

Article

Design and Control of a Reconfigurable Robot with Rolling and Flying Locomotion

Qing Chang ^{1,*}, Biao Yu ¹, Hongwei Ji ¹, Haifeng Li ², Tiantian Yuan ¹, Xiangyun Zhao ¹, Hongsheng Ren ² and Jinhao Zhan ¹

¹ School of Mechanical Engineering, Tianjin University of Commerce, No. 409 Guangrong Road, Beichen District, Tianjin 300134, China; yubiao@stumail.tjcu.edu.cn (B.Y.); jhwei@tjcu.edu.cn (H.J.); yuantiantian1@stumail.tjcu.edu.cn (T.Y.); zhaoxiangyun1@stumail.tjcu.edu.cn (X.Z.); zhanjinhao@stumail.tjcu.edu.cn (J.Z.)

² School of Information Engineering, Tianjin University of Commerce, No. 409 Guangrong Road, Beichen District, Tianjin 300134, China; lihaifeng@tjcu.edu.cn (H.L.); renhongsheng1@stumail.tjcu.edu.cn (H.R.)

* Correspondence: changqing@tjcu.edu.cn; Tel.: +86-022-2667-5774

Abstract: Given the continual rise in mission diversity and environmental complexity, the adept integration of a robot's aerial and terrestrial locomotion modes to address diverse application scenarios has evolved into a formidable challenge. In this paper, we design a reconfigurable airframe robot endowed with the dual functionalities of rolling and flying. This innovative design not only ensures a lightweight structure but also incorporates morphing capabilities facilitated by a slider-crank mechanism. Subsequently, a land-to-air transformation strategy for the robot is introduced, achieved through the coordinated movement of the robotic arm and the servo motor. To ensure stable control of the robot amid external wind disturbances, we leverage the collaboration between a Generative Adversarial Network (GAN) and a Nonlinear Model Predictive Control (NMPC) controller. After the wind force magnitude is predicted through the neural network, the robot's adeptness in flexible trajectory tracking is verified. Under simulated wind conditions of 12.1 m/s, the trajectory error consistently remains within the range of 10–15 cm, affirming the effectiveness of this control method.

Keywords: reconfigurable robot; umbrella mechanism; rolling and flying locomotion; Generative Adversarial Networks; external wind disturbances



Citation: Chang, Q.; Yu, B.; Ji, H.; Li, H.; Yuan, T.; Zhao, X.; Ren, H.; Zhan, J. Design and Control of a Reconfigurable Robot with Rolling and Flying Locomotion. *Actuators* **2024**, *13*, 27. <https://doi.org/10.3390/act13010027>

Academic Editor: Liang Sun

Received: 22 November 2023

Revised: 27 December 2023

Accepted: 29 December 2023

Published: 9 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, advances in autonomous navigation and computer vision technologies have greatly promoted the development of quadrotors. Characterized by their light weight, reliability and flexibility, robots have been used in a variety of fields such as search and rescue, military reconnaissance, agricultural inspection and power equipment inspection [1–6]. Due to their portability, ability to navigate in tight spaces and cost-effectiveness, micro robots have become increasingly popular in the market. However, as missions and environments become more complex, the adaptability of robots is challenged [7], especially with the requirement of a compact size with the ability to fly extended distances and be able to carry payloads. The limited flight efficiency of quadrotors, especially micro robots, not only limits the range of tasks they can perform, but also hinders the continued development and wider use of this robot category [8,9].

For the robots that need to perform a wide range of tasks, mixing aerial and terrestrial locomotion is necessary, which allows the robots to fly over obstacles normally and save energy by rolling on the ground [10–12]. The advantage is ground robots is that they can handle complex terrains, but the disadvantage of having complex linkages and multiple actuators, and having difficulty in crossing gullies that are several times as long as themselves, or reaching the heights of the terrain [13]. However, the combination in terrestrial

and aerial modes has led to a significant enhancement of the robot's range of motion [14,15], and it has been experimentally verified that the energy consumption of wheel-tracked actuation is about 5–10% that of a pure flight quadrotor.

In fact, some hybrid aerial and terrestrial locomotion robots have been developed. For example, JIA et al. [16] proposed a quadrotor with a passive reconfigurable mechanism. By using passive joints instead of additional actuators and servo motors, the design allows the robots to remain compact and lightweight, and in the ground motion modes, it is still supplied with propulsion through the propellers used in aerial locomotion. The FCSTAR robot [17] arranges a belt and gear reduction device at the end of a quadrotor arm to form a drive wheel system, and realizes the controllable folding of a deformable fuselage through a propeller arm mechanism to adjust the fuselage size during ground motion. However, these hybrid modal robots have the following problems: redundant mechanical structure [18], a small range of tasks, poor obstacle surmounting ability of ground mode [19], and easy damage of propellers exposed outside the fuselage. Table 1 illustrates the motion modes and rotor protection strategies of some existing multimodal robots.

Table 1. Summary of Hybrid Aerial-Ground Robots [13–17].

	Mode Descriptions	Rotor Protection
Baxter	Two modes of operation, aerial and terrestrial. Employs two novel hardware mechanisms: the M-Suspension and the Decoupled Transmission	Spherical Cage Protection
Hybrid aerial/terrestrial robot	A quadcopter with a mechanism for ground movement. Not use power dedicated to ground movement, and instead uses the flight mechanism of the quadcopter to achieve ground movement as well.	Rotor exposed outside None close mechanism
LEONARDO	Flying and walking. Using synchronized control of distributed electric thruster and a pair of multi-jointed legs, the two modes of flight and walking are interchanged.	Rotor exposed outside None close mechanism
Hybrid Terrestrial Quadrotor	Flying and rolling. The transitions between flight and rolling are accomplished with a highly dynamic maneuver, the robot remains compact and lightweight.	Rotor exposed outside None close mechanism
FCSTAR	Climbing walls and flying. By using thrust reversal and its 4-wheel drive, the robot can drive over steep slopes.	Rotor exposed outside None close mechanism

Therefore, this paper designs a hybrid robot with high adaptability to the external environment, compact structure, long endurance, and a wide range of executable tasks, which has two kinds of motion modes: aerial and rolling. The physical representation of the robot is shown in Figure 1. In the rolling mode, its two-side wheeled structure can realize the ground-air mode change through the umbrella-like closing mechanism. The mechanism keeps the robot's paddles protected inside when not in use, effectively reducing the risk of damage to the fragile paddles.

