s\alpha_3 h) \\ \quad - b\omega_4{}^2(c\beta_4 s\alpha_4 h) \sum_{i=1}^{i=4} s\beta_i s\alpha_i \omega_i \delta(i) \end{array} \right) \\ \left(\begin{array}{l} \sum_{i=1}^{4} c\alpha_i d\omega_i{}^2 \delta(i) - b\omega_1{}^2(-s\beta_1 s\alpha_1 l) \\ \quad - b\omega_2{}^2(c\beta_2 s\alpha_2 l) - b\omega_3{}^2(s\beta_3 s\alpha_3 l) \\ \quad - b\omega_4{}^2(-c\beta_4 s\alpha_4 l) \end{array} \right) \end{bmatrix}. \tag{16}$$

In order to check the correctness of our model, substitute α_i and β_i equal to zero in the dynamic equation (16) of the tiltrotors quadrotor, it becomes similar to that of conventional quadrotor.

3. DECENTRALIZATION OF TILTROTOR QUADROTOR

Consider a class of nonlinear MIMO system:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= h(x), \end{aligned} \tag{17}$$

where $x \in R^n$ is the state and $u \in R^m$ is the control input. Let the system (17) be subdivided into m subsystems $\sum_1$, $\sum_2$, …, $\sum_m$ with interactions block such that $y_i = h_i(x)$, $i = 1, 2, \ldots, m$ with $x = [x_1, x_2, \ldots, x_m]$, $x_i \in R^{vi}$, $1 \le i \le m$ and $\sum_{i=1}^{m} v_i = n$. Also, the control signal u is divided into m vectors as:

$$w_1 = \begin{pmatrix} u_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad w_2 = \begin{pmatrix} 0 \\ u_2 \\ \vdots \\ 0 \end{pmatrix}, \quad w_m = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ u_m \end{pmatrix}. \tag{18}$$

ith $w_i \in R^m$ and $\sum_{i=1}^{m} w_i = u$. Assuming that the functions $f(x)$ and $g(x)$ can be decomposed accordingly as:

$$\dot{x}_i = f_{vi}(x) + g_{vi}(x)u,$$

$$y_i = h_i(x), \ \forall\, 1 \le i \le m. \tag{19}$$

One more decomposition of $f_{vi}(x)$ and $g_{vi}(x)$ as follows:

$$\begin{aligned} f_{vi}(x) &= \psi_{vi1}(x) + \psi_{vi2}(x), \\ g_{vi}(x) &= \varphi_{vi1}(x) + \varphi_{vi2}(x), \end{aligned} \tag{20}$$

where $\psi_{vi1}(x)$ and $\varphi_{vi1}(x)$ depend on x_i only. Then $\sum_{si}$ can be written as:

$$\begin{aligned} \dot{x}_i =& [\psi_{vi1}(x) + \psi_{vi2}(x)] \\ &+ [\varphi_{vi1}(x) + \varphi_{vi2}(x)]\, u \\ =& \psi_{vi1}(x) + \varphi_{vi1}(x)\, w_i \\ &+ \left[\psi_{vi2}(x) + \varphi_{vi2}(x)\, w_i + \sum_{\substack{j=i \\ j\neq i}}^{m} g_{vi}(x)\, w_j \right], \\ y_i =& h_i(x), \ \forall\, 1 \le i \le m. \end{aligned} \tag{21}$$

Define

$$\begin{aligned} & F_i(x)\, \triangle_i(x,u,t) \\ &= \left[\psi_{vi2}(x) + \varphi_{vi2}(x)\, w_i + \sum_{\substack{j=1 \\ j\neq 1}}^{m} g_{vi}(x)\, w_j \right], \\ &\quad \forall\, 1 \le i \le m, \end{aligned} \tag{22}$$

where $F_i(x)$ depends only on x_i and $\triangle_i(x,u,t) \in R^{vi}$ contains all interactions and plant parameters. Then the global system, is transformed into the following set of subsystems:

$$\begin{aligned} \dot{x}_i &= \psi_{vi1}(x_i) + \psi_{vi2}(x_i)\, w_i + F_i(x)\, \triangle_i(x,u,t), \\ y_i &= h_i(x), \ \forall\, 1 \le i \le m, \end{aligned} \tag{23}$$

where $F_i(x)\, \triangle_i(x,u,t)$ considered the uncertainty term. However, the effect of the uncertainty term is left for future investigation in this work.

Now, consider the mathematical model of the tiltrotor quadrotor from (16) which can be re-written as:

$$\dot{X} = f(x) + g(x)\, u + l(t), \tag{24}$$

where

$$f(x) = \begin{pmatrix} x_2 \\ -K_1 x_2 \\ x_4 \\ -K_2 x_4 \\ x_6 \\ -K_3 x_6 - g \\ x_8 \\ -K_4 x_8 + x_{10} x_{12} I_1 \\ x_{10} \\ -K_5 x_{10} + x_8 x_{12} I_2 \\ x_{12} \\ -K_6 x_{12} + x_{10} x_8 I_3 \end{pmatrix}, \tag{25}$$

$$g(x) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/m(cx_{11}cx_9) & 1/m(-sx_{11}cx_7 + cx_{11}sx_9sx_7) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/m(sx_{11}cx_9) & 1/m(cx_{11}cx_7 + sx_{11}sx_9sx_7) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/m(-sx_9) & 1/m(cx_9sx_7) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/I_x & 0 \\ 0 & 0 & 0 & 1/I_y \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_z \end{pmatrix}, \tag{26}$$

$$l(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ (-\frac{I_R}{I_{uu}})(x_{12}S_2 + x_{10}S_3) \\ 0 \\ (-\frac{I_R}{I_{vv}})(x_{12}S_1 - x_8 S_3) \\ 0 \\ \left(-\frac{I_R}{I_{ww}}\right)(-x_{10}S_1 + x_8 S_2) \end{pmatrix}. \tag{27}$$

Define

$$\begin{aligned} u_a &= (1/m)(sx_{11}sx_7 + cx_{11}sx_9cx_7); \\ u_b &= (1/m)(-cx_{11}sx_7 + sx_{11}sx_9cx_7); \\ u_c &= (1/m)(cx_9cx_7); \\ u_d &= (1/m)(-sx_{11}cx_7 + cx_{11}sx_9sx_7); \\ u_e &= (1/m)(cx_{11}cx_7 + sx_{11}sx_9sx_7); \\ u_f &= (1/m)(cx_9sx_7); \\ u_g &= (1/m)(cx_{11}cx_9); \\ u_h &= (1/m)(sx_{11}cx_9), \ \text{and} \ u_i = (1/m)(-sx_9). \end{aligned} \tag{28}$$

In this work, S_1, S_2 and S_3 are neglected because I_R is negligibly small compared to the dynamics of the quadrotor. This will make $l(t) = 0$. Transforming (24) into m subsystems will be given in the following steps.

