Integrated SMC-MPC Design for Path Tracking of Quadcopters

Amirreza Bagherzadeh and Afshin Banazadeh*

Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran
Email: abb6024@psu.edu (A.B.); banazadeh@sharif.edu (A.B.)
*Corresponding author

Manuscript received June 5, 2023; revised July 30, 2023; accepted September 25, 2023; published April 8, 2024.

***Abstract*—A controller design with integration of Sliding Mode Control (SMC) and Model Predictive Control (MPC) has been proposed here for path-tracking and attitude control of multirotors. Sliding mode control has been used for tracking desired attitude and altitude under modeling errors and external disturbances. Model predictive control has also been designed to track the desired path in the horizontal plane of motion, which has been constrained to have feasible values of reference roll and pitch angles. In addition, some considerations have been made on the control design integration and thrust force constraints. This technique has been examined by tracking two different desired trajectories and checking the integral of the absolute error. It is shown that the proposed technique has outstanding performance under the external disturbances and control saturation.**

***Keywords*—Sliding Mode Control (SMC), Model Predictive Control (MPC), quadcopter, external disturbances**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have been used in many projects in military and civil sectors. More and more research efforts in companies and startups are being focused on using UAVs as their trump card in refereed projects. Among many types of UAVs, multirotors and specially quadcopters are spreading more rapidly than ever before. Simplicity in design and construction, vertical takeoff and landing capability, hovering ability, and precise path tracking are among the benefits that make the use of these devices, especially in photography, photogrammetry, monitoring of places, and making 3-D models from buildings [1, 2].

Stability and control of a quadcopter, as an under-actuated system is challenging due to highly coupled and nonlinear dynamics and uncertainty in parameters susceptible to external disturbances [3–5]. To conquer the problems, a large amount of research has been performed using various control strategies by linear and nonlinear techniques, including sliding mode control, model-based control, robust control and adaptive control [5, 6].

Sliding mode control, which is based on lyapunov stability theorem is simple and efficient. Elhennawy *et al*. [7] used second order sliding mode control to track the desired trajectory for quadcopter over the disturbances on the z-axis. Tripathy et al. have compared sliding mode controller and back stepping controller with the traditional PID controller [8]. Abrougui *et al*. [9] used two-stage sliding mode controller for a hexacopter. As the generality of the lyapunov theorem, many innovative and adaptive control designs are recently being developed. Ullah *et al*. used sliding mode control as a kernel in neural network in the presence of parameter uncertainties and disturbances and compared the results with the conventional Disturbance Observer-Based Sliding Mode Control (DOB-SMC) [10]. Alattas *et al.* [11] introduced a barrier function adaptive non-singular terminal sliding mode controller under matched disturbances. Eltayeb *et al*. [12] used Improved Adaptive Sliding Mode Controller (IASMC) with adaptive switching gain and compared the results with Adaptive Sliding Mode Controller (ASMC).

The strategies based on optimal control methodology are being used in several papers. Kawamura *et al.* designed control and guidance strategy with integrating a cost function, solved the optimization problem, and compared the results with a classical PID control [13]. Massé *et al.* developed a structured $\mathcal{H}_\infty$ optimal controller and compared the results with a classical LQR controller [14]. Ibrahim et al. derived offline approximate solution for an optimal learn-based control problem in the presence of wind blowing [15]. Oscar proposed an approach for quadcopter to fly inside the gates with solving the optimal control problem and showed its performance over this mission [16].

Model Predictive Control or moving horizon control is another strategy used by researchers. Three steps for model predictive control are; using the linear or nonlinear model to predict the system output along a future time horizon, optimization of performance index to calculate the control sequence, and a receding horizon strategy, so that at each step the horizon is been moved towards the future, which involves the application of the first control signal of the sequence calculated at each step [17].

Considering the type of the model used in the algorithm, MPCs are divided into two categories: Linear Model Predictive Control (LMPC) and Nonlinear Model Predictive Control (NMPC). Ganga *et al.* [18] used linear MPC and compared its results with a PID controller. It was shown that linear model predictive control is more efficient than PID control for tracking the desired trajectory. Islam *et al.* [19] used linear MPC under disturbances and model uncertainties. Eltrabyly *et al*. [20] designed an active Fault Tolerant Control (AFTC), which consisted of a Nonlinear Model Predictive Control (NMPC) with a Fault Detection and Diagnosis (FDD) module with respect to loss of effectiveness of the actuators. Merabti *et al.* [21] designed a nonlinear model predictive control with a particle swarm optimization algorithm to decrease the time cost of solving the optimization problem. Ru *et al*. [22] discussed about derivation and implementation of a nonlinear model predictive control for tracking reference trajectory. They transformed the system nonlinearities into pseudo-linear system matrix in the form of linear matrix

DOI: 10.7763/IJMO.2024.V14.849

inequality (LMI).

All those attempts have tried to design and implement a control algorithm to stabilize the quadcopter, and track the desired trajectory with the least tracking error. Most of the goals in the design of the controller for UAVs are; least tracking error, least control effort, maximum stability in flight, and least computation cost. However, all these could not be achieved by a single control technique.

Sliding mode controller is powerful in stabilizing the dynamics. It is also robust against the disturbances and unmodeled dynamics and has low computational costs. Nevertheless, it has low performance under the presence of actuator saturation and discretization in control loops. In addition, it depends on the given set-point. So, any change in the set-point will result in a high control effort and actuator saturation or high control gains, which may lead to instability or, at least, consume a large amount of energy.

Model predictive control is utilized for tracking the trajectory by prediction of system outputs at future time horizon. So, the possibility of saturation and the jump in actuator set-point will be decreased. In addition, it decreases the control effort in its optimization process. However, it has large computational cost, as it would take a large memory for computation, which would increase the response time and even will result in a failure. In addition, it is not robust to disturbances or the unmodeled dynamics due to dependency of this algorithm to the model.

This paper attempts to design an integrated double-loop SMC-MPC controller. The inner loop, which uses sliding mode control, serves to stabilize the quadcopter and control the attitude and the altitude. The outer loop, which uses linear model predictive control, serves to track the desired trajectory in the horizontal plane of motion by controlling the pitch and roll of the quadcopter.

II. Dynamic Modeling

To construct a six-degree of freedom nonlinear dynamic model of the quadcopter two main frames are considered; earth-fixed inertial frame as $\boldsymbol{E}$ and body-attached frame as $\boldsymbol{B}$. The aerodynamic forces are also assumed to be absorbed in disturbance forces. $[x \quad y \quad z]^T$ is the position vector of in the earth-fixed inertial frame and $[\phi \quad \theta \quad \psi]^T$ represents the roll, pitch and yaw angles. Reference frames and the attitude angles are illustrated in Fig. 1.

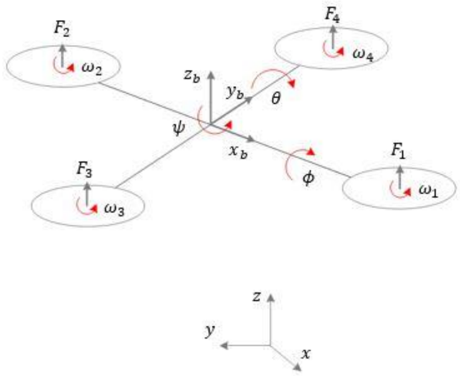

Fig. 1. Schematic of the quadcopter and the frames of reference.

A. Translational Dynamics

The translational dynamics of the quadcopter expressed in the inertial frame is given as follows.

