International Journal of Control, Automation and Systems 17(11) (2019) 2819-2832
http://dx.doi.org/10.1007/s12555-018-0860-9
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

A Model Predictive Control (MPC) Approach on Unit Quaternion Orientation Based Quadrotor for Trajectory Tracking

Maidul Islam*, Mohamed Okasha, and Erwin Sulaeman*

Abstract: The objective of this paper is to introduce with a quaternion orientation based quadrotor that can be controlled by Model Predictive Control (MPC). As MPC offers promising performance in different industrial applications, quadrotor can be another suitable platform for the application of MPC. The present study consistently adopts unit quaternion approach for quadrotor orientation in order to avoid any axes overlapping problem, widely known as singularity problem whereas Euler angle orientation approach is unable to resolve so. MPC works based on the minimal cost function that includes the attitude error and consequently, the cost function requires quaternion error in order to proceed with process of MPC. Therefore, the main contribution of this study is to introduce a newly developed cost function for MPC because by definition, quaternion error is remarkably different from the attitude error of Euler angle. As a result, a unit quaternion based quadrotor with MPC can ascertain a smooth singularity-free flight that is influenced by model uncertainty. MATLAB and Simulink environment has been used to validate the cost function for quaternion by simulating several trajectories.

Keywords: Constraint handling, cost function, disturbance and noise, path/trajectory tracking, quadrotor, quaternion.

1. INTRODUCTION

Multi-rotors have achieved a great taste from researchers and its users because of their various application fields like exploration, surveillance, mapping, package delivering, aerial filming etc. along with its Vertical Take-Off and Landing (VTOL) functionality [1]. Because of technological advancements, it widens the horizon for researchers of different fields like aeronautics, wireless communication, mechatronics, signal processing, automatic control etc. As most of its applications are trajectory tracking-oriented, robotic control community are interested to develop suitable control strategies in order to ensure smooth flights for multi-rotors.

Quadrotor is one of the most common choices among the researchers because it is simple to build. Apart from that, its omnidirectional agility and flighty movement with aggressive maneuver necessarily stir up interest among controller researchers. Structurally, it is developed using four electrical motors with a pair of clockwise and another pair of anti-clockwise propellers and these are mounted on at the end of a cross-linked frame. In reality, quadrotor encounters dynamic model complexity because it is an under-actuated system and additional model uncertainties like vibration, noise and disturbance, make the system more instable specially when it is used for outdoor applications [2, 3].

In order to overcome and simplify the problem of controller development, a mathematical model of quadrotor is necessarily developed that entails kinematic model and dynamic model. For dynamic model, most prominently Newton-Euler method and Euler-Lagrange method are available for quadrotor in the literatures [4]. Euler angle and quaternion are two different orientation systems that are popular among the researchers for quadrotor. As Euler angle orientation system is inefficacious at certain trajectories or maneuverings that associates gimbal lock or singularity, unit quaternion, a four-tuple orientation system draws more attention of the researchers' nowadays that reflects in different literatures [5, 6]. Fresk and Nikolakopoulos validated the special feature of quaternion experimentally by flipping the quadrotor practically [7].

Immediately the model uncertainties resolution can be considered after the successful completion of quadrotor model design and thus a suitable control approach is required that can deal with the unwanted noise and distur-

Manuscript received November 27, 2018; revised March 16, 2019 and May 8, 2019; accepted June 6, 2019. Recommended by Associate Editor Ning Sun under the direction of Editor Jay H. Lee. The support of Malaysian Ministry of Education through International Islamic University Malaysia under the research grant FRGS 17-036-0602 is gratefully acknowledged. A special thanks to Abdal Engineering Limited for assisting in research work.

Maidul Islam and Erwin Sulaeman are with the Department of Mechanical Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia (e-mails: esulaeman@iium.edu.my, mislam.dipu@gmail.com). Mohamed Okasha is with the Department of Mechanical Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia (e-mail: mokasha@iium.edu.my).
* Corresponding authors.

bance in the system along with multiple constraints at both inputs and outputs of the system. In literatures, significantly three different types of control approaches i.e., linear, nonlinear and learning-based are available for quadrotors [8]. Some commonly available control approaches like PID (Proportional-Integral-Derivative), LQR (Linear Quadratic Regulator), H∞ are considered as linear control approach while back-stepping, Sliding Mode Control (SMC), feedback linearization and MPC offer disposition of nonlinear plants supposedly that defines their category as nonlinear control approach [8, 9]. Some literatures introduce with learning-based control algorithm e.g., neural network and fuzzy logic that has the capability to compensate for gradual impact based on operational data [10].

For instance, Kehlenbeck tested the aggressive maneuver of his PID controlled quadrotor through a tunnel. In that study, unit quaternion representation has been adoted in order to explain the attitude of quadrotor and clinched at an optimal performance [11]. Reyes-Valeria *et al.* used gain-scheduling LQR control approach and finally showed that quaternion orientation can be a novel substitution of Euler-angle orientation [12]. Apart from that Djamel *et al.* applied H∞ theory and back-stepping approach at the same time in order to ensure the robust performance of the quadrotor with the help of quaternion [13]. Avidly Zha *et al.* proposes a nonlinear control strategy, Trajectory Linearization Control (TLC) using quaternion in order to ensure the robust stability that follows the functionality of gain-scheduling controller [14]. Besides, Tran and Ha introduced with a self-tuning new type of neural network controller, proportional double derivative-like neural network (PDDLNN) that combines proportional double derivative (PDD) and a recurrent neural network with a self-tuning technique, on a quaternion based quadrotor for attitude tracking [15].

In recent years, the application of MPC has achieved interest in the area of flight control specially among Unmanned Aerial Vehicles (UAVs). Notwithstanding its expensive computation, MPC has been adopted in such a fast dynamic area of UAV flight control because of some features of MPC as like constraints handling at multiple input multiple output (MIMO) system along with noise and disturbance rejection. In the field of process control, MPC proceeds with online optimization chronologically that allows to deal with model infidelity and helps the model signal to converge towards future reference [16]. Zhao and Go proposed a novel control algorithm wherein a robust feedback linearization contributed suppressing the tracking error towards MPC in order to improve the performance [17]. Some similar works also are found in literatures on MPC that shows the broader extent of contribution of MPC in quadrotor flight control [18–21].

The proposed MPC in the present study is designed for a quaternion-based quadrotor dynamic model whereas it is distinct from conventional MPC with Euler angle orientation system. Henceforth it impacts on the cost function of MPC control algorithm. In several literatures, MPC controller has been chosen because of its advantages for quaternion-based quadrotor where algebraic subtraction has been adopted in cost function to minimize the differences between two quaternion states [22–24]. However, by the definition of quaternion, algebraic subtraction is completely inappropriate with quaternion to find out the difference between two quaternions rather an approach namely "Quaternion Error" that is widely accepted [25,26]. Apart from that, some literatures adopt MPC with Euler angle in order to the design of plant partially or completely and then convert the Euler angle to quaternion and therefore, the study doesn't adopt quaternion completely [24]. In that approach, the controller computes the attitude using Euler angle and as a result, the system is still not free from singularity problem because of computational error. In a nutshell, the present study propounds a MPC controller for completely quaternion-based quadrotor plant and considers an improved cost function for MPC as per the definition of quaternion.

In the following, the paper has been structured as: Section 2 discusses about quaternion. Section 3 outlines quadrotor dynamic model. Afterwards, Section 4 delineates the control algorithm. Section 5 hence motivates for simulation results. Then the work culminates in Section 6 as conclusion.

2. QUATERNION MATH

2.1. Quaternion properties

As quaternion orientation is not similar to Euler-angle orientation system, the properties are essentially described to explain the dynamic model. Quaternion is defined as a hyper complex number and the rank is 4. A quaternion consists of a scalar, q_0 and three vector elements, $q_{1:3} = [q_1, q_2, q_3]^T$. Thence, it can be represented as [22]

$$\boldsymbol{q} = [q_0,\ q_1, q_2,\ q_3]^T. \tag{1}$$

In addition, the conjugate, norm and inverse of a quaternion can be denoted respectively as [27].

$$\bar{\boldsymbol{q}} = [q_0,\ -q_1, -q_2,\ -q_3]^T, \tag{2}$$

$$\|\boldsymbol{q}\| = \sqrt{{q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2}, \tag{3}$$

$$\boldsymbol{q}^{-1} = \frac{\bar{\boldsymbol{q}}}{\|\boldsymbol{q}\|}. \tag{4}$$

Furthermore, a unit quaternion is stipulated with such a condition, ${q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 = 1$ that means unit quaternion is the norm of quaternion [28]. As a result, the condition can help to derive conclusively that quaternion conjugate and quaternion inverse are equal as denoted, $\boldsymbol{q}^{-1} = \bar{\boldsymbol{q}}$.

