Científica

vol. 30, no. 2, July-December 2026, pp. 01-29
ISSN 2594-2921, Instituto Politécnico Nacional, MÉXICO
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

ARTICLE

Controlled Neural ODE and GRU Models for Physics-Based Multiclass Fault Detection in Quadrotor Drones

Modelos Neural ODE controlados y GRU para detección multiclase de fallas basada en física en drones cuadrorrotor

César **Solís Cervantes** [1]
Jorge **Morales Mercado** [2]
Carlos **Montelongo Vázquez** [3]
Gerardo **Sonck Martínez** [4]

Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Ciudad de México, MÉXICO

[1] ORCID: 0000-0003-1011-8005 / csolisc@ipn.mx
[2] ORCID: 0009-0002-0928-9907 / jomoralesme@ipn.mx
[3] ORCID: 0000-0002-5137-1093 / cmontelongov@ipn.mx

Tecnológico Nacional de México,
Instituto Tecnológico de Chihuahua,
Chihuahua, MÉXICO

[4] ORCID: 0000-0002-5214-1679 / arno.sm@chihuahua.tecnm.mx

Recibido 16/06/2026, aceptado 24/08/2026, publicado 31-08-2026.

Abstract

This paper presents a fault-diagnosis methodology for detecting progressive faults in quadrotor drones using a controlled Neural ODE network and a GRU recurrent baseline. Training and evaluation data were obtained from a numerical simulation of the physical system, including vehicle dynamics, battery electrical behavior, motor thermal response, and randomly scheduled long-duration degradations. The fault modes considered are battery degradation, loss of motor effectiveness, and overheating. Simulated telemetry was corrupted with measurement noise and slow sensor drift to approximate realistic operating conditions. The analysis compares a continuous latent-space formulation with a discrete recurrent architecture under the same temporal-window representation, class definitions, and evaluation criteria. The proposed framework provides a reproducible basis for studying early incipient fault diagnosis in unmanned aerial vehicles prior to experimental validation using flight logs.

Index terms: battery fault, controlled Neural ODE, drone fault detection, gated recurrent unit, overheating fault.

Resumen

Este trabajo presenta una metodología de diagnóstico para detectar fallas progresivas en drones cuadrorrotadores mediante una red Neural ODE controlada y una red GRU utilizada como referencia recurrente. La información de entrenamiento y evaluación se obtuvo mediante una simulación numérica que integra la dinámica de vuelo, el comportamiento eléctrico de la batería, la eficiencia de los motores, el calentamiento térmico y el ruido de los sensores. Las fallas consideradas corresponden a la degradación de la batería, la pérdida de efectividad del motor y el sobrecalentamiento, con inicios aleatorios y evoluciones graduales, para representar escenarios de larga duración. El estudio compara ambas arquitecturas bajo la misma partición de datos, las mismas clases y la misma función de pérdida, de modo que la evaluación se enfoque en su capacidad para extraer patrones temporales a partir de la telemetría. El manuscrito desarrolla los modelos físicos empleados para generar las señales, define la formulación de aprendizaje, describe la configuración de la simulación numérica y organiza los resultados mediante tablas y figuras que permiten interpretar la clasificación multiclase y la detección de operaciones anómalas.

Palabras clave: detección de fallas, dron cuadrorrotor, red Neural ODE, red GRU, simulación numérica.

I. INTRODUCTION

Fault detection in unmanned aerial vehicles is an important research problem because progressive degradation may occur over many flight samples before a safety threshold is violated. Existing unmanned aerial vehicle fault-diagnosis methods include model-based observers, data-driven classifiers, and fault-tolerant control strategies [1]. Data-driven process monitoring has also shown that temporal classifiers can detect weak deviations in nonlinear, noisy systems [2].

Model-based diagnosis remains attractive because it provides physical interpretability, but it can be sensitive to unmodeled aerodynamic coupling, battery sag, thermal effects, and nonstationary noise [3]. These limitations motivate hybrid studies in which a physically interpretable numerical simulator generates telemetry while learning models compare the temporal signatures associated with different fault mechanisms.

A. State of the Art

Quadrotor modeling has been extensively studied using nonlinear rigid-body dynamics, rotor thrust allocation, and trajectory-generation methods. These models are suitable for controlled numerical experiments because motor commands, attitude dynamics, and translational motion are explicitly coupled [5], [6], [7]. Battery and thermal subsystems are also relevant for fault diagnosis: equivalent-circuit battery models capture state-of-charge evolution and voltage sag, while lumped thermal models describe heat accumulation and dissipation in electrical devices [8], [9].

Neural Ordinary Differential Equations (Neural ODEs) model a network's hidden representation via a continuous-time differential equation [10]. Controlled neural differential models extend this idea by conditioning the latent dynamics on an input path [11]. GRU networks, in contrast, update a discrete hidden state using reset and update gates, thereby providing a compact recurrent baseline for sequence classification [12]. Related advances have also shown the effectiveness of combining Transformer architectures with GRU networks to identify nonlinear robotic kinematics through Product-of-Exponentials coordinates, highlighting hybrid sequence-learning models' ability to capture complex dynamic relationships and long-term dependencies in engineering systems [13]. These developments further motivate exploring deep temporal architectures for fault diagnosis in UAV applications.

B. Contributions

The main contributions of this work are: first, a numerical simulation framework that couples quadrotor rigid-body dynamics with battery, motor-effectiveness, and overheating fault models; second, a multiclass fault-detection formulation that compares a controlled Neural ODE classifier with a GRU baseline under identical data partitions and loss functions; third, a compact mathematical description of the state variables, fault scheduling, measurement model, classifiers, loss function, and metrics, together with a complete table of symbols; and fourth, a numerical evaluation with figures and tables that summarize training behavior, confusion matrices, class-level performance, any-fault detection behavior, and the evolution of the drone states and of the underlying fault mechanisms under each fault condition.

II. THEORETICAL FRAMEWORK AND MATHEMATICAL MODELS

This section presents the mathematical description of the simulated system, the fault mechanisms, and the learning architectures. All symbols used in the manuscript are collected in Table 1; each symbol is additionally defined in the text at its first appearance.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, MÉXICO
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

TABLE 1
NOMENCLATURE AND SYMBOL DEFINITIONS

Symbol	Description	Units
$\boldsymbol{p(t), v(t)}$	Position and velocity of the center of mass, inertial frame	m; m/s
$\boldsymbol{\eta = (\varphi, \theta, \psi)}$	Euler angles: roll, pitch, yaw	rad
$\boldsymbol{\omega_B}$	Angular velocity expressed in the body frame	rad/s
$\boldsymbol{x(t)}$	Augmented state vector (Eq. (1))	—
$\boldsymbol{R_B^I(\eta)}$	Rotation matrix from body frame to inertial frame	—
$\boldsymbol{R_x(\varphi), R_y(\theta), R_z(\psi)}$	Elementary rotation matrices about the x, y, z axes	—
$\boldsymbol{E(\eta)}$	Euler-angle-rate matrix mapping ω_B to $\dot{\eta}$	—
m	Rigid-body mass	kg
g	Gravitational acceleration	m/s²
$\boldsymbol{e_3}$	Unit vector along the inertial z axis	—
$\boldsymbol{J}$	Inertia matrix, body frame	kg·m²
$\boldsymbol{F_B, M_B}$	Total thrust and body-frame moment vector	N; N·m
$\boldsymbol{F_i\ (i = 1, \ldots, 4)}$	Thrust produced by rotor i	N
T	Total thrust, $T = \Sigma F_i$	N
L	Distance from the center of mass to each rotor	m
$\boldsymbol{\gamma}$	Yaw torque coefficient per unit thrust	—
$\boldsymbol{\tau_{roll}, \tau_{pitch}, \tau_{yaw}}$	Commanded roll, pitch, and yaw torques	N·m
$\boldsymbol{k_L, k_\Omega}$	Translational and rotational aerodynamic drag coefficients	N·s/m; N·m·s/rad
$\boldsymbol{SoC(t)}$	Battery state of charge	—
$\boldsymbol{V_b(t)}$	Battery terminal (bus) voltage	V
$\boldsymbol{V_{ocv}(SoC)}$	Battery open-circuit voltage	V
$\boldsymbol{I_b(t)}$	Battery bus current	A
$\boldsymbol{R_0}$	Nominal battery internal resistance	Ω
$\boldsymbol{C_{nom}}$	Nominal battery capacity	A·h
$\boldsymbol{P_{prop}, P_{av}}$	Total propeller power and avionics (static) power	W
$\boldsymbol{c_P}$	Propeller power coefficient	—
$\boldsymbol{a_P}$	Exponent of the thrust–power relation	—
$\boldsymbol{T_m(t)}$	Temperature of motor m ($m = 1, \ldots, 4$)	°C
$\boldsymbol{C_{th}}$	Lumped thermal capacity of a motor	J/K
$\boldsymbol{R_w}$	Motor winding electrical resistance	Ω
$\boldsymbol{I_m(t)}$	Current supplied to motor m	A
$\boldsymbol{c_F}$	Aerodynamic mechanical-loss coefficient	—
$\boldsymbol{h_{nom}}$	Nominal convective heat-transfer coefficient	W/K
$\boldsymbol{T_{amb}}$	Ambient temperature	°C
$\boldsymbol{Q_{\{f,m\}}(t)}$	Fault heat source of motor m	W
$\boldsymbol{s_k(t), k \in \{b, m, h\}}$	Severity of fault class k (battery, motor, overheat)	—
$\boldsymbol{t_{\{on,k\}}, T_{\{r,k\}}}$	Onset time and ramp duration of fault k	s
$\boldsymbol{a_R, a_C}$	Amplitudes of internal-resistance increase and capacity loss (battery)	—
$\boldsymbol{\Delta V, a_I}$	Additional voltage-sag amplitude and current-increase amplitude (battery)	V; —

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, MÉXICO
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

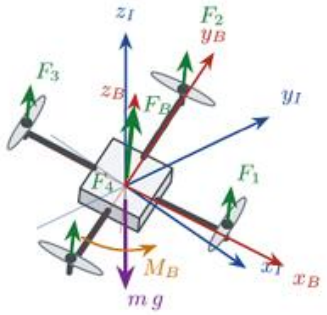
Symbol	Description	Units
a_M	Motor-effectiveness loss amplitude (motor fault)	—
m_m , m_h	Index of the faulty (motor) or overheated motor	—
$c_i(t)$	Effective thrust coefficient of rotor i	—
a_h , a_Q	Cooling-coefficient loss amplitude and fault heat amplitude (overheat)	—; W
h_{min}	Lower bound of the heat-transfer coefficient	W/K
λ_T , T_{der}	Thermal derating rate and derating onset temperature	1/°C; °C
a_d , d_{min}	Overheat-induced effectiveness-loss amplitude and its lower bound	—
$d_m(t)$	Thermal derating factor of motor m	—
$F_{\{i,cmd\}}$, $F_{\{i,act\}}$	Commanded and effective (actuated) thrust of rotor i	N
κ_V	Voltage-sensitivity coefficient	—
$y(t)$	Measured telemetry vector	—
C	Measurement matrix	—
$b(t)$	Slowly varying sensor bias	—
$n(t)$	Zero-mean Gaussian measurement noise	—
Σ_n, Σ_b	Noise covariance and bias random-walk covariance	—
$\varepsilon(t)$	Random-walk increment of the sensor bias	—
Δt	Sampling period	s
z_t, z_0, z_1	Continuous latent state and its initial/final values	—
E_φ, f_θ, g_ψ	Encoder, latent-dynamics, and classification-head networks	—
u_t	Time-interpolated telemetry (control) path	—
$\hat{y}$	Predicted class-probability vector	—
h_t	GRU hidden state	—
r_t, z_t (GRU)	Reset gate and update gate	—
W_r, W_z, W_h, W_o	GRU weight matrices	—
b_r, b_z, b_h, b_o	GRU bias vectors	—
$\sigma(\cdot)$, $tanh(\cdot)$, $\odot$	Sigmoid, hyperbolic tangent, and element-wise product	—
$L(\theta)$	Weighted cross-entropy loss	—
B, K	Mini-batch size and number of classes	—
$y_{\{b,k\}}$, $\dot{Z}_{\{b,k\}}$	One-hot target and predicted probability (sample b, class k)	—
w_k	Class weight (inverse frequency)	—
ε	Logarithm regularization constant	—
TP_k, FP_k, TN_k, FN_k	Confusion counts for class k	—
$P_k, R_k, F1_k, Brier_k$	Precision, recall, F1 score, and Brier score of class k	—

A. Quadrotor State and Rigid-Body Dynamics

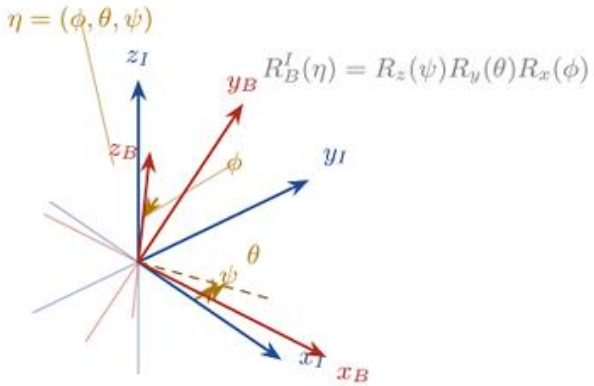
The simulator uses an augmented state that combines the quadrotor rigid-body state with the battery and thermal variables. Position and velocity are expressed in the inertial frame I; the attitude is parameterized by the Euler angles $\eta = (\varphi, \theta, \psi)$ (roll, pitch, and yaw, respectively); the angular velocity is expressed in the body frame B; the state of charge and the terminal voltage describe the battery; and each motor has an individual temperature state. Fig. 1 shows

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

a free-body diagram of the quadrotor with both the body-fixed frame B and the inertial reference frame I, the four rotor thrusts F_1, F_2, F_3, F_4, the total body-frame thrust F_B, the gravity force mg, and the body moments M_B. To avoid ambiguity between grouping symbols and algebraic operations, the augmented state is written as an ordered tuple in Eq. (1).