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}+mg\begin{bmatrix}0\\0\\1\end{bmatrix}=\boldsymbol{F}_T+\boldsymbol{F}_d \tag{1}$$

where, $\boldsymbol{F}_T$ is the net thrust force generated by four rotors expressed in the inertial frame:

$$\boldsymbol{F}_T=\boldsymbol{R}_{EB}\begin{bmatrix}0\\0\\F_Z\end{bmatrix} \tag{2}$$

here, $\boldsymbol{R}_{EB}$ is the transform matrix from the body–attached to the inertial frame as presented by Eq. (3).

$$\boldsymbol{R}_{EB}=\begin{bmatrix}\mathrm{c}\,\psi\,\mathrm{c}\,\theta & \mathrm{c}\,\psi\,\mathrm{s}\,\theta\,\mathrm{s}\,\phi-\mathrm{s}\,\psi\,\mathrm{c}\,\phi & \mathrm{c}\,\psi\,\mathrm{s}\,\theta\,\mathrm{c}\,\phi+\mathrm{s}\,\psi\,\mathrm{s}\,\phi\\ \mathrm{s}\,\psi\,\mathrm{c}\,\theta & \mathrm{s}\,\psi\,\mathrm{s}\,\theta\,\mathrm{s}\,\phi+\mathrm{c}\,\psi\,\mathrm{c}\,\phi & \mathrm{s}\,\psi\,\mathrm{s}\,\theta\,\mathrm{c}\,\phi-\mathrm{c}\,\psi\,\mathrm{s}\,\phi\\ -\mathrm{s}\,\theta & \mathrm{c}\,\theta\,\mathrm{s}\,\phi & \mathrm{c}\,\theta\,\mathrm{c}\,\phi\end{bmatrix} \tag{3}$$

where $c(\cdot)$ and $s(\cdot)$ are cos () and sin (), respectively. So, $\boldsymbol{F}_T$ will be simplified as follows.

$$\boldsymbol{F}_T=\begin{bmatrix}\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi\\ \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\\ \cos\theta\cos\phi\end{bmatrix} \tag{4}$$

In addition, the disturbance force $\boldsymbol{F}_d$ is given by

$$\boldsymbol{F}_d=\begin{bmatrix}F_{d_x}\\F_{d_y}\\F_{d_z}\end{bmatrix} \tag{5}$$

Finally, the translational dynamics of the quadcopter is given by Eqs. (6)–(8), [23]:

$$\ddot{x}=\frac{1}{m}(\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi)F_z+d_x \tag{6}$$

$$\ddot{y}=\frac{1}{m}(\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi)F_z+d_y \tag{7}$$

$$\ddot{z}=-g+\frac{1}{m}(\cos\theta\cos\phi)F_z+d_z \tag{8}$$

where, acceleration components due to disturbance forces are $d_x=\frac{F_{dx}}{m}$, $d_y=\frac{F_{dy}}{m}$ and $d_z=\frac{F_{dz}}{m}$.

B. Rotational Dynamics

The rotational dynamics of the quadcopter is expressed in terms of roll, pitch and yaw angles. First, the inertial matrix is assumed to be diagonal as is given by:

$$\boldsymbol{J}=\begin{bmatrix}I_{xx}&0&0\\0&I_{yy}&0\\0&0&I_{zz}\end{bmatrix} \tag{9}$$

The rotational dynamics is expressed by Eqs. (10)–(12) [24].

$$\ddot{\phi}=\dot{\theta}\dot{\psi}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right)-\frac{J_p}{I_{xx}}\dot{\theta}\Omega+\frac{\tau_\phi}{I_{xx}}+\frac{\tau_{d_\phi}}{I_{xx}} \tag{10}$$

$$\ddot{\theta}=\dot{\phi}\dot{\psi}\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right)+\frac{J_p}{I_{yy}}\dot{\phi}\Omega+\frac{\tau_\theta}{I_{yy}}+\frac{\tau_{d_\theta}}{I_{yy}} \tag{11}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right) + \frac{\tau_{\psi}}{I_{zz}} + \frac{\tau_{d_{\psi}}}{I_{zz}} \tag{12}$$

where $\Omega = \omega_4 + \omega_3 - \omega_2 - \omega_1$, J_r represents moment of inertia of the rotors, and τ_{d_ϕ}, τ_{d_θ} and $\tau_{d_\psi}$are the disturbance torques.

The four control inputs inside the dynamic of the quadcopter, expressed by rotational rate of the rotors, are given by Eqs. (13)–(16) [5].

$$\tau_{\phi} = lb(\omega_4^2 - \omega_3^2) \tag{13}$$

$$\tau_{\theta} = lb(\omega_2^2 - \omega_1^2) \tag{14}$$

$$\tau_{\psi} = d_f(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \tag{15}$$

$$F_z = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{16}$$

where ω_i is the rotational rate of each rotor, b is the thrust factor of the rotor, d_f is the frag factor of the rotor and l is the moment arm of the rotors.

The Eqs. (13)–(16) could be expressed in matrix form as

$$\begin{bmatrix}\tau_\phi\\ \tau_\theta\\ \tau_\psi\\ F_z\end{bmatrix} = \begin{bmatrix}0 & 0 & -lb & lb\\ -lb & lb & 0 & 0\\ d_f & d_f & -d_f & -d_f\\ b & b & b & b\end{bmatrix}\begin{bmatrix}\omega_1^2\\ \omega_2^2\\ \omega_3^2\\ \omega_4^2\end{bmatrix} = \boldsymbol{M}\begin{bmatrix}\omega_1^2\\ \omega_2^2\\ \omega_3^2\\ \omega_4^2\end{bmatrix} \tag{17}$$

Since, the matrix $\boldsymbol{M}$ is full-rank, the rotational rate of rotors could be expressed in terms of the four control inputs. This would be used in the inner loop control algorithm.

III. Controller Design

A double-loop controller, which uses a sliding mode controller for inner loop and a constrained model predictive controller for outer loop is purposed here to control the presented quadcopter. Fig. 2 shows a schematic of the controller architecture. This architecture is somehow similar to the design in Ref. [5, 23, 25] with two modifications; First, The authors of [23–25] have used unconstrained model predictive control for their controllers. Second, they all use control inputs like thrust or torque as the inputs to the "quadcopter dynamics" block, where in this work the reference rotational rates are given to the "quadcopter dynamics" block.

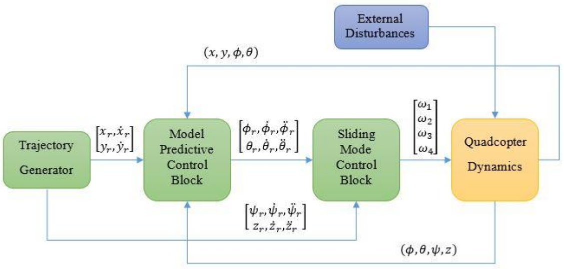
External Disturbances
(x, y, ϕ, θ)
Trajectory Generator
$\begin{bmatrix}x_r, \dot{x}_r\\ y_r, \dot{y}_r\end{bmatrix}$
Model Predictive Control Block
$\begin{bmatrix}\phi_r, \dot{\phi}_r, \ddot{\phi}_r\\ \theta_r, \dot{\theta}_r, \ddot{\theta}_r\end{bmatrix}$
Sliding Mode Control Block
$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix}$
Quadcopter Dynamics
$\begin{bmatrix}\psi_r, \dot{\psi}_r, \ddot{\psi}_r\\ z_r, \dot{z}_r, \ddot{z}_r\end{bmatrix}$
(ϕ, θ, ψ, z)

Fig. 2. Schematic of the proposed controller architecture.