However, in order to realize the switching of the two motion modes, the design of the umbrella retracting mechanism makes the center of gravity of the proposed robot shift upward significantly. This leads to the fact that the proposed robot is more prominently affected by wind and other external disturbances than a normal quadrotor robot in actual flight [20–22], which not only makes it difficult to ensure that the robot accurately tracks the reference trajectory, but also may cause the robots to lose control of flight, resulting in incalculable consequences. Therefore, a high-performance controller with strong anti-jamming capability is needed.

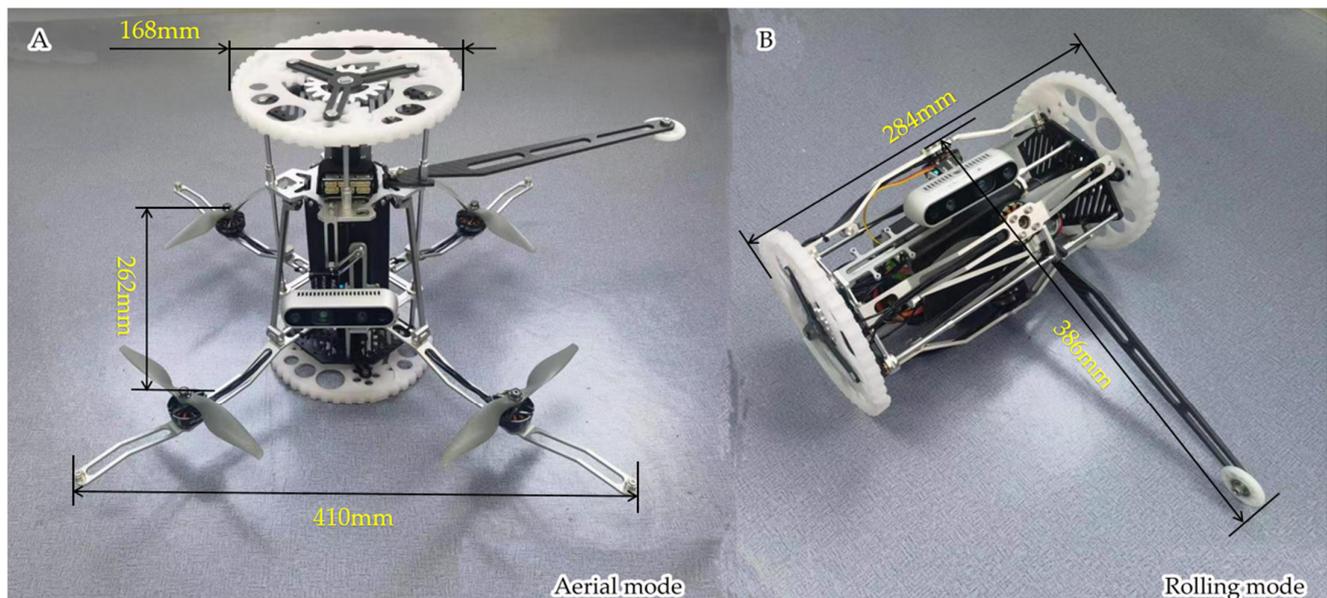


Figure 1. Motion modes of the robot. (A): aerial mode; (B): rolling mode.

Considering the existing controllers, such as cascade controllers [23], linear quadratic regulator controllers [24], and model predictive control (MPC) controllers [25], a majority of them do not meet expectations for quadrotor control in wind-disturbed environments. To address the above challenges, this paper introduces a novel flight control system called GAN-NMPC, which combines Nonlinear Model Predictive Control (NMPC) with Generative Adversarial Networks (GAN) [26–28]. The GAN is able to estimate the wind force in the wind field environment based on the robot's attitude, which subsequently provides the aerodynamic rationale for the NMPC controller.

The structure of this paper is as follows. We introduce the mechanical structure and modal switching principle of this robot in Section 2. In Section 3, we present the motion and dynamics modeling analysis of this robot. In Section 4, an anti-interference performance good controller is proposed, which is applicable to the reconfigurable robot designed in this paper. It consists of an NMPC controller and GAN. In addition, Section 5 describes the wind farm environment construction and GAN training experiments, on the basis of which simulation experiments for robot trajectory tracking are conducted and results are presented. The conclusions and suggestions for future work are shown in Section 6.

2. Design of the Reconfigurable Robot

2.1. Mechanical Design

The robot is composed of a quad-rotor aircraft and a wheel ground-driven robot. Its 3D models are shown in Figures 2 and 3. It consists of two symmetrical wheel mechanisms, four symmetrically placed “X” shaped arms and a main body. In ground mode, the wheels are symmetrically distributed on both sides of the fuselage and are controlled by two independent drive motors. In aerial mode, the flight components, namely the blade plane of the propeller, are placed vertically on the fuselage. When the linear motor drives the arm to fold upwards, both the propeller and the drive motor can be cleverly stored inside the U-shaped slot of the arm, making the folded body of the robot more compact and able to easily maneuver in narrow channels.