Quaternion multiplication, let, between two different quaternion $\boldsymbol{p}$ and $\boldsymbol{q}$ can be described with the following equation and symbolically the multiplication is denoted by $\otimes$. Noted that quaternion multiplication is not commutative that means $\boldsymbol{p} \otimes \boldsymbol{q} \neq \boldsymbol{q} \otimes \boldsymbol{p}$ [6]

$$\boldsymbol{p}\otimes\boldsymbol{q} = \begin{pmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & -p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}. \tag{5}$$

Moreover, as Euler's theorem infers a rotational angle, α and a unit vector of three dimensions, $e = ie_1 + je_2 + ke_3$, quaternion represents it as follows [24]:

$$\boldsymbol{q} = \left[\cos\left(\frac{\alpha}{2}\right), e_1 \sin\left(\frac{\alpha}{2}\right), e_2 \sin\left(\frac{\alpha}{2}\right), e_3 \sin\left(\frac{\alpha}{2}\right)\right]^T. \tag{6}$$

2.2. Quaternion derivative

The derivative of quaternion can be revised as the following algebraic equation wherein the significance of quaternion derivatives come across in order to define angular velocities of quadrotor, $\omega = [p, q, r]^T$ and it considered under Kinematics [29].

$$\dot{\boldsymbol{q}} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ p \\ q \\ r \end{pmatrix}. \tag{7}$$

3. SYSTEM DESIGN

A quadrotor, in general, is subjugated by its mass and inertia wherein it is subjected to gravitational force. In addition, balance-counterbalance of the forces and torques that work on the quadrotor make the dynamics more complex during flight in 3D space.

However, in order to describe the dynamics of quadrotor, two frame of references, Earth-fixed frame and Body frame, are denoted as respectively **E** and **B** in Fig. 1, are necessarily considered. The significance of Earth-fixed frame appears when it is required to measure the position and gravitational force on the quadrotor. Meanwhile, the estimation of thrust and angular velocities of quadrotor mandate the introduction of Body frame as well.

However, in the present study, $\varsigma = [x, y, z]^T$ and $\xi = [\phi, \theta, \psi]^T$ have been used in order to explain the position and rotational movements such as roll, pitch and yaw respectively, in 3D space. Noted that $\xi = [\phi, \theta, \psi]^T$ describes Euler angle rotations that helps to comprehend the rotational movement easily.

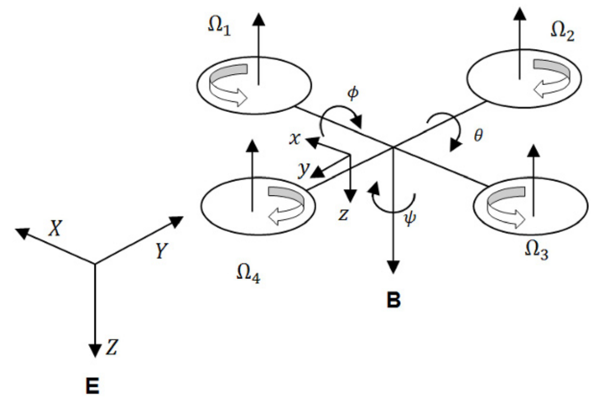


Fig. 1. Configuration of quadrotor [5].

3.1. Kinematics

As the cooperation between Earth-fixed frame and Body frame is required in order to define the complete system, the introduction of a rotational transformation matrix can make it successful. In such a case, if it is considered that the quadrotor has a rotation along X, Y and Z axis then it can be formulated as [7].

$$Q_x = \begin{bmatrix} 1-2\left({q_2}^2+{q_3}^2\right) \\ 2(q_1q_2+q_0q_3) \\ 2(q_1q_3-q_0q_2) \end{bmatrix}, \tag{8}$$

$$Q_y = \begin{bmatrix} 2(q_1q_2-q_0q_3) \\ 1-2\left({q_1}^2+{q_3}^2\right) \\ 2(q_2q_3+q_0q_1) \end{bmatrix}, \tag{9}$$

$$Q_z = \begin{bmatrix} 2(q_1q_3+q_0q_2) \\ 2(q_2q_3-q_0q_1) \\ 1-2\left({q_1}^2+{q_2}^2\right) \end{bmatrix}. \tag{10}$$

Therefore, a final transformation matrix can be formulated through the consolidation of the aforementioned equations (8)-(10) wherein it is applicable for Earth-fixed frame to Body frame transformation as [30]

$$Q_{EB} = \begin{bmatrix} {Q_x}^T \\ {Q_y}^T \\ {Q_z}^T \end{bmatrix} = \begin{bmatrix} 1-2({q_2}^2+{q_3}^2) & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2({q_1}^2+{q_3}^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2({q_1}^2+{q_2}^2) \end{bmatrix}. \tag{11}$$

In order to make quaternion more lucid explicitly, the following equations can be introduced that maintains relation between Euler-angle and quaternion orientation as follows [31]:

$$\boldsymbol{q}=\begin{bmatrix} -\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)+\cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)+\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ -\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)+\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)+\cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \end{bmatrix}, \tag{12}$$

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}=\begin{bmatrix}\operatorname{atan2}(2\,(q_1q_2+q_0q_3)\,,1-2\left(q_1{}^2+q_2{}^2\right)\,) \\ \operatorname{asin}(2(q_0q_2-q_1q_3)) \\ \operatorname{atan2}(2\,(q_1q_2+q_0q_3)\,,\;1-2\left(q_2{}^2+q_3{}^2\right))\end{bmatrix}. \tag{13}$$

3.2. Dynamics

The operations of quadrotor are, in general, performed by four different movements. The movements are achieved by force differences or torque balances that is as known as control inputs in control engineering. However, the control inputs can be achieved by thrust balance with the help of the motors. For instance, U_2 depends on the produced thrust of motor 4 and 2 and it is liable for roll movement while U_3 can be achieved by balancing the thrust of motor 3 and 1 that impacts on pitch movement. In addition, U_4 and U_1 are the outcomes of the thrust of all motors while U_4 and U_1 are responsible for yaw and vertical movement respectively as follows:

$$U_1=k_f\left(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2\right), \tag{14}$$

$$U_2=k_f(\Omega_4^2-\Omega_2^2), \tag{15}$$

$$U_3=k_f\left(-\Omega_3^2+\Omega_1^2\right), \tag{16}$$

$$U_4=k_M(\Omega_1^2-\Omega_2^2+\Omega_3^2-\Omega_4^2), \tag{17}$$

where Ω_i =Angular velocity of i motor

Hence the initial conditions and nominal parameters has been introduced in Table 1 beforehand that will be helpful for later discussion about the work.

In dynamic model, Newton's second law can help to achieve the translational motion formula as follows [30]:

$$m\begin{pmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{pmatrix}=\begin{pmatrix}0\\0\\mg\end{pmatrix}+Q_{EB}\begin{pmatrix}0\\0\\-U_1\end{pmatrix}-k_t\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{pmatrix}. \tag{18}$$

Subsequently, another important formula, Newton-Euler equation is required to be introduced for rotational motion of quadrotor as follows where ω_r is the relative speed of the motor [30].

$$J\dot{\omega}=-\omega\times J\omega-\omega\times\begin{pmatrix}0\\0\\J_r\Omega_r\end{pmatrix}+\begin{pmatrix}lU_2\\lU_3\\U_4\end{pmatrix}-k_r\begin{pmatrix}p\\q\\r\end{pmatrix}, \tag{19}$$

where

$$\omega=[p,\;\;q,\;\;r]^T,$$

Table 1. Parameters and initial conditions for simulation [32].

Symbol	Value	Symbol	Value	Symbol	Value
J_x	7.5e-3 kg.m^2	k_{t_x}	0.1 Ns/m	k_{r_x}	0.1 Nm.s
J_y	7.5e-3 kg.m^2	k_{t_y}	0.1 Ns/m	k_{r_y}	0.1 Nm.s
J_z	1.3e-2 kg.m^2	k_{t_z}	0.15 Ns/m	k_{r_z}	0.15 Nm.s
J_r	6e-5 kg.m^2	k_f	3.13e-5 Ns^2	g	9.81 m/s^2
l	0.23 m	k_M	7.5e-7 Nms^2	m	0.65 kg

$\dot{\omega}$ = Derivative of angular velocity of quadrotor with respect to time,

$\Omega_r=-\Omega_1+\Omega_2-\Omega_3+\Omega_4.$

Finally, the following equations can be formulated from aforementioned quaternion derivative, translational motion equation and rotational motion equation as follows. However, the equations represents the complete dynamic model of quadrotor.

$$\ddot{x}=\frac{-1}{m}\left[k_{t_x}\dot{x}+U_1(2q_0q_2+2q_1q_3)\right], \tag{20}$$

$$\ddot{y}=\frac{-1}{m}\left[k_{t_y}\dot{y}-U_1(2q_0q_2-2q_1q_3)\right], \tag{21}$$

$$\ddot{z}=\frac{-1}{m}\left[k_{t_z}\dot{z}-mg+U_1(2q_0{}^2+2q_3{}^2-1)\right], \tag{22}$$

$$\dot{q}_0=\frac{1}{2}\left[-pq_1-qq_2-rq_3\right], \tag{23}$$

$$\dot{q}_1=\frac{1}{2}\left[pq_0-qq_3+rq_2\right], \tag{24}$$

$$\dot{q}_2=\frac{1}{2}\left[pq_3+qq_0-rq_0\right], \tag{25}$$

$$\dot{q}_3=\frac{1}{2}\left[-pq_2+qq_1+rq_0\right], \tag{26}$$

$$\dot{p}=\frac{-1}{I_x}\left[k_{r_x}p-lU_2-J_yqr+J_zqr+J_rq\Omega_r\right], \tag{27}$$

$$\dot{q}=\frac{-1}{I_y}\left[-k_{r_y}q+lU_3-J_xpr+J_zpr+J_rp\Omega_r\right], \tag{28}$$

$$\dot{r}=\frac{-1}{I_z}[U_4-k_{r_z}r+J_xpq-J_ypq]. \tag{29}$$

Significantly, equations (20)-(29) have been finally derived using MATLAB code. In this work, aerodynamic drag and moment have been considered for higher accuracy that sometimes are neglected in different literatures.