(a) Free-body diagram: rotor thrusts $F_1 \dots F_4$, resultant F_B, gravity $m\,g$, moment M_B, body frame B and inertial frame I.



(b) Attitude: Euler angles (ZYX) $\eta = (\phi, \theta, \psi)$ and rotation matrix $R_B^I(\eta) = R_z(\psi)R_y(\theta)R_x(\phi)$.

Fig. 1. Free-body diagram of the quadrotor showing the inertial reference frame (I), the body-fixed frame (B), the Euler attitude angles ($\eta = (\phi, \theta, \psi)$), the rotor thrusts ($F_1$,...,$F_4$), the resultant body-frame thrust (F_B), the gravitational force (mg), and the body moment (M_B).

$$x(t) = (p(t), v(t), \eta(t), \omega_B(t), SoC(t), V_b(t), T_m(t)), \quad m = 1, \dots, 4 \tag{1}$$

Here $p(t) \in \mathbb{R}^3$ is the position, $v(t) \in \mathbb{R}^3$ the velocity, $\eta(t)$ the attitude, $\omega_B(t)$ the body angular velocity, $SoC(t)$ the battery state of charge, $V_b(t)$ the terminal voltage, and $T_m(t)$ the temperature of motor m ($m = 1, \dots, 4$). The compact rigid-body model in Eq. (2) combines translational kinematics, translational dynamics, attitude kinematics, and rotational dynamics. In Eq. (2), $R_B^I(\eta)$ maps body-frame vectors to the inertial frame, $E(\eta)$ maps the body angular velocity to the Euler-angle rates, J is the inertia matrix, $e_3 = (0, 0, 1)^T$ is the unit vector along the inertial z axis, k_L and k_Ω are the translational and rotational aerodynamic drag coefficients, respectively, and M_B is the body-frame

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

moment vector. The rotation matrix is parameterized by the Euler angles as $R_B^I(\eta) = R_z(\psi)R_y(\theta)R_x(\varphi)$, where R_x, R_y, and R_z are the elementary rotation matrices about the x, y, and z axes, respectively.

$$\begin{aligned} \dot{p} &= v \\ m\dot{v} &= R_B^I(\eta)F_B - mge_3 - k_L\, v \\ \dot{\eta} &= E(\eta)\,\omega_B \\ J\dot{\omega}_B &= M_B - \omega_B \times (J\omega_B) - k_\Omega \omega_B \\ E(\eta) &= \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta \end{bmatrix} \\ R_B^I(\eta) &= R_z(\psi)R_y(\theta)R_x(\varphi) \end{aligned} \tag{2}$$

Each rotor produces a thrust that is proportional to the square of its rotational speed, and the four thrusts are combined by a static mixer that distributes the total thrust T and the commanded torques τ_{roll}, τ_{pitch}, and τ_{yaw} among the rotors. The distance from the center of mass to each rotor is L, and γ is the yaw torque coefficient per unit thrust. Equation (3) defines the body-frame force and moment vectors used by the flight dynamics: the total thrust is $F_B = e_3\, \Sigma\, F_i$, and the body moments are generated by the differential thrusts according to the last two lines of Eq. (3).

$$\begin{aligned} F_1 &= \frac{T}{4} - \frac{\tau_{pitch}}{2L} + \frac{\tau_{yaw}}{4\gamma} \\ F_2 &= \frac{T}{4} + \frac{\tau_{roll}}{2L} - \frac{\tau_{yaw}}{4\gamma} \\ F_3 &= \frac{T}{4} + \frac{\tau_{pitch}}{2L} + \frac{\tau_{yaw}}{4\gamma} \\ F_4 &= \frac{T}{4} - \frac{\tau_{roll}}{2L} - \frac{\tau_{yaw}}{4\gamma} \\ M_B &= \left(L(F_2 - F_4),\quad L(F_3 - F_1),\quad \gamma(F_1 - F_2 + F_3 - F_4)\right)^T \\ F_B &= e_3(F_1 + F_2 + F_3 + F_4) \end{aligned} \tag{3}$$

B. Battery, Motor, and Overheating Dynamics

The battery model represents the state-of-charge depletion, the voltage sag, and the electrical demand required to sustain the commanded thrust. In Eq. (4), $SoC(t)$ is the state of charge, C_{nom} is the nominal capacity, $V_{ocv}(SoC)$ is the open-circuit voltage, R_0 is the nominal internal resistance, $I_b(t)$ is the bus current, P_{prop} is the total electrical power absorbed by the four propellers, P_{av} is the avionics (static) power, c_P is the propeller power coefficient, and a_P is the exponent of the thrust–power relation. The effective capacity and the internal resistance vary when a battery fault is active, as described in Eq. (7).

$$\begin{aligned} \frac{d\mathrm{SoC}}{dt} &= -\frac{I_b}{C_{nom}} \\ V_b(t) &= V_{ocv}(\mathrm{SoC}) - I_b R_0 \\ I_b &= \frac{P_{prop} + P_{av}}{V_b} \\ P_{prop} &= c_P \sum_{i=1}^{4} \max(F_i, 0)^{a_P} \end{aligned} \tag{4}$$

The thermal model is lumped and independent for each motor. In Eq. (5), C_{th} is the thermal capacity of the motor, $T_m(t)$ is its temperature, R_w is the winding electrical resistance, $I_m(t)$ is the current supplied to the motor, c_F is the aerodynamic mechanical-loss coefficient, h_{nom} is the nominal convective heat-transfer coefficient, and T_{amb} is the ambient temperature. Heating increases with the electrical and mechanical losses, whereas convective cooling removes heat relative to the ambient temperature. The fault heat source $Q_{\{f,m\}}(t)$ is inactive during nominal operation and becomes nonzero when overheating is scheduled.

$$\begin{aligned} C_{th}\frac{dT_m}{dt} &= R_w {I_m}^2 + c_F \max(F_m, 0)^2 - h_{nom}(T_m - T_{amb}), \quad m = 1, \dots, 4 \\ I_m &= \frac{F_m I_b}{\sum_{i=1}^{4} F_i} \end{aligned} \tag{5}$$

C. Random Progressive Fault Scheduling and Measurement Model

Each faulty trajectory contains one active fault class. The onset time, the ramp duration, and the final severity are random, which prevents the classifiers from learning a fixed fault instant. The saturation operator in Eq. (6) clips its argument below zero and above one, producing a gradual ramp that remains bounded. In Eq. (6), $s_k(t)$ is the severity of the active fault of class $k \in \{b, m, h\}$ (battery, motor, overheat), $s_{\{max,k\}}$ is its final severity, $t_{\{on,k\}}$ is the onset time, and $T_{\{r,k\}}$ is the ramp duration.

$$\begin{aligned} \mathrm{sat}(u) &= \min\{\max(u, 0), 1\} \\ s_k(t) &= s_{max,k}\,\mathrm{sat}\frac{(t - t_{on,k})}{T_{r,k}}, \quad k \in \{b, m, h\} \end{aligned} \tag{6}$$

The battery fault increases the internal resistance, reduces the effective capacity, and adds an extra voltage sag. The motor fault reduces the effectiveness of one randomly selected motor. The corresponding mechanisms are summarized in Eq. (7). In Eq. (7), a_R and a_C are the relative amplitudes of the internal-resistance increase and the capacity loss, ΔV is the additional voltage-sag amplitude, a_I is the current-increase amplitude, a_M is the motor-effectiveness loss amplitude, and m_m is the index of the faulty motor. The coefficient ($c_i(t)$) is the effective thrust coefficient of rotor (i): it equals one for healthy rotors and is reduced by a factor (a_M) when rotor (i) is the faulty one (($i = m_m$)). Here, (1) denotes the vector of ones, and the subscript ($i = m_m$) indicates that the motor-effectiveness loss is applied only to the faulty rotor.

$$R_{int}(t) = R_0\big(1 + a_R s_b(t)\big),\ C_{eff}(t) = C_{nom}\big(1 - a_C s_b(t)\big),\ \Delta V_f(t) = \Delta V s_b(t)$$
$$I_b(t) = \frac{P_{prop} + P_{av}}{V_b}\big(1 + a_I s_b(t)\big) \tag{7}$$
$$c_i(t) = 1 - a_M s_m(t) 1_{i = m_m}$$

The overheating fault reduces the local cooling coefficient of one randomly selected motor, injects additional heat, and activates a thermal derating mechanism that reduces motor effectiveness when the temperature exceeds a safe limit. These mechanisms are summarized in Eq. (8). In Eq. (8), a_h and a_Q are the cooling-coefficient loss amplitude and the fault heat amplitude, respectively, h_{min} is the lower bound of the heat-transfer coefficient, m_h is the index of the overheated motor, λ_T and T_{der} are the thermal derating rate and the derating onset temperature, a_d is the overheat-induced effectiveness-loss amplitude, d_{min} is the lower bound of the derating factor, $F_{\{i,cmd\}}$ is the commanded thrust, $F_{\{i,act\}}$ is the effective (actuated) thrust, and κ_V is the voltage-sensitivity coefficient.

$$h_m(t) = \max\{h_{nom}\big(1 - a_h s_h(t) 1_{m = m_h}\big), h_{min}\},\ Q_{f,m}(t) = a_Q s_h(t) 1_{m = m_h}$$
$$d_m(t) = \max\big\{(1 - \lambda_T \max(T_m - T_{der}, 0))\big(1 - a_d s_h(t) 1_{m = m_h}\big),\ d_{min}\big\} \tag{8}$$
$$F_{i,act} = c_i(t) d_m(t) \kappa_V F_{i,cmd}$$

The observable telemetry is generated from the augmented state via the measurement matrix C. The noise term $n(t)$ models zero-mean Gaussian measurement noise with covariance Σ_n, whereas $b(t)$ is a slowly varying bias driven by the random-walk increment $\varepsilon(t)$ with covariance Σ_b. This distinction is useful because prolonged faults may be confused with slow sensor drift.

$$y(t) = Cx(t) + b(t) + n(t)$$
$$n(t) \sim \mathrm{N}(0, \Sigma_n),\ b(t) = b(t - \Delta t) + \sqrt{\Delta t}\, \varepsilon(t),\ \varepsilon(t) \sim \mathrm{N}(0, \Sigma_b) \tag{9}$$

D. Controlled Neural ODE Classifier

The controlled Neural ODE architecture is shown in Fig. 2. A normalized telemetry window is encoded into an initial latent state z_0, propagated by a continuous-time latent dynamics model, and mapped to class probabilities. In Eq. (10), E_φ denotes the encoder network with parameters φ, f_θ denotes the latent-dynamics network with parameters θ, g_ψ denotes the classification head with parameters ψ, and u_t is the time-interpolated telemetry path associated with the same window [10], [11]. The latent state is integrated over the normalized time interval [0, 1] with an adaptive ODE solver, so that the number of internal steps adapts to the local complexity of the trajectory.