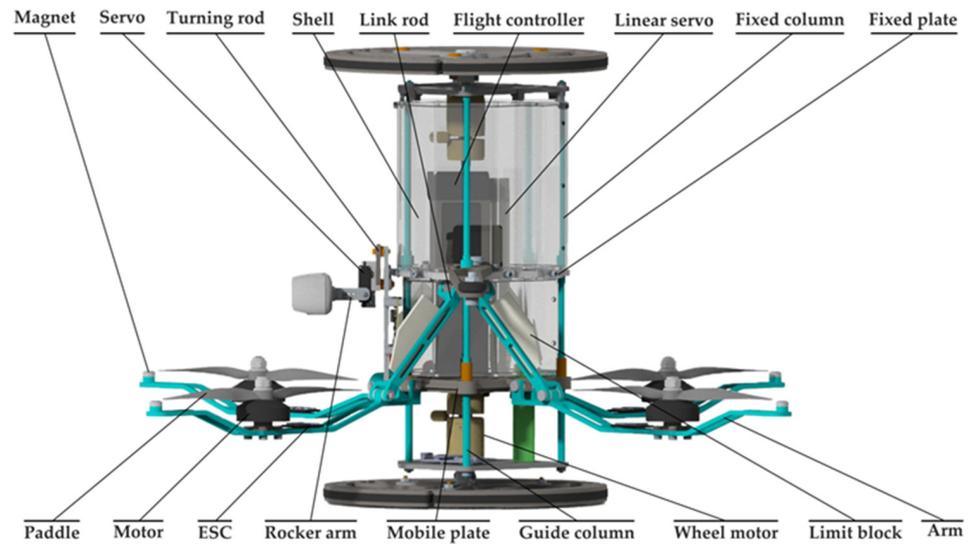


Figure 2. The 3D model of the robot in flight state layout.

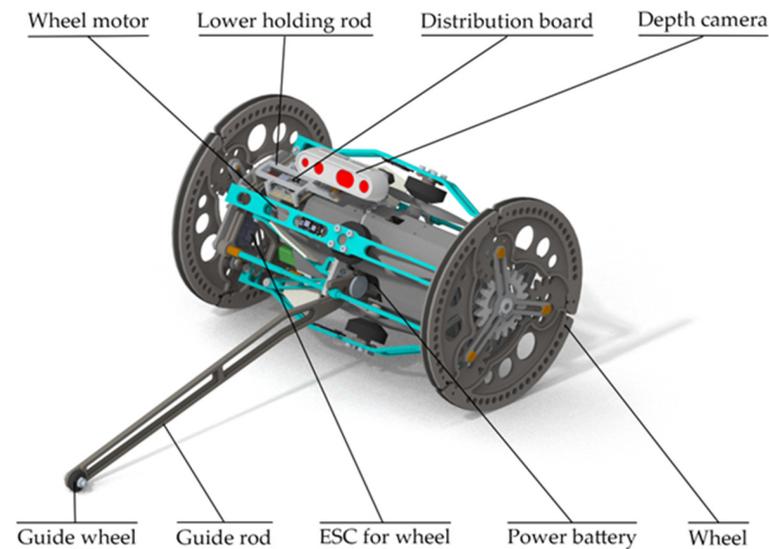


Figure 3. The 3D model of the robot in rolling state layout.

Table 2 lists the physical parameters of the robot. The four symmetrical arms of the robot can freely swing within the x - z plane, with a swinging range of 0 – 90° , without interference with other components during the swinging process. To achieve flight stability, we have chosen the reliable PX4 flight controller control system to ensure the robot operates stably in various motion modes.

2.2. Mode Switching

The innovative robot introduced in this paper achieves mode switching through the arm-retracting mechanism. During the transition from ground to air, the four arms unfold, forming a quadcopter configuration. Conversely, when transitioning from air to ground, the arms retract to a position parallel to the robot's Z -axis. This mode switching not only addresses the challenge of carrying a traditional quadcopter but also establishes a rotor protection mechanism. Folding the arms shields fragile flight components from external collisions and interference. The switching process between rolling and flying modes is illustrated in Figure 4. In a rolling state, the arms automatically expand. Upon ground contact, the main body of the robot is supported. When the arms are fully deployed, the center of gravity shifts to the other side of the landing point. Under the influence of gravity and inertia forces, the main body automatically stands upright, transitioning to the flying

state. When the robot in flight needs to switch to the ground state, the arm will start to close under the drive of servo motors after receiving the corresponding command. Controlling the rotation of one side propeller generates lateral thrust, causing the main body to tilt to one side. During the tilting process, all arms simultaneously fold. When the center of gravity moves to the other side of the landing point, the propellers immediately stop rotating and hit the limit block under the influence of inertia to cease motion, while the arms continue to fold. Under the force of gravity, the main body continues to descend until the arms are fully rotated into place, completing the transition to the rolling state.

Table 2. Physical parameters of the hybrid robot.

Component	Parameters
Mass (with battery)	1.86 kg
Folded size	284 × 168 × 168 mm ³
Unfolded size	410 × 410 × 284 mm ³
Propeller size (maximum boundary)	170 mm × 20 mm
Minimum pass size	390 mm × 200 mm
Battery of robot	6 S, 22.2 V, 2700 mAh
Path planning and decision response time	≤200 ms
Maximum flight time	≥15 min
Mode switching time	≤5 s
Rolling speed	1.86 m/s
Creep speed	0.87 m/s
Flight speed	10.66 m/s

