2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Fault-Tolerant Quadcopter Control Integrating Dynamic Equilibrium Analysis and Nonlinear Model Predictive Control*

Daisuke Muroki[1], Yuta Asano[2], Takumi Noro[2], Takeya Shima[2] and Toshiyuki Ohtsuka[1]

Abstract— **This study proposes a novel fault-tolerant control strategy for a quadcopter by integrating dynamic equilibrium analysis with nonlinear model predictive control (NMPC). First, we formulate an optimization problem based on dynamic equilibrium analysis. The feasibility of maintaining a constant altitude under a fault condition is determined by solving this problem. This analysis yields an optimal target that minimizes yaw angular velocity, while satisfying physical limitations. The target then serves as an online-updated reference for NMPC. This integrated approach enables a single controller to manage various rotor faults with a single fixed set of controller weights. Furthermore, the controller explicitly handles angular velocity constraints within the sensor limits, enhancing its reliability in a real quadcopter. The effectiveness of this strategy is verified through numerical simulations using a model based on an actual quadcopter, which enables visualization of operational bounds for various fault conditions. Simulations also demonstrate that stable flight can be achieved, while satisfying constraints even after a severe rotor fault.**

I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs), particularly multicopters have been widely used in industries including agriculture, logistics, and infrastructure inspection [1]. A major challenge in the practical application of multicopters is the high probability of accidents including crashes due to faults in their propulsion systems, composed of propellers and motors. In particular, quadcopters with four rotors are widely used, but lack inherent redundancy, making them vulnerable to propulsion system faults. Therefore, it is essential to implement fault-tolerant control (FTC) methods to handle propulsion system faults. This study aims to develop an effective FTC method for handling various rotor faults in a quadcopter.

FTC for quadcopters is generally classified as either passive fault-tolerant control (PFTC) or active fault-tolerant control (AFTC). The former is a robust control approach that can handle minor faults (malfunctions) such as a slight loss of effectiveness, but cannot manage severe faults [2], [3]. In contrast, AFTC performs fault detection and diagnosis (FDD) and sacrifices yaw rotation control based on fault information. This approach allows a quadcopter to maintain flight during severe faults such as the complete fault of one rotor [4], [5], [6], [7], [8]. However, prior studies had limitations in their scope and design, as they often relied on controller switching, model linearization, or specific fault scenarios. Although some studies used nonlinear model predictive control (NMPC) to handle a nonlinear model directly without linearization [7], [9], they did not consider online fault diagnosis or the transition process after fault occurrence.

To address these issues, a previous study [10] proposed an FTC method capable of handling diverse faults with a single controller, regardless of the degree or location by integrating fault diagnosis based on online thrust estimation into NMPC. However, three main challenges remained for the practical application of this method. First, they did not quantitatively specify the range of faults for which the altitude could be maintained. Second, the reference values for the thrust and angular velocity were set as constants based on the premise that yaw control was not sacrificed. Consequently, two sets of weights were used depending on the fault severity, and the weights were changed during severe faults to sacrifice yaw control. However, this method relied on the inherent robustness of NMPC rather than explicitly computing an optimal fault-tolerant attitude. Finally, the angular velocity constraints arising from the quadcopter's gyro sensor characteristics described in [8], [9] were not considered. Large yaw angular velocities can occur during FTC, potentially exceeding sensor limits. Thus, it is necessary to constrain the angular velocity within the sensor limits.

Therefore, this study proposes a new FTC framework that fully addresses these three issues. First, for a quadcopter under a fault condition, we formulate the dynamic equilibrium state [9], in which a quadcopter maintains stable altitude by sacrificing yaw control. By solving an optimization problem to minimize the angular velocity subject to the constraints, we determine whether the quadcopter can maintain altitude under physical limitations on thrust and angular velocity for any fault pattern. This analysis helps prevent accidents by enabling the quadcopter to identify unrecoverable situations in advance and avoid controls that could cause further instability. Next, if this analysis determines the possibility of maintaining altitude, the control reference for NMPC is updated online by the optimal solution. This eliminates the need for fault-dependent weight adjustments and provides a well-defined optimal reference, while minimizing the sacrifice of yaw angular velocity. Finally, using a more realistic model considering air drag, we handle angular velocity constraints by adding a penalty term to the NMPC performance index. This is essential for preventing saturation of the actual

*This work was supported in part by JSPS KAKENHI Grant Number JP22H01510 and JP23K22780.

[1]Graduate School of Informatics, Kyoto University, 36-1 Yoshida-Honmachi, Sakyo ward, Kyoto city, Kyoto 606-8501, JAPAN muroki.daisuke.26m@st.kyoto-u.ac.jp and ohtsuka@i.kyoto-u.ac.jp

[2]Advanced Technology R&D Center, Mitsubishi Electric Corporation, 8-1-1 Tsukaguchi-Honmachi, Amagasaki city, Hyogo 661-8661, JAPAN Asano.Yuta@dw.mitsubishielectric.co.jp, Noro.Takumi@dr.mitsubishielectric.co.jp and Shima.Takeya@cb.mitsubishielectric.co.jp

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

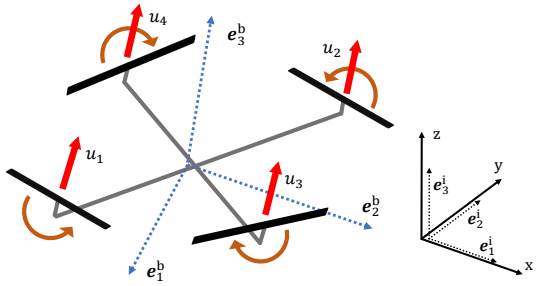

Fig. 1. Quadcopter schematic and coordinate system.