Decomposing $f(x)$ as:

$$f(x) = \begin{pmatrix} f_{v1} \\ f_{v2} \\ f_{v3} \\ f_{v4} \end{pmatrix} = \begin{pmatrix} \psi_{v11} + \psi_{v12} \\ \psi_{v21} + \psi_{v22} \\ \psi_{v31} + \psi_{v32} \\ \psi_{v41} + \psi_{v42} \end{pmatrix}, \tag{29}$$

where

$$\psi_{v11} = \begin{pmatrix} x_2 & -K_1 x_2 & x_4 & -K_2 x_4 & x_6 & -K_3 x_6 \end{pmatrix}^T;$$

$$\psi_{v12} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -g \end{pmatrix}^T,$$
$$\psi_{v21} = \begin{pmatrix} x_8 \\ -K_4 x_8 \end{pmatrix}; \quad \psi_{v22} = \begin{pmatrix} 0 \\ x_{11}x_{12}I_1 \end{pmatrix},$$
$$\psi_{v31} = \begin{pmatrix} x_{10} \\ -K_5 x_{10} \end{pmatrix}; \quad \psi_{v32} = \begin{pmatrix} 0 \\ x_{10}x_{12}I_2 \end{pmatrix},$$
$$\psi_{v41} = \begin{pmatrix} x_{12} \\ -K_6 x_{12} \end{pmatrix}; \quad \psi_{v42} = \begin{pmatrix} 0 \\ x_{10}x_{11}I_3 \end{pmatrix}.$$

Also decomposing $g(x)$ as:

$$g(x) = \begin{pmatrix} g_{v1} \\ g_{v2} \\ g_{v3} \\ g_{v4} \end{pmatrix} = \begin{pmatrix} \phi_{v11} + \phi_{v12} \\ \phi_{v21} + \phi_{v22} \\ \phi_{v31} + \phi_{v32} \\ \phi_{v41} + \phi_{v42} \end{pmatrix}, \tag{30}$$

where

$$\phi_{v11} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ u_g & u_d & u_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ u_h & u_e & u_b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ u_i & u_f & u_c & 0 & 0 & 0 \end{pmatrix}.$$

u_a, u_b, u_c, u_d, u_e, u_f, u_g, u_h and u_i are given in (28), and

$$\phi_{v12} = 0_{6x6},$$
$$\phi_{v21} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_x & 0 & 0 \end{pmatrix}; \quad \phi_{v21} = 0_{6x2},$$
$$\phi_{v31} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_y & 0 \end{pmatrix}; \quad \phi_{v32} = 0_{6x2},$$
$$\phi_{v41} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_z \end{pmatrix}; \quad \phi_{v42} = 0_{6x2}.$$

Define

$$[w_1 = \begin{pmatrix} u_1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T;$$
$$w_2 = \begin{pmatrix} 1 & u_2 & 0 & 0 & 0 & 0 \end{pmatrix}^T;$$
$$w_3 = \begin{pmatrix} 0 & 0 & u_3 & 0 & 0 & 0 \end{pmatrix}^T;$$
$$w_4 = \begin{pmatrix} 0 & 0 & 0 & u_4 & 0 & 0 \end{pmatrix}^T;$$
$$w_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & u_5 & 0 \end{pmatrix}^T;$$
$$w_6 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & u_6 \end{pmatrix}^T.$$

Then, following the decentralized approach, each subsystem is given as follows:

$\tilde{S}_1$ can be written as:

$$\dot{\tilde{x}}_1 = \begin{pmatrix} x_2 & -K_1 x_2 & x_4 & -K_2 x_4 & x_6 & -K_3 x_6 \end{pmatrix}^T + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ (u_g) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ (u_h) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ (u_i) & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_1$$
$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & u_d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & u_e & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -g & u_f & 0 & 0 & 0 & 0 \end{pmatrix} w_2$$
$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_c & 0 & 0 & 0 \end{pmatrix} w_3, \tag{31}$$

which is further decomposed in three subsystems:

$$\tilde{S}_{11};\ \dot{\tilde{x}}_{11} = \begin{pmatrix} x_2 \\ -K_1 x_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ u_g & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_1 + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & u_d & 0 & 0 & 0 & 0 \end{pmatrix} w_2 + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_a & 0 & 0 & 0 \end{pmatrix} w_3,$$
$$\tilde{S}_{12};\ \dot{\tilde{x}}_{12} = \begin{pmatrix} x_4 \\ -K_2 x_4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & u_e & 0 & 0 & 0 & 0 \end{pmatrix} w_2 + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ u_h & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_1 + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_b & 0 & 0 & 0 \end{pmatrix} w_3,$$
$$\tilde{S}_{13};\ \dot{\tilde{x}}_{13} = \begin{pmatrix} x_6 \\ -K_3 x_6 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & u_c & 0 & 0 & 0 \end{pmatrix} w_3 + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ u_i & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_1 + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -g & u_f & 0 & 0 & 0 & 0 \end{pmatrix} w_2.$$

Similarly, $\tilde{S}_2$, $\tilde{S}_3$, $\tilde{S}_4$ are given as:

$$\tilde{S}_2;\ \dot{\tilde{x}}_2 = \begin{pmatrix} x_8 \\ -K_4 x_8 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_x & 0 & 0 \end{pmatrix} w_4 + \begin{pmatrix} 0 \\ x_{11}x_{12}I_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_4, \tag{32}$$
$$\tilde{S}_3;\ \dot{\tilde{x}}_3 = \begin{pmatrix} x_{10} \\ -K_5 x_{10} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_y & 0 \end{pmatrix} w_5$$

$$+\begin{pmatrix} 0 \\ x_{10}x_{12}I_2 \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_5, \tag{33}$$

$$\tilde{S}_4;\ \widetilde{\dot{x}}_4 = \begin{pmatrix} x_{12} \\ -K_6 x_{12} \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_z \end{pmatrix} w_6 + \begin{pmatrix} 0 \\ x_{10}x_{11}I_3 \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_6. \tag{34}$$

4. WIND GUST MODELLING

The modeling approach used in this work is based on [13]. This approach takes the following into consideration:

- The effect of wind velocity change (increasing or decreasing).
- Gust duration.
- Wind velocity change with respect to altitude.
- Wind direction change.

The wind force expression depending on the effective influence area on the quadrotor is also derived. However, this model is suitable, based on the finding in [32], the effect of wind gust in small quadrotors is significantly correlated to the rate of increase or duration of a gust rather than the magnitude of the gust.

At any point in time the effect of wind felt at the different elements of the body is assumed to have equal magnitude and direction. The wind model velocity takes the form (35):

$$|V| = \begin{cases} V_{oi}, & t \le t_{0i}, \\ V_{0i}+\dfrac{|V_{mi}-V_{0i}|}{2}\left(1-\cos\left(\dfrac{\pi(t-t_{0i})}{d_{ni}-t_{0i}}\right)\right), & t_{0i} < t \le d_{ni},\ V_{mi} \ge V_{0i}, \\ V_{0i}+\dfrac{|V_{mi}-V_{0i}|}{2}\left(cos\left(\dfrac{\pi(t-t_{0i})}{d_{ni}-t_{0i}}\right)-1\right), & t_{0i} < t \le d_{ni},\ V_{mi} < V_{0i}, \\ V_{mi}, & t \le t_m, \end{cases} \tag{35}$$

where

- t_m; represents the maximum flight time.
- n represents a discrete random variable to determine the number of wind steps for t_m.
- V_{oi}; represents the wind velocity before each step.
- t_{oi}; represents a discrete random variable to determine each wind step start.
- d_{ni}; represents a discrete random variable to determine each duration of gust.