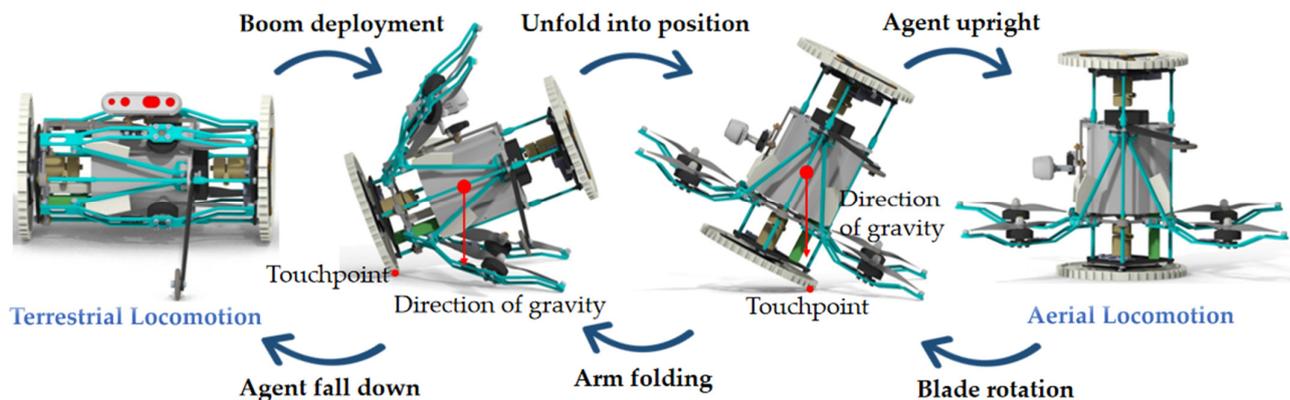


Figure 4. The switching process between rolling and flying.

A sliding link mechanism similar to the umbrella closing mechanism is used to achieve the folding and expansion of the four arms, as shown in Figure 5. In this mechanism, the upper linear servo set on a fixed plane drives the lower moving plane to move up and down in the guide column to realize the retraction and extension of the arm. The moving plane is connected to the short and long connecting rods through a rotating shaft, and the long connecting rod is also connected to the fixed plane through a rotating shaft. When the moving plane moves downwards, an integral part of the arm and short connecting rod will rotate upwards and retract. On the contrary, when the moving plane moves upwards, the arm will unfold downwards. The design of this slider linkage mechanism enables the arm to automatically retract and unfold during mode switching, providing convenience and flexibility for the robot's multiple functions.

2.3. Control Scheme

There are two different modes for the robot: a regular flight mode and a ground mode. In flight mode, the flight controller uses its accelerometer and gyroscope sensors to provide closed-loop correction commands, controlling the robot's flight. In ground mode, the flight controller disconnects the motor arms of the robot through pulse-width modulation (PWM)

switches, and the wheel motors receive instructions directly from a programmable remote controller through a secondary receiver, enabling the robot to roll. Additionally, a linear servo is controlled by the programmable controller to achieve mode switching of the robot. The remote controller can program the PWM for mixing motor commands, allowing the robot to perform multiple functions. The control system scheme is shown in Figure 6. Commands for movement and deformation control of the robot are given by a human operator.

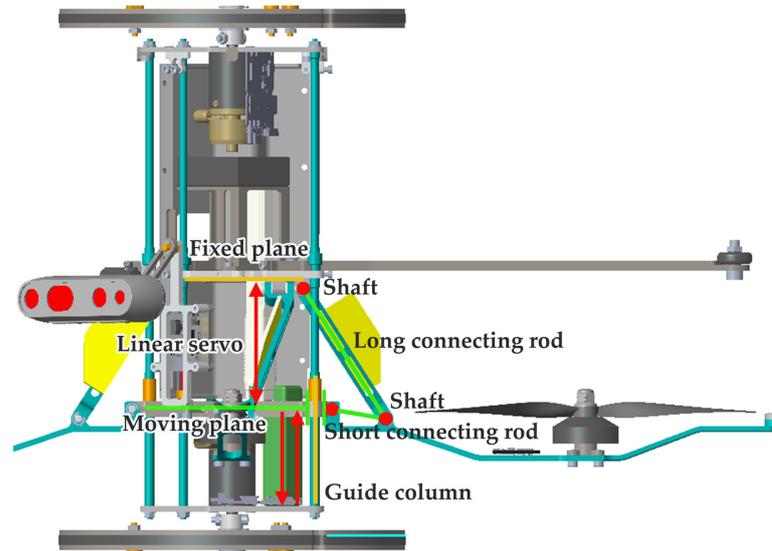


Figure 5. Slider linkage mechanism for arm extension.