The block entitled "Trajectory Generator" in Fig. 2, generates the reference position and velocity in horizontal plane of motion, to the outer control loop and the reference height and heading angle and their velocity and acceleration to the inner control loop. Then, the outer control loop generates the reference roll and pitch angles and its velocity and acceleration to the inner loop. The inner loop generates the reference rotational rates of the rotors. For practical applications, the sliding mode controller in the inner loop should run faster than the model predictive control in the outer loop [5].

A. Sliding Mode Controller, Inner Loop

The sliding mode control for adjusting the orientation and height of the quadcopter is discussed here. Consider the rotaional dynamics of the roll angle as

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right) - \frac{J_P}{I_{xx}}\dot{\theta}\Omega + \frac{\tau_\phi}{I_{xx}} + d_\phi \tag{18}$$

where, $d_\phi = \frac{\tau_{d_\phi}}{I_{xx}}$ is the rotaional roll acceleration due to disturbances. The objective is to control the roll angle with τ_ϕ. In order to apply the lyapunov theorem, the terms due to rotors inertia and external disturbances may cause degradation of the performance because these terms are unknown. So, it is considered that

$$\left|\frac{J_P}{I_{xx}}\dot{\theta}\Omega\right| \leq f_1;\ \left|d_\phi\right| \leq f_2 \tag{19}$$

where, f_1 and f_2 are some known boundaries.

Next, define a sliding manifold as

$$s_\phi = \dot{e}_\phi + \lambda_\phi e_\phi \tag{20}$$

where, the error states are defined as $e_\phi = \phi - \phi_r$ and $\dot{e}_\phi = \dot{\phi} - \dot{\phi}_r$. The time derivative of $s_\phi$is given by

$$\dot{s}_\phi = \dot{\theta}\dot{\psi}I_1 - \frac{J_P}{I_{xx}}\dot{\theta}\Omega + \frac{\tau_\phi}{I_{xx}} + d_\phi - \ddot{\phi}_r + \lambda_\phi\dot{e}_\phi \tag{21}$$

where, $I_1 = \frac{I_{yy}-I_{zz}}{I_{xx}}$. Therefore, the best approximation for control law is to drive s_ϕ to zero. So, the $\tau_\phi$would be as follows [26].

$$\tau_\phi = I_{xx}\left(-\dot{\theta}\dot{\psi}I_1 + \ddot{\phi}_r - \lambda_\phi\dot{e}_\phi - k_\phi sgn(s_\phi)\right) \tag{22}$$

where, the function $\mathrm{sgn}(x)$ is the sign function of x.

To determine the value k_ϕ to stabilize the roll dyanmics, we define the laypunov function as

$$V_\phi = \frac{1}{2}s_\phi^2 \tag{23}$$

This function is definitely positive. Its time derivative is given by

$$\dot{V}_\phi = s_\phi\dot{s}_\phi = s_\phi\left(-\frac{J_P}{I_{xx}}\dot{\theta}\Omega + d_\phi - k_\phi sgn(s_\phi)\right) \tag{24}$$

so,

$$\dot{V}_\phi \leq (f_1 + f_2 - k_\phi)\left|s_\phi\right| \tag{25}$$

If the value $k_\phi = f_1 + f_2 + \eta_\phi$ is chosen where η_ϕ is a positive parameter, then the time derivative of the lyapunov function is definitely negative and therefore, the roll dynamics would be stabilized. So, the control law would bring the system to the sliding line in finite-time.

On sliding line, the error states are governed by [5]

$$\dot{e}_\phi + \lambda_\phi e_\phi = 0 \Rightarrow e_\phi(t) = e_\phi(0)\exp\left(-\lambda_\phi t\right) \quad (26)$$

Hence the tracking error converges to zero exponentially. Therefore $\phi \to \phi_r$ and $\dot{\phi} \to \dot{\phi}_r$.

Following the same procedure, control laws for the pitch, yaw and height are given by

$$\tau_\theta = I_{yy}\left(-\dot{\phi}\dot{\psi}I_2 + \ddot{\theta}_r - \lambda_\theta \dot{e}_\theta - k_\theta\, sgn(s_\theta)\right) \quad (27)$$

$$\tau_\psi = I_{zz}\left(-\dot{\phi}\dot{\theta}I_3 + \ddot{\psi}_r - \lambda_\psi \dot{e}_\psi - k_\psi\, sgn(s_\psi)\right) \quad (28)$$

$$F_z = \frac{m}{\cos\phi\cos\theta}\left(g + \ddot{z}_r - \lambda_z \dot{e}_z - k_z\, sgn(s_z)\right) \quad (29)$$

with

$$I_2 = \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right);\ e_\theta = \theta - \theta_r;\ s_\theta = \dot{e}_\theta + \lambda_\theta e_\theta \quad (30)$$

$$I_3 = \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right);\ e_\psi = \psi - \psi_r;\ s_\psi = \dot{e}_\psi + \lambda_\psi e_\psi \quad (31)$$

$$e_z = z - z_r;\ s_z = \dot{e}_z + \lambda_z e_z \quad (32)$$

$$k_\theta = f_3 + f_4 + \eta_\theta;\ k_\psi = f_5 + \eta_\psi;\ k_z = f_6 + \eta_z \quad (33)$$

where the following holds for the f_is:

$$\left|\frac{J_p}{I_{yy}}\dot{\phi}\Omega\right| \le f_3;\ \left|d_\phi\right| \le f_4;\ \left|d_\psi\right| \le f_5;\ \left|d_z\right| \le f_6 \quad (34)$$

One of the major drawback of using the sliding mode control is the issue of *chattering*. It is notable that if the paramenter $s_{(\cdot)}$ is oscillating around the zero, the term $\mathrm{sgn}(s_{(\cdot)})$ is oscillating between −1 and 1. There are many ways to overcome this problem. A promising one is to approximate the sign function as [27]

$$\mathrm{sgn}\left(s_{(\cdot)}\right) = \frac{2}{\pi}\arctan\left(\mu s_{(\cdot)}\right) \quad (35)$$

where, $\mu > 0$ is a positive constant and the approximation error can be minimized by increaseing μ.

Therefore, the four control inputs are calculated by the above equations and the rotational speed of rotors can be calculated from these inputs. By use of the following equations

$$\begin{bmatrix}\widehat{\omega}_1\\ \widehat{\omega}_2\\ \widehat{\omega}_3\\ \widehat{\omega}_4\end{bmatrix} = \begin{bmatrix}0 & 0 & -lb & lb\\ -lb & lb & 0 & 0\\ d_f & d_f & -d_f & -d_f\\ b & b & b & b\end{bmatrix}^{-1}\begin{bmatrix}\tau_\phi\\ \tau_\theta\\ \tau_\psi\\ F_z\end{bmatrix} \quad (36)$$

the values of $\widehat{\omega}_i$ will be calculated. If the $\widehat{\omega}_i$ is positive, then the value of ω_i can be calculated as $\omega_i = \sqrt{\widehat{\omega}_i}$**.** In the simulation model, where $\widehat{\omega}_i$ may sometimes be negative, it will be set as zero. In addition, the rotaional speed of rotors has a saturaion value in reality. In quadcopters, the maximum thrust is usually set as a function of the weight. Therefore, it can be assumed that $\omega_{max} = \sqrt{\frac{mg}{2b}}$. Finally, the rotational rate of rotors is given by

$$\omega_i = \begin{cases}\sqrt{\frac{mg}{2b}}; & \widehat{\omega}_i \ge \frac{mg}{2b}\\ \sqrt{\widehat{\omega}_i}\,; & \frac{mg}{2b} > \widehat{\omega}_i > 0\\ 0; & \widehat{\omega}_i \le 0\end{cases} \quad (37)$$

B. The Model Predictive Control: Outer Loop

To design constrained linear model predictive control two virtual inputs are defined as follows [5].

$$u_x = \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \quad (38)$$

$$u_y = \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \quad (39)$$

With these definitions, we have

$$u_x^2 + u_y^2 = 1 - \cos^2\theta\cos^2\phi \le 1. \quad (40)$$

Hence the virtual control inputs should be constrained to be on or inside the unit circle in $\mathbb{R}^2$ to generate a feasible value for the reference angles θ and ϕ.