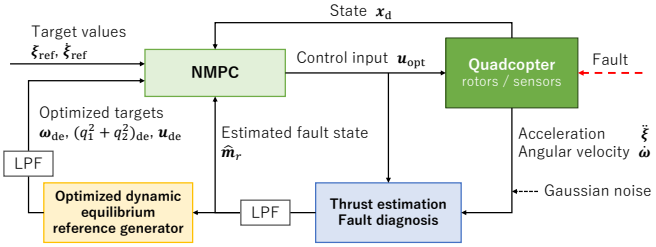

Fig. 2. System configuration of FTC. The system integrates fault diagnosis and dynamic equilibrium state optimization with NMPC. LPF denotes a low-pass filter used to process noisy sensor measurements.

gyro sensor and enhancing control reliability during rapid rotational motion after a fault.

The remainder of this paper is organized as follows. Section II describes the state equation of a quadcopter, including air drag. Section III details the proposed control system, including the fault diagnosis, dynamic equilibrium state optimization, and the FTC method that integrates these components with NMPC. Section IV visualizes the achievable range of FTC for two-rotor faults and presents simulation results of FTC using this method. Finally, Section V concludes the paper and discusses future work.

II. MODELING OF QUADCOPTER

This section presents the dynamic model of a quadcopter. Quaternions [11] are used to represent the quadcopter's attitude to avoid the gimbal lock problem associated with Euler angles. The components of a quaternion and its conjugate are defined as vectors

$$\boldsymbol{q} = \begin{pmatrix} q_0 & q_1 & q_2 & q_3 \end{pmatrix}^\top, \quad \boldsymbol{q}^\dagger = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{pmatrix}^\top. \tag{1}$$

Next, as shown in Fig. 1, the rotors are numbered from 1 to 4. We define two coordinate systems: an inertial frame with basis vectors $\{\boldsymbol{e}_1^{\mathrm{i}}, \boldsymbol{e}_2^{\mathrm{i}}, \boldsymbol{e}_3^{\mathrm{i}}\}$ and a body frame with basis vectors $\{\boldsymbol{e}_1^{\mathrm{b}}, \boldsymbol{e}_2^{\mathrm{b}}, \boldsymbol{e}_3^{\mathrm{b}}\}$. The origin of the body frame is located at the quadcopter's center of gravity, and its position in the inertial frame is denoted by $\boldsymbol{\xi} = \begin{pmatrix} x & y & z \end{pmatrix}^\top \in \mathbb{R}^3$. The angular velocity vector about the body-frame axes is denoted by $\boldsymbol{\omega} = \begin{pmatrix} \omega_1 & \omega_2 & \omega_3 \end{pmatrix}^\top \in \mathbb{R}^3$. Let the state vector of the quadcopter be $\boldsymbol{x}_{\mathrm{d}} = \begin{pmatrix} \boldsymbol{\xi}^{\mathrm{T}} & \dot{\boldsymbol{\xi}}^\top & \boldsymbol{\omega}^\top & \boldsymbol{q}^\top \end{pmatrix}^\top \in \mathbb{R}^{13}$ and the rotor thrust vector be $\boldsymbol{u} = \begin{pmatrix} u_1 & u_2 & u_3 & u_4 \end{pmatrix}^\top$. Then, using the quadcopter mass m and inertia matrix $\boldsymbol{J} \in \mathbb{R}^{3\times3}$, the state equation of the quadcopter considering air drag is expressed as

$$\dot{\boldsymbol{x}}_{\mathrm{d}} = \begin{pmatrix} \dot{\boldsymbol{\xi}} \\ \frac{1}{m}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{b}} - g\boldsymbol{e}_3^{\mathrm{i}} \\ \boldsymbol{J}^{-1}(\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{T}\boldsymbol{u} - \boldsymbol{D}\boldsymbol{\omega}) \\ \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q} \end{pmatrix}, \tag{2}$$

where $\boldsymbol{Q}$ is the transformation matrix from the body frame to the inertial frame, $\boldsymbol{F}^{\mathrm{b}} = \begin{pmatrix} 0 & 0 & \sum_{i=1}^{4} u_i \end{pmatrix}^\top$ is the thrust generated by the quadcopter in the body frame, g is the gravitational acceleration, and $\boldsymbol{\Omega}$ is the skew-symmetric matrix related to the time derivative of the quaternion. The term $\boldsymbol{D}\boldsymbol{\omega}$ represents the effect of air drag in the rotational direction. This effect is non-negligible at large angular velocities and is modeled based on [12]. $\boldsymbol{D} \in \mathbb{R}^{3\times3}$ is a diagonal matrix, in which the diagonal elements are the drag coefficients d_1, d_2, and d_3. Furthermore, $\boldsymbol{T}$ is expressed as

$$\boldsymbol{T} = \begin{pmatrix} r_{1,y} & r_{2,y} & r_{3,y} & r_{4,y} \\ -r_{1,x} & -r_{2,x} & -r_{3,x} & -r_{4,x} \\ -k & -k & k & k \end{pmatrix}, \tag{3}$$

using the x and y components of the i-th rotor's coordinates, $r_{i,x}$, $r_{i,y}$, and a constant k representing the relationship between the rotor thrust and reaction torque.

III. CONTROL SYSTEM

In this study, the control of the quadcopter's position, velocity, attitude, and angular velocity is formulated as an optimal control problem. Fig. 2 provides a block diagram of the overall system configuration. To handle the nonlinear model of the quadcopter directly, we employ nonlinear model predictive control (NMPC) [13], which solves an optimal control problem, while predicting future responses at each time step for the nonlinear model. This section briefly summarizes NMPC, a fault diagnosis method, dynamic equilibrium state optimization under faults, and their integration into the proposed FTC method.