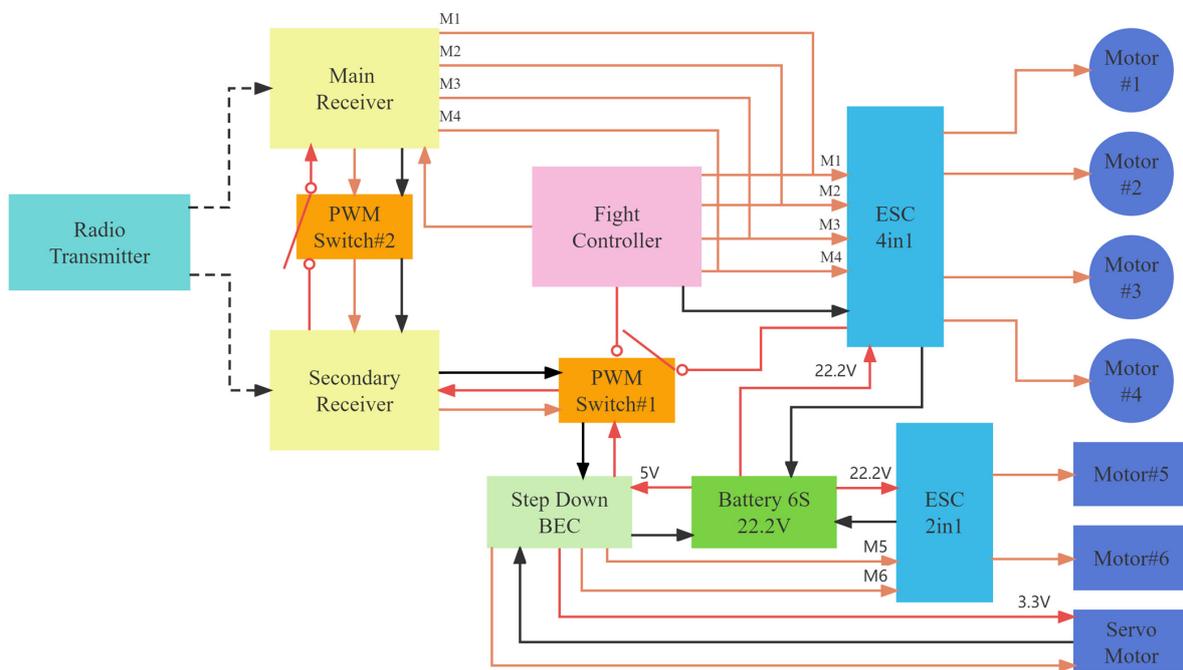


Figure 6. The control scheme of the hybrid drone. In flight mode, switch 1 is closed and switch 2 is opened, the flight controller only controls the arm motors, and in the diagram, ESC 4in1 represents the electronic speed controller for the four arm motors. In ground mode, switch 1 is opened and switch 2 is closed, the flight controller only controls the wheel motors and the servo motor, with ESC 2in1 representing the electronic speed controller for the two wheel motors, and BEC being the depicted battery eliminator circuit. Note that in the diagram that the solid lines represent electrical signals, while dashed lines represent wireless signals.

3. Robot Modeling

In order to conduct more extensive research on the proposed robot in this paper, it is necessary to gain a deeper understanding of the flight characteristics of the quadrotor robot and a more precise description of its spatial motion. First, in this section, we establish corresponding coordinate systems and further investigate the flight principle, motion rules, and forces and torque applied to the proposed robots in this paper. Its dynamic model is illustrated in Figure 7. These elements serve as the groundwork for formulating the dynamic equations and designing the control system for the robot.

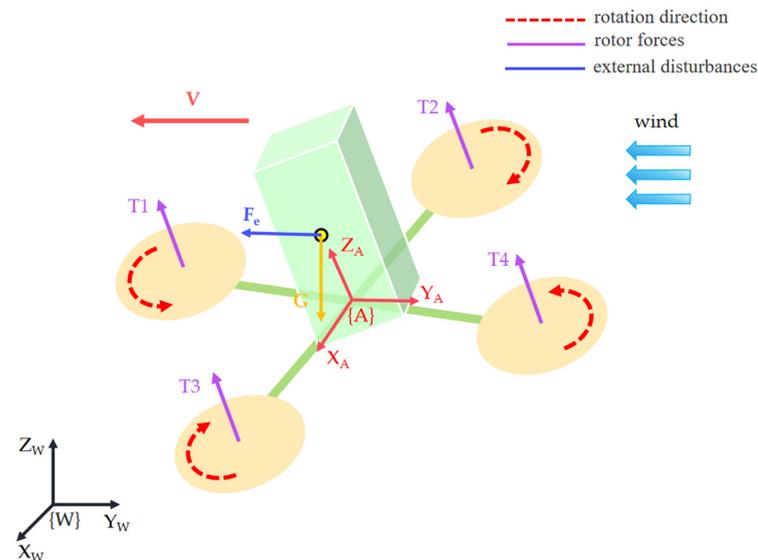


Figure 7. The dynamic model of the hybrid robot.

3.1. Robot Kinematics

Coordinate systems need to be established to better describe the conversion relationship between various variables in the quadrotor robot control system, providing a basis for subsequent dynamic modeling of the robot. The coordinate systems used in this paper are as follows:

1. Body Coordinate System

The origin of the body coordinate system is located at the geometric center of the robot, with the X_A axis defined as the direction of robot flight velocity, the Y_A axis determined by the right-hand rule, and the Z_A axis being perpendicular to the horizontal plane and pointing upwards.

2. World Coordinate System

The world coordinate system, also known as the inertial frame, is mainly used to represent the motion state of the quadrotor robot relative to the ground. It is usually centered on a point on the ground, with the X_W axis indicating east direction, the Y_W axis determined by the right-hand rule, and the Z_W axis being perpendicular to the ground and pointing upwards.

The following variables are used to describe the dynamics and kinematics of a quadcopter:

$\eta_1 = [x \ y \ z]^T$ —position of the origin of $\{A\}$ measured in $\{W\}$;

$\eta_2 = [\phi \ \theta \ \psi]^T$ —angles of roll (ϕ), pitch (θ) and yaw (ψ) that parametrize locally the orientation of $\{A\}$ with respect to $\{W\}$;

$v_1 = [u \ v \ w]^T$ —linear velocity of the origin of $\{A\}$ relative to $\{W\}$ expressed in $\{A\}$ (i.e., agent-fixed linear velocity);

$v_2 = [p \ q \ h]^T$ —angular velocity of $\{A\}$ relative to $\{W\}$ expressed in $\{A\}$ (i.e., agent-fixed angular velocity);

$h_W = [x_W \ y_W \ z_W]^T$ —distance from the origin of $\{A\}$ to the robot's center of mass.

Because the robot's position and attitude are constantly changing during flight, the coordinates of the robot's body coordinate system and the ground coordinate system are continuously evolving. This paper uses roll angle, pitch angle, and yaw angle to represent the angles of three-axis rotation in the process of transforming the body coordinate system to the ground coordinate system. First, rotate φ angle around the X_A axis of the body coordinate system to obtain a new coordinate system $OX_{A1}Y_AZ_A$. Then, rotate θ angle around the Y_A axis in the newly obtained coordinate system to obtain another new coordinate system $OX_{A1}Y_{A1}Z_A$. Finally, rotate ψ angle around the Z_A axis in the most recent coordinate system to align with the ground coordinate system.