To navigate the position of the quadcopter in the horizontal plane of motion, the angles θ and ϕ have to be controlled. The dynamics of the vehicles in this plane is given by:

$$\frac{d}{dt}\begin{bmatrix}x\\ \dot{x}\\ y\\ \dot{y}\end{bmatrix} = \begin{bmatrix}\dot{x}\\ \frac{u_x F_z}{m} + d_x\\ \dot{y}\\ \frac{u_y F_z}{m} + d_y\end{bmatrix} \quad (41)$$

So

$$\frac{d}{dt}\begin{bmatrix}x\\ \dot{x}\\ y\\ \dot{y}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ y\\ \dot{y}\end{bmatrix} + \begin{bmatrix}0 & 0\\ \frac{F_z}{m} & 0\\ 0 & 0\\ 0 & \frac{F_z}{m}\end{bmatrix}\begin{bmatrix}u_x\\ u_y\end{bmatrix} + \begin{bmatrix}0\\ d_x\\ 0\\ d_y\end{bmatrix} \quad (42)$$

Using eular discretization with the assumption of state vector $\boldsymbol{\xi} = [x \quad \dot{x} \quad y \quad \dot{y}]^T$, input vector $\boldsymbol{u}_{xy} = [u_x \quad u_y]^T$, the time step Δt, and ignoring the disturbance terms, the above dynamics can be expressed in an equivalent form as follows.

$$\boldsymbol{\xi}(k+1) = \boldsymbol{A}_d\boldsymbol{\xi}(k) + \boldsymbol{B}_d\boldsymbol{u}_{xy}(k) \quad (43)$$

where,

$$\boldsymbol{A}_d = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B}_d = \begin{bmatrix} 0 & 0 \\ \frac{F_z \Delta t}{m} & 0 \\ 0 & 0 \\ 0 & \frac{F_z \Delta t}{m} \end{bmatrix} \tag{44}$$

The output of discretized dynamic equation is given by

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{\xi}(k) \tag{45}$$

where the matrix $\boldsymbol{C}$ is the identity matrix $\boldsymbol{I}$. Therefore, the predict for the next N-1 step is given by

$$\begin{gathered} \Xi(k) = \boldsymbol{F}\boldsymbol{\xi}(k) + \boldsymbol{H}\boldsymbol{U}_{xy}(k) \\ \boldsymbol{Y}(k) = \overline{\boldsymbol{C}}\Xi(k) \end{gathered} \tag{46}$$

where,

$$\Xi(k) = \begin{bmatrix} \boldsymbol{\xi}(k) \\ \boldsymbol{\xi}(k+1) \\ \vdots \\ \boldsymbol{\xi}(k+N-1) \end{bmatrix} \tag{47}$$

$$\boldsymbol{U}_{xy}(k) = \begin{bmatrix} \boldsymbol{u}_{xy}(k) \\ \boldsymbol{u}_{xy}(k+1) \\ \vdots \\ \boldsymbol{u}_{xy}(k+N-1) \end{bmatrix} \tag{48}$$

$$\boldsymbol{Y}(k) = \begin{bmatrix} \boldsymbol{y}(k) \\ \boldsymbol{y}(k+1) \\ \vdots \\ \boldsymbol{y}(k+N-1) \end{bmatrix} \tag{49}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A}_d \\ \boldsymbol{A}_d^2 \\ \vdots \\ \boldsymbol{A}_d^{N-1} \end{bmatrix} \tag{50}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{O} & \cdot & \cdot & \cdot & \boldsymbol{O} \\ \boldsymbol{B}_d & \boldsymbol{O} & \cdot & \cdot & \boldsymbol{O} \\ \boldsymbol{A}_d\boldsymbol{B}_d & \boldsymbol{B}_d & \dots & \cdot & \boldsymbol{O} \\ \vdots & \vdots & \cdot & \cdot & \cdot \\ \boldsymbol{A}_d^{N-2}\boldsymbol{B}_d & 3\boldsymbol{B}_d & \dots & \boldsymbol{B}_d & \boldsymbol{O} \end{bmatrix} \tag{51}$$

$$\overline{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & & & \\ & \boldsymbol{C} & & \\ & & \cdot & \\ & & & \boldsymbol{C} \end{bmatrix} \tag{52}$$

where the matrix $\boldsymbol{I}$ is the identity matrix and the matrix $\boldsymbol{O}$ is the null matrix of appropriate dimensions. N is the prediction or control horizon. Note that the prediction and control horizons are considered the same for this study. The final step's states and the corresponding outputs are given by

$$\boldsymbol{\xi}(k+N) = \boldsymbol{A}_d^N\boldsymbol{\xi}(k) + \overline{\boldsymbol{B}}\boldsymbol{U}_{xy}(k) \tag{53}$$

$$\boldsymbol{y}(k+N) = \boldsymbol{C}\boldsymbol{\xi}(k+N) \tag{54}$$

where $\overline{\boldsymbol{B}} = [\boldsymbol{A}_d^{N-1}\boldsymbol{B}_d \quad \boldsymbol{A}_d^{N-2}\boldsymbol{B}_d \quad \cdots \quad \boldsymbol{A}_d\boldsymbol{B}_d \quad \boldsymbol{B}_d]$.

The model predictive control uses a cost function to find an optimal input vector at each sampling interval. This solution must be chosen in the way that the predicted outputs, drived from the prediction horizon N, are driven to track the desired trajectory $\boldsymbol{y_r}(k)$, while it should minimize the required control effort $\boldsymbol{u}_{xy}(k)$. In this work, the cost function panalizes the weighted norm of the diffrence between the current output states and the desired trajectory, and the weighted norm of control inputs. The cost function for this algorithm is given by

$$\boldsymbol{J} = \left|\left|\boldsymbol{Y}(\boldsymbol{k}) - \boldsymbol{Y}_r(\boldsymbol{k})\right|\right|_{\overline{\boldsymbol{Q}}_N} + \left|\left|\boldsymbol{U}_{xy}(\boldsymbol{k})\right|\right|_{\overline{\boldsymbol{R}}_N} + \left|\left|\boldsymbol{y}(\boldsymbol{k}+\boldsymbol{N}) - \boldsymbol{y}_r(\boldsymbol{k}+\boldsymbol{N})\right|\right|_{\overline{\boldsymbol{Q}}_f} \tag{55}$$

where, $\overline{\boldsymbol{Q}}_{\boldsymbol{f}} \geq 0$ and $\boldsymbol{Y_r}(k)$, $\overline{\boldsymbol{Q}}_{\boldsymbol{N}}$ and $\overline{\boldsymbol{R}}_N$are given by

$$\boldsymbol{Y}_r(k) = \begin{bmatrix} \boldsymbol{y_r}(k) \\ \boldsymbol{y_r}(k+1) \\ \vdots \\ \boldsymbol{y_r}(k+N-1) \end{bmatrix} \tag{56}$$

$$\overline{\boldsymbol{Q}}_{\boldsymbol{N}} = \begin{bmatrix} \boldsymbol{Q} & & & \\ & \boldsymbol{Q} & & \\ & & \cdot & \\ & & & \boldsymbol{Q} \end{bmatrix} \tag{57}$$

$$\overline{\boldsymbol{R}}_{\boldsymbol{N}} = \begin{bmatrix} \boldsymbol{R} & & & \\ & \boldsymbol{R} & & \\ & & \cdot & \\ & & & \boldsymbol{R} \end{bmatrix} \tag{58}$$

where, the matrices $\boldsymbol{Q} \geq 0$ and $\boldsymbol{R} > 0$ are the diagonal matrices and the weighted norm of the vector is given by:

$$\left|\left|\boldsymbol{X}\right|\right|_{\boldsymbol{W}} = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}. \tag{59}$$