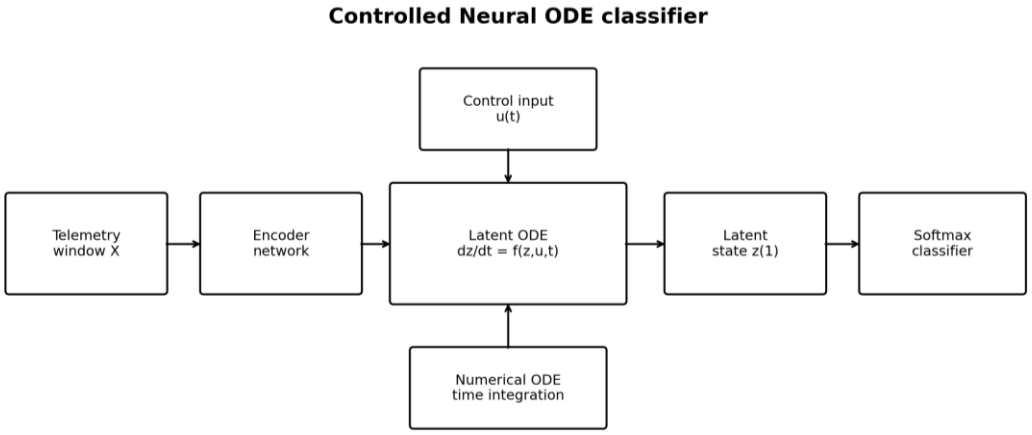


Fig. 2. Controlled Neural ODE architecture used to classify temporal telemetry windows.

$$z_0 = E_\varphi(x_t) \tag{10}$$
$$\frac{dz_t}{dt} = f_\theta(z_t, u_t), \quad 0 \le t \le 1$$
$$z_1 = z_0 + \int_0^1 f_\theta(z_t, u_t)\, dt$$
$$\hat{y} = g_\psi(z_1)$$

E. GRU Baseline

The GRU baseline, shown in Fig. 3, processes the same telemetry window sample by sample. Its reset gate r_t controls how much past information participates in the candidate state, and its update gate z_t controls the interpolation between the candidate state and the previous hidden state h_{t-1} [12]. In Eq. (11), W_r, W_z, W_h, and W_o are the weight matrices of the reset, update, candidate, and output layers, b_r, b_z, b_h, and b_o are the corresponding bias vectors, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes the element-wise product.

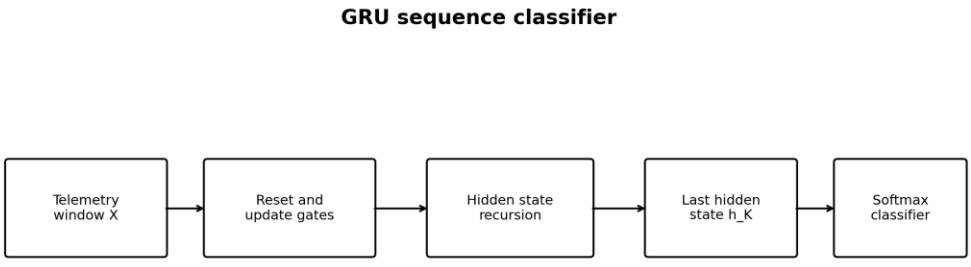


Fig. 3. GRU architecture used as the recurrent baseline for the same telemetry windows.

$$
\begin{aligned}
r_t &= \sigma(W_r[h_{t-1};\ x_t] + b_r) \\
z_t &= \sigma(W_z[h_{t-1};\ x_t] + b_z) \\
\tilde{h}_t &= \tanh(W_h[r_t \odot h_{t-1};\ x_t] + b_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \\
\hat{y} &= W_o h_L + b_o
\end{aligned}
\tag{11}
$$

F. Loss Function and Evaluation Metrics

Both classifiers are trained with the same weighted cross-entropy loss, given in Eq. (12). In Eq. (12), B is the mini-batch size, K is the number of classes, $y_{\{b,k\}}$ is the one-hot target of sample b for class k, $\dot{Z}_{\{b,k\}}$ is the predicted class probability, w_k is the class weight, and ε is a small positive constant that prevents evaluating the logarithm at zero. The inverse-frequency class weights compensate for class imbalance while preserving the same objective for both networks [14], [15].

$$
L(\theta) = -\frac{1}{B}\sum_{b=1}^{B}\sum_{k=1}^{K} w_k y_{b,k}\, log\left(\widehat{p_{b,k}} + \varepsilon\right) \tag{12}
$$

The evaluation uses accuracy, balanced accuracy, macro precision, macro recall, macro specificity, macro F1, weighted F1, multiclass Matthews correlation coefficient, ROC-AUC, PR-AUC, Brier score, and detection delay. Class precision P_k, class recall R_k, class F1 $F1_k$, and the Brier score $Brier_k$ are standard evaluation definitions [16], [17], [18]; they are given in Eq. (13) for clarity.

$$
\begin{aligned}
P_k &= \frac{TP_k}{TP_k + FP_k}, \quad R_k = \frac{TP_k}{TP_k + FN_k} \\
F1_k &= \frac{2P_k R_k}{P_k + R_k}, \quad Brier_k = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{p_{b,k}} - y_{b,k}\right)^2
\end{aligned}
\tag{13}
$$

III. METHODOLOGY

A numerical simulation was carried out to generate all training, validation, and test sequences. The simulation consisted of closed-loop quadrotor flights following smooth references, with random initial conditions and randomly scheduled faults. The learning models did not receive the internal fault variables directly; instead, they received noisy telemetry windows generated through the measurement model of Eq. (9). This design forces the Neural ODE and GRU classifiers to infer degradation from temporal patterns rather than from explicit fault labels in the input.

TABLE 2
NUMERICAL SIMULATIONS AND TRAINING CONFIGURATION

Item	Value
Final simulated time	16 s
Sampling period	0.04 s (25 Hz)
Simulated trajectories	240
Fault trajectory ratio	0.75
Fault types	Battery degradation; motor-effectiveness loss; overheating
Fault onset range	0.14 to 0.43 of the flight duration
Fault ramp range	0.30 to 0.62 of the flight duration
Window length and stride	80 samples; 10 samples
Train/validation/test split	0.70 / 0.15 / 0.15 by trajectory
Classes	Healthy; BatteryFault; MotorFault; OverheatFault
Test Window	1320
Evaluated fault trajectories	31 (zero missed)
Training epochs	14
Mini-batch size	96
Neural ODE latent dimension	40
GRU hidden units	80

The preprocessing pipeline normalizes each telemetry channel using training-set statistics only, constructs overlapping windows, assigns a window label based on the active fault severity threshold, and trains both classifiers using identical mini-batches and class weights. The decision rule selects the largest posterior probability for multiclass diagnosis and aggregates all nonhealthy probabilities for any-fault detection.

The flight sample shown in Fig. 4 illustrates the gradual nature of the simulated faults. The visible interval between onset and strong degradation is intentional because the target problem is incipient fault detection over prolonged operating periods.

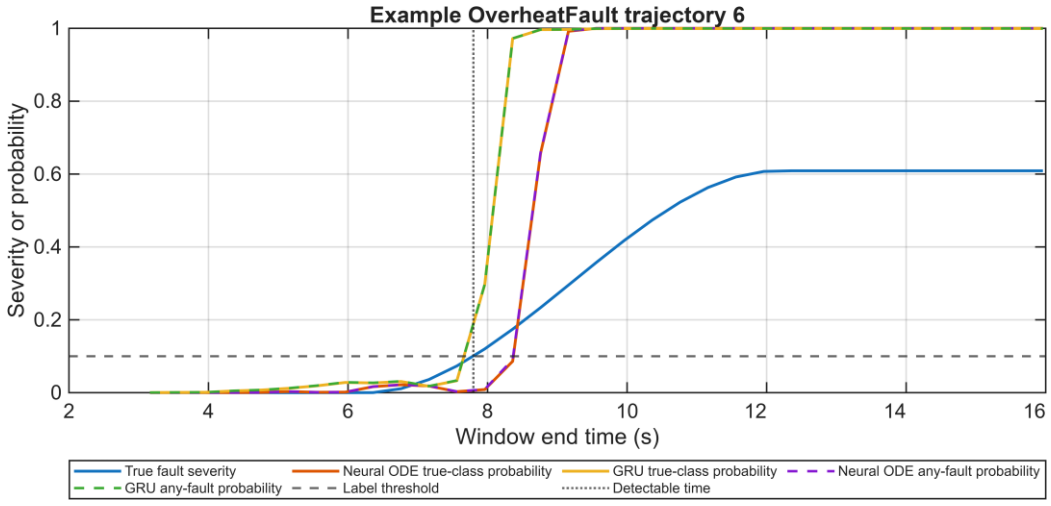


Fig. 4. Example simulated trajectory with battery, motor, and overheating variables under random fault scheduling.

IV. NUMERICAL RESULTS

A. Training and Validation Behavior

The training curves in Fig. 5 show stable convergence for both models, while Fig. 6 shows the evolution of the validation macro F1. These plots confirm that the comparison is based not only on a terminal metric but also on consistent behavior during optimization.

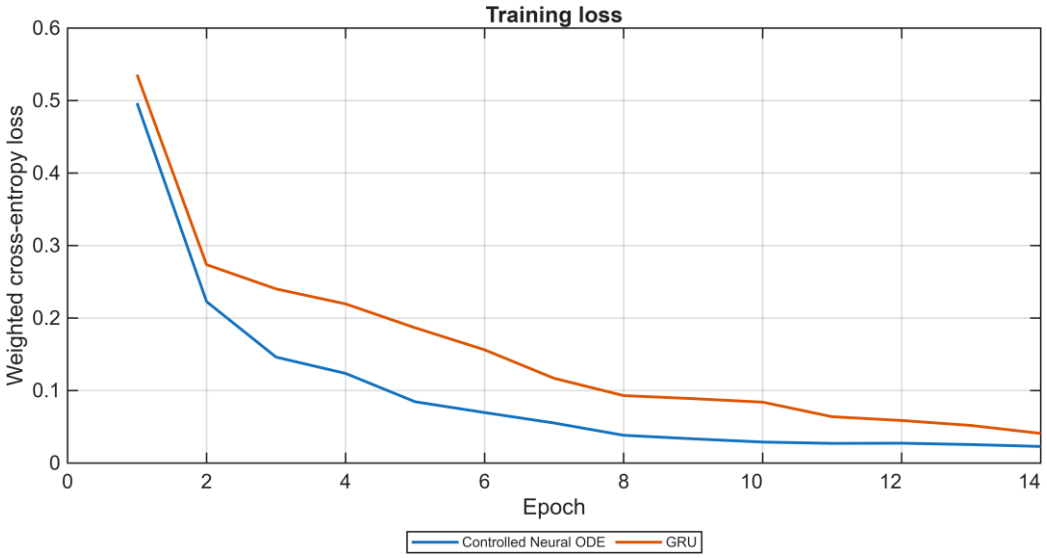


Fig. 5. Training loss curves for the Controlled Neural ODE and GRU classifiers.