First, the basic rotation axis formulas are provided:

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix} \quad (1)$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \quad (2)$$

$$R_z(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Then, based on the above rotation sequence, the transformation matrix from body frame to inertial frame was obtained:

$$\begin{aligned} {}^W_A R &= R_z(\psi)R_y(\theta)R_x(\varphi) \\ &= \begin{bmatrix} c\theta c\psi & -c\varphi s\psi + s\varphi s\theta c\psi & s\varphi s\psi + c\varphi s\theta c\psi \\ c\theta s\psi & c\varphi c\psi + s\varphi s\theta s\psi & -s\varphi c\psi + c\varphi s\theta s\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \end{aligned} \quad (4)$$

where $c = \cos$ and $s = \sin$.

The transformation matrix between the two reference systems is obtained by matrix multiplication of the basic orthogonal rotation matrix. The agent-fixed linear velocity vector v_1 and the position velocity vector $\dot{\eta}_1$ can be related through the transformation matrix ${}^W_A R(\eta_2)$:

$$\dot{\eta}_1 = \frac{d\eta_1}{dt} = {}^W_A R(\eta_2)v_1 = \begin{bmatrix} c\theta c\psi & -c\varphi s\psi + s\varphi s\theta c\psi & s\varphi s\psi + c\varphi s\theta c\psi \\ c\theta s\psi & c\varphi c\psi + s\varphi s\theta s\psi & -s\varphi c\psi + c\varphi s\theta s\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5)$$

The Agent-fixed angular velocity vector v_2 and Euler rate vector $\dot{\eta}_2$ are connected through a transformation matrix $Q(\eta_2)$

$$\dot{\eta}_2 = Q(\eta_2)v_2, Q(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ \theta & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{s\phi}{c\theta} \end{bmatrix} \quad (6)$$

The simple expression of system kinematics is as follows:

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} {}^W_A R(\eta_2) & 0 \\ 0 & Q(\eta_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \dot{\eta} = J_R(\eta)v \quad (7)$$

3.2. Robot Dynamics

For the six degrees of freedom (DOF) motion of a robot, factors that affect this motion mainly include body and propeller deformation, rigid structure of the robot, and changes in the center of gravity. Here, we make the following assumptions to simplify the system model of the robot:

1. The center of gravity of the robot coincides with the centroid, and the mass of the robot remains unchanged during the dynamic process.
2. The rotational inertia of the quadcopter is assumed to be zero.
3. The robot body does not deform and is structurally symmetrical during motion.

In the inertial coordinate system, the following equations are established using Newton's laws for the translation and rotation motions of the robot:

$$m\ddot{X} = F \quad (8)$$

$$I_{3 \times 3}\ddot{\omega} + \dot{\omega} \times I_{3 \times 3}\dot{\omega} = M \quad (9)$$

where m represents the mass of the robot, F represents the net external force acting on the robot, \ddot{X} represents the linear acceleration of the robot in the three axes, $I_{3 \times 3}$ represents the moment of inertia of the robot, $\ddot{\omega}$ represents the angular acceleration of the robot in the three axes, $\dot{\omega}$ represents the angular velocity of the robot in the three axes, and M represents the net external torque acting on the robot.

Then, this paper presents a set of rigid body motion equations derived from Euler's theorem. Consider the aircraft-fixed coordinate system frame $\{A\}$ rotating with angular velocity $\omega = [\omega_1 \ \omega_2 \ \omega_3]$ about the ground-fixed coordinate system frame $\{W\}$.

$$I_A = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, I_A = I_A^T > 0 \quad (10)$$

Since the main axis of the fuselage's fixed coordinate system is aligned with the robot's center axis, $I_{xy} = I_{yx} = I_{xz} = I_{zx} = I_{zy} = I_{yz} = 0$.

Therefore, we present the robot dynamics equations in the body frame as follows:

$$\begin{cases} F_x = m[\dot{u} - vh + wq - x_W(q^2 + h^2) + y_W(pq - \dot{h}) + z_W(ph + \dot{q})] \\ F_y = m[\dot{v} - wp + uh + x_W(qp + \dot{h}) - y_W(p^2 + h^2) + z_W(qh - \dot{p})] \\ F_z = m[\dot{w} - up + vp + x_W(hp - \dot{q}t) + y_W(hq - \dot{p}) - z_W(q^2 + p^2)t] \\ T_\phi = I_{xx}\dot{p} + (I_{zz} - I_{yy})qh + m[y_W(\dot{w} - uq + vp) - z_W(\dot{v} - wp + uh)] \\ T_\theta = I_{yy}\dot{q} + (I_{xx} - I_{zz})qh + m[z_W(\dot{u} - vh + wq) - x_W(\dot{w} - up + vp)] \\ T_\psi = I_{zz}\dot{h} + (I_{yy} - I_{xx})qh + m[x_W(\dot{v} - wp + uh) - y_W(\dot{u} - vh + wq)] \end{cases} \quad (11)$$

Next, nonlinear dynamic equations in the inertial frame of the quadrotor's six DOF can be expressed in a compact form as follows:

$$M\dot{v} + C(v)v + M_e(\eta) + g_W(\eta) = \tau \quad (12)$$

where $\eta = [\eta_1 \ \eta_2]^T$, is a vector representation of position and direction, $v = [v_1 \ v_2]^T$ is the vector of linear and angular velocities, M is the mass and inertia matrix of the robot, matrix $C(v)$ consists mainly of Coriolis and centripetal terms, $M_e(\eta)$ represents the vector of wind forces and moments, $g_W(\eta)$ represents the vector of gravitational forces and

moments, and control inputs are given as vector τ . The mass and inertia matrix of the robot M is defined as follows:

$$M = \begin{bmatrix} m & 0 & 0 & 0 & mz_W & -my_W \\ 0 & m & 0 & -mz_W & 0 & mx_W \\ 0 & 0 & m & my_W & -mx_W & 0 \\ 0 & -mz_W & my_W & I_x & 0 & 0 \\ m & 0 & -mx_W & 0 & I_y & 0 \\ -my_W & mx_W & 0 & 0 & 0 & I_z \end{bmatrix}, M = M^T > 0 \quad (13)$$