In a nutshell, the states for quadrotor are

$$X=x,y,z,\;\dot{x},\dot{y},\;\dot{z},\phi,\theta,\psi,\dot{\phi},\dot{\theta},\dot{\psi}.$$

3.3. Model identification

In this study, the parameters of quadrotor have been obtained using CAD model based on [32]. In addition, a Frequency Domain approach, MATLAB System Identification Toolbox has been adopted for rotor dynamics. Moreover, a simulator OS4 has also been developed depending on the present model that helps to find out sensor data, delay and noise based on [32]. Hence the model has been experimentally validated in the literature [32]. Furthermore, the model has been also mathematically validated by considering open loop simulation as mentioned in Appendix A.

4. MODEL PREDICTIVE CONTROL DESIGN

This section approaches the short description of Model Predictive Control algorithm with a view to giving a concise idea about its working procedure. MPC is especially chosen when it is a matter of carrying out high-level of trajectory tracking along with optimal control and requires online optimization under multivariable constrained as aforementioned [33, 34].

MPC achieved its popularity because of mainly its predicting behavior. Here, some mathematical equations have been developed in order to achieve the predicting behavior of MPC. The prediction carries out throughout a predefined time duration. MPC works by following the algorithm: i) Step 1: a linear state-space model is required to develop at first, ii) Step 2: at a certain time, **t**, future system outputs are predicted depending on the previous inputs and outputs behavior over a prediction horizon, iii) Step 3: the most desired behavior have been adopted from control signals, iv) Step 4: then over a predefined time interval, the control signals are applied and v) Step 5: afterwards it moves to the next time interval and then it repeats from step 2 again.

A figure can be offered to comprehend the complete procedure of MPC as follows in Fig. 2. During the working procedure, MPC gets familiar with the behavior of past inputs and outputs along with it receives the predicted future inputs in order to generate predicted outputs for each time period. Then, the predicted behavior is analogized to the desired trajectory in order to estimate the errors. An optimizer receives the errors and minimize the cost

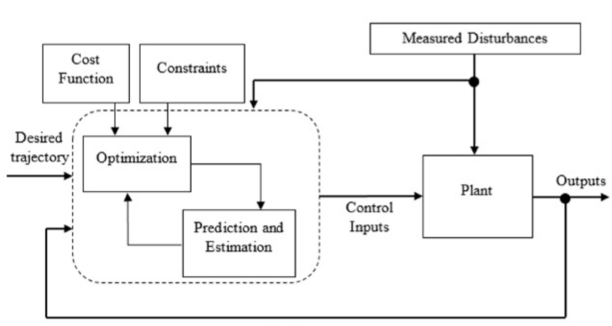


Fig. 2. MPC working flow diagram.

through cost function. As a result, the subsequent procedures assist to generate predicted future inputs that is provided as feedback for next up cycle.

Significantly, the role of MPC is to compute manipulated variables, U in order to optimize future outputs Y.

A step by step mathematical derivation can help to illustrate the complete process of MPC algorithm for quadrotor. Therefore, the section can be organized with two different sub-sections i.e., Methodological development and Numerical development. Here, methodological development will show the development of MPC algorithm mathematically and meanwhile, numerical development will demonstrate the numerical values that have been adopted for the designed model at different periods.

4.1. Methodological development

4.1.1 Plant model

A nonlinear system can be written in the form

$$\dot{X} = f(X(t), U(t)), \tag{30}$$

where $X(t)\varepsilon R^n$ denotes the states of the system and $U(t)\varepsilon R^m$ denotes system inputs.

In order to proceed according to the design procedure of MPC, a linear discrete-time system around a nominal point, more specifically a hovering point, has been derived as follows in (31) and (32) where nominal states and nominal control inputs are denoted as X_T and U_T respectively.

$$\Delta X_{k+1+i} = E\Delta X_{k+i} + F\Delta U_{k+i}, \tag{31}$$

$$\Delta Y_{k+i} = G\Delta X_{k+i} + D\Delta U_{k+i}, \tag{32}$$

where $i = 1, 2, 3, ..., N$, $\Delta X_k = X_k - X_T$ and $\Delta U_k = U_k - U_T$.

Here, current sample, state matrix, input matrix, system outputs, output matrix and feedforward matrix are respectively symbolized in the equations by k, $E \in R^{n\times n}$, $F \in R^{n\times m}$, $Y \in R^p$, $G \in R^{p\times n}$ and $D \in R^{p\times m}$. When the system is designed based on quaternion for quadrotor, then $n = 13$ and $m = 4$.

As MPC predicts the future behavior of the states within a prediction horizon, N gradually, it necessitates an estimator as follows [21]:

$$\begin{bmatrix} \Delta X_k \\ \Delta X_{k+1} \\ \Delta X_{k+2} \\ \vdots \\ \Delta X_{k+N-1} \end{bmatrix} = \begin{bmatrix} I \\ E \\ E^2 \\ \vdots \\ E^{N-1} \end{bmatrix} \Delta X_k + \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ F & 0 & \cdots & 0 & 0 \\ EF & F & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ E^{N-2}F & E^{N-3}F & \cdots & F & 0 \end{bmatrix} \times \begin{bmatrix} \Delta U_k \\ \Delta U_{k+1} \\ \Delta U_{k+2} \\ \vdots \\ \Delta U_{k+N-1} \end{bmatrix}, \tag{33}$$

$$\begin{bmatrix} \Delta Y_k \\ \Delta Y_{k+1} \\ \Delta Y_{k+2} \\ \vdots \\ \Delta Y_{k+N-1} \end{bmatrix} = \begin{bmatrix} G & 0 & 0 & \cdots & 0 \\ 0 & G & 0 & \cdots & 0 \\ 0 & 0 & G & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & G \end{bmatrix} \begin{bmatrix} \Delta X_k \\ \Delta X_{k+1} \\ \Delta X_{k+2} \\ \vdots \\ \Delta X_{k+N-1} \end{bmatrix} + \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ 0 & D & 0 & \cdots & 0 \\ 0 & 0 & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D \end{bmatrix} \begin{bmatrix} \Delta U_k \\ \Delta U_{k+1} \\ \Delta U_{k+2} \\ \vdots \\ \Delta U_{k+N-1} \end{bmatrix}. \quad (34)$$

The equations can be analogous to (35) and (36) in short form when the significance of D is considerable.

$$\Delta X_k = E_m \Delta X_k + F_m \Delta U_k, \quad (35)$$

$$\Delta Y_k = G_m \Delta X_k. \quad (36)$$

4.1.2 Control design

The MPC optimizer incorporates cost function to achieve minimal cost. However, the cost function can be minimized using two different gain matrices $\widehat{W}_U$ and $\widehat{W}_Y$ depending on the error between desired states and current states as follows: Noted that the following equation can be directly used on Euler angle based quadrotor where algebraic subtraction is adopted to calculate cost function in several literatures [34, 35].

$$J(\Delta x, \Delta u) = \left[(\Delta Y_k - \Delta Y_k^r) \hat{W}_Y \right]^2 + \left[(\Delta U_k \hat{W}_{\Delta U}) \right]^2, \quad (37)$$

where

$$\hat{W}_U = \begin{bmatrix} W_{U|0,1} & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \ddots & 0 & \ddots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & W_{U|0,m} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & W_{U|N-1,1} & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & W_{U|N-1,m} \end{bmatrix},$$

$$\hat{W}_Y = \begin{bmatrix} W_{Y|0,1} & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \ddots & 0 & \ddots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & W_{Y|0,m} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & W_{Y|N-1,1} & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & W_{Y|N-1,m} \end{bmatrix}.$$