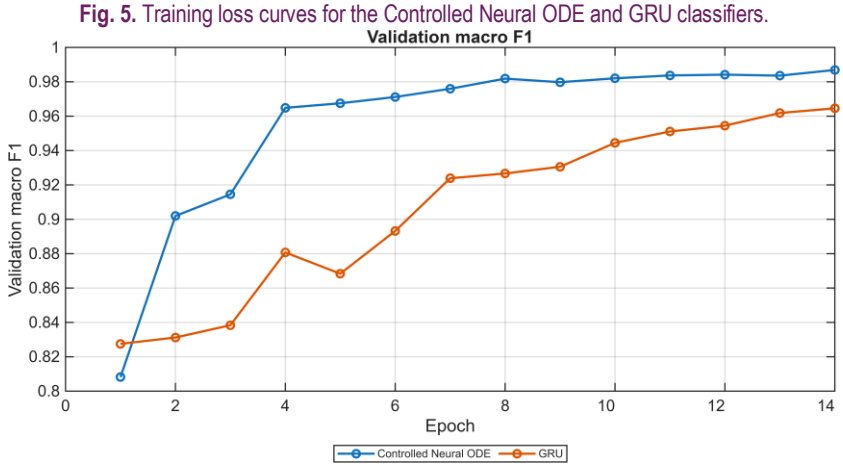


Fig. 6. Validation macro F1 obtained during training for both classifiers.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, MÉXICO
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

B. Test-Set Classification Performance

TABLE 3
SUMMARY OF TEST-SET METRICS

Model	Accuracy	Macro F1	Weighted F1	MCC	Any-fault F1	Any-fault ROC-AUC	Mean delay (s)	Missed faults
Neural ODE	0.9818	0.9828	0.9818	0.9743	0.9846	0.9976	0.401	0
GRU	0.9667	0.9702	0.9665	0.9528	0.9705	0.9918	0.374	0

Table 3 indicates that both models achieve high multiclass accuracy and any-fault detection performance on the test windows, with zero missed fault trajectories among the 31 fault trajectories evaluated. The Controlled Neural ODE provides the highest accuracy (0.9818), macro F1 (0.9828), weighted F1 (0.9818), and multiclass MCC (0.9743), as well as the highest any-fault ROC-AUC (0.9976); the GRU attains a slightly shorter mean detection delay (0.374 s against 0.401 s). Fig. 7 summarizes the main metrics graphically.

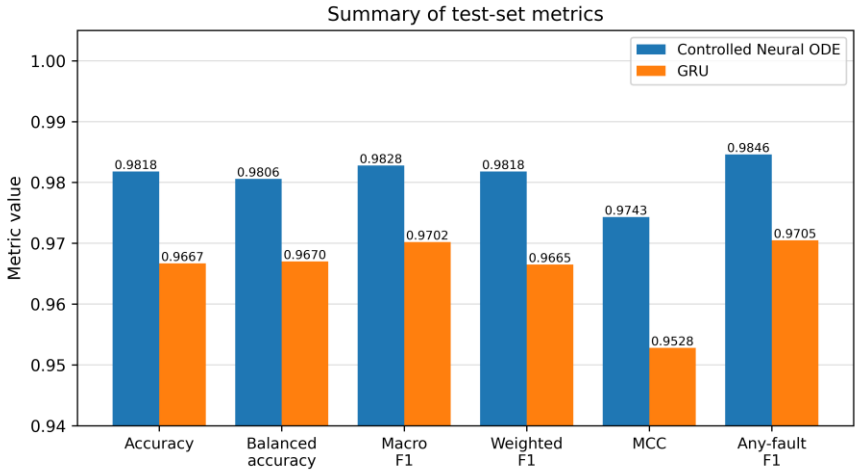


Fig. 7. Summary metric comparison between the Controlled Neural ODE and GRU classifiers.

TABLE 4
PER-CLASS F1 SCORES

Model	Healthy	Battery	Motor	Overheat
Neural ODE	0.9799	0.9909	0.9819	0.9786
GRU	0.9617	0.9977	0.9899	0.9314

Table 4 and Fig. 10 show the per-class F1 scores. In this configuration, the GRU achieves the highest per-class F1 for the battery (0.9977) and motor (0.9899) faults, while the Controlled Neural ODE outperforms the GRU for the healthy (0.9799 against 0.9617) and overheat (0.9786 against 0.9314) classes. The higher sensitivity of the GRU to the electrical signature of the battery fault, together with its response to the asymmetric-thrust signature of the motor fault, explains its per-class advantage in these two classes; the Neural ODE, in turn, yields a more balanced multiclass behavior, which is reflected in its superior macro F1 and MCC.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, MÉXICO
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

The confusion matrices in Fig. 8 and Fig. 9 show that most errors occur near the class boundaries rather than uniformly across all fault types. This behavior is expected because windows near the severity threshold contain transitional signatures.

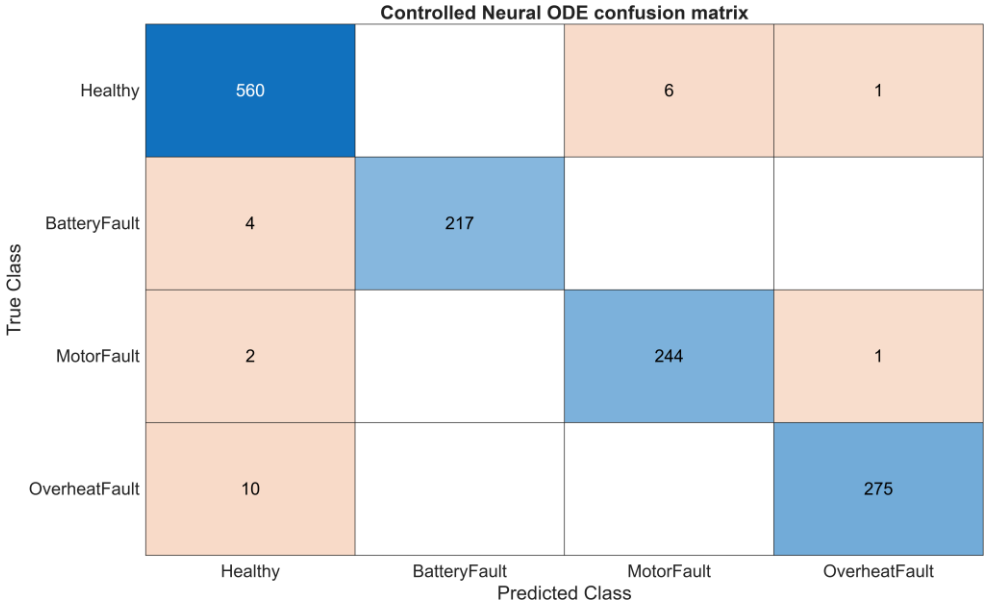


Fig. 8. Multiclass confusion matrix for the Controlled Neural ODE classifier.

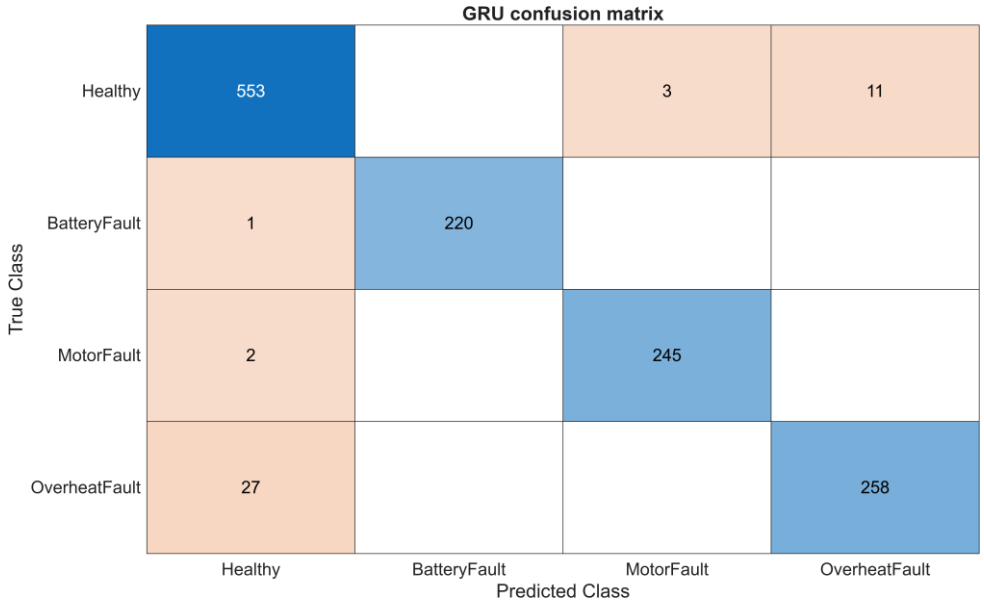


Fig. 9. Multiclass confusion matrix for the GRU classifier.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

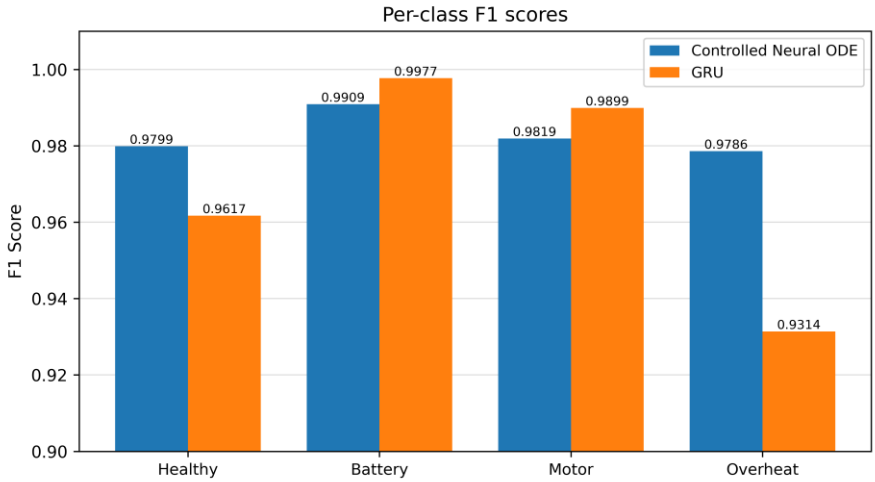


Fig. 10. Per-class F1 comparison for the healthy, battery, motor, and overheating classes.

C. Any-Fault Detection

Finally, the any-fault ROC curves in Fig. 11 evaluate binary discrimination between healthy and faulty operation after aggregating the three nonhealthy classes. This view complements the multiclass tables by evaluating whether the models detect anomalous behavior before assigning a specific fault category. The any-fault ROC-AUC is 0.9976 for the Neural ODE and 0.9918 for the GRU, and both models detect all 31 fault trajectories in the test set.

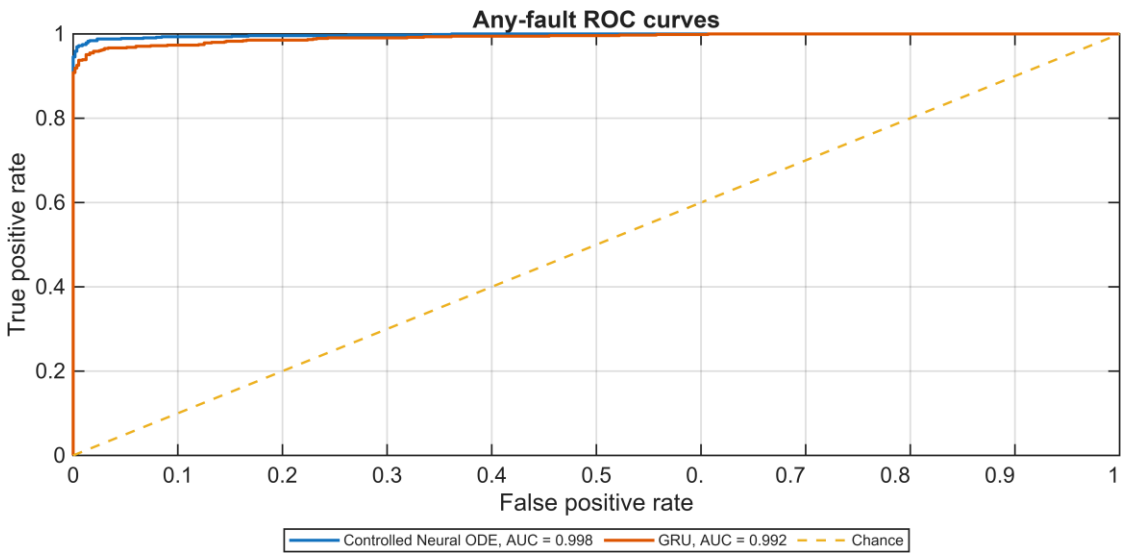


Fig. 11. Any-fault ROC curves obtained by aggregating battery, motor, and overheating faults into a single nonhealthy class.

D. State Evolution and Fault-Mechanism Analysis

To make explicit the physical mechanisms that underlie the classification results, this subsection presents the temporal evolution of the drone states and of the fault variables for one representative test trajectory of each fault class: test trajectory 20 for battery degradation, test trajectory 21 for loss of motor effectiveness, and test trajectory 6 for overheating. Table 5 summarizes the onset time, detectable time, maximum severity, and extreme values of the fault variables for these trajectories.