The above matrix representation is unique. For the matrix $C(v)$, there are a large number of parameters, and using mathematical methods, it is parameterized so that $C(v)$ is skew-symmetric.

$$C(v) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ m(y_W\dot{\theta} + z_W\dot{\psi}) & -m(x_W\dot{\theta} - \dot{z}) & -m(x_W\dot{\psi} + \dot{y}) \\ -m(y_W\dot{\phi} + \dot{z}) & m(z_W\dot{\psi} + x_W\dot{\phi}) & -m(y_W\dot{\psi} - \dot{x}) \\ -m(z_W\dot{\phi} - \dot{y}) & -m(z_W\dot{\theta} + \dot{x}) & m(x_W\dot{\phi} + y_W\dot{\theta}) \\ m(y_W\dot{\theta} + z_W\dot{\psi}) & -m(x_W\dot{\theta} - \dot{z}) & -m(x_W\dot{\psi} + \dot{y}) \\ -m(y_W\dot{\phi} + \dot{z}) & m(z_W\dot{\psi} + x_W\dot{\phi}) & -m(y_W\dot{\psi} - \dot{x}) \\ -m(z_W\dot{\phi} - \dot{y}) & -m(z_W\dot{\theta} + \dot{x}) & m(x_W\dot{\phi} + y_W\dot{\theta}) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (14)$$

Then, gravity and its moments $g_W(\eta)$ are shown as follows:

$$f_W(\eta_2) = {}^W_A R^{-1}(\phi, \theta) \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (15)$$

$$g_W(\eta_2) = - \begin{bmatrix} f_W(\eta_2) \\ h_W \times f_W(\eta_2) \end{bmatrix} \quad (16)$$

$F_e = [F_{ex} \quad F_{ey} \quad F_{ez}]^T$ represents the wind force,

$$f_e(\eta_2) = {}^W_A R^{-1}(\phi, \theta, \psi) \begin{bmatrix} F_{ex} \\ F_{ey} \\ F_{ez} \end{bmatrix} \quad (17)$$

$$M_e(\eta_2) = - \begin{bmatrix} f_e(\eta_2) \\ h_W \times f_e(\eta_2) \end{bmatrix} \quad (18)$$

where g denotes the gravitational constant and the control input the vector τ is represented as follows:

$$f_{\tau}(\eta_2) = {}^W_A R^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix}$$

$$\tau(\eta_2, U) = \begin{bmatrix} f_{\tau}(\eta_2) \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (19)$$

where $U = [U_1 \ U_2 \ U_3 \ U_4]^T$ represents the position system input control variables, including the robot's altitude, roll, pitch, and yaw control values,

$$\begin{cases} U_1 = k(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = k(\Omega_4^2 - \Omega_2^2) \\ U_3 = k(\Omega_1^2 - \Omega_3^2) \\ U_4 = k_d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{cases} \quad (20)$$

where Ω represents the rotor speeds for each rotor, and $\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$.

3.3. Turbulent Wind Field Modeling

Generally speaking, the turbulent wind field of the atmosphere can be derived from conditions such as continuity equations, Navier-Stokes equations, and so on. However, the method of calculating the turbulent wind field model by the previous way is very computationally intensive and difficult to realize in practice. Therefore, a suitable stochastic model is needed to simulate the turbulent wind field model.

In this paper, we choose to design the turbulent wind field through the Dryden model. First of all, the numerical simulation of the model is conducted. Through the generation of Gauss-distributed random signals, in accordance with the known spectrum design filter, the use of filters will be converted from white noise signal to atmospheric turbulence signals, to complete the simulation of the turbulent wind field.

The working principle is shown in Figure 8. $G(s)$ denotes the shaping filter transfer function, $m_i(r)$ is defined as the Gaussian distributed white noise signal, $u(r)$, $v(r)$ and $w(r)$ denotes the velocity components of the turbulent signal in the three directions, respectively.

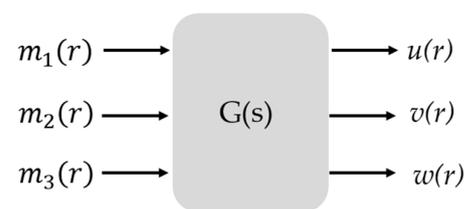


Figure 8. The working principle.

Next, the turbulent wind field is described by using the Dryden model, where the velocity of the turbulent flow is the main factor affecting the motion of the vehicle, and the relationship between spatial and temporal frequencies is $\Omega = \omega/V$, when the flight velocity of the quadrotor is V ; L_u , L_v and L_w represent the turbulence scale and σ_u , σ_v and σ_w indicate the turbulence intensity.

From the relation $\Phi(\omega) = \frac{1}{V}\Phi(\Omega) = \frac{1}{V}\Phi(\frac{\omega}{V})$, the time spectrum of the velocity is given as follows:

$$\left\{ \begin{array}{l} \Phi_u(\omega) = \sigma_u^2 \frac{L_u}{\pi V} \frac{1}{1+(L_u \frac{\omega}{V})^2} \\ \Phi_v(\omega) = \sigma_v^2 \frac{L_v}{\pi V} \frac{1+12(L_v \frac{\omega}{V})^2}{[1+4(L_v \frac{\omega}{V})^2]^2} \\ \Phi_w(\omega) = \sigma_w^2 \frac{L_w}{\pi V} \frac{1+12(L_w \frac{\omega}{V})^2}{[1+4(L_w \frac{\omega}{V})^2]^2} \\ \Phi_{V_x}(\omega) = \frac{(\frac{\omega}{V})^2}{1+(\frac{3b}{\pi V}\omega)^2} \Phi_v(\omega) \\ \Phi_{W_x}(\omega) = \frac{(\frac{\omega}{V})^2}{1+(\frac{4b}{\pi V}\omega)^2} \Phi_w(\omega) \\ \Phi_{W_y}(\omega) = \frac{\sigma_w^2}{L_w V} \frac{0.2(\frac{\pi L_w}{2b})^{1/3}}{1+(\frac{4b}{\pi}\Omega)^2} \end{array} \right. \quad (21)$$