A. Formulation of Nonlinear Model Predictive Control

NMPC is applicable to a nonlinear system described by the following state equation:

$$\dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}(t)\big), \tag{4}$$

where $\boldsymbol{x}(t)$ is the state vector, $\boldsymbol{u}(t)$ is the control input vector and $\boldsymbol{p}(t)$ is the time-varying parameter vector. In NMPC, a time series of the control input $\boldsymbol{u}(\tau)$ is computed to minimize the performance index J over a finite prediction horizon $[t, t+T]$ with the initial state given by the current state $\boldsymbol{x}(t)$:

$$J = \varphi\big(\boldsymbol{x}(t+T), \boldsymbol{p}(t+T)\big) + \int_t^{t+T} L\big(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{p}(\tau)\big)d\tau, \tag{5}$$

where φ is the terminal cost and L is the stage cost. This optimization yields the optimal control function $\boldsymbol{u}_{\mathrm{opt}}(\tau)$ $(t \le \tau \le t+T)$. In NMPC, only the initial value $\boldsymbol{u}(t) = \boldsymbol{u}_{\mathrm{opt}}(t)$

is applied to the system, and the optimization is solved at each control cycle to achieve feedback control.

One advantage of NMPC is its ability to explicitly handle input constraints. Each rotor thrust u_i of the quadcopter is subject to the inequality constraint, $u_{\min} \leq u_i \leq u_{\max}$, where $u_{\min}$ and $u_{\max}$ are the minimum and maximum rotor thrusts, respectively. In this study, the C/GMRES method [14] is used as a fast numerical solver to implement NMPC online.

B. Thrust Estimation and Fault Diagnosis by NMPC

Based on [10], this subsection presents a method for diagnosing faults in a quadcopter. First, the actual thrust of each rotor, $\hat{u}_i$, is estimated in real time by formulating the relationship between acceleration/angular acceleration and thrust, derived from the quadcopter's equations of motion, as a least squares problem [10, equations (4)-(5)]. Next, a parameter $\boldsymbol{m}_{\mathrm{r}} = \begin{pmatrix} m_{\mathrm{r}1} & m_{\mathrm{r}2} & m_{\mathrm{r}3} & m_{\mathrm{r}4} \end{pmatrix}^\top$ is defined to represent the effectiveness of each rotor as a continuous value. The estimated effectiveness of each rotor, $\hat{\boldsymbol{m}}_{\mathrm{r}}(t)$, is defined in each control cycle as the ratio of the estimated thrust to the NMPC control input from the previous calculation time step:

$$\hat{m}_{\mathrm{r}i}(t) = \frac{\hat{u}_i(t - \Delta t)}{u_{\mathrm{opt},i}(t - \Delta t)} \qquad (i = 1, \ldots, 4), \tag{6}$$

where the change in the fault state is considered negligible as Δt is sufficiently small. This allows for real-time determination of the fault occurrence time, the location of the faulty rotor, and the degree of effectiveness loss for each rotor. Here, fault diagnosis refers to estimation of the fault state $\hat{m}_{\mathrm{r}i}$. This method leverages the physical constraint that the motors of a multicopter cannot be reversed, i.e., the rotors cannot output negative thrust. The NMPC constraints ensure $u_{\mathrm{opt},i} \geq u_{\min} > 0$, and therefore $\hat{m}_{\mathrm{r}i} \geq 0$. If $\hat{m}_{\mathrm{r}i}$ exceeds 1, it represents an abnormal condition in which, for example, a motor or circuit malfunction causes a larger thrust to be generated than commanded.

C. Dynamic Equilibrium State Optimization and Feasibility Analysis of FTC

In this subsection, we formulate the dynamic equilibrium states under fault conditions. This formulation serves two purposes: first, to determine whether the quadcopter can maintain its altitude, while satisfying physical constraints, and second, to determine the optimal reference for NMPC.

For a more physically intuitive representation of attitude, we introduce ZYX Euler angles $\boldsymbol{\eta} = \begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^\top$. For a hovering multicopter under FTC that allows yaw rotation, a dynamic equilibrium state [9] exists in which the height z, roll angle ϕ, pitch angle θ, yaw angular velocity $\dot{\psi}$, and control inputs u_i $(i = 1, \ldots, 4)$ are constant. In this state, even with thrust asymmetry due to faults, moment and time-averaged horizontal force equilibrium is maintained through tilt and yaw rotation. Although yaw rotation complicates onboard camera operation and manual control, it is necessary to maintain flight.

Let the steady-state values of each component in the dynamic equilibrium state be z_{de}, θ_{de}, ϕ_{de}, $\dot{\psi}_{\mathrm{de}}$, and $u_{i\mathrm{de}}$ $(i = 1, \ldots, 4)$. The angular velocity vector in the dynamic equilibrium state is constant and expressed as

$$\boldsymbol{\omega}_{\mathrm{de}} = \dot{\psi}_{\mathrm{de}} \begin{pmatrix} -\sin\theta_{\mathrm{de}} \\ \sin\phi_{\mathrm{de}}\cos\theta_{\mathrm{de}} \\ \cos\phi_{\mathrm{de}}\cos\theta_{\mathrm{de}} \end{pmatrix} = \mathrm{const.} \tag{7}$$

Furthermore, as the angular acceleration and vertical acceleration are zero in the dynamic equilibrium state, from the state equation (2), the dynamic equilibrium conditions for the estimated fault state can be derived as

$$\begin{aligned} \boldsymbol{T}\hat{\boldsymbol{m}}_{\mathrm{r}} \odot \boldsymbol{u}_{\mathrm{de}} - \boldsymbol{D}\boldsymbol{\omega}_{\mathrm{de}} &= \omega_{\mathrm{de}} \times \boldsymbol{J}\omega_{\mathrm{de}}, \\ \sum_{i=1}^{4} \hat{m}_{\mathrm{r}i} u_{i\mathrm{de}} \cos\phi_{\mathrm{de}} \cos\theta_{\mathrm{de}} &= mg, \end{aligned} \tag{8}$$