The martix $\overline{\boldsymbol{Q}}_{\boldsymbol{f}}$ would be determined from the solution of the discreate algebratic ricatti equation given by

$$\boldsymbol{P} = \boldsymbol{A}_d^T\boldsymbol{P}\boldsymbol{A}_d + \boldsymbol{A}_d^T\boldsymbol{P}\boldsymbol{B}_d\left(\boldsymbol{R} + \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{B}_d\right)^{-1}\boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{A}_d + \boldsymbol{Q} \tag{60}$$

Therefore, the martix $\overline{\boldsymbol{Q}}_f$is equal to the solution for the equation above. In MATLAB, the function ‘idare’ is utilized to solve discreate algebratic riccatti equation. Finally the cost function is expressed in quadratic form as follows.

$$\begin{aligned} &\boldsymbol{J} = \boldsymbol{U}_{xy}^T\left[\boldsymbol{H}^T\overline{\boldsymbol{C}}^T\overline{\boldsymbol{Q}}_N\overline{\boldsymbol{C}}\boldsymbol{H} + \overline{\boldsymbol{R}}_N + \overline{\boldsymbol{B}}^T\boldsymbol{C}^T\boldsymbol{Q}_f\boldsymbol{C}\overline{\boldsymbol{B}}\right]\boldsymbol{U}_{xy} + 2\Big(\big(\overline{\boldsymbol{C}}\boldsymbol{F}\boldsymbol{\xi}(k) - \\ &\boldsymbol{Y}_r(k)\big)^T\overline{\boldsymbol{Q}}_N\overline{\boldsymbol{C}}\boldsymbol{H} + \left(\boldsymbol{C}\boldsymbol{A}_d^N\boldsymbol{\xi}(k) - \boldsymbol{y_r}(k+N)\right)\boldsymbol{Q}_f\boldsymbol{C}\overline{\boldsymbol{B}}\Big)\boldsymbol{U}_{xy} + \\ &\left(\overline{\boldsymbol{C}}\boldsymbol{F}\boldsymbol{\xi}(k) - \boldsymbol{Y}_r(k)\right)^T\overline{\boldsymbol{Q}}_N\left(\overline{\boldsymbol{C}}\boldsymbol{F}\boldsymbol{\xi}(k) - \boldsymbol{Y}_r(k)\right) + \Big(\boldsymbol{C}\boldsymbol{A}_d^N\boldsymbol{\xi}(k) - \\ &\boldsymbol{y_r}(k+N)\Big)^T\overline{\boldsymbol{Q}}_f\left(\boldsymbol{C}\boldsymbol{A}_d^N\boldsymbol{\xi}(k) - \boldsymbol{y_r}(k+N)\right) \end{aligned} \tag{61}$$

This form of the cost function is suitable for a quadratic programming solver to solve the optimization problem. In MATLAB, the function ‘quadprog’ is utilized to solve this problem, where the input constraints are defined as follows.

$$\begin{bmatrix} u_l \\ u_l \\ \vdots \\ u_l \end{bmatrix} \leq \boldsymbol{U}_{xy} \leq \begin{bmatrix} u_m \\ u_m \\ \vdots \\ u_m \end{bmatrix} \tag{62}$$

Hence, the optimal input vector $\boldsymbol{U}_{xy}(k)$ is calculated and the first two elements, $\boldsymbol{u}_{xy}(k) = [u_1 \quad u_2]^T$are selected for generating the reference values of the roll and pitch angles. By using the Eqs. (38)–(39), the reference roll and pitch angles are calculated as:

$$\theta_r = \pm \tan^{-1}\left[\frac{(u_1 \cos\psi + u_2 \sin\psi)^2}{1-(u_1^2+u_2^2)}\right]^{1/2} \tag{63}$$

$$\cos(\phi_r) = \pm[1-(u_1^2+u_2^2)+(u_1\cos\psi+u_2\sin\psi)^2]^{1/2} \tag{64}$$

Therefore, the specific reference values for θ_r and ϕ_r could not be determined. In this study, the following strategy is utilized for assigning the correct sign for the reference angles to be tracked.

First, consider the case, where ψ is driven to be near zero. Therefore, the virtual controllers of the quadcopter are approximated by $u_x \approx \sin\theta_r \cos\phi_r$ and $u_y \approx -\sin\phi_r$. Consequently, the references in the pitch and roll angles are calculated using the Eqs. (63) and (64) by taking only the positive values of the right hand side.

Let's consider the two reference values as θ_c and ϕ_c as $\theta_c \in (0, \frac{\pi}{2})$ and $\phi_c \in (0, \frac{\pi}{2})$. Using these reference values, the virtual controls u_x and u_y are calculated. With the assumption that ψ is approximately zero, it would result in a positive u_x and a negetive u_y. Then, it is checked if the signs of u_x and u_y match the corresponding signs of u_1 and u_2. If the sign of u_x matches with u_1, the reference value to be tracked for the pitch angle is assigned as $\theta_r = \theta_c$. Otherwise, we assign $\theta_r = -\theta_c$. Similarly, If the sign of u_y matches with u_2, the reference value to be tracked for the pitch angle is assigned as $\phi_r = \phi_c$. Otherwise, we assign $\phi_r = -\phi_c$ [5].

Note that in this work, the pitch and roll rate and acceleration are assumed to be zero at any reference waypoint.

IV. Simulation Results

In this section, simulation results are presented for the proposed MPC-SMC controller. Parameters of the quadcopter and the controller are presented in Tables 1–2.

Table 1. Parameters of the quadcopter

Parameter	Value	Parameter	Value
m	0.74 kg	b	2.9×10^{-5} N×s^2
I_{xx}	0.004 kg-m^2	d_f	1.1×10^{-}6 N×m×s^2
I_{yy}	0.004 kg-m^2	l	0.21 m
I_{zz}	0.0084 kg-m^2		

Table 2. Parameters of the controller

Parameter	Value	Parameter	Value
$\lambda_{(\cdot)}$	5	μ	10
$\eta_{(\cdot)}$	50	Δt	0.01 s
$\boldsymbol{Q}$	100$\boldsymbol{I}$	u_l	–0.2
$\boldsymbol{R}$	$\boldsymbol{I}$	u_m	0.2
N	10		

In this work, the disturbance forces, applied to the model, are assumed to be [5]

$$F_d = \begin{bmatrix} F_{d_x} \\ F_{d_y} \\ F_{d_z} \end{bmatrix} = u(t-20)\exp(-0.5(t-20))\begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \end{bmatrix} \tag{65}$$

$$\tau_d = u(t-20)\exp(-0.5(t-20))\begin{bmatrix} 0.01\sin(t-20) \\ 0.01\sin(2t-40) \\ -0.015\cos(t-20) \end{bmatrix} \tag{66}$$

where, $u(t)$ is the Heaviside function.

The f_i values used in the controller are presented in Table 3.