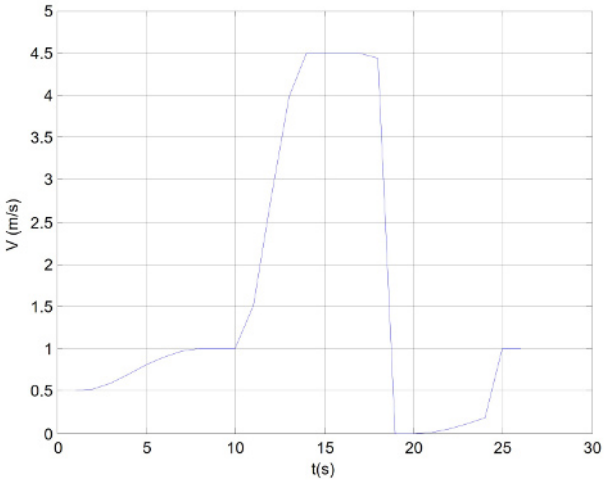

Fig. 2. Simulation showing wind velocity before and after wind gust.

- V_m; represents each gust magnitude.

Simulation example of the model given in [13], for $t_0 = [0;9;16;19]$ s, $V_m = [1;4.5;0;1]$ m/s, $d_n = [7;5;2;5]$ s, $t_m = 25$ s and $V_0 = 0.5$ m/s is shown in Fig. 2.

However the following limitations apply when generating random values:

$$n \in \left[0, \frac{t_m}{10},\right];\ d_i \in [0, t_{i+1}-t_i];\ v_i \in [0, V_{max}];$$
$$\frac{|v_i - v_{i-1}|}{d_i} < a\ (a: \text{ restriction of the rate of step rise}),$$

v_i; represents a discrete random variable to determine each gust magnitude.

Also, the point at which wind blows as a wind direction be the azimuth (Ψ_w), measured from the north through east.

Wind direction changes at each wind velocity step given by:

$$\Psi_{w(i+1)} = \Psi_{wi} \pm \Delta\Psi_{wi}, \tag{36}$$

where $\Delta\Psi_w$ is the random value of wind direction change.

Since the wind velocity changes with altitude, the average wind velocity is determined by:

$$V_{cz} = V_{oz}\left(\frac{z}{z_o}\right)^p, \tag{37}$$

where

V_{cz}; wind velocity at the altitude of z,

V_{0z}; specified wind velocity at the altitude of z_o,

p; energetic wind profile index.

The wind force is given by:

$$F_w = S_e A {V_{cz}}^2, \tag{38}$$

where

S_e; effective area influenced by the wind and A is a conversion factor to Nm2.

With reference to the influence force, we decompose into the following components for more appropriate or easier application:

$$F_{wx} = S_e A {V_{cz}}^2 \cos(\Psi_w); \quad F_{wy} = S_e A {V_{cz}}^2 \sin(\Psi_w). \quad (39)$$

For simplicity, the quadrotor surface area is represented as a cylinder. So the surface area:

$$S_k = \mu 2\pi rh + \sigma 2\pi r^2. \quad (40)$$

The right hand representing the sum of lateral area and bases and μ, σ representing the fill factors here. Therefore if wind affects only half of the quadrotor the effective area will be given by:

$$\begin{aligned} S_{ex} &= \mu \pi rh \cos(\theta) + \sigma \pi r^2 \sin(\theta); \\ S_{ey} &= \mu \pi rh \cos(\varphi) + \sigma \pi r^2 \sin(\varphi) \end{aligned} \quad (41)$$

with θ and φ representing the pitch and yaw angles.

5. BACKSTEPPING CONTROL OF DECENTRALIZED SYSTEM

The theory of backstepping is given in [33]. Each subsystem is in the form below:

$$\begin{aligned} \tilde{S}_i : \dot{x}_i &= \psi_{vi1}(x) + \varphi_{vi1}(x) w_i + F_i(x) \triangle_i(x,u,t), \\ & i = 1,\ 2,\ 3,\ 4, \end{aligned} \quad (42)$$

where $F_i(x) \triangle_i(x,u,t)$ is considered as the uncertainty term. $F_i(x)$ depends only on x_i and $\triangle_i(x,u,t) \in R^{vi}$ contains all interactions and plant parameters. Then each subsystem is treated separately without the uncertainty term to generate control inputs using backstepping method.

For $\tilde{S}_2$:

$$\begin{aligned} \tilde{S}_2;\ \widetilde{\dot{x}}_2 = &\begin{pmatrix} x_8 \\ -K_4 x_8 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_x & 0 & 0 \end{pmatrix} w_4 \\ &+ \begin{pmatrix} 0 \\ x_{11} x_{12} I_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} w_4. \end{aligned}$$

Without the uncertainty term, $\tilde{S}_2$ represents the roll angle subsystem which is now in a strict feedback form:

$$\widetilde{\dot{x}}_2 = \begin{pmatrix} x_8 \\ -K_4 x_8 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_x & 0 & 0 \end{pmatrix} w_4.$$

Extracting those gives:

$$\begin{aligned} \dot{x}_7 &= x_8, \\ \dot{x}_8 &= -K_4 x_8 + u_4 (1/I_x). \end{aligned}$$

To achieve a change of state, we add and subtract u_{x7}; where u_{x7} is a function of x_7. Thus we have:

$$\begin{aligned} \dot{x}_7 =& x_8 + u_{x7} - u_{x7} \\ =& u_{x7} + (x_8 - u_{x7}) = u_{x7} + \mathrm{e}_2. \end{aligned} \quad (43)$$

So that,

$$\dot{\mathrm{e}}_2 = \dot{x}_8 - \dot{u}_{x7} = -K_4 x_8 + u_4 (1/I_x) - \dot{u}_{x7} = v_2.$$

Define the Lyapunov function $V(x_7, e_2)$:

$$V = \frac{1}{2}\left(x_7 - x_7^d\right) + \frac{1}{2} e_2^2. \quad (44)$$

The time derivative,

$$\begin{aligned} \dot{V} &= \left(x_7 - x_7^d\right) \dot{x}_7 + e_2 \dot{e}_2 \\ &= \left(x_7 - x_7^d\right)(u_{x7} + e_2) + e_2 v_2 \\ &= \left(x_7 - x_7^d\right)(u_{x7}) + e_2 (x_7 - x_7^d + v_2). \end{aligned}$$

To ensure $\dot{V}$ is negative definite in order to guarantee stability:

$$u_{x7} = -k_{21}\left(x_7 - x_7^d\right),$$

and

$$v_2 = -k_{22} e_2 - \left(x_7 - x_7^d\right); \quad k_{21},\ k_{22} > 0.$$

Therefore,

$$\dot{V} = -k_{21}\left(x_7 - x_7^d\right)^2 + e_2[\left(x_7 - x_7^d\right) - k_{22} e_2 - \left(x_7 - x_7^d\right)].$$

Hence,

$$\dot{V} = -k_{21} e^2 - k_{22} e_2^2 < 0.$$

So,

$$\begin{aligned} u_4 &= I_x (K_4 x_8 + \dot{u}_{x7} + v_2), \\ u_4 &= I_x (K_4 x_8 + \dot{u}_{x7} + \left(-k_{22} e_2 - \left(x_7 - x_7^d\right)\right)). \end{aligned}$$

But, $\dot{u}_{x7} = -k_{21} \dot{x}_7 = -k_{21} x_8$ and $e_2 = \dot{x}_7 - u_{x7} = x_8 + k_{21}\left(x_7 - x_7^d\right)$.