TABLE 5
PER-CLASS F1 SCORES

Fault class	Trajectory	Onset (s)	Detectable (s)	Max severity	Min SoC	Min V_b (V)	Max I_b (A)	Faulty motor	Min effectiveness	Max T_m (°C)
Battery Fault	20	2.41	5.08	0.491	0.823	10.68	172.6	—	—	—
Motor Fault	21	3.41	5.88	0.614	0.838	10.68	186.7	3	0.093	40.7
Overheat Fault	6	6.29	7.80	0.609	0.842	10.68	138.2	4	—	60.5

Fig. 12 provides a global overview: for each fault class, it shows the evolution of the position and the any-fault detection probability obtained with the Neural ODE and the GRU, together with the fault onset (dotted line) and the detectable time (dashed line). In all three cases, the detection probability crosses the decision threshold within one to two seconds after the fault becomes active, which is consistent with the mean detection delays reported in Table 3.

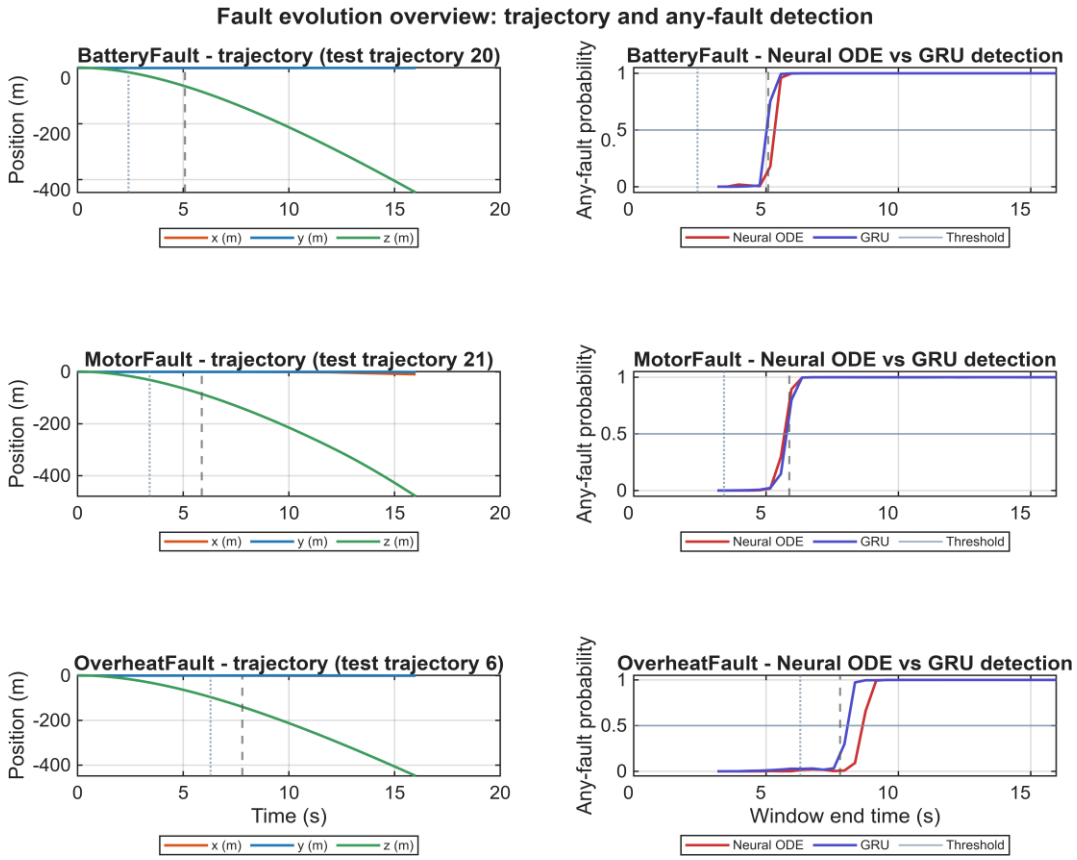


Fig. 12. Fault evolution overview: position of the quadrotor and any-fault detection probability of the Neural ODE and the GRU under battery degradation, motor-effectiveness loss, and overheating. The dotted and dashed lines mark the fault onset and the detectable time, respectively.

For the battery-fault trajectory, the degradation begins at $t_{on} = 2.41\ s$ and the any-fault probability exceeds the decision threshold at 5.08 s (Figs. 13–17). The loss of effective capacity and the increase of the internal resistance force the battery to supply up to 172.6 A in order to maintain the commanded thrust; consequently, the terminal voltage reaches a minimum of 10.68 V while the state of charge decreases to 0.823 (Fig. 15). The resulting electrical stress is the dominant signature of this fault class and explains the very high per-class F1 obtained by both models (0.9909 for the Neural ODE and 0.9977 for the GRU). The mechanical states (Figs. 13 and 14) show a slow drift in altitude and a mild attitude perturbation, consistent with the battery fault being an energy-source fault rather than an actuator fault.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

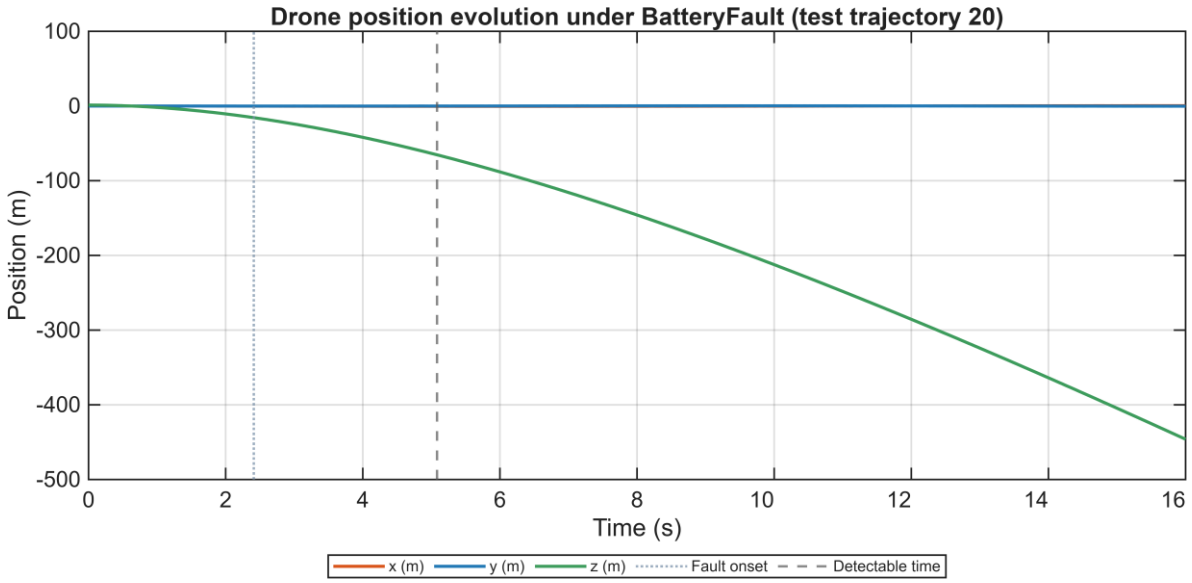


Fig. 13. Position evolution of the quadrotor under battery degradation (test trajectory 20)

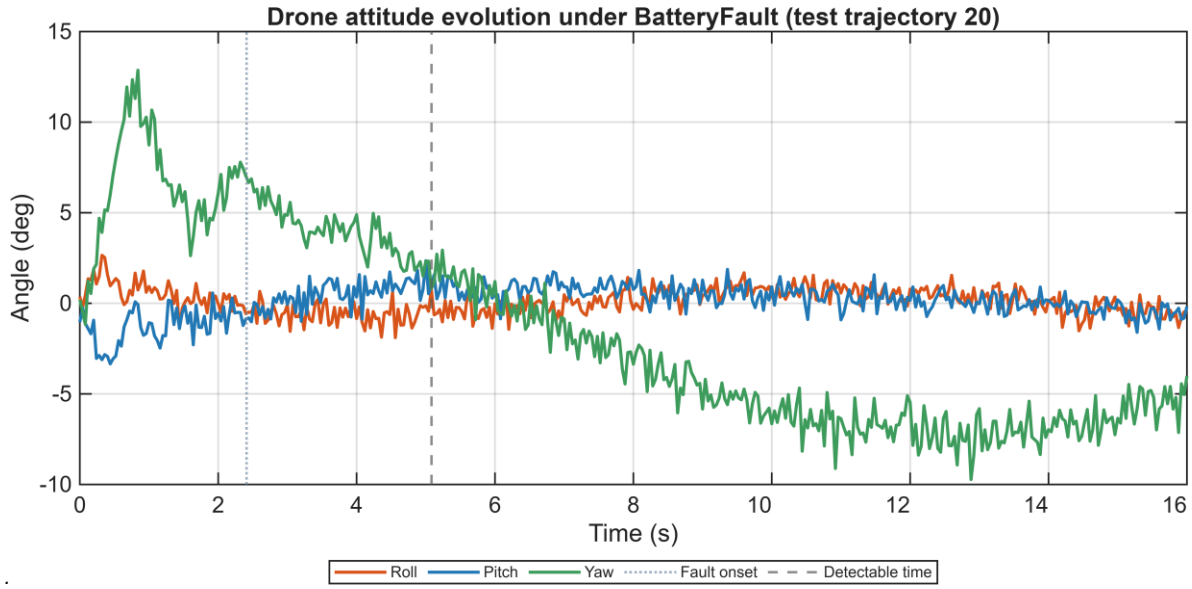


Fig. 14. Attitude evolution of the quadrotor under battery degradation (test trajectory 20).

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

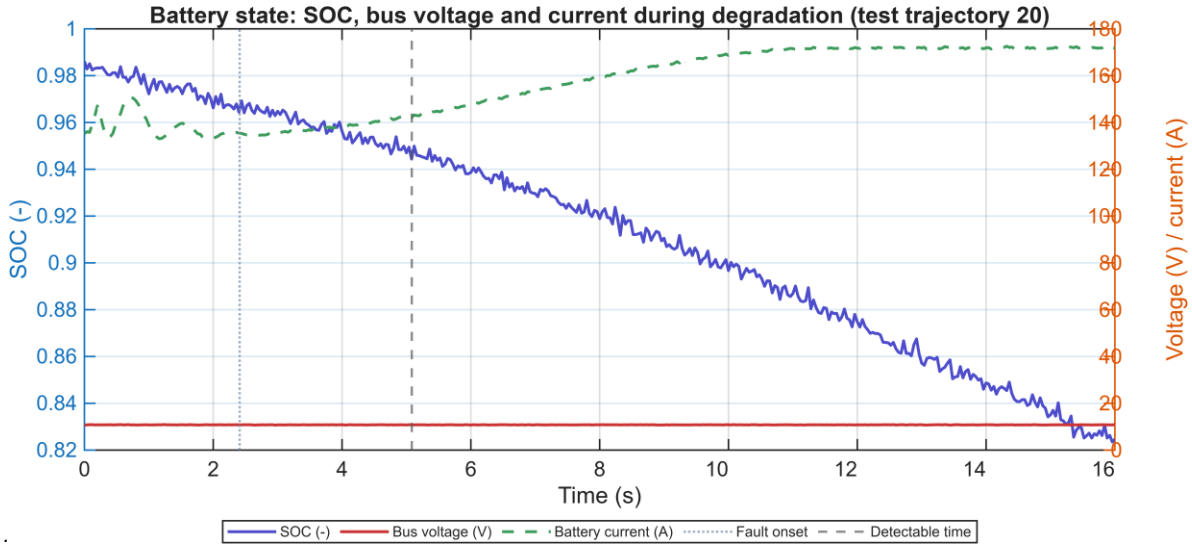


Fig. 15. Battery state of charge and terminal voltage during battery degradation (test trajectory 20).