where $\odot$ represents the element-wise product of vectors (Hadamard product). The dynamic equilibrium state equation (8) comprises four equations with seven unknown variables. Therefore, the solution is not unique and has three degrees of freedom. Therefore, to select an optimal equilibrium state from the set of feasible solutions, we solve the following optimization problem. The objective is to minimize the yaw angular velocity, while satisfying the physical constraints:

$$\begin{aligned} J_{\mathrm{de}} &= \omega_{3\mathrm{de}}^2 + \beta \sum_{i=1}^{4} (\hat{m}_{\mathrm{r}i} u_{i\mathrm{de}})^2 \\ &= (\dot{\psi}_{\mathrm{de}} \cos\phi_{\mathrm{de}} \cos\theta_{\mathrm{de}})^2 + \beta \sum_{i=1}^{4} (\hat{m}_{\mathrm{r}i} u_{i\mathrm{de}})^2, \\ u_{\min} \leq u_{i\mathrm{de}} &\leq u_{\max}, \ |\phi_{\mathrm{de}}| \leq \phi_{\max}, \ |\theta_{\mathrm{de}}| \leq \theta_{\max}, \end{aligned} \tag{9}$$

where β is a sufficiently small positive number, and the term $\beta \sum_{i=1}^{4} (\hat{m}_{\mathrm{r}i} u_{i\mathrm{de}})^2$ is introduced to improve the convergence of the solver. The constraints on ϕ_{de} and θ_{de} prevent the quadcopter from tilting excessively. This is because it is difficult to control a large change in attitude after a fault, which increases the risk of instability. Furthermore, the upper limits $\phi_{\max}$ and $\theta_{\max}$ are set to always satisfy the condition that the yaw angular velocity component is larger than the other axis angular velocity components, $|\omega_{3\mathrm{de}}| \geq |\omega_{1\mathrm{de}}|, |\omega_{2\mathrm{de}}|$. A detailed derivation of this condition is provided in the APPENDIX.

By solving the optimization problem in equation (9) subject to the conditions in equation (8), the dynamic equilibrium state and the corresponding control input values for a fault condition are determined. If a dynamic equilibrium state exists and the obtained angular velocity satisfies the constraint $|\omega_{3\mathrm{de}}| \leq \omega_{\max}$, then from the condition shown in the APPENDIX, $|\omega_{1\mathrm{de}}|, |\omega_{2\mathrm{de}}| \leq \omega_{\max}$ holds and the quadcopter can maintain altitude, while respecting thrust and angular velocity constraints. Here, $\omega_{\max}$ is the angular velocity upper bound due to the gyro sensor and depends on its properties. Conversely, if no dynamic equilibrium state exists, the quadcopter is unable to maintain altitude, while satisfying the given thrust constraints.

D. FTC Integrating Fault Diagnosis and Dynamic Equilibrium States into NMPC

By integrating the fault diagnosis and the optimal dynamic equilibrium state described in the previous sections into NMPC, we propose a method to effectively perform FTC for various faults without controller switching or changing the performance index weights.

The proposed method has two key components. The first is the incorporation of the fault state into the NMPC prediction model. Based on the estimated effectiveness $\hat{\boldsymbol{m}}_{\mathrm{r}}$, the actual thrust generated by each rotor is modeled as $\hat{m}_{\mathrm{r}i}u_i$. Therefore, by replacing the control input $\boldsymbol{u}(t)$ with $\hat{\boldsymbol{m}}_{\mathrm{r}} \odot \boldsymbol{u}$ in the state equation (2), the system's state equation incorporating the fault state is expressed as

$$\dot{\boldsymbol{x}}_{\mathrm{d}} = \begin{pmatrix} \dot{\boldsymbol{\xi}} \\ \frac{1}{m}\boldsymbol{Q}\begin{pmatrix} 0 & 0 & \sum_{i=1}^{4} \hat{m}_{\mathrm{r}i}u_i \end{pmatrix}^{\top} - g\boldsymbol{e}_3^{\mathrm{i}} \\ \boldsymbol{J}^{-1}(\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{T}\hat{\boldsymbol{m}}_{\mathrm{r}} \odot \boldsymbol{u} - \boldsymbol{D}\boldsymbol{\omega}) \\ \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q} \end{pmatrix}. \quad (10)$$

By using the state equation (10) for optimization in NMPC, we calculate the control input $\boldsymbol{u}_{\mathrm{opt}}$ by considering the effectiveness of each rotor $\hat{\boldsymbol{m}}_{\mathrm{r}}$. Over the prediction horizon, the change in the fault state is assumed negligible. Thus, $\hat{\boldsymbol{m}}_{\mathrm{r}}$ is treated as a constant.

The second component is the online update of the NMPC references using the results from the dynamic equilibrium state optimization. Specifically, $\boldsymbol{\omega}_{\mathrm{de}}$ and $\boldsymbol{u}_{\mathrm{de}}$ are the reference values for the angular velocity and control input, respectively. The attitude reference is derived from the relationship between quaternions and Euler angles, and is set as $\left(q_1^2 + q_2^2\right)_{\mathrm{de}} = (1 - \cos\phi_{\mathrm{de}} \cos\theta_{\mathrm{de}})/2$. As the optimization of the dynamic equilibrium state may exceed one control cycle time, these optimal states are pre-computed for a grid of assumed rotor effectiveness values $\hat{\boldsymbol{m}}_{\mathrm{r}}$. The optimal reference is generated in real time by interpolating the pre-calculated data based on the currently estimated rotor effectiveness. This provides an optimal reference for maintaining altitude, regardless of the presence of a fault. For a severe fault, as the performance index J_{de} penalizes the square of the yaw angular velocity, the resulting reference effectively results in the minimum necessary yaw angular velocity. The position and velocity references can be set to track an arbitrary trajectory $\boldsymbol{x}_{\mathrm{ref}}(t) = \begin{pmatrix} \boldsymbol{\xi}_{\mathrm{ref}}^{\top}(t) & \dot{\boldsymbol{\xi}}_{\mathrm{ref}}^{\top}(t) \end{pmatrix}^{\top} \in \mathbb{R}^6$.