This gives

$$\begin{aligned} u_4 =& I_x (K_4 x_8 - k_{21} x_8 + (-k_{22}(x_8 + k_{21}\left(x_7 - x_7^d\right)) \\ & - \left(x_7 - x_7^d\right))), \\ u_4 =& I_x ((K_4 - k_{21} - k_{22}) x_8 \\ & + (-k_{22} k_{21} - 1)\left(x_7 - x_7^d\right)). \end{aligned} \quad (45)$$

Similarly two other inputs are generated from $\tilde{S}_3$ and $\tilde{S}_4$ for pitch and yaw respectively:

$$\begin{aligned} u_5 =& I_y ((K_5 - k_{31} - k_{32}) x_{10} \\ & + (-k_{32} k_{31} - 1)\left(x_9 - x_9^d\right)), \end{aligned} \quad (46)$$

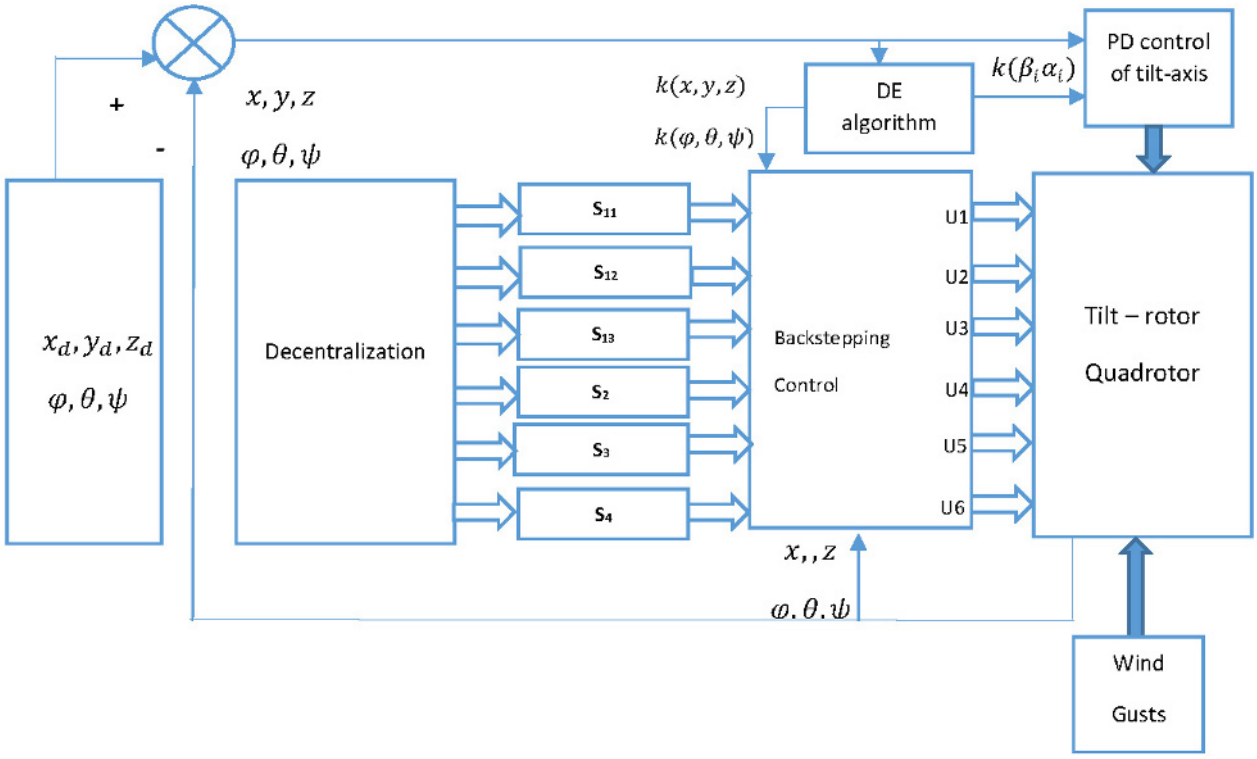

Fig. 3. Block diagram of decentralized backstepping control.

$$u_6 = I_z((K_6 - k_{41} - k_{42})x_{12} + (-k_{42}k_{41} - 1)\left(x_{11} - x_{11}^d\right)). \quad (47)$$

Now, for subsystem $\tilde{S}_1$, which comprises $\tilde{S}_{11}$, $\tilde{S}_{12}$ and $\tilde{S}_{13}$ are used to generate three inputs; u_1, u_2 and u_3.

For instance, backstepping control is applied to S_{13} as follows:

$$\tilde{S}_{11}; \tilde{\dot{x}}_{11} = \begin{pmatrix} x_2 \\ -K_1 x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ u_g u_1 \end{pmatrix},$$
$$[\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -K_1 x_2 + u_g u_1.$$

Again, to achieve a change of state, we add and subtract u_{x1}; where u_{x1} is a function of x_1 we have:

$$\begin{aligned} \dot{x}_1 &= x_2 + u_{x1} - u_{x1} \\ &= u_{x1} + (x_2 - u_{x1} \\ &= u_{x1} + e_1. \end{aligned} \quad (48)$$

So that,

$$\dot{e}_1 = \dot{x}_2 - \dot{u}_{x1} = -K_1 x_2 + u_g u_1 - \dot{u}_{x1} = v_1.$$

The Lyapunov function is similar as previously defined, so, to ensure V is negative definite;

$$u_{x1} = -k_{11}\left(x_1 - x_1^d\right) \text{ and}$$
$$v_1 = -k_{12}e_1 - \left(x_1 - x_1^d\right); \; k_{11}, k_{12} > 0.$$

From

$$v_1 = -K_1 x_2 + u_g u_1 - \dot{u}_{x1},$$
$$u_1 = (1/u_g)(v_1 + K_1 x_2 + \dot{u}_{x1}).$$

But, $\dot{u}_{x1} = -k_{11}\dot{x}_1 = -k_{11}x_2$ and $e_1 = \dot{x}_1 - u_{x1} = x_2 + k_{11}\left(x_1 - x_1^d\right)$

$$\begin{aligned} v_1 &= -k_{12}(x_2 + k_{11}\left(x_1 - x_1^d\right)) - \left(x_1 - x_1^d\right), \\ u_1 &= (1/u_g)(-k_{12}(x_2 + k_{11}\left(x_1 - x_1^d\right)) - \left(x_1 - x_1^d\right) \\ &\quad + K_1 x_2 - k_{11}x_2), \\ u_1 &= (1/u_g)((-k_{12}k_{11} - 1)\left(x_1 - x_1^d\right) \\ &\quad + (K_1 - k_{11} - k_{12})x_2). \end{aligned} \quad (49)$$

Similarly, from $\tilde{S}_{12}$ and $\tilde{S}_{13}$ we have:

$$\begin{aligned} u_2 &= (1/u_e)((-k_{62}k_{61} - 1)\left(x_3 - x_3^d\right) \\ &\quad + (K_2 - k_{61} - k_{62})x_4), \end{aligned} \quad (50)$$

$$\begin{aligned} u_3 &= (1/u_c)((-k_{72}k_{71} - 1)\left(x_5 - x_5^d\right) \\ &\quad + (K_3 - k_{71} - k_{72})x_6). \end{aligned} \quad (51)$$

Fig. 3 shows the structure of the decentralized-backstepping system.The backstepping method in conjunction with Lyapunov function introduced the controller parameters k_{ij}. These control inputs are not a function of the orientations, α_i and β_i. In subsequent sections, we will introduce a PD control of the orientation angles which is followed by the optimization technique for all the controller parameters.