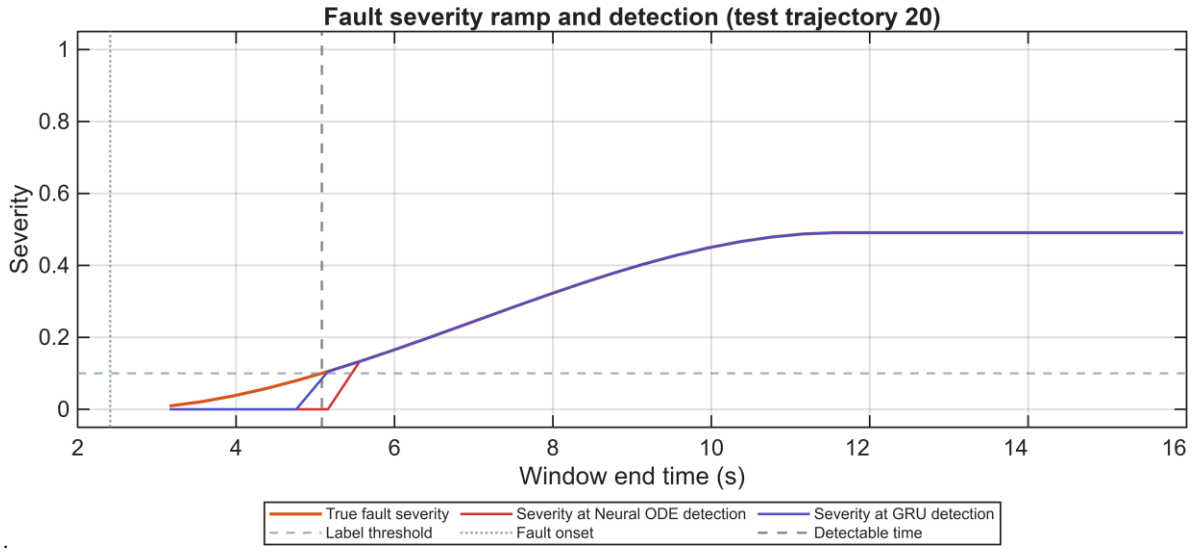


Fig. 16. Fault-severity evolution for the battery-degradation case (test trajectory 20).

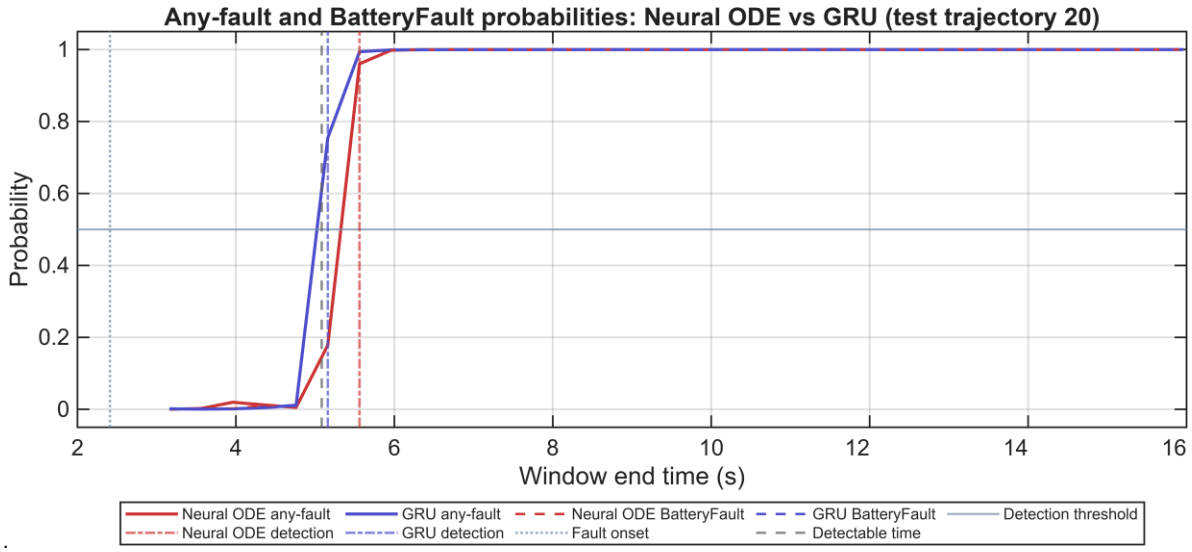


Fig. 17. Any-fault detection probability under battery degradation for the Neural ODE and the GRU (test trajectory 20).

For the motor-fault trajectory, the loss of effectiveness of motor 3 begins at 3.41 s and becomes detectable at 5.88 s (Figs. 18–22). The effective coefficient of the faulty rotor decreases to 0.093 (Fig. 20), which produces an asymmetric thrust distribution and a characteristic attitude deviation (Fig. 19) together with a lateral drift of the position (Fig. 18). The healthy motors must compensate the lost thrust, and the required bus current reaches 186.7 A; the associated electrical heating also explains the moderate temperature increase (40.7 °C) even though no overheating fault is scheduled for this trajectory.

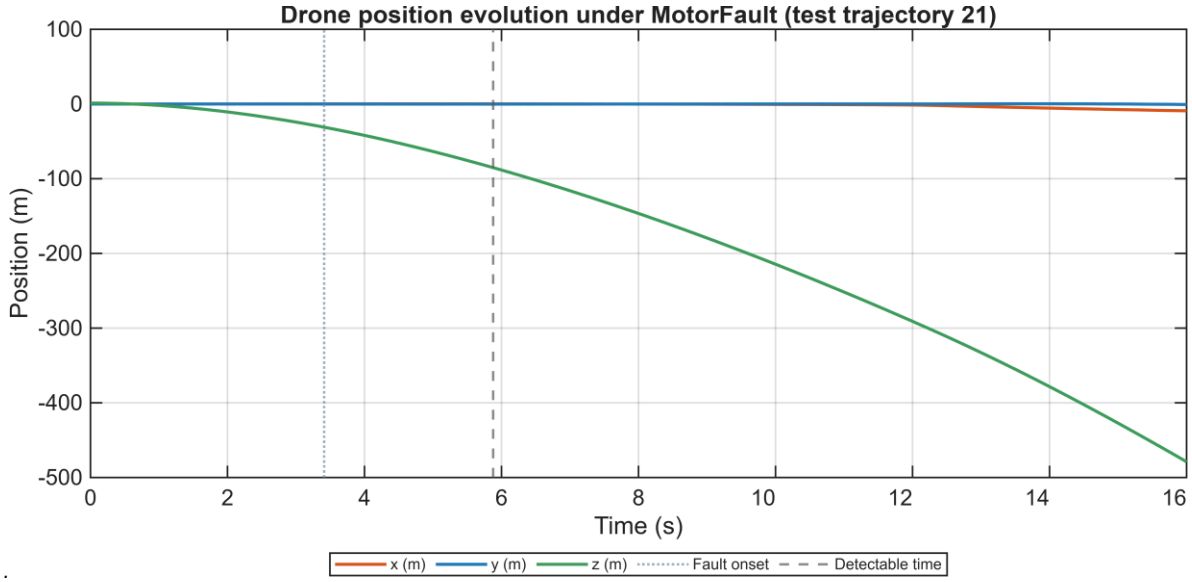


Fig. 18. Position evolution of the quadrotor under motor-effectiveness loss (test trajectory 21).

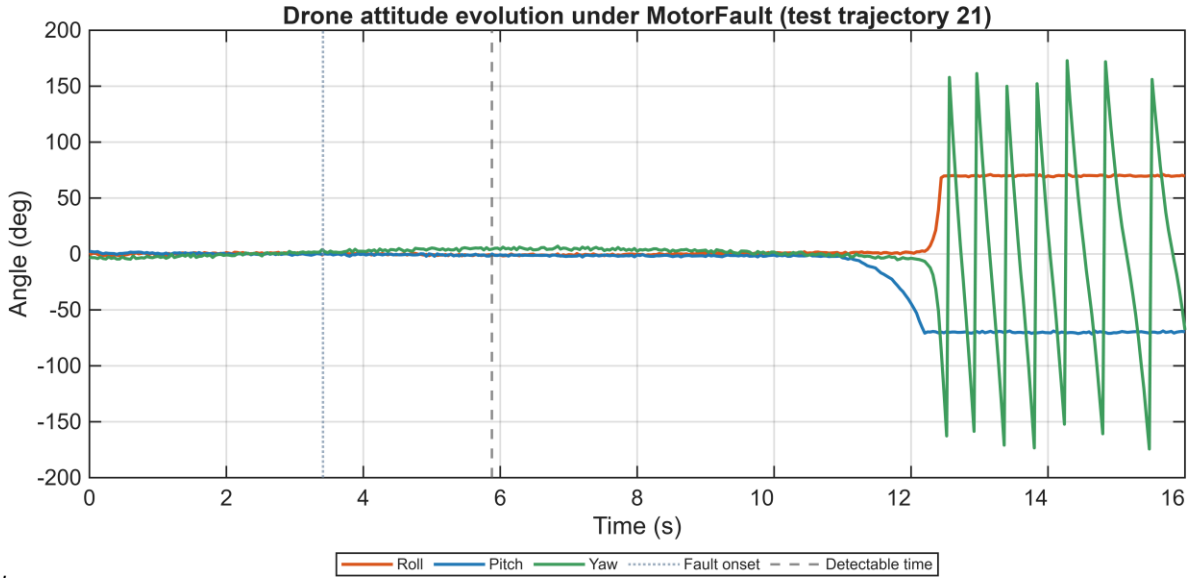


Fig. 19. Attitude evolution of the quadrotor under motor-effectiveness loss (test trajectory 21).

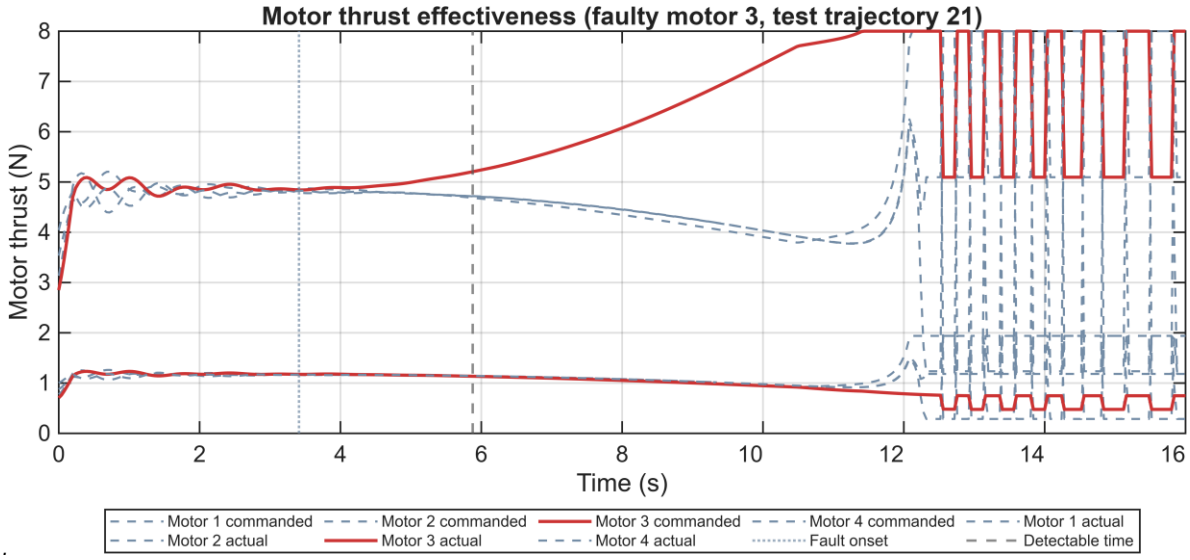


Fig. 20. Motor effectiveness of the four rotors during the actuator fault (test trajectory 21, motor 3).