Furthermore, to incorporate the angular velocity constraints imposed by the gyro sensor, a penalty term is added to the NMPC performance index. The penalty function is defined as near-zero when the angular velocities are within the feasible limits, increasing sharply near the boundaries, and saturates at a value γ beyond the boundaries. This formulation enables NMPC to adjust the control input to satisfy the angular velocity constraints over the prediction horizon, allowing the solver to continue functioning and find a feasible solution even if the state temporarily violates the constraints:

$$p = \gamma \sum_{i=1}^{3} \Bigg(\frac{1 + \tanh(k_p(\omega_i - \omega_{\max}))}{2} + \frac{1 + \tanh(k_p(-\omega_{\max} - \omega_i))}{2} \Bigg), \quad (11)$$

where γ is the penalty parameter, and k_p is a positive constant representing the slope of the penalty function near the bounds.

Accordingly, the terminal cost φ and stage cost L in the performance index are defined as

$$\varphi(\boldsymbol{x}_{\mathrm{d}}) = \sum_{i=1}^{6} s_i(x_{i\mathrm{d}} - x_{i\mathrm{ref}})^2 + \sum_{i=1}^{3} s_{i+6}(\omega_i - \omega_{i\mathrm{de}})^2 + s_{10}\Big((q_1^2 + q_2^2) - (q_1^2 + q_2^2)_{\mathrm{de}}\Big)^2 + p, \quad (12)$$

$$L(\boldsymbol{x}_{\mathrm{d}}, \boldsymbol{u}) = \sum_{i=1}^{6} s_i(x_{i\mathrm{d}} - x_{i\mathrm{ref}})^2 + \sum_{i=1}^{3} s_{i+6}(\omega_i - \omega_{i\mathrm{de}})^2 + s_{10}\Big((q_1^2 + q_2^2) - (q_1^2 + q_2^2)_{\mathrm{de}}\Big)^2 + \sum_{i=1}^{4} r_i(u_i - u_{i\mathrm{de}})^2 - \sum_{i=1}^{4} r_i' u_i' + p, \quad (13)$$

where s_i $(i = 1, \ldots, 10)$, r_i $(i = 1, \ldots, 4)$, and r_i' $(i = 1, \ldots, 4)$ are the weights for the state, control input, and dummy input, respectively. Note that a dummy input u_i' is introduced to handle inequality constraints in the numerical solution of the NMPC. The formulated performance index attempts to find the control input $\boldsymbol{u}_{\mathrm{opt}}(\tau)$ that minimizes the error between the predicted states and inputs and their dynamically determined references, which are based on the estimated fault state and optimal dynamic equilibrium state over the prediction horizon. State reference tracking is achieved by the first three terms of the terminal cost (12) and stage cost (13), while thrust tracking is penalized by the fourth term of the stage cost (13). The last term of the stage cost (13) is included to uniquely determine the sign of the dummy input u_i' and prevent NMPC solver failures.

IV. SIMULATION

The effectiveness of the proposed method for dynamic equilibrium analysis and FTC was validated through simulations using a model of Coex's Clover4.2 quadcopter [15]. The physical parameters determined from system identification experiments are shown in TABLE I. All simulations were executed on a virtual Linux environment built with WSL on a general-purpose laptop (CPU: AMD Ryzen 7 6800U 2.70 GHz, RAM: 16GB, OS: Windows 11 Home).

A. Feasibility Analysis by Dynamic Equilibrium State Optimization

The feasibility of FTC was confirmed through dynamic equilibrium state optimization, as described in Section III-C. The effectiveness $\hat{m}_{\mathrm{r}i}$ of two rotors was varied from 0 to 1, while the remaining two rotors were assumed to be healthy ($\hat{m}_{\mathrm{r}i} = 1$). Subsequently, we performed optimization

TABLE I

PHYSICAL PARAMETERS OF COEX'S CLOVER4.2 QUADCOPTER

Param	Meaning	Value
m	Quadcopter mass	$0.817\,\mathrm{kg}$
$u_{\max}$	Maximum thrust of a rotor	$10.1\,\mathrm{N}$
$u_{\min}$	Minimum thrust of a rotor	$0.141\,\mathrm{N}$
$\omega_{\max}$	Maximum angular velocity of sensor	$4.36\,\mathrm{rad/s}$
l	Distance from center of gravity to each rotor	$0.117\,\mathrm{m}$
k	Proportional coefficient between reaction torque around yaw axis and thrust	$0.0088\,\mathrm{m}$
J_{xx}	Diagonal components of moment of inertia	$0.0043\,\mathrm{kg\cdot m^2}$
J_{yy}		$0.0045\,\mathrm{kg\cdot m^2}$
J_{zz}		$0.0069\,\mathrm{kg\cdot m^2}$
d_1	Rotational drag coefficient	$0.005\,\mathrm{N\cdot m\cdot s}$
d_2		$0.005\,\mathrm{N\cdot m\cdot s}$
d_3		$0.010\,\mathrm{N\cdot m\cdot s}$

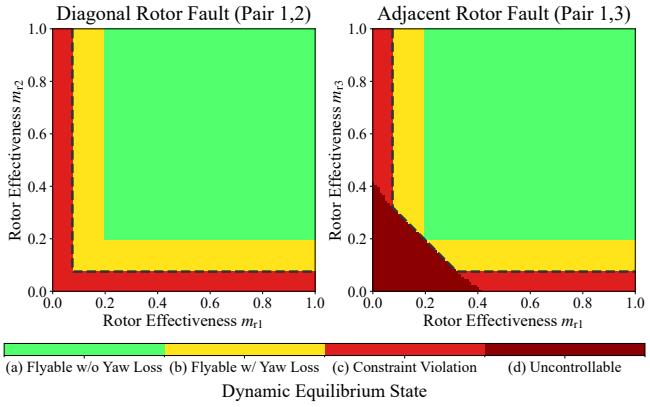

Fig. 3. Feasibility classification of FTC under two-rotor faults based on dynamic equilibrium state optimization. (a) Flyable without yaw loss: sufficient rotor effectiveness to maintain altitude with yaw control. (b) Flyable with yaw loss: altitude maintained with yaw rotation. (c) Constraint violation: dynamic equilibrium exists but $|\omega_{3\mathrm{de}}|$ exceeds $\omega_{\max}$. (d) Uncontrollable: no feasible dynamic equilibrium state under thrust constraints. The dashed lines indicate the boundaries between feasible and infeasible flight regions.