Noted that quaternion error cannot be calculated by simple subtraction rather than its vector formula. Quaternion error is derived according to the formula as follows [11]:

$$q_{err} = \begin{bmatrix} q_{0,\mathrm{d}} & q_{1,\mathrm{d}} & q_{2,\mathrm{d}} & q_{3,\mathrm{d}} \\ -q_{1,\mathrm{d}} & q_{0,\mathrm{d}} & -q_{3,\mathrm{d}} & -q_{2,\mathrm{d}} \\ -q_{2,\mathrm{d}} & -q_{3,\mathrm{d}} & q_{0,\mathrm{d}} & q_{1,\mathrm{d}} \\ -q_{3,\mathrm{d}} & q_{2,\mathrm{d}} & -q_{1,\mathrm{d}} & q_{0,\mathrm{d}} \end{bmatrix} \begin{bmatrix} q_{0,\mathrm{a}} \\ q_{1,\mathrm{a}} \\ q_{2,\mathrm{a}} \\ q_{3,\mathrm{a}} \end{bmatrix}. \quad (38)$$

Therefore, the use of the Quaternion Error will impact on the previous cost function such as that in the present work it is proposed a new cost function for the quaternion-orientation based MPC controller as shown

$$J(\Delta x, \Delta u) = \left[\begin{bmatrix} \left(\Delta Y_{p,k} - \Delta Y_{p,k}^r\right) \\ \left(\Delta Y_{v,k} - \Delta Y_{v,k}^r\right) \\ \left(\Delta Y_{q,k}\right) \\ \left(\Delta Y_{\omega,k} - \Delta Y_{\omega,k}^r\right) \end{bmatrix} \hat{W}_{Y|\,13\times 13} \right]^2 + \left[\begin{bmatrix} \Delta U_{1,k} \\ \Delta U_{2,k} \\ \Delta U_{3,k} \\ \Delta U_{4,\,k} \end{bmatrix} \hat{W}_{\Delta U|\,4\times 4} \right]^2 + \left[\begin{bmatrix} \Delta \dot{U}_{1,k} \\ \Delta \dot{U}_{2,k} \\ \Delta \dot{U}_{3,k} \\ \Delta \dot{U}_{4,\,k} \end{bmatrix} \hat{W}_{\Delta \dot{U}|\,4\times 4} \right]^2, \quad (39)$$

where

$$\Delta Y_{p,k} = [\Delta Y_{1,k},\ \Delta Y_{2,k},\ \Delta Y_{3,k}]^T,$$
$$\Delta Y_{v,k} = [\Delta Y_{4,k},\ \Delta Y_{5,k},\ \Delta Y_{6,k}]^T,$$
$$\Delta Y_{q,k} = q_{err,\ k},$$
$$\Delta Y_{\omega,k} = [\Delta Y_{11,k},\ \Delta Y_{12,k},\ \Delta Y_{13,k}]^T.$$

Noted that $\Delta Y_{p,k}$, $\Delta Y_{v,k}$ and $\Delta Y_{\omega,k}$ indicate the differences between the current and desired positions, velocities and angular velocities respectively along x, y and z axes.

Here, in (38), an additional part, rate of control changes, $\Delta \dot{U}$ has been added in order improve the performance of cost function.

4.1.3 Quadratic programming

The role of a quadratic programming is to solve the optimization problem wherein it minimizes the cost function through a feasible search direction ΔU [36]

$$f(x) = \frac{1}{2} \Lambda^T H \Lambda + \varepsilon^T \Lambda, \quad (40)$$

where $\boldsymbol{A}X \geq b$, Λ denotes the solution vector, H is considered as Hessian matrix, $\boldsymbol{A}$ is a matrix of linear constraint efficient and ε and b are vectors.

4.1.4 Constraint handling

Since the motor of quadrotor has a limited rotation per minute (rpm), it will generate rpm until the certain limit and hence force will be generated accordingly. Consequently, the control inputs are bound to the limits. Hence, constraints are required to be handled at control input in order to design the control algorithm properly. However, when an upper bound, u_{ub} and a lower bound, u_{lb} at the control inputs where $U_{lb} \leq U_{k+i} \leq U_{ub}$ for $i = 0,\ 1,\ 2,\ \ldots,$

$N-1$. Therefore, the constraints at control inputs can be described as follows, where $I_{m\times m}$ is an identity matrix

$$\begin{bmatrix} I_{m\times m} \\ -I_{m\times m} \end{bmatrix} \Delta U_{k+i} \leq \begin{bmatrix} U_{ub}-\Delta U_T \\ -(U_{lb}-\Delta U_T) \end{bmatrix}. \tag{41}$$

Equation (41) can be represented by (42) that represents the complete procedure specifically on bounded limits.

$$I_U \Delta U_k \leq \Delta U_b, \tag{42}$$

where

$$I_U = \begin{bmatrix} \begin{bmatrix} I_{m\times m} \\ -I_{m\times m} \end{bmatrix} & 0 & \cdots & 0 \\ 0 & \begin{bmatrix} I_{m\times m} \\ -I_{m\times m} \end{bmatrix} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \begin{bmatrix} I_{m\times m} \\ -I_{m\times m} \end{bmatrix} \end{bmatrix},$$

$$\Delta U_b = \begin{bmatrix} \begin{bmatrix} U_{ub}-\Delta U_T \\ -(U_{lb}-\Delta U_T) \end{bmatrix} \\ \begin{bmatrix} U_{ub}-\Delta U_T \\ -(U_{lb}-\Delta U_T) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} U_{ub}-\Delta U_T \\ -(U_{lb}-\Delta U_T) \end{bmatrix} \end{bmatrix}.$$

4.2. Numerical development

Prior to proceed on simulation part, essentially MPC is is validated based on some factors like the stability of plant, sample time, control horizon, prediction horizon, overshoot and settling time around a certain point. Therefore, firstly, an operating point is required to be selected for linearization and here, the linearized model has been designed at a hovering point, z = 1m. Now it must be confirmed that the system is stable around the operating point before going to further actions and it can also be confirmed through the poles of the plant. When the real numbers of the poles of the plant are confirmed as placed in left side of s-plane, it indicates the real numbers are negative and therefore, it can be confirmed that the plant is stable where the poles are $[0, 0, 0, -0.1538, -0.1538, -0.2308, 0, 0, 0, -13.3333, -13.3333, -11.5385]$.

Along with, sampling time, prediction horizon and control horizon is necessarily chosen suitably based on some trials wherein suitability were subjective to overshoot and settling time at a certain tracking point as considered $x=3$ m, $y=2$ m, $z=-5$ m, $q_0=0.991$ and $q_3=0.131$. The following Tables 2 and 3 help to choose proper prediction horizon, control horizon and sampling time.

From Table 2, two pairs e.g., N = 20, C = 2 and N = 25, C = 2, have been considered as the best suited N and C based on the performance of the states. In additon, sampling time is also a required factor for the MPC designing.

Table 2. Effects of prediction horizon, N and control horizon, C on the settling time and overshoot along axes.

N	C	Position					
		x-axis		*y*-axis		*z*-axis	
		Settling time (s)	Over shoot (%)	Settling time (s)	Over shoot (%)	Settling time (s)	Over shoot (%)
15	2	5.1	3.5	7.0	10.4	6.2	4.7
15	3	12.9	7.5	19.3	30.8	7.1	0
20	**2**	**4.4**	**1.6**	**8.1**	**3.9**	**4.9**	**2.2**
20	3	7.1	7.6	10.9	11.1	6.4	0
25	**2**	**5.5**	**1.5**	**9.2**	**2.8**	**4.2**	**0.0**
25	3	4.3	4.8	9.1	7.7	6.3	0.0
30	2	6.6	1.5	10.3	2.3	5.4	0.0
35	2	7.6	1.4	7.9	1.9	6.2	0.0
40	2	9.2	1.1	9.0	1.4	4.1	0
45	2	9.9	1.0	10.2	1.1	7.5	0

Based on the best-suited N and C, several trials have also been considered in order to determine suitable sampleing time as mentioned in Table 3.

A quick review on the overall process can give an idea about the most suitable sampling time, 0.20 s, prediction horizon, 25 and control horizon, 2 are determined through a trade-off between overshoot and settling time wherein more than 5% overshoot has not been tolerated.