6. WIND FORCE CANCELLATION

Forces are generated in different axis due to tilting of the quadortor. This concept has dual advantages in that, it can be used to calcel wind gust whilst keeping the orientation as desired [25]. However, for both rotors 1 and 3, $\beta_{1,3}$ angles are set to zero while $\alpha_{1,3}$ angles are allowed the flexiblity of tilting so as to move in the x-direction and concurrently cancel wind disturbance in the same direction. For rotors 2 and 4, the concept is the same only that $\beta_{2,4}$ are now set to 90 degrees so that the $\alpha_{2,4}$ can now be directed towards the y direction when needed. The PD controller used to effectively control the orientations α_i is given by;

$$\alpha_{1,3} = K_{p\alpha_{1,3}} e_x + K_{d\alpha_{1,3}} \left(e_{ix} - e_{(i-1)x}\right), \quad (52)$$

$$\alpha_{2,4} = K_{p\alpha_{2,4}} e_y + K_{d\alpha_{1,3}} \left(e_{iy} - e_{(i-1)y}\right), \quad (53)$$

where $e_r = r_d - r$, for $r = (x, y)$, in the body-fixed frame. K_p and K_d are the proportional and derivative control gains, and they are given in Table 2.

7. CONTROLLER PARAMETERS OPTIMIZATION TECHNIQUE

A significant evolution of optimization theory have emanated over time. A recent algorithm for evolutionary algorithm called the Differential Evolution (DE) was first introduced by Rainer Storn and Keneth Price in 1995 [29].

In brief, DE works in the following way; First, we randomly select and initialize control parameters and then evaluate the objective function. Thereafter, the following processes will be executed so long as the stopping criteria is not met; For each individual in the population an offspring of controller parameters is created using the weighted difference of parent solutions. Finally, the fitter vector between the parent and offspring is passed on to the next iteration of the algorithm. Subsequently the controller parameters are passed on to the controller.

Fig. 4 shows the flowchart for the process of DE. First, it starts, with (N_p) initial population generated randomly between two bounds $(\mathrm{X_{j,max}}, \mathrm{X_{j,min}})$. Each solution (X) comprises of (D) elements which is the dimension of the problem (number of problem parameters needed to be optimized). Additional factors need to be defined, such as generation number or iteration (N_g), mutation factor $(F \in [0,1])$ which control the convergence speed, and crossover factor $(CR \in [0,1])$ which plays role in the smoothness of the convergence and also ensures the diversity of the solutions in order not to be trapped in a local minimum during the optimization process. Let G be the number of generations.

$$G^i = \left[X_1^i, X_2^i, \ldots, X_{N_p}^i\right], \quad (54)$$

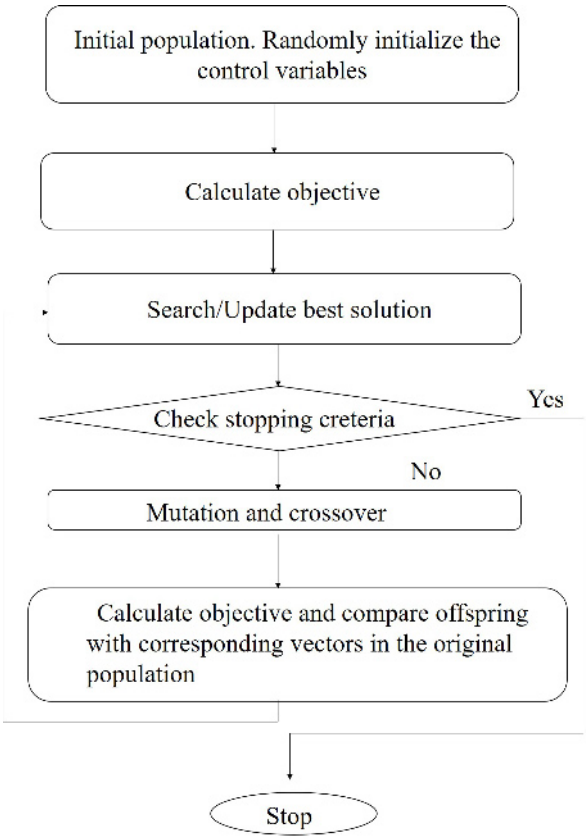

Fig. 4. DE Flowchart.

where i is the generation number, and each solution has (D) parameters which are the controller gains for this case.

$$X_n^i = [X_{n1}, X_{n2}, \ldots, X_{nD}],$$
$$X_{i,j} = X_{j,min} + \text{random number}\,(X_{j,max} - X_{j,min}). \quad (55)$$

In the next two steps, the fitness or objective function for each solution will be calculated, and according to it the best solution among the population will be nominated. A good set of control parameters will result in minimization of performance criteria which includes overshoot, rise time, settling time and steady state error. The objective function is selected based on the minimum Integral Square Error (ISE), the error being the difference between the desired response and actual response. The ISE of a step response is computed at each iteration until a minimum error that guarantees the best performance criteria is achieved:

$$ISE = \int_0^\infty e(t)^2 dt$$
$$\underset{X=k_{ij}}{\text{minimize}}\; ISE(X). \quad (56)$$

Then the stopping criteria will be checked which may result in terminating or continuing to the next step. This step includes mutation and crossover processes which are

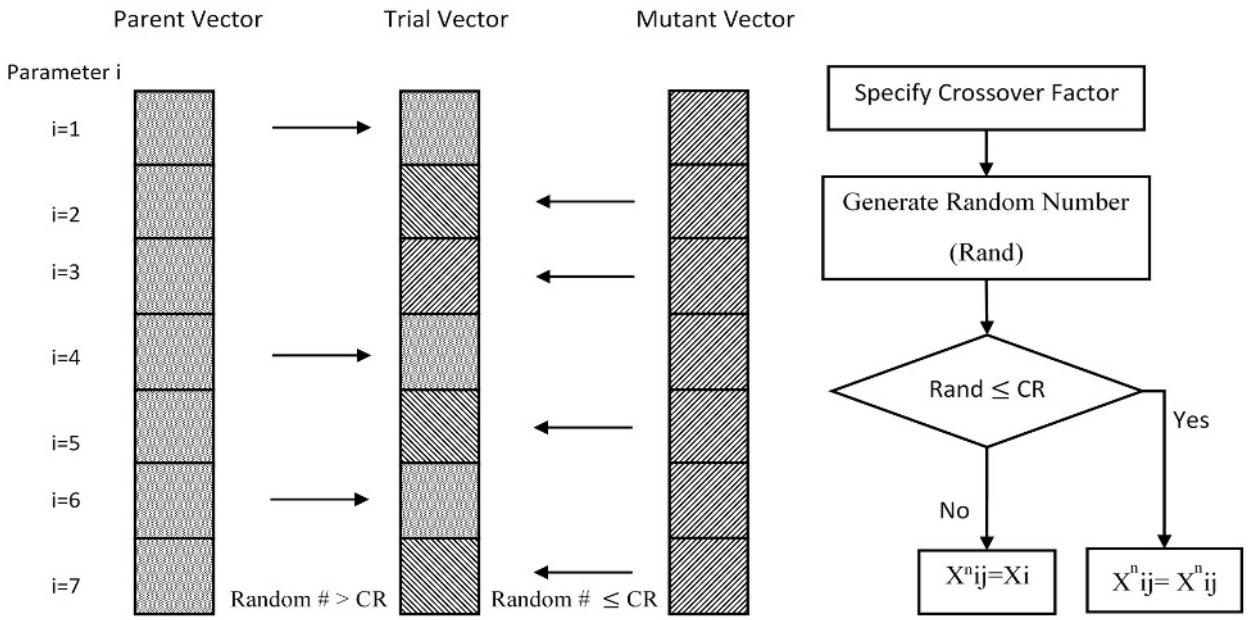

Fig. 5. Crossover procedure.