to find the dynamic equilibrium state. Two fault cases were considered based on the quadcopter's symmetry: one where the faulty rotors were diagonal to each other, and another where they were adjacent. First, we verified the existence of a solution to the optimization problem. Then, if a solution existed, the resulting angular velocity was checked to satisfy the constraint. Based on these criteria, the feasibility of FTC was comprehensively determined. The values of $\hat{m}_{\mathrm{r}i}(i = 1, 2 \text{ or } 1, 3)$ for the two faulty rotors were discretized into a 101×101 grid, and the optimization was performed at each point. The sequential least squares programming (SLSQP) method was used for optimization. To improve convergence, 50 random initial points were provided, and the solution yielding the smallest performance index was adopted.

The results of classifying the feasibility of FTC are shown in Fig. 3. The optimization time for each initial point is $12.7\,\mathrm{ms}$ on average. The left panel shows the results for diagonal rotor faults, and the right panel shows the results for adjacent rotor faults. In both cases, it is clear that for minor faults, a dynamic equilibrium state can be achieved by

TABLE II

WEIGHTS OF THE PERFORMANCE INDEX IN NMPC

Param	Value	Param	Value
$s_{1,2}$	8.35×10^4	s_3	7.51×10^4
$s_{4,5}$	3.87×10^3	s_6	1.06×10^4
$s_{7,8}$	2.52×10^2	s_9	2.50
s_{10}	1.93		
$r_{1,\ldots,4}$	2.34	$r'_{1,\ldots,4}$	0.100

TABLE III

PARAMETERS OF NMPC

Param	Meaning	Value
T_{f}	Final length of the prediction horizon	$0.167\,\mathrm{s}$
α_{NMPC}	Parameter related to the length of the time-varying prediction horizon	$2\,\mathrm{s}^{-1}$
$k_{\max}$	Maximum number of GMRES iterations	20
N	Number of discretization grids	50
ζ	Stabilization parameter of C/GMRES	$200\,\mathrm{s}^{-1}$

increasing the control inputs of the faulty rotors without sacrificing yaw control. The condition for a rotor with reduced effectiveness to produce the thrust required for stationary hovering is $\hat{m}_{\mathrm{r}i} \geq \frac{mg}{4u_{\max}} \approx 0.199$. This result agrees well with the results shown in Fig. 3, suggesting that this simple physical constraint is dominant. Conversely, for severe faults, a dynamic equilibrium state can be achieved by allowing yaw rotation, and when $\hat{m}_{\mathrm{r}i}$ deviates from 1, the yaw angular velocity also increases and eventually exceeds the constraint. When two adjacent rotors fail severely and simultaneously, a dynamic equilibrium state may not exist. This is because the remaining two adjacent rotors are unable to generate the necessary torque to control the roll and pitch attitudes. In contrast, when two diagonal rotors fail completely, the quadcopter can maintain a dynamic equilibrium state. This is because the remaining two rotors are located diagonally, allowing them to control the pitch and roll attitude, and yaw angular velocity is balanced by air drag.

The results in Fig. 3 are physically consistent and demonstrate that the feasibility of FTC can be evaluated by optimizing the dynamic equilibrium state. Although this section is focused on two-rotor faults, the feasibility of FTC for faults involving three or four rotors can be evaluated likewise.

B. Simulation of FTC Targeting Dynamic Equilibrium State

To validate the effectiveness of the FTC strategy, we performed a simulation of FTC using NMPC with the reference from the dynamic equilibrium state and employed a penalty term for angular velocity constraints as shown in Fig. 2. For the NMPC implementation, we used AutoGenU for Jupyter [16], [17], an automatic code generation tool for NMPC using the C/GMRES method. The optimal control input $\boldsymbol{u}_{\mathrm{opt}}$ was calculated in real time by calling a C++ library generated by AutoGenU through a Python interface. Fault diagnosis was performed in real time, concurrently with the NMPC calculations. Identical model parameters were used for the simulation and NMPC. The weights for the state and control inputs in the performance index were

set to constants independent of the fault state, as shown in TABLE II. For the penalty term (11), the gain and slope were set to $\gamma = 1000$ and $k_p = 50$, respectively. The NMPC parameters are listed in TABLE III. The sampling period was set to $5\,\mathrm{ms}$, as prior tests on a Raspberry Pi 4 [18] showed $1.7\,\mathrm{ms}$ computation time. To clearly observe changes in the position and attitude, the reference and initial state were set to stationary hovering. Under this condition, we verified that the proposed method maintained a hovering state, while minimizing the yaw angular velocity for various rotor faults.