In methodological development section, the limits of the control inputs have been discussed in Fig. 3 because it is sometimes very important in order to save the electronic devices specially in quadrotor it is considered as motors from overvoltage. According to motor specification, it can generate 600 rad/s smoothly without any risk of burn out [23]. Hence, from (14)-(17), it can be fringed around the control inputs as follows:

$$0 < U_1 < 45.0720, \tag{43}$$
$$-11.2680 < U_2 < 11.2680, \tag{44}$$
$$-11.2680 < U_3 < 11.2680, \tag{45}$$
$$-0.54 < U_4 < 0.54. \tag{46}$$

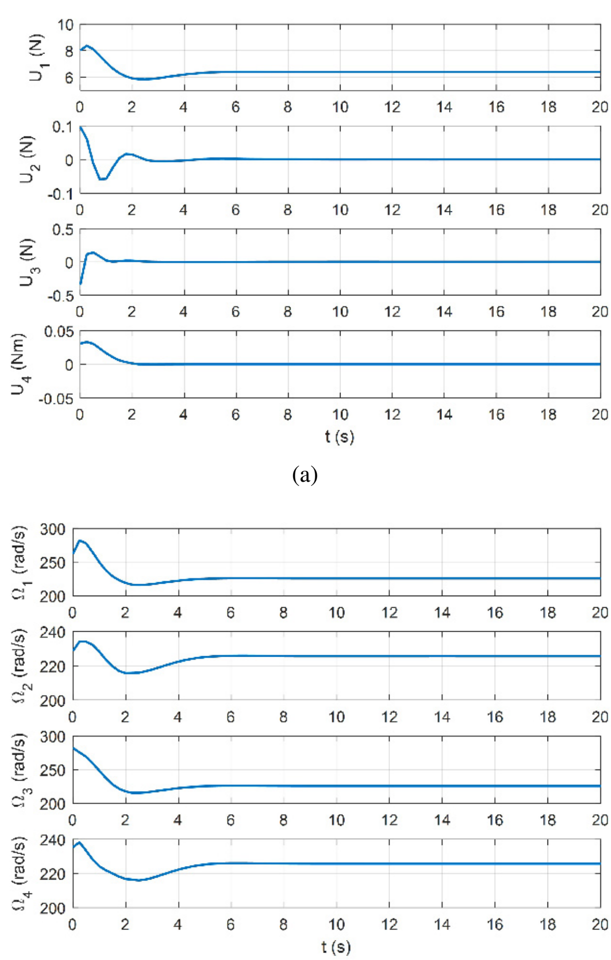
Now, another challenge is to find out the gain matrices properly after fulfilling the preliminary requirement of MPC algorithm. Finally, the gain matrices have been found out as follows:

$$\hat{W}_{\Delta U} = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix},$$

$$\hat{W}_{\Delta \dot{U}} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix},$$

Table 3. Effects of sampling time, prediction horizon and control horizon on the settling time and overshoot along axes.

Sample time	N	C	Position					
			x-axis		y-axis		z-axis	
			Settling time (s)	Over shoot (%)	Settling time (s)	Over shoot (%)	Settling time (s)	Over shoot (%)
0.10	20	2	5.7	5.1	8.1	23.9	5.0	1.0
0.15	20	2	4.9	3.9	9.6	22.1	6.3	3.6
0.20	20	2	4.8	2.3	7.2	6.9	6.0	4.2
0.25	20	2	4.4	1.7	8.1	3.9	4.9	2.2
0.30	20	2	5.25	1.6	9.1	3.1	3.8	0.0
0.35	20	2	6.1	1.6	10.1	2.5	4.9	0.0
0.40	20	2	6.9	1.5	10.9	2.2	5.7	0.0
0.10	25	2	4.6	4.0	11.4	25.5	5.5	2.1
0.15	25	2	5.0	2.9	6.9	10.5	6.2	4.1
0.20	**25**	**2**	**4.4**	**1.5**	**8.0**	**3.9**	**3.4**	**1.8**
0.25	25	2	5.5	1.6	9.2	2.8	4.2	0.0
0.30	25	2	6.5	1.5	10.4	2.3	5.4	0.0
0.35	25	2	7.6	1.4	7.9	1.9	6.2	0.0
0.40	25	2	8.6	1.3	9.1	1.6	6.9	0



(a)

(b)

Fig. 3. (a) Control inputs and (b) angular velocities of the motor at $x = 3$ m, $y = 2$ m and $z = -5$ m.

$$\hat{W}_Y = diag([10^{-1}, 5\times10^{-3}, 5\times10^{-2}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 3\times10^{-2}, 0, 0, 0]).$$

5. SIMULATION

The performance of controller can be checked finally through some devised trajectories such as square trajectory in Fig. 4, double helix in Fig. 5 and lissajous trajectory in Fig. 6 after the complete development of controller Subsequently, the performances of the controller along square trajectory, double helix trajectory and lissajous trajectory have been illustrated in chronological order.

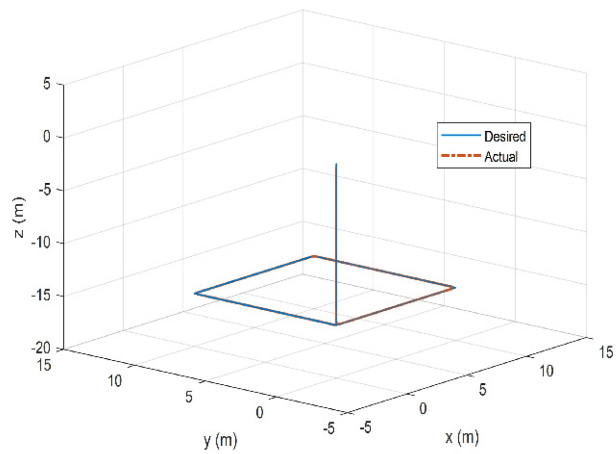
For square trajectory, it is considered that quadrotor starts to fly upwards for 15 m along z-axis and it returns back again to its initial condition after traveling a square trajectory of flight duration, 300 s.

In double helix trajectory, the quadrotor starts its flight for 230 s from ground to 28.2 m height following a helical turn and then moves 5 m along x-axis in order to complete next helical turn.

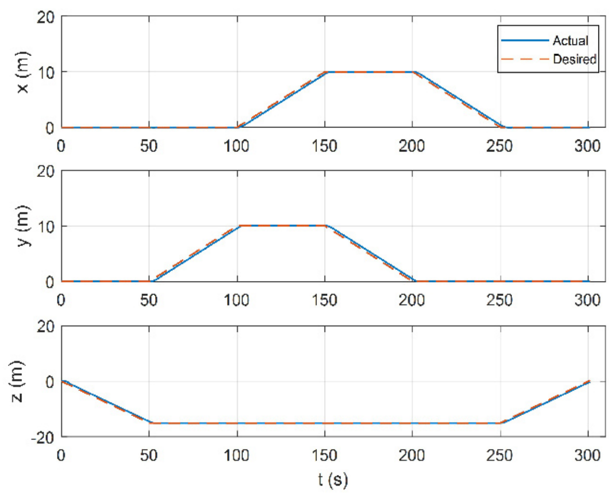
Next, a lissajous trajectory has been developed also in order to check the controller's performance finally wherein quadrotor needs agile movement in such a trajectory.

Fig. 7 portrays the angular velocities while Fig. 8 illustrates the control inputs in different trajectories that confirms that MPC is successfully able to maintain the constraints in control inputs.

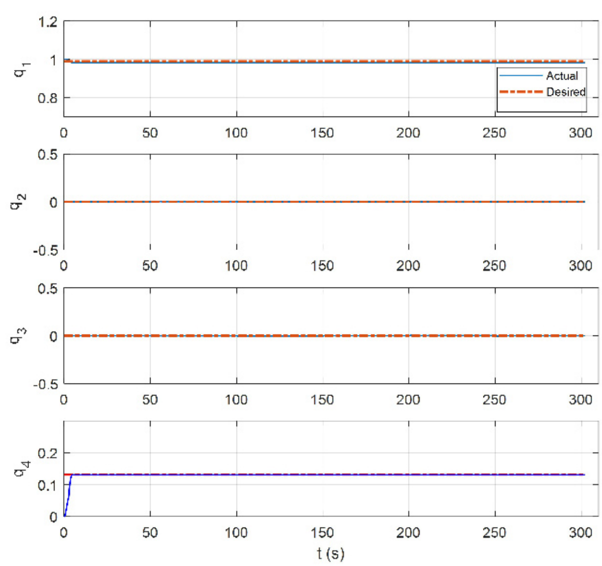
Meanwhile, it is also required to verify the tracking error of MPC in order to understand the performance of MPC according to different situations. In consequence, Root Means Square Error (RMSE) can help to find out the trajectory error along each axis that is one of the most



(a)

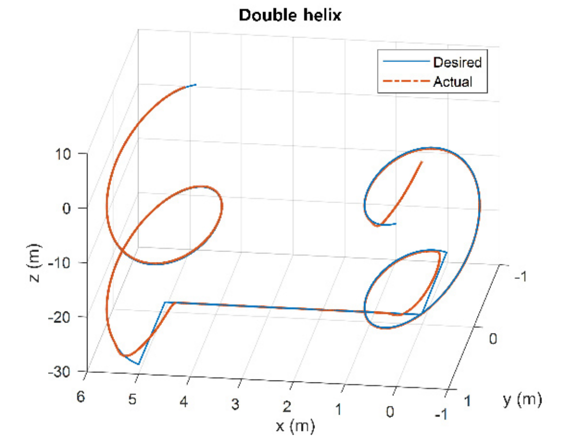


(b)

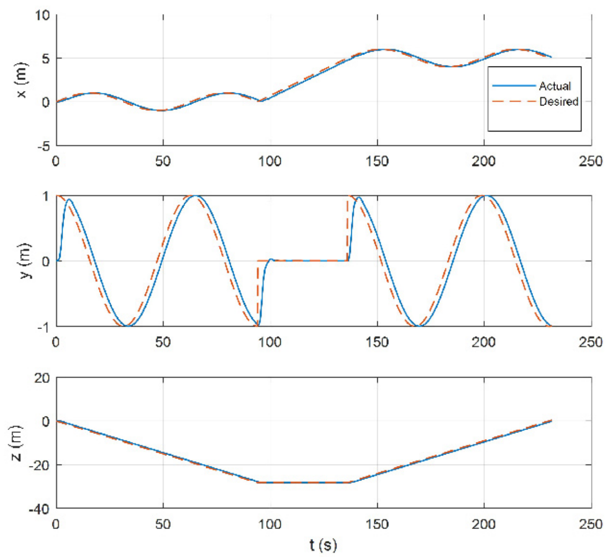


(c)