the heart of the differential evolution algorithm. Here, a variant vector solution (offspring) is generated for each solution in the population by using the following formula:

$$V_i^{(G+1)} = X_i^{(G)} + F\left(X_{best}^{(G)} - X_i^{(G)}\right) + F(X_{r1}^{(G)} - X_{r2}^{(G)}), \quad (57)$$

where $X_{r1}^{(G)}$ and $X_{r2}^{(G)}$ are randomly selected solution vectors from the current generation (different from each other and the corresponding X_i) and $X_{best}^{(G)}$ is the solution achieving best fitness function.

Then a trial solution will be generated by copying the parameters form the parent solution or the offspring solution based on randomly generated probability and the crossover factor. Fig. 5 which is the crossover procedure illustrates the process of generating the trial solution, where a random probability number between [0, 1] is generated and then compared to the crossover factor. If the random number is found to be larger than the crossover, then the trial solution will take the parameter from the parent and from the offspring otherwise. In one solution this procedure will be repeated (D) times until the trial solution is formed.

In the last step, there will be Np trial solutions corresponding to the original population. The fitness function will be calculated for them. The new generation will be formed by comparing the parent solution to the trial solution and takes the one which has the best fitness function as the member (new parent) for the new generation. The whole procedures will be repeated again and again until the stopping criteria is satisfied or the generation (iteration number) number is reached.

Table 1. DE algorithm parameters.

Cross over factor (CR)	0.5
Mutation factor (F)	0.5
Generation	50
Population size	25

Table 2. Controller parameters.

Backstepping controller	Gains
k_{ij}; $i = 1:6$, $j = 2$	35.1197, 33.3426, 38.8252, 14.2395, 40.6587,24.4382
k_{ij}; $i = 1:6$, $j = 2$	0.0684, 0.0114, 2.0, 0.0262, 0.0262, 0.01
PD controller	Gains
K_p, K_d	0.0226, 2.9097

As long as the number of solutions and iterations gets larger, the possibility to reach the global minimum increases. However, the resulting parameters used to achieve the minimum ISE are selected as the controller parameters. The DE algorithm parameters and controller gains derived are given in Tables 1 and 2, respectively.

Table 3. Prameters of the quadrotor.

Parameter	Definition	Value	Unit
g	Acceleration due to gravity	9.81	m/s^2
m	Mass	0.5	Kg
r, h	radius, height	0.2	m
$I_x = I_y$	x, y inertia	4.85×10^{-3}	$kg.m^2$
I_z	z inertia	8.81×10^{-3}	$kg.m^2$
I_R	Rotor inertia	3.36×10^{-5}	$kg.m^2$
b	Trust factor	2.92×10^{-6}	kg.m
d	Drag factor	1.12×10^{-7}	$kg.m^2$
K_1, K_2, K_3	Drag coefficients	0.01	Ns/m
K_4, K_5, K_6	Drag coefficients	0.012	Ns/m
A	Rate of wind velocity	0.61	Nm^2
μ, σ	Fill factors	0.2, 0.4	M

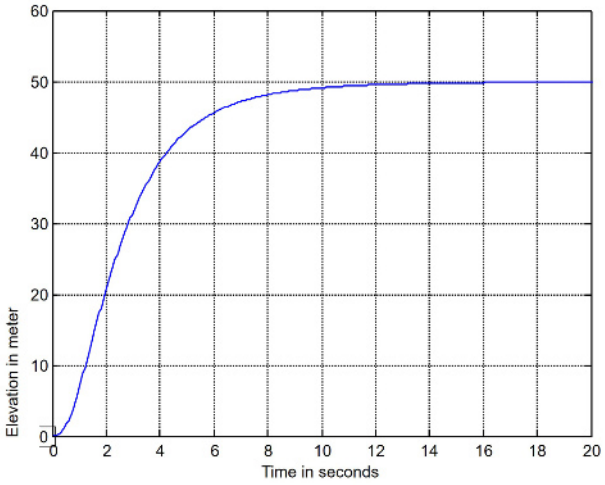

Fig. 6. Elevation to $[0,0,50]$ m under wind gust.

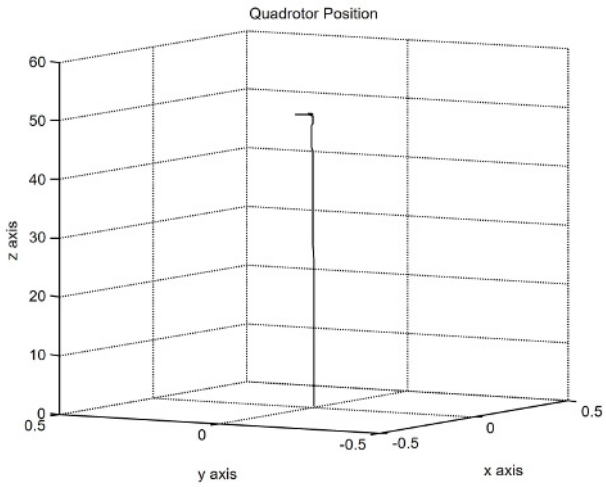

Fig. 7. 3D plot of elevation to $[0,0,50]$ m under wind.

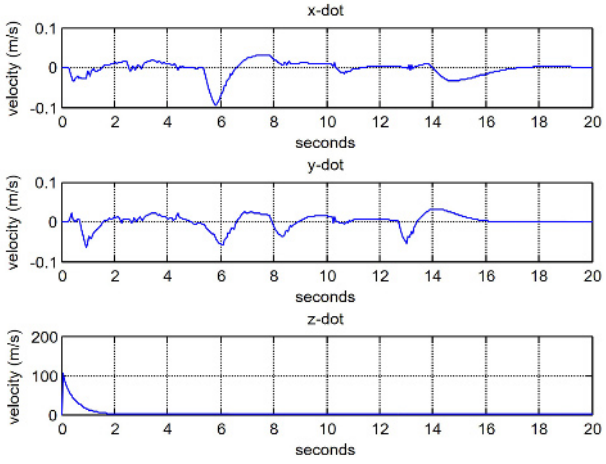

Fig. 8. Translational rates of elevation to $[0,0,50]$ m under wind gust.

8. SIMULATION RESULTS AND DISCUSSION

Simulation is carried out in MATLAB environment and the control and dynamic parameters used are given in the Table 3.

Quadrotor commanded move to the following coordinate $[0,0,50]$ m. Fig. 6, shows that the target height is reached and the tiltrotor quadrotor is able to hover at 50 m. Fig. 7 depicts the 3D plot which shows a location tracking of the coordinate with a negligibly small error in all translational axes.