In a 20-second simulation, at $t = 3\,\mathrm{s}$, the effectiveness of one rotor was reduced by 40%. Subsequently, at $t = 10\,\mathrm{s}$, the effectiveness of an adjacent rotor was additionally reduced by 90%. Based on the results from the previous section (Fig. 3), a dynamic equilibrium state can be achieved for both fault conditions. For the single rotor with 40% fault, control is achievable with zero yaw angular velocity. For the two-rotor fault, control is achievable by allowing some yaw rotation. The simulation results are shown in Fig. 4. In each plot, the fault initiation times $t = 3,\ 10\,\mathrm{s}$ are indicated by dotted lines. The gray shaded areas represent regions where the angular velocity and control input constraints are violated. The references in the performance index are indicated by dashed lines. For the minor fault in the first rotor, the position is maintained with zero yaw angular velocity, except for the transient at the moment of occurrence of the fault. For the subsequent severe fault in the adjacent rotor, the position is maintained without violating the angular velocity constraints, while allowing yaw rotation. The maximum position error after the fault is less than $1\,\mathrm{cm}$. In the steady state after each fault, the angular velocity and control input track the reference values derived from the dynamic equilibrium state, indicating that the controller successfully minimizes the yaw angular velocity. Furthermore, the references obtained from the dynamic equilibrium state optimization change smoothly, and the interpolation errors are negligible.

To validate the effectiveness of the proposed method in minimizing yaw angular velocity, comparative simulations were performed under two alternative scenarios: one with a fixed reference corresponding to a stationary hovering, and another with the same fixed reference, but without the penalty term for angular velocity constraints. The root mean square error (RMSE) in the translational direction is shown in TABLE IV, and time histories of the yaw angular velocity are shown in Fig. 5. The results show that the position control RMSE is less than $2\,\mathrm{mm}$ for all methods, which is practically negligible. Furthermore, the maximum position error is less than $1\,\mathrm{cm}$ for all methods, indicating that all three controllers are capable of maintaining a hovering state. In contrast, Fig. 5 reveals significant differences in yaw control. In the minor fault case, by using the dynamic equilibrium state as a reference, the yaw angular velocity is maintained at zero. Setting the reference to the hovering state results in unnecessary yaw angular velocity. In the severe fault case, the yaw angular velocity is larger when using the hovering state reference compared to those from the dynamic equilibrium state. With the penalty term, the angular velocity is kept within the constraint; while the constraint is violated without the penalty term. Violation of this angular velocity constraint can result in sensor saturation and lead to control instability.

TABLE IV

RMSE OF X, Y, AND Z AXES UNDER DIFFERENT CONTROLS

Control	RMSE		
	$x\,(\mathrm{m})$	$y\,(\mathrm{m})$	$z\,(\mathrm{m})$
W/ DES	1.08×10^{-3}	1.18×10^{-3}	1.30×10^{-3}
W/O DES	1.33×10^{-3}	0.34×10^{-3}	1.27×10^{-3}
W/O DES&penalty	1.35×10^{-3}	0.35×10^{-3}	1.24×10^{-3}

Note: W/ DES is the proposed method with references from Dynamic Equilibrium State; W/O DES is the control with a fixed hovering reference; W/O DES&penalty is the control with a fixed hovering reference and without the penalty term.

The above simulation results demonstrate that the proposed method with a single set of weights can maintain the position, while minimizing yaw angular velocity for both minor and severe faults. Furthermore, the effectiveness of the penalty term in preventing angular velocity constraint violations is also confirmed.

V. CONCLUSION

In this study, we propose an FTC method for quadcopters, effective against various faults, that integrates dynamic equilibrium analysis with NMPC. The proposed method first quantitatively evaluates the feasibility of FTC under thrust and angular velocity constraints for any given fault condition by optimizing the quadcopter's dynamic equilibrium state. This process clarifies the range of faults within which altitude can be maintained, as demonstrated by simulations. By using these optimization results as dynamic references for NMPC, two key contributions are made. First, it is no longer necessary to change the weights of the performance index according to the fault state as a single fixed set of weights can handle both minor and severe faults. Second, the practical reliability is enhanced by explicitly handling sensor constraints to maintain an attitude that minimizes the yaw angular velocity. Simulations have demonstrated that the proposed method achieves stable control, while satisfying the sensor constraints, even under severe fault conditions that would cause conventional methods to violate these constraints. Future work will include experimental validation on a real quadcopter and the development of a safe landing control strategy for cases in which a dynamic equilibrium state cannot be achieved.

APPENDIX
DERIVATION OF ATTITUDE ANGLE CONSTRAINTS

The attitude angle constraints used in the main text, $|\phi_{\mathrm{de}}| \leq \phi_{\max}$ and $|\theta_{\mathrm{de}}| \leq \theta_{\max}$, are imposed to ensure that the angular velocity component around the yaw axis is greater than or equal to the other two components, $|\omega_{3\mathrm{de}}| \geq |\omega_{1\mathrm{de}}|, |\omega_{2\mathrm{de}}|$. Here, we make the constraints for $|\phi_{\mathrm{de}}|$ and $|\theta_{\mathrm{de}}|$ the same, setting $\phi_{\max} = \theta_{\max} = \tau_{\max}$. As the rotors can produce only positive thrust, we assume $|\phi_{\mathrm{de}}| < \pi/2$