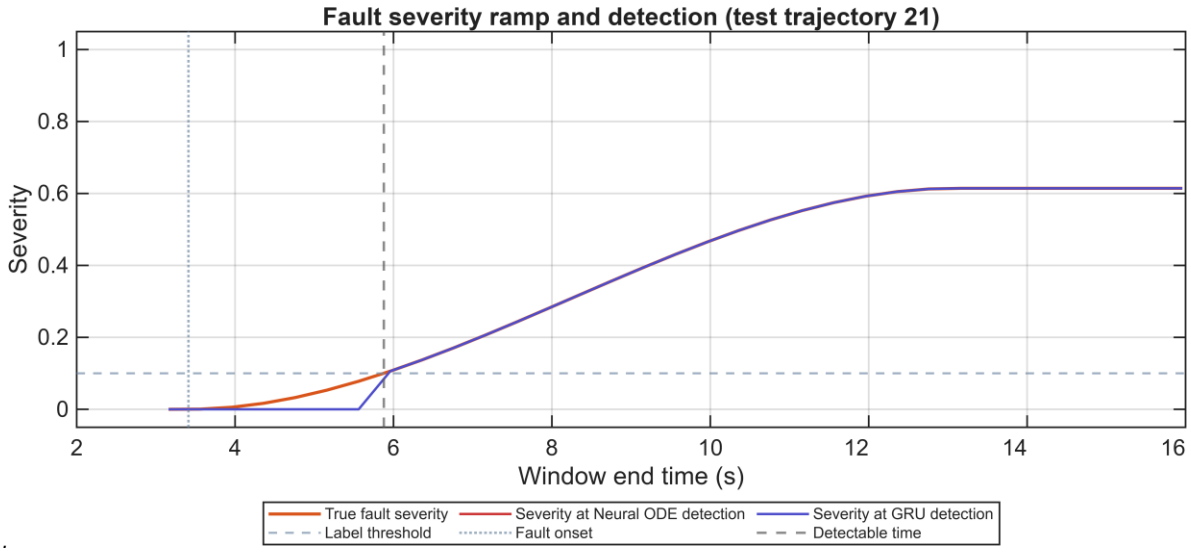


Fig. 21. Fault-severity evolution for the motor-effectiveness loss case (test trajectory 21).

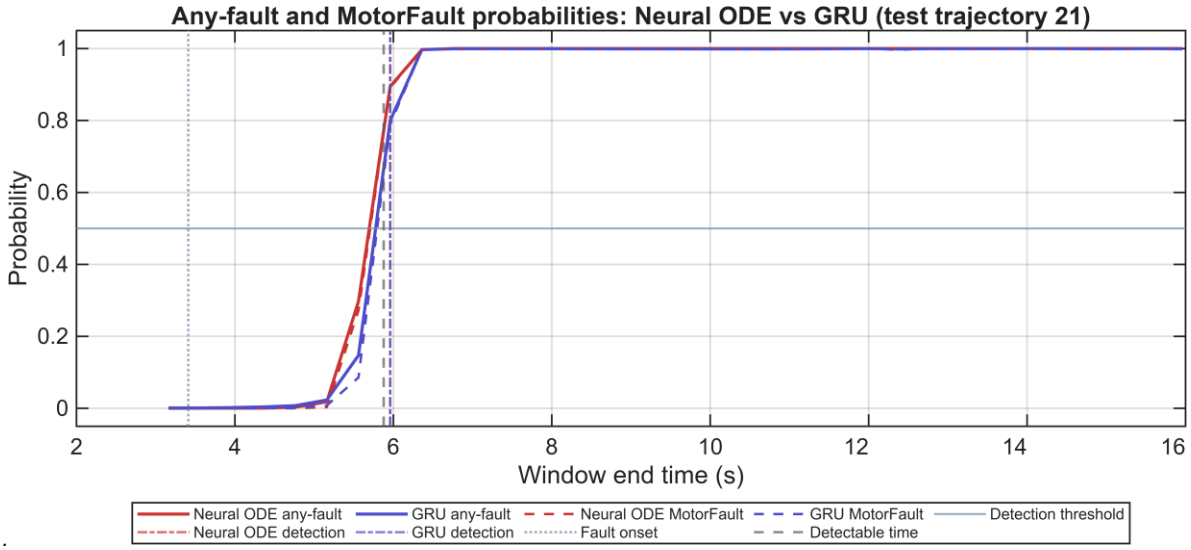


Fig. 22. Any-fault detection probability under motor-effectiveness loss for the Neural ODE and the GRU (test trajectory 21).

For the overheating trajectory, the motor 4 fault begins at 6.29 s and is detected at 7.80 s (Figs. 23–27). The reduction in the local cooling coefficient, together with the injected heat, raises motor 4's temperature to 60.5 $°C$ (Fig. 25), which in turn activates the thermal derating factor and reduces the effective thrust of the affected rotor. Because the thermal dynamics are slower than the electrical ones, the interval between onset and detection is the largest of the three classes (1.51 s), which is consistent with the slightly lower per-class F1 of the overheat fault (0.9786 for the Neural ODE and 0.9314 for the GRU).

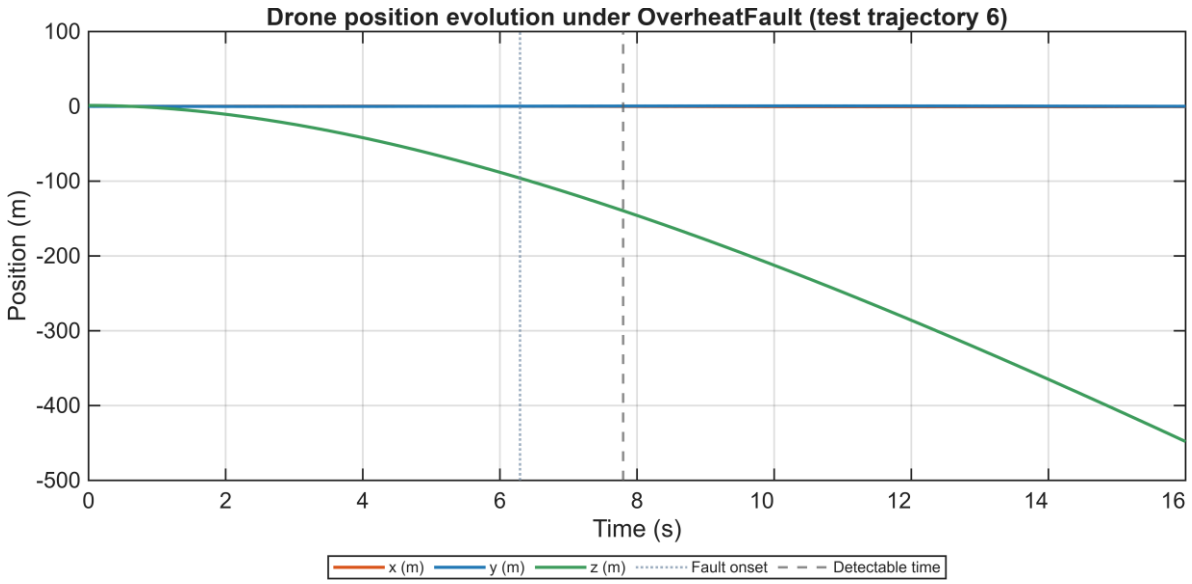


Fig. 23. Position evolution of the quadrotor under overheating (test trajectory 6).

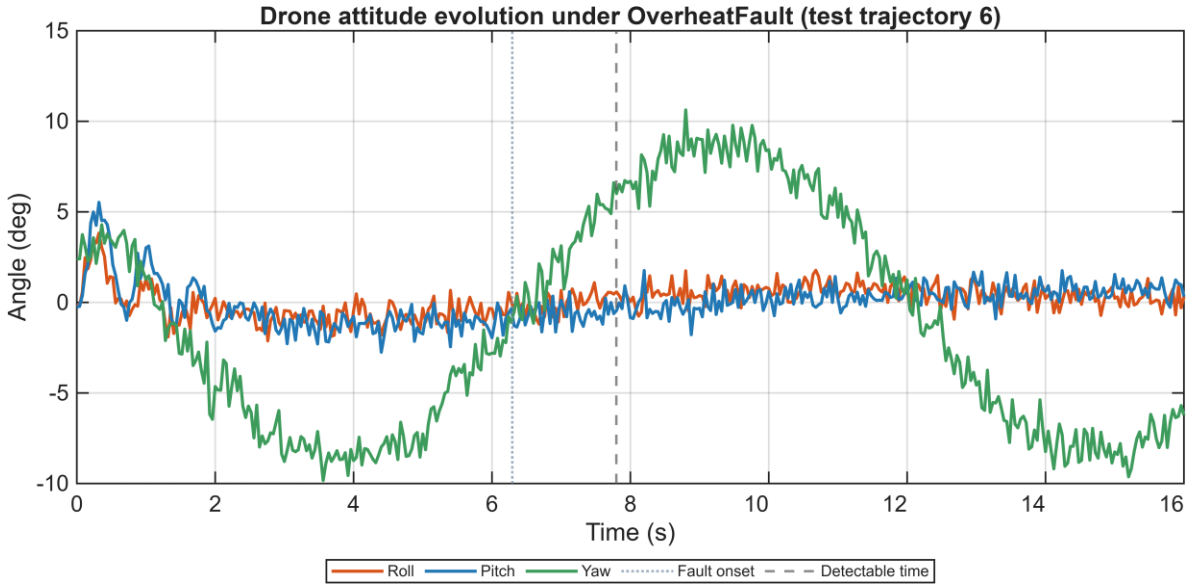


Fig. 24. Attitude evolution of the quadrotor under overheating (test trajectory 6).

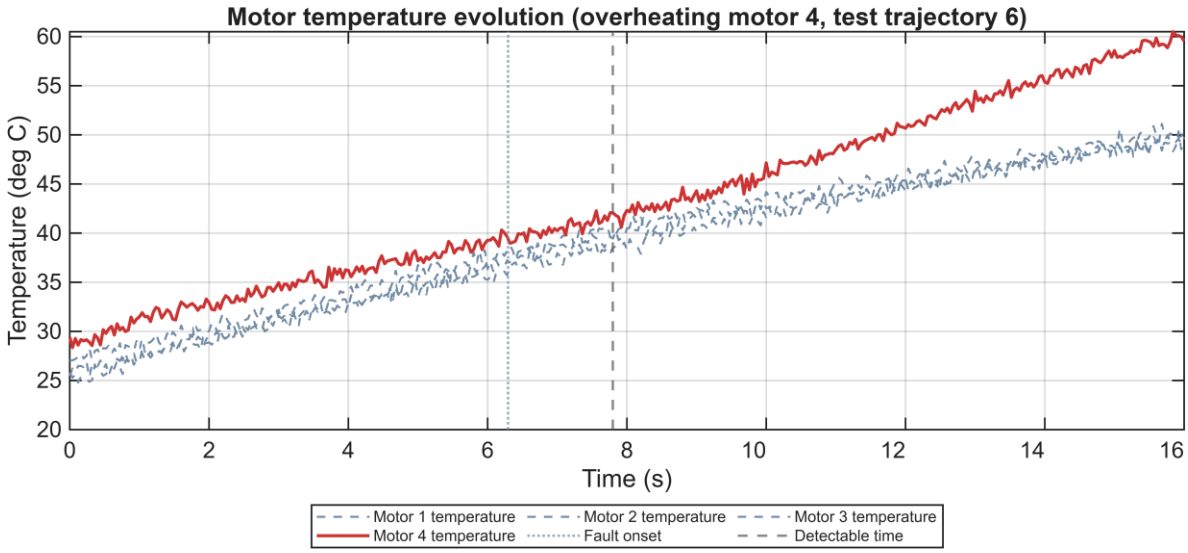


Fig. 25. Motor temperature during the overheating fault (test trajectory 6, motor 4).

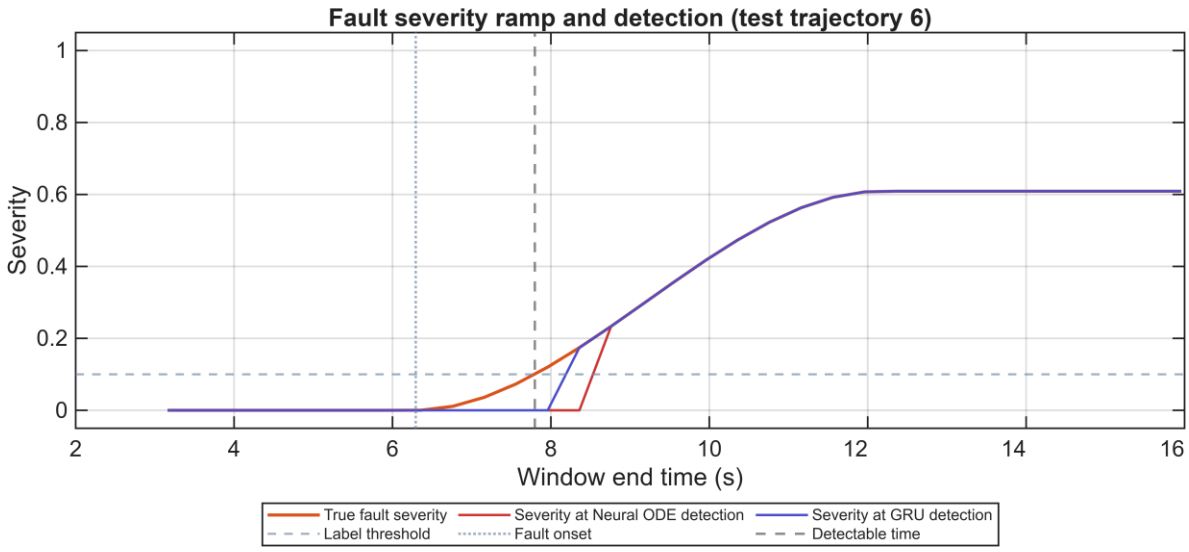


Fig. 26. Fault-severity evolution for the overheating case (test trajectory 6).