Table 3. f_i values

Parameters	Value
f_1	360
f_2	0.01
f_3	360
f_4	0.01
f_5	0.015
f_6	0.1

This simulation is performed in Simulink/MATLAB, with the use of "ode45" and millisecond time step. The time-step of the outer loop control is set to be 0.01 seconds, so the outer loop has been set to work 10 times slower than the inner loop. In this work, two different cases have been considered with two different time-varying trajectories to illustrate the effectiveness of the proposed controller.

A. Case-I

In this case the desired trajectory adopted from Ref. [5] is given by:

$$x_r = a\cos(\omega t) - 3;\ \dot{x}_r = -a\omega\sin(\omega t) \tag{67}$$

$$y_r = a\sin(\omega t);\ \dot{y}_r = a\omega\cos(\omega t) \tag{68}$$

$$z_r = z_v t;\quad \dot{z}_r = z_v;\quad \ddot{z}_r = 0 \tag{69}$$

where $a = 4\,m$, $\omega = \frac{\pi}{30} rad/s$ and $z_v = 0.05\,m/s$. Therefore, the desired trajectory is a helix. The desired yaw angle is zero. The initial position of the quadcopter is $(x_0.y_0.z_0) = (-1.0.5.\ 0)\,m$ and the initial orientation of the quadcopter is $(\phi_0.\theta_0.\psi_0) = (0.\ 0.\ 0.5)\,rad$.

The trajectory tracking performance of the controller are shown in Figs. 3 and 4. The error in tracking the Euler angles are shown in Fig. 5. These figures shows that the controller could achieved satisfactory tracking performance. The Integral Absolute Error (IAE) indices obtained for the simulation results are shown in Table 4 and have been compared with the indices of Ref. [5]. These results shows that the purposed controller has better performance in position tracking with respect to a little increase in the pitch and yaw error.

The required forces and torques for the purposed SMC-MPC are shown in Fig. 6 and the rotor thrust forces are shown in Fig. 7. At first, the thrust forces are oscillating between zero and the maximum thrust, which is caused by tracking the required speed by the quadcopter.

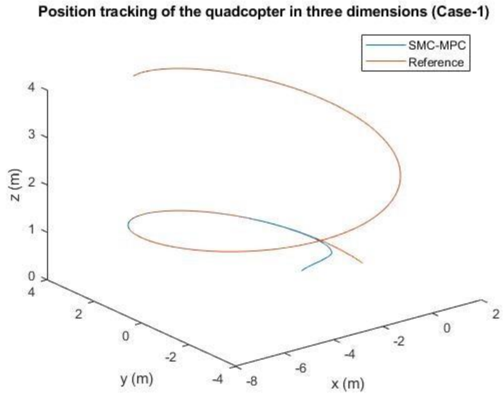

Fig. 3. Position tracking of the quadcopter in three dimensions (Case-I).

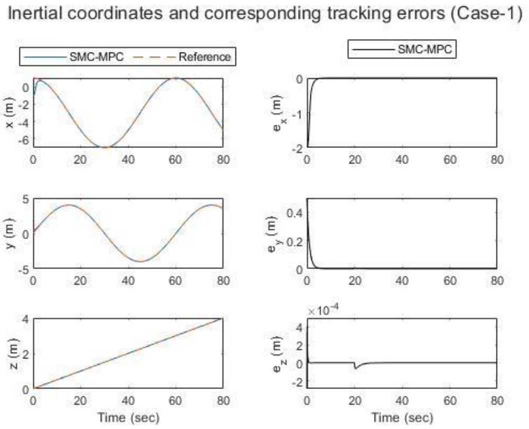

Fig. 4. Inertial coordinates and corresponding tracking errors (Case-I).

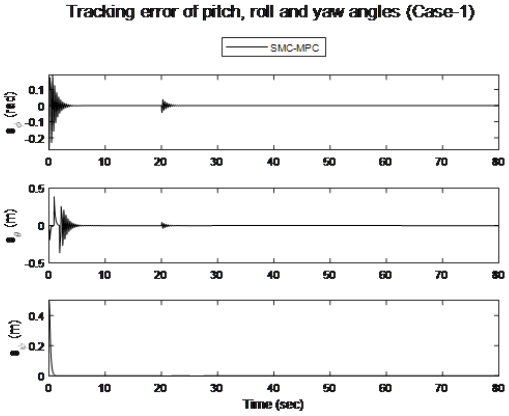

Fig. 5. Euler angles tracking error (Case-I).

A less-powered fluctuation has also been found at the moment the disturbance occurs. As seen in these figures, the disturbance has a big effect, mostly on roll and pitch dynamics. Finally, the time histories of the virtual control inputs for this simulation are given in Fig. 8.

Table 4. IAE performance indices (Case-I)

States	SMC-MPC	SMC-MPC* (in Ref [5])
x	2.7935	3.5952
y	0.5029	8.3154
z	0.00024845	0.0182
ϕ	0.2243	0.4555
θ	0.4060	0.3869
ψ	0.1463	0.1165

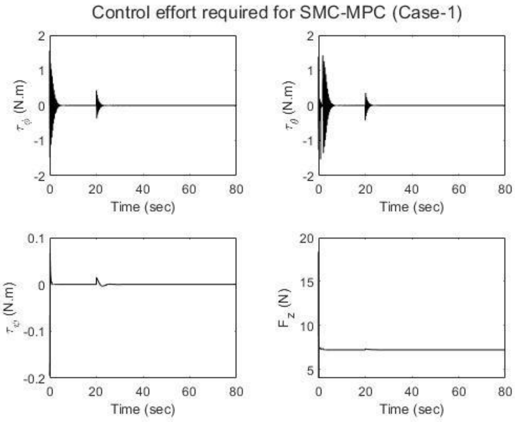

Fig. 6. Control effort required for the integrated SMC-MPC (Case-I).

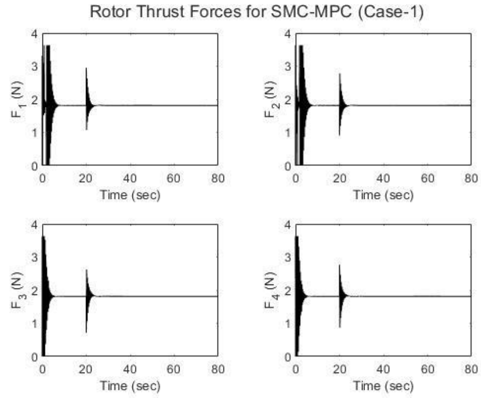

Fig. 7. Rotor Thrust forces for the integrated SMC-MPC (Case-I)

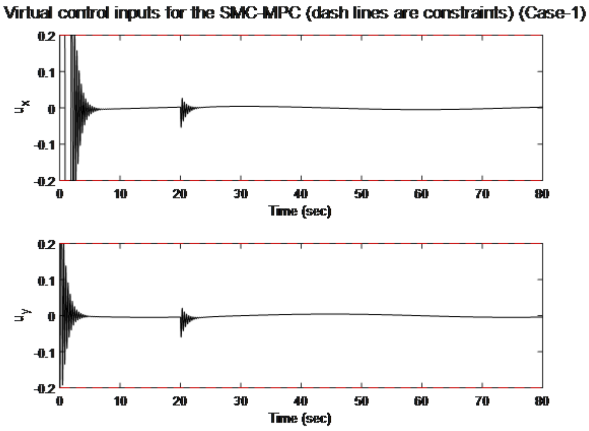

Fig. 8. Virtual control inputs for the integrated SMC-MPC (dash lines are constraints) (Case-I).