As earlier mentioned, wind gust is seen to cause random fluctuation in the velocities as shown in Fig. 8, which depicts the translational rates, although negligible. However, the angular orientation is not affected due to the feature of the quadrotor. Fig. 9. shows the orientations α_i as controlled by the PD controller. The β_i orientation angle are given in Fig. 10, which shows the values that are set in order to achieve desired trajectory. Figs. 11-13 shows the backstepping control input to the quadrotor for the translational orientation, although the rotational inputs are unaffected as because they are not triggered.

9. CONCLUSION

A decentralized control approach is successfully demonstrated on a tilt-rotor quadrotor under wind disturbance. A Differential evolution (DE) optimization technique was first established to compute controller gains based on Integral Square Error. Different simulation modes show how the tilt-rotor quadrotor system are decoupled and controlled independently. The results shows that the system exhibits robustness against wind gust disturbance. The future work is to build a prototype of the tilt-rotor quadrotor system and use different nonlinear control techniques for comparisons.

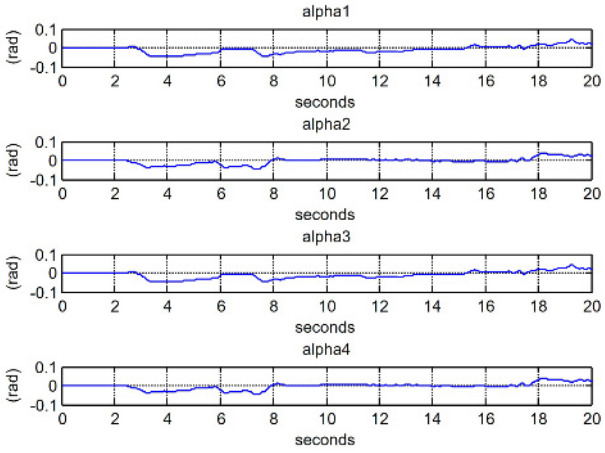

Fig. 9. Orientation angles α_i's.

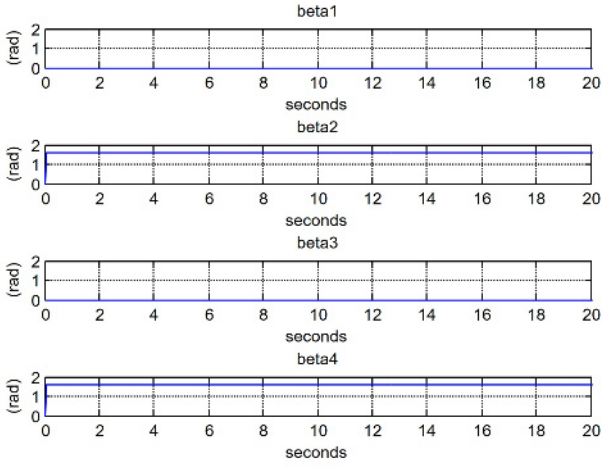

Fig. 10. Orientation angles β_i's.

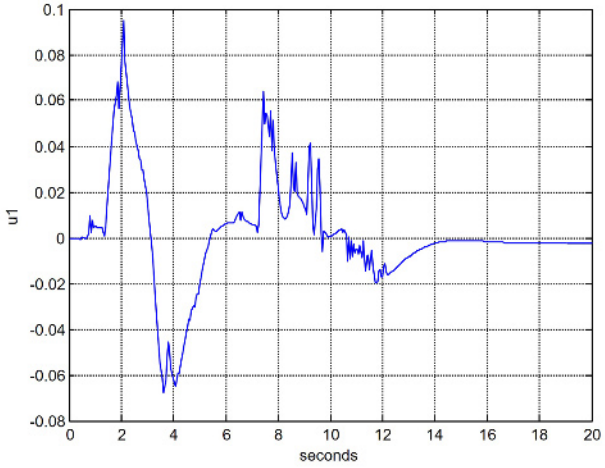

Fig. 11. Input for z direction.

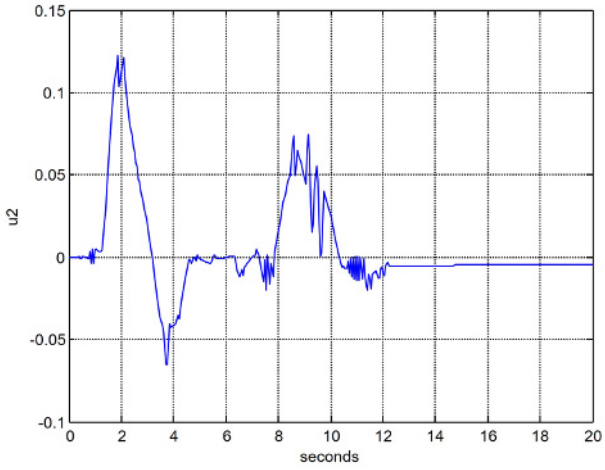

Fig. 12. Input 1 for x direction.

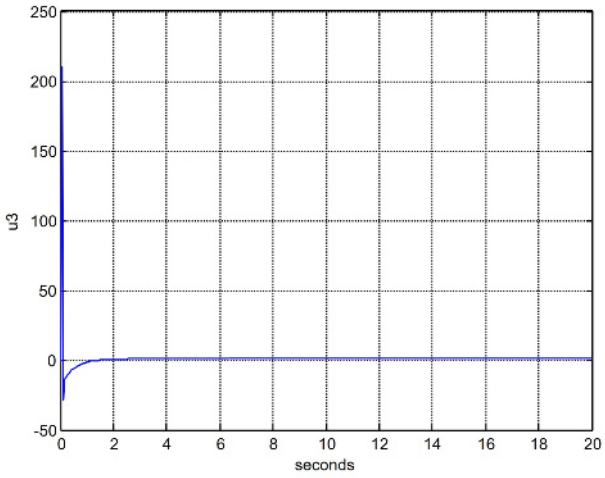

Fig. 13. Input for y direction.

REFERENCES

[1] B. Samir,*Design and Control of Quadrotors with Application to Autonomous Flying*, Ph.D. Thesis, Ecole Polytechnique Federale de Lausanne, 2007.

[2] P. Paul, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 7, no. 18, pp. 691-699, 2010.

[3] V. Holger, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," *Proc. of the IEEE International Conference on Mechatronics*, pp. 1-6, 2009.

[4] A. Erdinç, J. P. Ostrowski, and C. J. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," *International Journal of Robotics Research*, vol. 5, no. 24, pp. 329-341, 2005.

[5] R., Markus, H. H. Bülthoff, and P. R. Giordano, "Modeling and control of a quadrotor UAV with tilting propellers," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4606-4613, 2012.

[6] K. T. Öner, E. Çetinsoy, M. Ünel, M. F. Akşit, I. Kandemir, and K. Gülez, "Dynamic model and control of a new

quadrotor unmanned aerial vehicle with tilt-wing mechanism," *Proc. of the World Academy of Science, Engineering and Technology*, vol. 2, no. 9, 2008.

[7] E. Mahmoud, M. Elshafei, A. A. Saif, and M. F. Al-Malki, "Quadrotor helicopter with tilting rotors: modeling and simulation," *Proc. of the IEEE World Congress on Computer and Information Technology (WCCIT)*, pp. 1-5, 2013.

[8] S. Fatih and E. Altug, "Adaptive control of a tilt-roll rotor quadrotor UAV," *Proc. of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1132-1137, 2014.