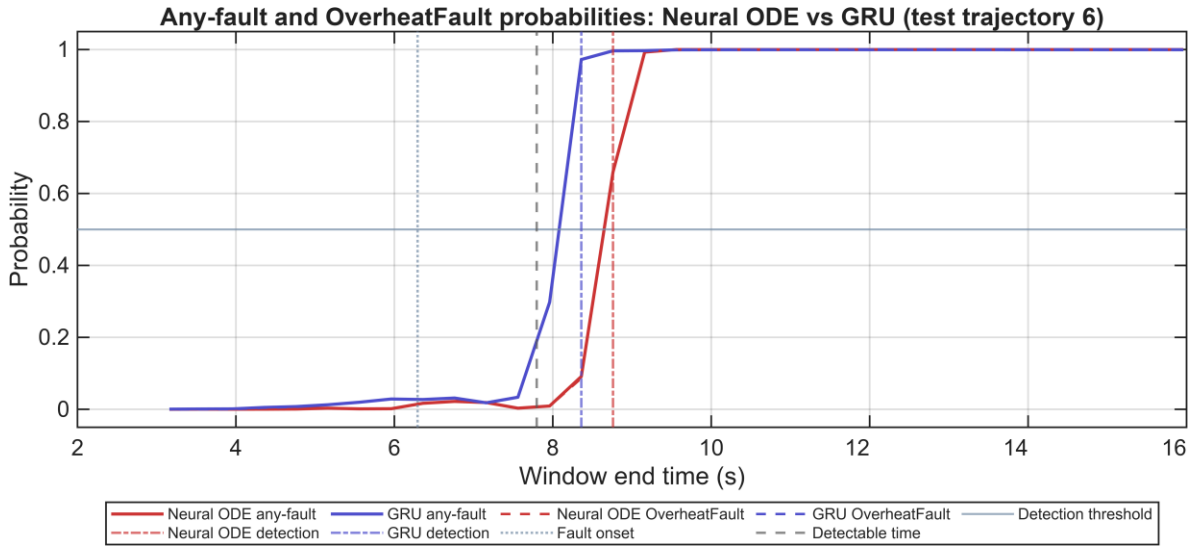


Fig. 27. Any-fault detection probability under overheating for the Neural ODE and the GRU (test trajectory 6).

E. Extension to Structural and Mechanical Faults

Although the present study focuses on progressive (gradually evolving) faults, the proposed framework can be extended to structural and mechanical fault scenarios such as propeller damage, rotor imbalance, and hard motor failure. The key observation is that, in the present formulation, the effect of a mechanical fault ultimately manifests as a change in the thrust produced by one or more rotors and in the energy balance of the electrical and thermal subsystems. Consequently, the same telemetry-driven learning architecture can be applied, provided that the fault signature is observable in the measured states.

Propeller damage (for example, blade erosion or tip fracture) reduces the effective thrust coefficient of the affected rotor and typically increases the torque required for a given thrust. In the present model, this behavior is analogous to the loss of motor effectiveness captured by the coefficient $c_i(t)$ in Eq. (7): a damaged propeller behaves as a partially effective actuator, and its incipient degradation produces a signature qualitatively similar to that of the motor fault. The distinguishing feature is the presence of an additional parasitic torque on the affected axes and, when the damage is severe, an imbalance between the opposing rotors that excites roll–yaw oscillations. Because the Neural ODE and the GRU operate directly on the full telemetry window, such oscillatory signatures are in principle capturable without modifying the architecture. A targeted extension would generate propeller-damage trajectories in the simulator by replacing a rotor's constant thrust coefficient with a slowly decreasing or periodically modulated coefficient, and adding the corresponding parasitic torque to the moment vector M_B in Eq. (3).

Rotor imbalance, on the other hand, produces a characteristic harmonic excitation at the shaft frequency (1×) and its harmonics (2×), which appear in the attitude channels as narrowband oscillations and, in a physical platform, in the vibration signals. Because the present simulator operates at a sampling rate of 25 Hz, it cannot resolve these high-frequency components; extending the framework to rotor imbalance therefore requires a higher-rate simulation of the aerodynamic forces of the rotating system and frequency-domain features, such as the spectral power in the 1× and 2×

bands, as additional input channels. The continuous latent dynamics of the Neural ODE is well suited to this purpose because it can be conditioned on such band-limited features without modifying the classification objective.

A hard motor failure (sudden total loss of thrust) lies outside the scope of progressive degradation, since the severity variable $s_k(t)$ in Eq. (6) assumes a bounded ramp between zero and one. Nevertheless, the any-fault aggregation used in Section IV-C can be directly reused for a binary healthy-versus-failure decision: the resulting anomaly probability would typically cross the threshold within the first one or two windows after the failure, as the thrust deficit produces an immediate attitude deviation. A natural extension is to couple the detector with a reconfiguration layer that re-solves the thrust allocation of Eq. (3) over the remaining rotors, which is a standard fault-tolerant control problem [4], [5]. In summary, the learning layer of the proposed framework is independent of the fault model: extending it to structural and mechanical faults requires faithful simulation of the corresponding signatures and, for high-frequency phenomena, the acquisition or simulation of additional channels at an adequate sampling rate.

V. Conclusions and Future Work

This work presented a physics-based numerical study of multiclass fault detection in quadrotor drones using a controlled Neural ODE and a GRU baseline. The simulator included rigid-body flight dynamics, a battery electrical model, motor thermal dynamics, and three randomly scheduled progressive fault mechanisms. The mathematical model was written in compact, numbered equations that define the system, the fault mechanisms, the learning architectures, and the loss function; a complete table of symbols was included, and the force-allocation and battery-electrical models were separated into distinct numbered equations. Standard metric definitions were provided without numbering to maintain a concise reading flow.

The numerical results show that both architectures can identify battery degradation, motor-effectiveness loss, overheating, and healthy operation from noisy telemetry windows, with zero missed fault trajectories in the test set. The Neural ODE obtained the strongest overall multiclass indicators in the simulated test set, while the GRU achieved the highest per-class F1 for the battery and motor faults and produced a slightly shorter mean detection delay. The detailed analysis of the state evolution and of the fault mechanisms confirmed that the electrical signature dominates the battery fault, the asymmetric thrust distribution dominates the motor fault, and the thermal dynamics govern the overheat fault, and that the proposed framework can be extended to structural and mechanical fault scenarios such as propeller damage, rotor imbalance, and hard motor failure without modifying the learning architecture.

Future work should validate the proposed methodology using flight-log data, consider simultaneous multi-fault scenarios, account for wind and actuator-saturation uncertainty, incorporate the structural and mechanical fault mechanisms discussed in Section IV-E, compare fixed-step and adaptive ODE solvers under identical accuracy constraints, and assess deployment on embedded hardware with strict memory and latency limits.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04

CRediT (*Contributor Roles Taxonomy*)

Author's contributions: Conceptualization: C**SC**; Methodology: C**SC**; Software: C**SC**; Investigation: C**SC**; Writing - original draft: C**SC**; Writing - review and editing: J**MM**, C**MV**, J**SM**; Supervision: J**MM**; Formal analysis: J**MM**, J**SM**; Project administration: C**MV**.

Funding: This research was supported by the Instituto Politécnico Nacional (IPN) through the Secretaría de Investigación y Posgrado (SIP) under Grant Nos. IND-2026-0112, IND-2026-0228, and Multti-2026-0035-M3.

Data Availability Statement: All numerical results, figures, configuration tables, and source files required to reproduce the results presented in this manuscript are publicly available in the supplementary repository at: https://la.mathworks.com/matlabcentral/fileexchange/184063-neural-ode-multiclass-fault-detection-in-quadrotor-drones

Acknowledgements: The author gratefully acknowledges the institutional support provided by the Instituto Politécnico Nacional (IPN) and the recognition and support received from the National System of Researchers and Researchers (Sistema Nacional de Investigadoras e Investigadores, SNII) of Mexico.

Conflict of interest: The authors declare that there is no conflict of interest.

REFERENCES

[1] G. K. Fourlas, G. C. Karras, "A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles," *Machines*, vol. 9, no. 9, art. no. 197, Sep. 2021, doi: https://doi.org/10.3390/machines9090197.

[2] S. Yin, S. X. Ding, X. Xie, H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418-6428, Nov. 2014, doi: https://doi.org/10.1109/TIE.2014.2301773.

[3] R. Isermann, "Model-based fault-detection and diagnosis - status and applications," *Annual Reviews in Control*, vol. 29, no. 1, pp. 71-85, 2005, doi: https://doi.org/10.1016/j.arcontrol.2004.12.002.

[4] N. P. Nguyen, 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, Mar. 2019, doi: https://doi.org/10.3390/en12061139.

[5] P. Pounds, R. Mahony, P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, no. 7, pp. 691-699, Jul. 2010, doi: https://doi.org/10.1016/j.conengprac.2010.02.008.

[6] G. M. Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," in *AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head*, SC, USA, Aug. 2007, doi: https://doi.org/10.2514/6.2007-6461.

[7] D. Mellinger, V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 2520-2525, doi: https://doi.org/10.1109/ICRA.2011.5980409.

[8] O. Tremblay, L.-A. Dessaint, "Experimental validation of a battery dynamic model for EV applications," *World Electric Vehicle Journal*, vol. 3, no. 2, pp. 289-298, 2009, doi: https://doi.org/10.3390/wevj3020289.

[9] C. Forgez, D. Vinh Do, G. Friedrich, M. Morcrette, C. Delacourt, "Thermal modeling of a cylindrical LiFePO4/graphite lithium-ion battery," *Journal of Power Sources*, vol. 195, no. 9, pp. 2961-2968, May 2010, doi: https://doi.org/10.1016/j.jpowsour.2009.10.105.

[10] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 6572-6583, doi: https://doi.org/10.48550/arXiv.1806.07366.

[11] P. Kidger, J. Morrill, J. Foster, T. Lyons, "Neural controlled differential equations for irregular time series," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6696-6707, doi: https://doi.org/10.48550/arXiv.2005.08926.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, Oct. 2014, pp. 1724-1734, doi: https://doi.org/10.3115/v1/D14-1179.

[13] C. Solis, J. Morales, C. Montelongo, S. Palomino, "Transformer- and GRU-Based Identification of Open-Chain Robot Kinematics Using Product-of-Exponentials Coordinates," *Technologies*, vol. 14, no. 6, p. 333, 2026, doi: https://doi.org/10.3390/technologies14060333.

[14] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, CA, USA, 2015, doi: https://doi.org/10.48550/arXiv.1412.6980.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, Feb. 2020, doi: https://doi.org/10.1109/TPAMI.2018.2858826.

[16] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, Jun. 2006, doi: https://doi.org/10.1016/j.patrec.2005.10.010.

[17] T. Saito, M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, art. no. e0118432, Mar. 2015, doi: https://doi.org/10.1371/journal.pone.0118432.

[18] D. Chicco, G. Jurman, "The advantages of the Matthews correlation coefficient over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, art. no. 6, Jan. 2020, doi: https://doi.org/10.1186/s12864-019-6413-7.

Científica, vol. 30, no. 2, pp. 01-29, July-December 2026,
ISSN 2594-2921, Instituto Politécnico Nacional, México
e300204 | DOI: 10.46842/ipn.cien.v30n2a04