B. Case-II

In this case, the desired trajectories, which are also adopted from the Refs. [5, 23] and [25] are as follows.

$$x_r = \begin{cases} 0.5\cos(0.5t) & 0 \le t < 4\pi \\ 0.5 & 4\pi \le t < 20 \\ 0.25t - 4.5 & 20 \le t < 30 \\ 3 & 30 \le t \le 80 \end{cases} \tag{70}$$

$$y_r = \begin{cases} 0.5\sin(0.5t) & 0 \le t < 4\pi \\ 0.25t - \pi & 4\pi \le t < 20 \\ 5 - \pi & 20 \le t < 30 \\ -0.25t + 12.5 - \pi & 30 \le t < 40 \\ 2.5 - \pi & 40 \le t \le 80 \end{cases} \tag{71}$$

$$z_r = \begin{cases} 0.125t + 1 & 0 \le t < 4\pi \\ 0.5\pi + 1 & 4\pi \le t < 40 \\ \exp(-0.2t + 8.944) & 40 \le t < 80 \end{cases} \quad (72)$$

The desired yaw angle is zero. The initial position and orientation are the same as Case-I. Trajectory tracking performance of the controller is shown in Figs. 9 and 10. The error in tracking the Euler angles is shown in Fig. 11. These results show that the controller can achieve satisfactory tracking performance.

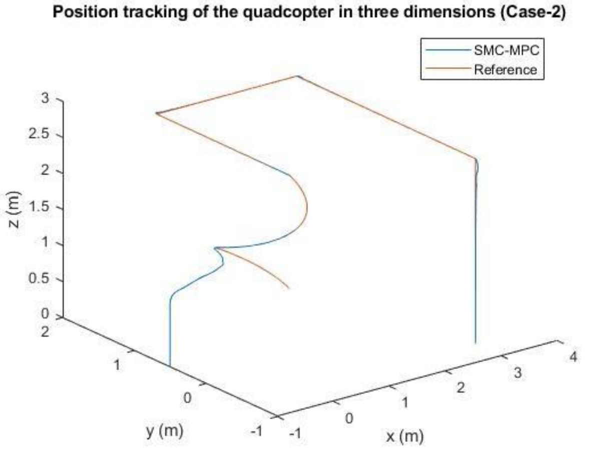

Fig. 9. Position tracking of the quadcopter in three dimensions. (Case-II).

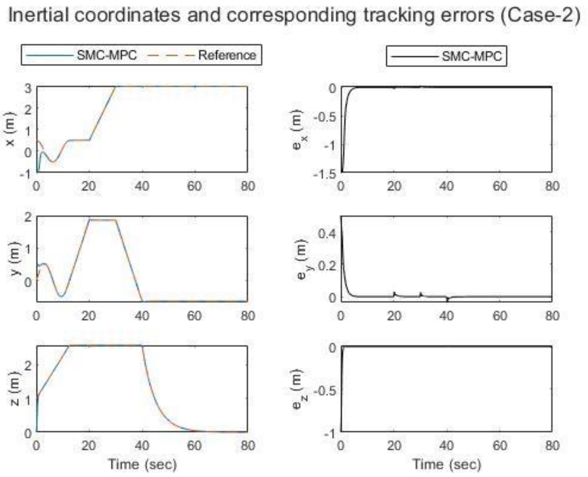

Fig. 10. Inertial coordinated and corresponding tracking errors (Case-II).

The integral absolute error (IAE) indices obtained for the simulation results are shown in Table 5 and have been compared with the indices from Ref. [5]. These results shows that the purposed controller in this paper has improved position tracking performance in some aspects.

Table 5. IAE performance indexes (Case-Ii)

States	SMC-MPC	SMC-MPC* (in Ref [5])
x	2.5280	2.9011
y	0.6298	1.0845
z	0.3809	0.2768
ϕ	1.3073	1.7371
θ	1.0141	1.2441
ψ	0.3763	0.1165

The required force and torques for the purposed SMC-MPC are shown in Fig. 12 and the rotor thrust forces are shown in Fig. 13. As seen in these figures, there is a large fluctuation at the moment that the trajectory is changing (12, 20, 30 and 40 seconds). From the Fig. 12, at 12 and 40 seconds the required net thrust gets negative. Therefore, the rotors will stop working, which is more realistic. Finally, the time histories of the virtual control inputs for this simulation scenario are given in Fig. 14.

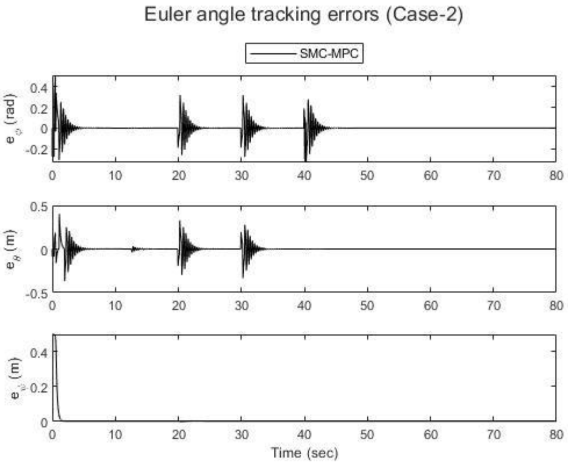

Fig. 11. Euler angle tracking errors (Case-II).

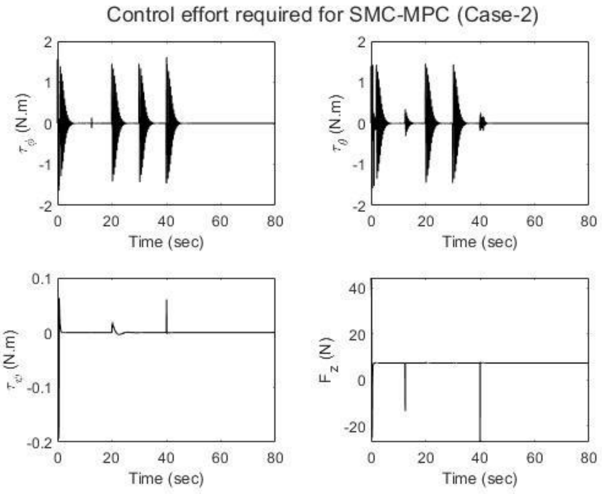

Fig. 12. Control effort required for SMC-MPC (Case-II).

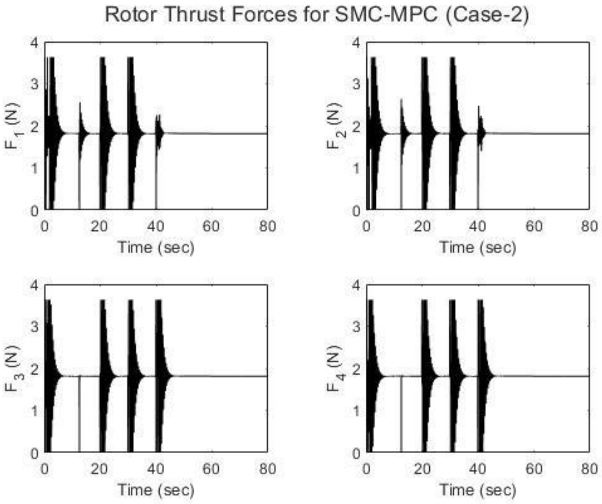

Fig. 13. Rotor Thrust Forces for SMC-MPC (Case-II).