[9] H. Minh-Duc, T. Hamel, and C. Samson, "Control of VTOL vehicles with thrust-direction tilting," arXiv preprint, arXiv:1308.0191, 2013.

[10] C. Nicol, C. J. B. Macnab, and A. Ramirez-Serrano, "Robust adaptive control of a quadrotor helicopter," *Mechatronics*, vol. 6, no. 21, pp. 927-938, 2011.

[11] S. L. Waslander and C. Wang, "Wind disturbance estimation and rejection for quadrotor position control," *Proc. of the AIAA Unlimited Conference on Infotech@ Aerospace and AIAA Unmanned*, pp. 1983, 2009.

[12] C. Yan-min, Y. He, and M. Zhou, "Decentralized PID neural network control for a quadrotor helicopter subjected to wind disturbance," *Journal of Central South University*, no. 22, pp. 168-179, 2015.

[13] V. V. Solovyev, V. I. Finaev, Zargaryan Y. A. Shapovalov O. Igor, and D. A. Beloglazov, "Simulation of wind effect on a quadrotor flight," *ARPN Journal of Engineering and Applied Science*, vol. 10, pp 1535-1538, 2015.

[14] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances," *Control Engineering Practice*, vol. 19, no. 10, pp 1195-1207, 2011.

[15] B. Samir and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2247-2252, 2005.

[16] M. Tarek and A. Benallegue, "Control of a quadrotor mini-helicopter via full state backstepping technique," *Proc. of the 45th IEEE Conference on Decision and Control*, pp. 1515-1520, 2006.

[17] Y. Yu, Y. Guo, X. Pan, and C. Sun, "Robust backstepping tracking control of uncertain MIMO nonlinear systems with application to quadrotor UAVs," *Proc. of the IEEE International Conference on Information and Automation*, pp. 2868-2873, 2015.

[18] Y. Li and G. Wang, "Quad-rotor airship modeling and simulation based on backstepping control," *International Journal of Control and Automation*, vol. 5, no. 6, pp 369-384, 2013.

[19] B. M. Ariffanan, A. Husain, and K. A. Danapalasingam, "Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 2, no. 79, pp 295-321, 2015.

[20] Y. Chen, Y. He, and M. Zhou, "Decentralized PID neural network control for a quadrotor helicopter subjected to wind disturbance," *Journal of Central South University*, no. 22, pp 168-179, 2015.

[21] H. Liu, D. Derawi, J. Kim, and Y. Zhong, "Robust optimal attitude control of hexarotor robotic vehicles," *Nonlinear Dynamics*, vol. 4, no. 74, pp 1155-1168, 2013.

[22] C. Hancer, K. T. Oner, E. Sirimoglu, E. Cetinsoy, and M. Unel, "Robust position control of a tilt-wing quadrotor," *Proc. of the 49th IEEE Conference on Decision and Control (CDC)*, pp. 4908-4913, 2010.

[23] A. Belhani, F. Mehazzem, and K. Belarbi, "Decentralized backstepping controller for a class of nonlinear multivariable systems," *ICGST ASCE Journal*, vol. 2, no. 6, pp 41-48, 2006.

[24] D. Mujahid, A. A. Saif, M. Al-Malki, and M. El Shafie, "Decentralized backstepping control of quadrotor," *Proc. of the 4th International Conference on Underwater System Technology: Theory and Applications (USYS'12)*, December 2012.

[25] A. Aliyu, M. Elshafei, A. A. Saif, and M. Dhaifullah, "Performance evaluation of quadrotor with tilted rotors under wind gusts," *Proc. of the IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 294-299, 2016.

[26] A. A. Saif, "Feedback linearization control of quadrotor with tiltable rotors under wind gusts," *International Journal of Advanced and Applied Sciences*, vol. 4, no. 10, pp. 150-159, 2017.

[27] V. Jakob and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," *Proc. of the Congress on Evolutionary Computation*, vol. 2, pp. 1980-1987, 2004.

[28] P. Sandra and T. Krink, "Differential evolution and particle swarm optimisation in partitional clustering," *Computational Statistics & Data Analysis*, vol. 5 no. 50, pp 1220-1247, 2006.

[29] V. V. Truong, "A comparison of particle swarm optimization and differential evolution," *International Journal on Soft Computing*, vol. 3 no. 3, p. 13, 2012.

[30] E. Julian, J. M. Lopez-Guede, and M. Graña, "Particle swarm optimization quadrotor control for cooperative aerial transportation of deformable linear objects," *Cybernetics and Systems*, vol. 1-2, no. 47, pp. 4-16, 2016.

[31] B. Vahid and A. Khosravi, "Robust PID controller design based on H∞ theory and a novel constrained artificial bee colony algorithm," *Transactions of the Institute of Measurement and Control*, vol. 1, no. 40, pp. 202-209, 2016.

[32] H. Manal, P. W. Quimby, S. Chang, K. Jackson, and M. L. Cummings, "Wind gust alerting for supervisory control of a micro aerial vehicle," *Proc. of the IEEE Conference on Aerospace*, pp. 1-7, 2011.

[33] H. K. Khalil, *Nonlinear Systems*, 3rd edition, Prentice Hall, Upper Saddle River, New Jersey, 2002.

Abdul-Wahid A. Saif received his Ph.D. from control and and Instrumentation Group, Department of Engineering, Leicester University, U.K., an M.Sc. from Systems Engineering Department, and a B.Sc. from Physics Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran S.A. Dr. Saif is currently an Associate Professor of Control and Instrumentation in Systems Engineering Department (SE) at KFUPM. Dr. Saif research interest is Simultaneous and Strong Stabilization, Robust Control and H_∞-optimization, Wire and Wireless Networked Control, Instrumentation and Computer Control. Dr. Saif taught several courses in Modeling and Simulation, Digital Control, Digital Systems, Microprocessor and Microcontrollers in automation, Optimization, Numerical Methods, PLC's, Process Control and Control System Design. Dr. Saif has published more than 75 papers in reputable journals and conferences.

Abdulrahman Aliyu received the B.Eng. degree in Electrical Engineering from Bayero University Kano, Nigeria. He also recieved his M.Sc. degree in Systems and Control Engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in 2016, where he is currently working towards a Ph.D. degree in the same feild. His research areas includes, Integer/Fractional Order Control, Robotics, Multi-agent and Complex Systems, Artificial Intelligence and Machine Learning.

Mujahed Al Dhaifallah received his B.S. and M.S. degrees in systems engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and his Ph.D. degree in electrical and computer engineering from University of Calgary, Calgary, Canada. He has been an assistant professor of systems engineering with King Fahd University of Petroleum and Minerals, Saudi Arabia since 2009. Currently, he is working as dean of college of engineering at Wadi Al-Dawaser, Prince Sattam bin Abdulaziz University, Saudi Arabia. His main research in the field of Systems Identification, Control and Machine learning.

Moustafa Elshafei obtained his Ph.D. in Electrical Engineering from McGill in 1982 (Dean List), Canada. Since then he has accumulated over 24 years of Academic experience and 9 years of industrial experience. He is inventor/coinventor of 20 USA and international patents, Author/Co-author of 3 books, and published over 150 articles in international journals and professional conferences in the fields of intelligent instrumentation, digital signal processing, artificial intelligence, and industrial control/ automation systems.