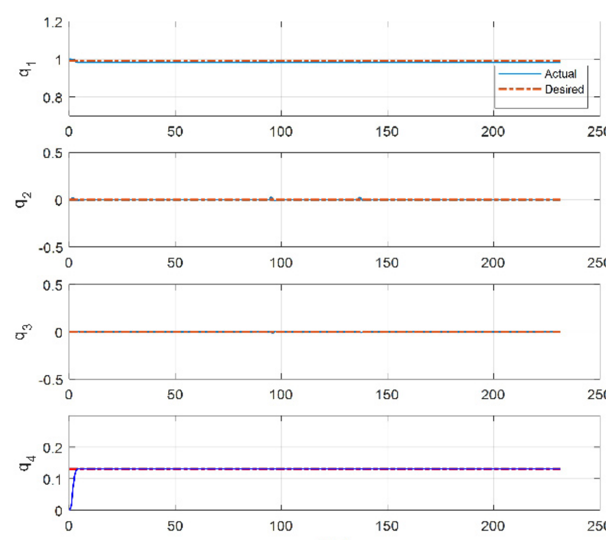
Fig. 4. (a) 3D position, (b) position, and (c) attitude of square trajectory.



(a)

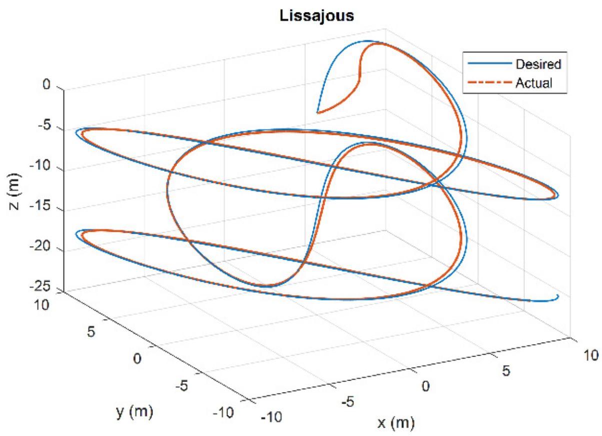


(b)

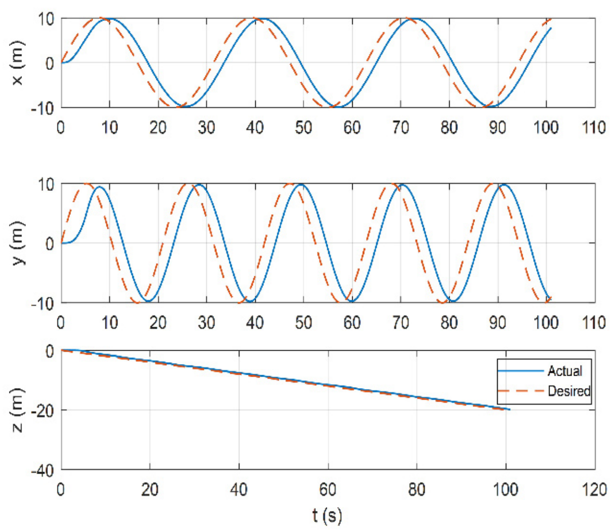


(c)

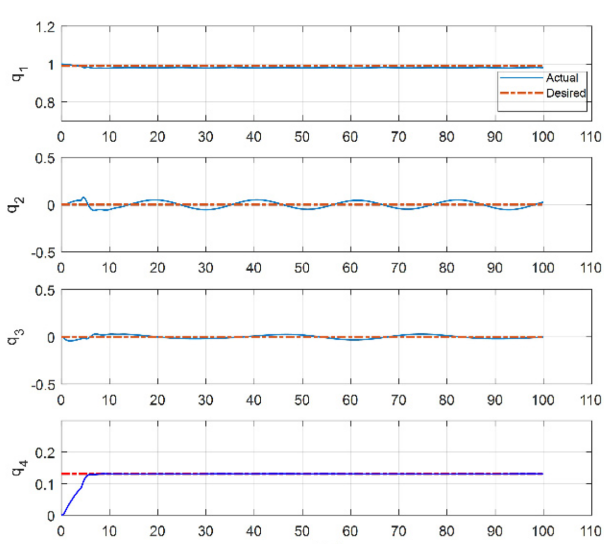
Fig. 5. (a) 3D position, (b) position (c) attitude of double helix trajectory.



(a)

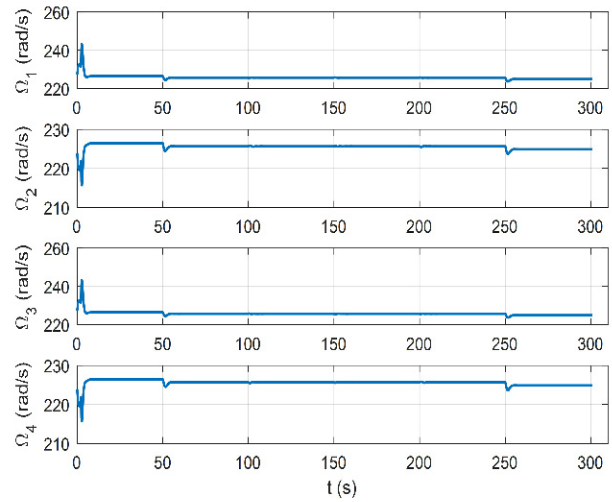


(b)

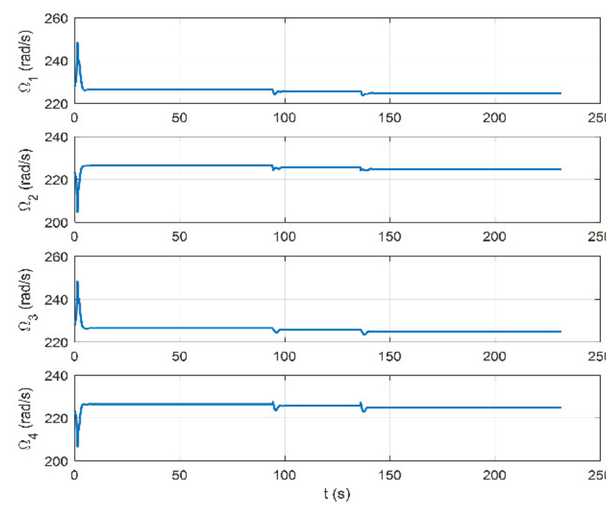


(c)

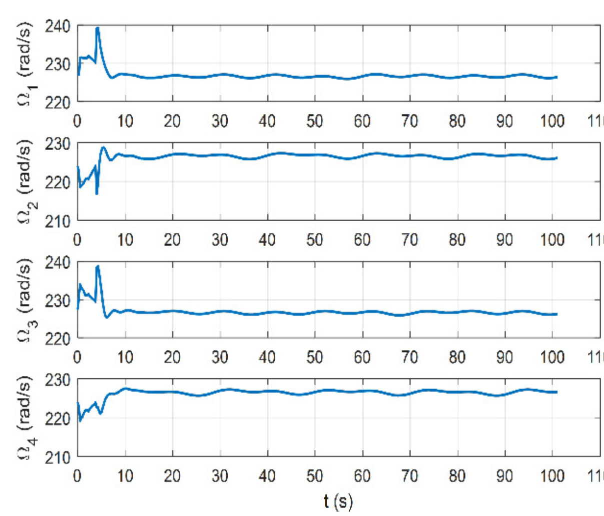
Fig. 6. (a) 3D position, (b) position, and (c) attitude of Lissajous trajectory.