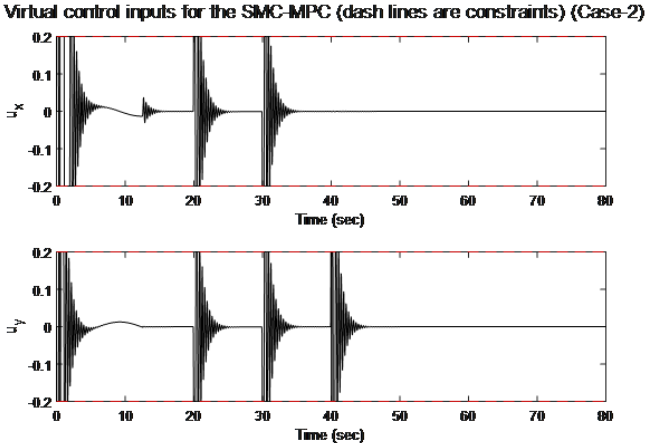

Fig. 14. Virtual control inputs for the SMC-MPC (dash lines are constraints) (Case-II).

V. Conclusion

In this research an integrated SMC-MPC controller is developed for quadcopters to control the attitude and to track the desired path. For attitude and altitude control, sliding mode controller is designed. For tracking the desired trajectory in horizontal plane of motion, model predictive controller is designed by solving a constrained optimization problem. Here, two virtual controls are derived, which generate the reference roll and pitch angles based on the assumption that the yaw angle is near zero. Furthermore, this controller is examined against two distinct trajectories. It is shown that the purposed controller has outstanding performance in stabilizing and tracking the desired trajectory. In addition, the integral absolute error indices have been used to compare with the existing literature. It was found that the performance of the proposed controller has been equal or better in spite of some limitations on the rotor thrust.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Amirreza Bagherzadeh conducted the mathematical analysis of the controller, performed simulations, and drafted the article. Afshin Banazadeh reviewed the results, revised the manuscript, both authors contributed to the conceptualization of the study. and approved the final version for publication.

References

[1] Z. Zhang, S. Li, and H. Dai, "Application of UAV aerial photogrammetry in beach change monitoring – Take zhoushan island for example," in *Proc. International Conference on Optics and Machine Vision (ICOMV 2022*), vol. 12173, Guangzhou, China, 2022.

[2] O. Dorosh, Y. Butenko, H. Kolisnyk, A. Dorosh, and I. Kupriianchyk, "The use of UAVs: development, perspectives and application," *AgroLife Scientific Journal*, vol. 10, pp. 63−71, 2021.

[3] A. K. Shastry, M. T. Bhargavapuri, M. Kothari, and S. R. Sahoo, "Quaternion base adaptive control for package delivery using variable-pitch quadcopters," in *Proc. 2018 Indian Control Conference (ICC)*, pp. 340−345, Kanpur, India, 2018.

[4] M. S. Esmail, M. H. Merzban, A. A. M. Khalaf, H. F. A. Hamad, and A. I. Hussein, "Attitude and altitude tracking controller for quadcopter dynamical systems," *IEEE Access*, vol. 10, pp. 53344–53358, 2022.

[5] D. Bhayyacharjee and K. Subbarao, "Robust control strategy for quadcopters using sliding mode control and model predictive control," in *Proc. AIAA SciTech 2020 Forum*, p. 2071, Orlando, Florida, USA, 2020.

[6] D. Asadi, K. Ahmadi, and S. Y. Nabavi, "Fault-tolerant trajectory tracking control of a quadcopter in persence of a motor fault," *International Journal of Aeronautical and Space Sciences*, vol. 23, pp. 129–142, 2022.

[7] A. M. Elhennawy and M. K. Habib, "Tracjectory tracking of a quadcopter flying vehicle using sliding mode control," in *Proc. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6264–6269, Beijing, China, 2017.

[8] V. K. Tripathi, L. Behera, and N. Verma, "Design of sliding mode and backstepping controllers for a quadcopter." in *Proc. 39th national systems conference (NSC)*, pp. 1–6, 2015.

[9] H. Abrougui, S. Nejim, A. Charrada, and H. Dallagi, "Sliding mode control for hexacopter autopilot design," in *Proc. 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 445–449, 2022.

[10] M. Ullah, C. Zhao, H. Maqsood, *et al*., "Adaptive neural-sliding mode control of a quadrotor vehicle with uncertainties and disturbances compensation," in *Proc. 2022 2nd International Conference on Artificial Intelligence (ICAI)*, pp. 38–45, Islamabad, Pakistan, 2022.

[11] K. A. Alattas, O. Mofid, A. K. Alanazi *et al*., "Barrier function adaptive nonsingular terminal sliding mode control approach for quad-rotor unmanned aerial vehicles," *Sensors*, vol. 22, no. 3, p. 909, 2022.

[12] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. A. M. Eltoum, and S. El-Ferik, "An improved design of an adaptive sliding mode controller for chattering attenuation and trajectory tracking of the quadcopter UAV," *IEEE Access*, vol. 8, pp. 205968−205979, 2020.

[13] E. Kawamura and D. Azimov, "Integrated optimal control and explicit guidance for quadcopters," in *Proc. 2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 513−522, Atlanta, GA, USA, 2019.

[14] C. Massé, O. Gougeon, D. Nguyen, and D. Saussié, "Modeling and control of a quadcopter flying in a wind field: A comparison between LQR and structured $\mathscr{H}\infty$ control techniques," in *Proc. 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1408−1417, Dallas, TX, USA, 2018.

[15] M. Ibrahim, C. Kallies, and R. Findeisen, "Learning-supported approximated optimal control for autonomous vehicles in the presence of state dependent uncertainties," in *Proc. 2020 European Control Conference (ECC)*, pp. 338−343, St. Petersburg, Russia, 2020.

[16] O. E. Ramos, "Minimum-time optimal control for a quadcopter trajectory through gates with arbitrary pose," in *Proc. 2021 IEEE XXVIII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pp. 1–4, 2021.

[17] K. S. Holkar and L. M. Waghmare. "An overview of model predictive control," *International Journal of Control and Automation*, 3, no. 4, pp. 47–63, 2020.

[18] G. Ganga and M. M. Dharmana, "MPC controller for trajectory tracking control of quadcopter," in *Proc. 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–6, Kollam, India, 2017.

[19] M. Islam, M. Okasha, and M. M. Idres, "Dynamics and control of quadcopter using linear model predictive control approach," in *Proc. IOP conference series: materials science and engineering*, vol. 270, no. 1, p. 012007, IOP Publishing, 2017

[20] A. Eltrabyly, D. Ichalal, and S. Mammar, "Fault-tolerant model predictive control trajectory tracking for a quadcopter with 4 faulty actuators," *IFAC-PapersOnLine*, vol. 54, no. 4, pp. 141−146, 2021.

[21] H. Merabti, I. Bouchachi, and K. Belarbi, "Nonlinear model predictive control of quadcopter," in *Proc.* 2*015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 208–211, Monastir, Tunisia, 2015.

[22] P. Ru and K. Subbarao, "Nonlinear model predictive control for unmanned aerial vehicles," *Aerospace*, vol. 4, no. 2, p. 31, 2017.

[23] V. R. Guilherme, G. O. Manuel and R. R. Francisco, "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[24] A. Das, K. Subbarao, and F. Lewis, "Dynamic inversion with zero-dynamics stabilisation for quadrotor control," *IET control theory & applications*, vol. 3, no. 3, pp. 303–314, 2009

[25] D. Ma, Y. Xia, T. Li, and K. Chang, "Active disturbance rejection and predictive control strategy for a quadrotor helicopter," *IET control theory & applications*, vol. 10, no. 17, pp. 2213−2222, 2016.

[26] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ, USA: Prentice Hall, 1991.

[27] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *Proc.* 45th *IEEE Conference on Decision and Control*, pp. 4957–4962, San Diego, CA, USA, 2006.