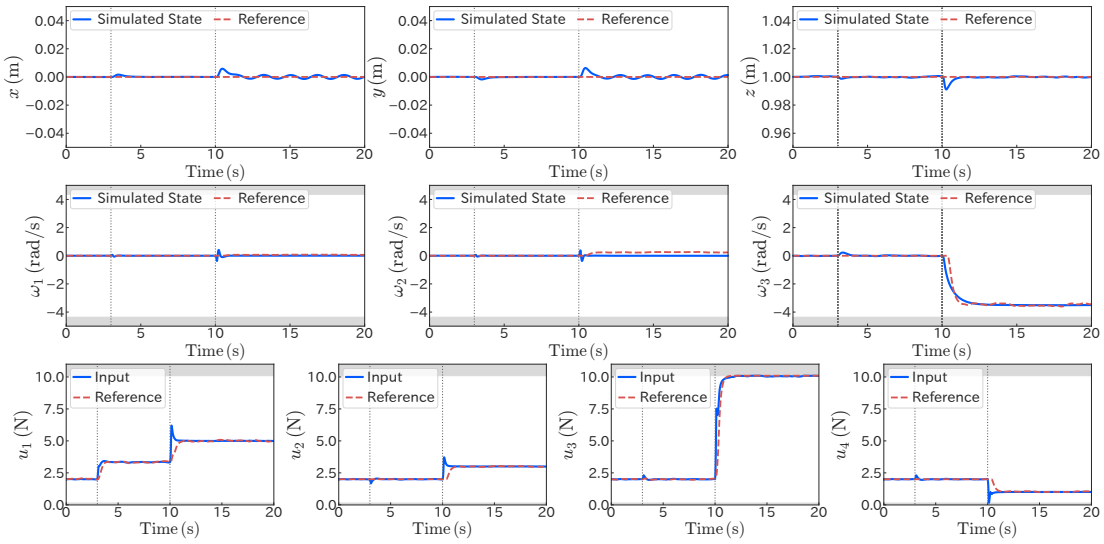

Fig. 4. Time histories of position, angular velocity, and control inputs using the proposed NMPC with references from dynamic equilibrium state. A 40% effectiveness loss is introduced to one rotor at $t = 3\,\mathrm{s}$, followed by a 90% loss to an adjacent rotor at $t = 10\,\mathrm{s}$.

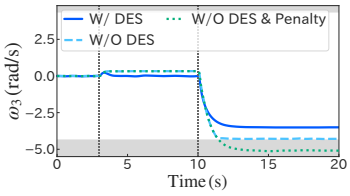

Fig. 5. Comparison of yaw angular velocity, demonstrating the superiority of the proposed method in minimizing rotation during single and two-rotor faults. The fault scenario is identical to that in Fig. 4.

and $|\theta_{\mathrm{de}}| < \pi/2$. From the angular velocity equation (7) for dynamic equilibrium, the aforementioned condition, $|\omega_{3\mathrm{de}}| \geq |\omega_{1\mathrm{de}}|, |\omega_{2\mathrm{de}}|$, is equivalent to the following two inequalities being satisfied for all angles within the constraint range.

$$\cos\phi_{\mathrm{de}} \geq |\tan\theta_{\mathrm{de}}| \quad \text{and} \quad 1 \geq |\tan\phi_{\mathrm{de}}|. \tag{14}$$

To always satisfy these inequalities, the most stringent condition, $\cos\tau_{\max} \geq \tan\tau_{\max}$, must be satisfied (in this case, $1 \geq |\tan\tau_{\max}|$ is also satisfied). Solving this inequality yields the following condition.

$$\sin\tau_{\max} \leq (\sqrt{5}-1)/2. \tag{15}$$

Therefore, the upper bound $\tau_{\max}$ is set as

$$\tau_{\max} \leq \arcsin\left(\frac{\sqrt{5}-1}{2}\right) \approx 0.666\,\mathrm{rad}. \tag{16}$$

References

[1] L. R. García Carrillo, A. E. Dzul López, R. Lozano, and C. Pégard, *Quad Rotorcraft Control: Vision-Based Hovering and Navigation*, ser. Advances in Industrial Control. London: Springer-Verlag, 2012.

[2] T. Li, Y. Zhang, and B. W. Gordon, "Passive and active nonlinear fault-tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 227, no. 1, pp. 12–23, 2013.

[3] B. Wang and Y. Zhang, "Adaptive sliding mode fault-tolerant control for an unmanned aerial vehicle," *Unmanned Systems*, vol. 5, no. 4, pp. 209–221, 2017.

[4] A. Freddi, A. Lanzon, and S. Longhi, "A feedback linearization approach to fault tolerance in quadrotor vehicles," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 5413–5418, 2011.

[5] N. P. Nguyen and S. K. Hong, "Fault diagnosis and fault-tolerant control scheme for quadcopter UAVs with a total loss of actuator," *Energies*, vol. 12, no. 6, art. no. 1139, 2019.

[6] W. Jung and H. Bang, "Fault and failure tolerant model predictive control of quadrotor UAV," *International Journal of Aeronautical and Space Sciences*, vol. 22, pp. 663–675, 2021.

[7] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, "Nonlinear MPC for quadrotor fault-tolerant control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5047–5054, 2022.

[8] J. Yeom, R. B. TMB, G. Li, and G. Loianno, "Experimental system design of an active fault-tolerant quadrotor," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 814–821.

[9] Y. Aoki, Y. Asano, A. Honda, N. Motooka, K. Hoshino, and T. Ohtsuka, "Nonlinear model predictive control for hexacopter with failed rotors based on quaternions—simulations and hardware experiments—," *Mechanical Engineering Journal*, vol. 8, no. 5, paper 21–00204, 2021.

[10] D. Muroki, Y. Asano, T. Noro, T. Shima, and T. Ohtsuka, "Fault-tolerant control of quadcopter integrating rotor thrust estimation and nonlinear model predictive control," *Transactions of the Institute of Systems, Control and Information Engineers*, vol. 69, no. 6, pp. 109–118, 2025, (in Japanese).

[11] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, ser. Intelligent Robotics and Autonomous Agents series. Cambridge: MIT Press, 2005, pp. 489–498.

[12] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8189–8195, 2017.

[13] T. Ohtsuka, Ed., *Practical Applications of Control by Real-Time Optimization*. Tokyo: Corona Publishing, 2015, (in Japanese).

[14] T. Ohtsuka, *Introduction to Nonlinear Optimal Control*. Tokyo: Corona Publishing, 2011, (in Japanese).

[15] "COEX Clover," accessed Jul. 30, 2025. [Online]. Available: https://clover.coex.tech/en/

[16] "AutoGenU for Jupyter," accessed Jul. 30, 2025. [Online]. Available: https://github.com/ohtsukalab/autogenu-jupyter

[17] S. Katayama and T. Ohtsuka, "Automatic code generation tool for nonlinear model predictive control with Jupyter," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7033–7040, 2020.

[18] "Raspberry Pi 4," accessed Jul. 30, 2025. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/