(a)

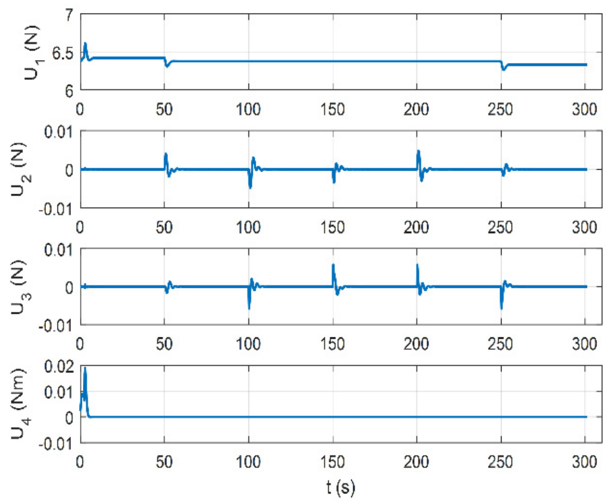


(b)

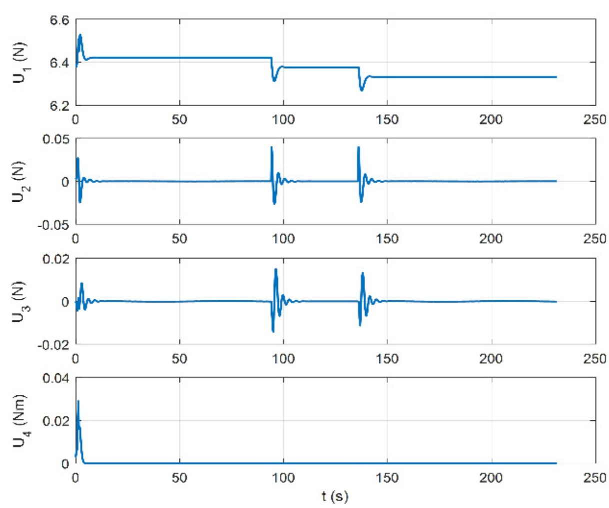


(c)

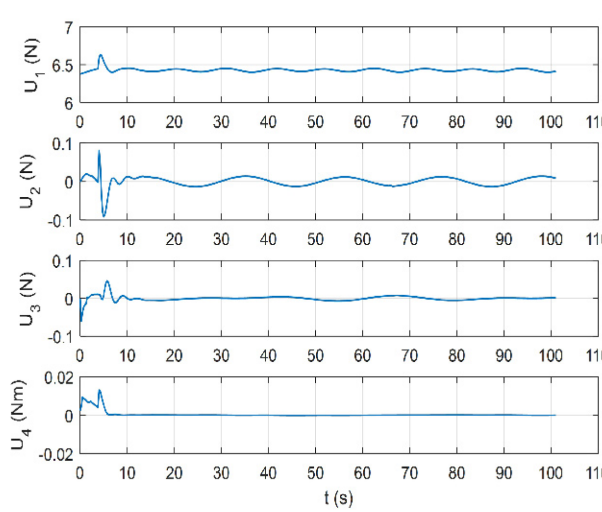
Fig. 7. Angular velocities in (a) square trajectory, (b) double helix trajectory, and (c) Lissajous trajectory.



(a)



(b)



(c)

Fig. 8. Control inputs in (a) square trajectory, (b) double helix trajectory, and (c) Lissajous trajectory.

commonly used methods. RMSE can be expressed by the following equation:

$$\mathrm{RMSE} = \sqrt{\left(\frac{(y_a - y_e)^2}{\text{length of } y}\right)},$$

where

y_a = Actual values of y,

y_e = Estimated values of y.

Table 4 shows the RMSE of the trajectories along three axes for two different circumstances e.g., under disturbance and without disturbance. Noted that additional disturbances, $[0.5, 0.3, 0.2, 0.1]$ have been considered at control input that is known to the MPC model based on the literature [37] while the earlier considered disturbances in equation 18 and 19 are unknown to MPC model. From the Table 4, it is noticeable that the difference of RMSE between two different situations are very insignificant. Moreover, in both the circumstances, surprisingly the RMSEs along all axes are less than 5%.

However, from the figures, a delay from the controller's response can be noticed because of the consideration of air drag. As this study proposes realistic plant with MPC controller, air drag and additional disturbance have been considered while the controller is not introduced with the air drag in advance. Noted that similar delay response because of air drag is also found in literatures [38].

6. CONCLUSION

This work shows the successful development of MPC controller using quaternion on quadrotor platform. Moreover, in this study, quadrotor successfully tracks the trajectories avoiding disturbance with satisfactory tracking error through maintaining constraints at control inputs.

Necessarily, the quadrotor cannot avoid the disturbances that exist in the plant and is not introduced to the controller and however, as a result, the system shows delay during tracking.

Significantly it is mentioned that the mathematical model of quadrotor has been developed considering aerodynamic moment and drag that sometimes are found ne-

Table 4. RMSE for different trajectories.

Trajectories	RMSE without disturbances			RMSE with disturbances		
	x-axis (%)	y-axis (%)	z-axis (%)	x-axis (%)	y-axis (%)	z-axis (%)
Square	0.1	0.1	0.0	0.1	0.1	0.0
Double helix	1.2	4.1	0.1	1.2	4.1	0.1
Lissajous	2.2	1.6	1.5	2.2	1.6	1.6

glected in literatures [39, 40]. The presence of aerodynamic moment and drag help to comprehend the system in real-time situation. Noted that the numerical value may vary from current value.

In addition, the adoption of quaternion with MPC in this work gives novelty and makes it appreciably different from other works on quadrotor where most of the works are found based on Euler angle orientation system. It is noticed during simulation that quaternion requires less computation time with less tracking error comparing to previous study on Euler-angle based MPC approach.

However, it is also acknowledged that MPC takes much computation time while other controller like LQR or PID requires less time. Meanwhile, it is conceded that the constraints handling and disturbances handling with such a minimal tracking error cannot be expected form the controllers as well.

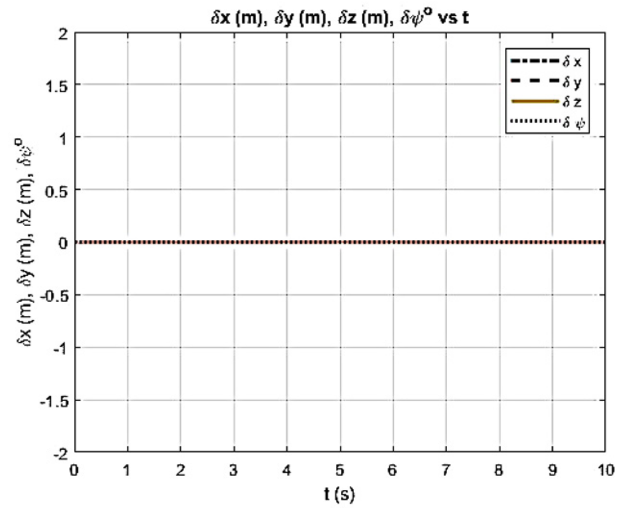
As the work is based on simulations, an important direction as future research is to perform real-time experiment. Additionally, gain-scheduling MPC approach can also be investigated in future work to deal with nonlinear system in more realistically.

NOMENCLATURE

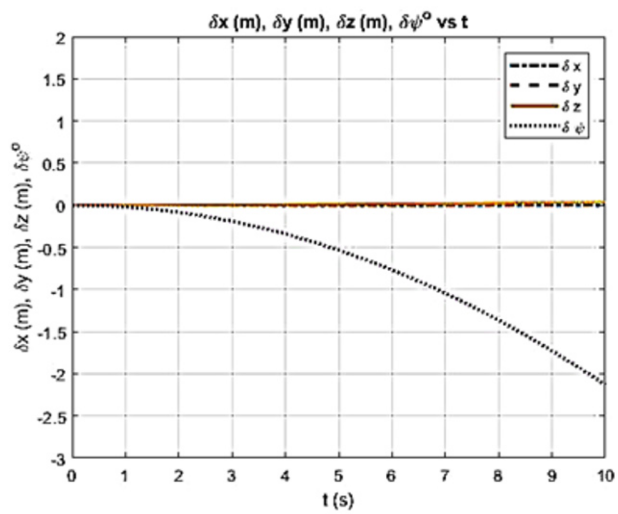
The following symbols are used in this paper:
g =Gravity,
J =Moment of inertia,
J_r =Inertia of motor,
k_f =Thrust coefficient,
k_M =Moment coefficient,
k_r =Aerodynamic moment drag coefficient,
k_t =Aerodynamic thrust drag coefficient,
l =Arm length,
m =Mass of quadcopter,
Ω_i =Angular velocity of i motor.

APPENDIX A: OPEN LOOP SIMULATION

Two open simulations are accomplished for both linear and nonlinear model of quadrotor where U_1, U_2, U_3 and U_4 were considered as $[mg, 0, 0, 0.0001]^T$ and $[mg+0.001, 0, 0, 0.0001]^T$ in order to check the compatibility of linear model with nonlinear model. Significantly, from Fig. 9(b), it is found that the behavior of linear model started to change after 1.5s because the operating point is different. In contrast, in Fig. 9(a), linear model behaves very smoothly and almost the responses are same to nonlinear model as their operating points are same. Noted that yaw angle according to Euler angle orientation is considered 15° (in quaternion $q_0 = 0.991$ and $q_3 = 0.131$) in order to show the angle difference easily. Here, δx, δy, δz and $\delta\psi^\circ$ shows the difference between linear model and nonlinear model along position x, y, z and yaw angle, ψ°.



(a)



(b)

Fig. 9. Open loop simulation where control inputs are (a) $[mg, 0, 0, 0.0001]^T$, (b) $[mg+0.001, 0, 0, 0.0001]^T$.

REFERENCES

[1] P. Foehn and D. Scaramuzza, "Onboard state dependent LQR for Agile Quadrotors," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 6566-6572, 2018.

[2] E. Hernandez-Martinez, G. Fernandez-Anaya, E. Ferreira, J. Flores-Godoy, and A. Lopez-Gonzalez, "Trajectory tracking of a quadcopter UAV with optimal translational control," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 226-231, 2015.

[3] A. K. Shastry, M. T. Bhargavapuri, M. Kothari, and S. R. Sahoo, "Quaternion based adaptive control for package delivery using variable-pitch quadrotors," *Proc. of Indian Control Conference (ICC)*, IEEE, pp. 340-345, 2018.

[4] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," *Abstract and Applied Analysis*, vol. 2014, Article ID 320526, 16 pages, 2014.

[5] H. Parwana, J. S. Patrikar, and M. Kothari, "A novel fully quaternion based nonlinear attitude and position controller," *Proc. of AIAA Guidance, Navigation, and Control Conference*, p. 1587, 2018.

[6] S. Swarnkar, H. Parwana, M. Kothari, and A. Abhishek, "Biplane-quadrotor tail-sitter UAV: flight dynamics and control," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1049-1067, 2018.

[7] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," *Proc. of European Control Conference (ECC)*, IEEE, pp. 3864-3869, 2013.

[8] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315-378, 2012.

[9] M. Islam, M. Okasha, and M. M. Idres, "Trajectory tracking in quadrotor platform by using PD controller and LQR control approach," *IOP Conference Series: Materials Science and Engineering*, vol. 260, no. 1, p. 012026, 2017.

[10] C. J. Ostafew, *Learning-based Control for Autonomous Mobile Robots*, Doctor of Philosophy, University of Toronto, Canada, 2016.

[11] A. Kehlenbeck, *Quaternion-based Control for Aggressive Trajectory Tracking with a Micro-quadrotor UAV*, University of Maryland, College Park, 2014.

[12] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard, "LQR control for a quadrotor using unit quaternions: Modeling and simulation," *Proc. of International Conference on Electronics, Communications and Computing (CONIELECOMP)*, IEEE, pp. 172-178, 2013.

[13] K. Djamel, M. Abdellah, and A. Benallegue, "Attitude optimal backstepping controller based quaternion for a uav," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[14] C. Zha, X. Ding, Y. Yu, and X. Wang, "Quaternion-based nonlinear trajectory tracking control of a quadrotor unmanned aerial vehicle," *Chinese Journal of Mechanical Engineering*, vol. 30, no. 1, pp. 77-92, 2017.

[15] T.-T. Tran and C. Ha, "Self-tuning proportional double derivative-like neural network controller for a quadrotor," *International Journal of Aeronautical and Space Sciences*, vol. 19, no. 4, pp. 976-985, 2018.

[16] C. Liu, H. Lu, and W.-H. Chen, "An explicit MPC for quadrotor trajectory tracking," *Proc. of the 34th Chinese Control Conference (CCC)*, IEEE, pp. 4055-4060, 2015.

[17] W. Zhao and T. H. Go, "Quadcopter formation flight control combining MPC and robust feedback linearization," *Journal of the Franklin Institute*, vol. 351, no. 3, pp. 1335-1355, 2014.

[18] M. Bangura and R. Mahony, "Real-time model predictive control for quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11773-11780, 2014.

[19] M. Chipofya, D. J. Lee, and K. T. Chong, "Trajectory tracking and stabilization of a quadrotor using model predictive control of Laguerre functions," *Abstract and Applied Analysis*, vol. 2015, Article ID 916864, 11 pages, 2015.

[20] K. Alexis, G. Nikolakopoulos, and A. Tzes, "On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances," *Asian Journal of Control*, vol. 16, no. 1, pp. 209-224, 2014.

[21] M. Islam, M. Okasha, and M. Idres, "Dynamics and control of quadcopter using linear model predictive control approach," *IOP Conference Series: Materials Science and Engineering*, vol. 270, no. 1, p. 012007, 2017.

[22] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 528-535, 2016.

[23] C. Kanellakis, S. S. Mansouri, and G. Nikolakopoulos, "Dynamic visual sensing based on MPC controlled UAVs," *Proc. of the 25th Mediterranean Conference on Control and Automation (MED)*, IEEE, pp. 1201-1206, 2017.

[24] T. Engelhardt, T. Konrad, B. Schäfer, and D. Abel, "Flatness-based control for a quadrotor camera helicopter using model predictive control trajectory generation," *Proc. of the 24th Mediterranean Conference on Control and Automation (MED)*, IEEE, pp. 852-859, 2016.

[25] A. Chovancová, T. Fico, P. Hubinský, and F. Duchoň, "Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator," *Robotics and Autonomous Systems*, vol. 79, pp. 87-98, 2016.

[26] J.-F. Guerrero-Castellanos, J. J. Téllez-Guzmán, S. Durand, N. Marchand, J. Alvarez-Muñoz, and V. R. Gonzalez-Diaz, "Attitude stabilization of a quadrotor by means of event-triggered nonlinear control," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 123-135, 2014.

[27] A. Sudbery, "Quaternionic analysis," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 85, no. 2, pp. 199-225, Cambridge University Press, 1979.

[28] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1-35, 2006.

[29] A. Chovancová, T. Fico, L'. Chovanec, and P. Hubinsk, "Mathematical modelling and parameter identification of quadrotor (a survey)," *Procedia Engineering*, vol. 96, pp. 172-181, 2014.

[30] S. Lindblom and A. Lundmark, *Modelling and Control of a Hexarotor UAV*, Linköpings universitet, 2015.

[31] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

[32] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," 2007.

[33] J. Zhang, X. Cheng, and J. Zhu, "Control of a laboratory 3-DOF helicopter: Explicit model predictive approach," *International Journal of Control, Automation and Systems*, vol. 14, no. 2, pp. 389-399, 2016.

[34] M. H. Murillo, A. C. Limache, P. S. R. Fredini, and L. L. Giovanini, "Generalized nonlinear optimal predictive control using iterative state-space trajectories: Applications to autonomous flight of UAVs," *International Journal of Control, Automation and Systems*, vol. 13, no. 2, pp. 361-370, 2015.

[35] A.-W. A. Saif, A. Aliyu, M. Al Dhaifallah, and M. Elshafei, "Decentralized Backstepping Control of a Quadrotor with Tilted-rotor under Wind Gusts," *International Journal of Control, Automation and Systems*, vol. 16, no. 5, pp. 2458-2472, 2018.

[36] Mathworks, *QP Solver*, 2018. Available: https://www.mathworks.com/help/mpc/ug/qp-solver.html

[37] I. Kugelberg, *Black-box Modeling and Attitude Control of a Quadcopter*, Master of Science Thesis, Linköping University, 2016.

[38] Y. Wang, A. Ramirez-Jaime, F. Xu, and V. Puig, "Nonlinear model predictive control with constraint satisfactions for a quadcopter," *Journal of Physics: Conference Series*, vol. 783, no. 1, p. 012025, 2017.

[39] W. Zhu, H. Du, Y. Cheng, and Z. Chu, "Hovering control for quadrotor aircraft based on finite-time control algorithm," *Nonlinear Dynamics*, vol. 88, no. 4, pp. 2359-2369, 2017.

[40] L. V. Santana, A. S. Brandão, and M. Sarcinelli-Filho, "Navigation and cooperative control using the ar. drone quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 327-350, 2016.

Maidul Islam completed his BSc. degree in Aeronautical Engineering from Military Institute of Science and Technology, Bangladesh in 2015. Later he obtained his MSc. in Mechanical Engineering from International Islamic University Malaysia, Malaysia in 2019. Now he is a Ph.D. candidate at the same university. His research interest goes to control system, smart material and UAV (Unmanned Aerial Vehicle). In his academic career, he could manage to publish 8 articles according to his research interests.

Mohamed Okasha obtained his Bachelor and Master degrees from Aerospace Dept. at Cairo University, Egypt. He started his career as a system engineer at the National Authority of Remote Sensing and Space Sciences (NARSS). He received a 2 year on-job-training from Youshnoi Design office at Dnepropetrovsk, Ukraine on dynamic and control of space vehicles. He received a Ph.D. from Mechanical and Aerospace Engineering from Old Dominion University, USA. Later, he worked as a research associate at Yonsei University, Korea. Currently, he is an Assistant Professor at Mechanical Engineering Dept. IIUM.

Erwin Sulaeman earned his Bachelor degree from Institut Teknologi Bandung, Indonesia in 1986. He started his research interest in structural dynamics, unsteady aerodynamics and aeroelasticity since 1987 when he worked for IPTN, Indonesian Aircraft Industry until 2010. During the time, he received scholarship from the Indonesian Ministry of Research and Technology to pursue his post graduate studies. He was graduated in 1993 in Master in Aerospace Engineering from the University of Dayton, USA. In 2002 he obtained his Ph.D. degree in Aerospace Engineering degree from Virginia Polytechnic Institute and State University, USA. Since 2010, he has been working as an Associate Professor in Mechanical Engineering Department of IIUM. He also served as the Section Editor of IIUM Engineering Journal.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.