Since the movement region of the quadrotor belongs to low altitude and the flight altitude is low, the relationship between the turbulence intensity and the turbulence scale under the condition of low altitude can be expressed as follows [29]:

$$2L_w = h \quad (22)$$

$$L_u = 2L_v = \frac{h}{(0.177 + 0.000823h)^{1.2}} \quad (23)$$

$$\sigma_w = 0.1u_{20} \quad (24)$$

$$\frac{\sigma_u}{\sigma_w} = \frac{\sigma_v}{\sigma_u} = \frac{1}{(0.177 + 0.000823h)^{0.4}} \quad (25)$$

where in, u_{20} denotes the wind speed at a height of 6 m, and h denotes the height.

The white noise signal is output through a filter with $G(s)$ as the transfer function, and the spectral function of the output sequence $x(t)$ can be expressed as follows:

$$\Phi(\omega) = |G(i\omega)|^2 = G^*(i\omega)G(i\omega) \quad (26)$$

After decomposing the turbulence spectral function to obtain the transfer function of the filter used to generate the given spectrum, the simplified turbulent velocity transfer function is as follows:

$$\left\{ \begin{array}{l} G_u(s) = \frac{K_u}{T_u s + 1} \\ K_u = \sigma_u \sqrt{\frac{L_u}{\pi V}}, T_u = \frac{L_u}{V} \end{array} \right. \quad (27)$$

$$\left\{ \begin{array}{l} G_v(s) = \frac{K_v}{T_v s + 1} \\ K_v = \sigma_v \sqrt{\frac{L_v}{\pi V}}, T_v = \frac{2}{\sqrt{3}} \frac{L_u}{V} \end{array} \right. \quad (28)$$

$$\left\{ \begin{array}{l} G_w(s) = \frac{K_w}{T_w s + 1} \\ K_w = \sigma_w \sqrt{\frac{L_w}{\pi V}}, T_w = \frac{2}{\sqrt{3}} \frac{L_w}{V} \end{array} \right. \quad (29)$$

4. Control Design

From the dynamics and kinematics analysis in Section 3, it can be observed that, for the underdriven quadrotor model with six degrees of freedom and four control inputs, there exists a strong coupling relationship between the control quantities of the vehicle. Therefore, the designed controller should effectively control the position of the quadrotor

during flight and simultaneously ensure stable control of its attitude under the influence of wind. To achieve this goal, this paper utilizes the GAN-NMPC augmentation controller, and the flowchart is presented in Figure 9. Due to the limited data of wind training samples in real environments, a turbulent wind field model is constructed in Simulink to supplement the training samples. Then, we import the data into the GAN network for wind prediction.

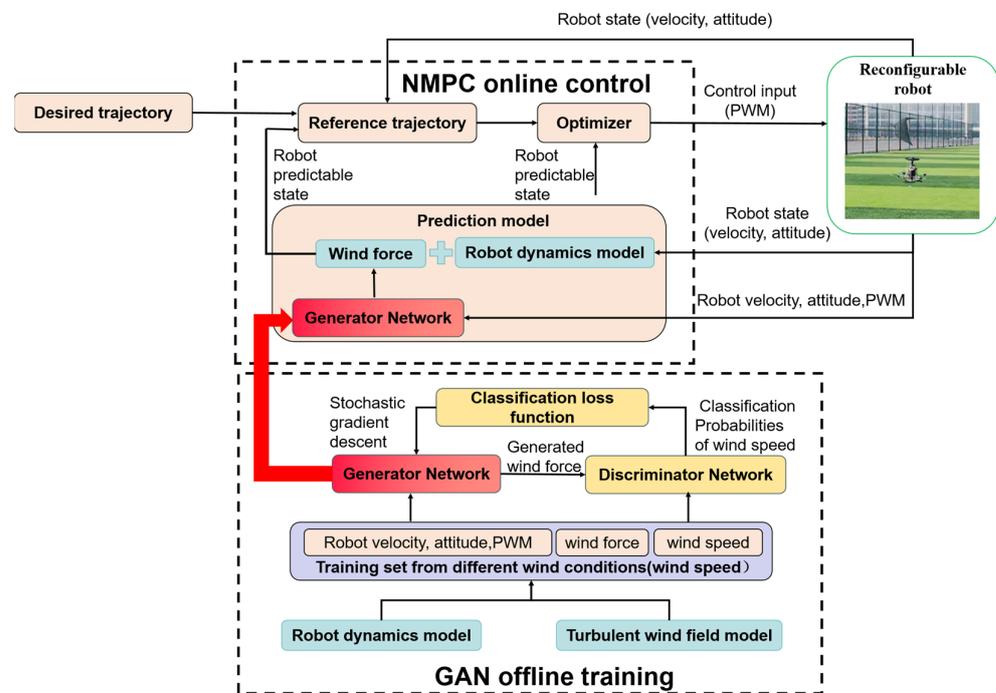


Figure 9. The control flowchart of the GAN-NMPC controller. Input the reference trajectory to the NMPC controller and introduce the robot's position, velocity and attitude of the previous moment and feed it back to the NMPC, Input the robot state to the trained generator network to estimate the wind force, and then add the aerodynamic effects to the dynamics and kinematics model. The NMPC will carry out the next moment of control.

In the turbulent wind field, the intensity and direction of the wind force acting on a robot undergo frequent variations. Conventional nonlinear control techniques, such as adaptive control, are primarily suited for scenarios where state variables change gradually or remain constant, posing challenges in identifying rapidly fluctuating wind forces affecting a robot. Hence, this study employs a data-driven machine learning approach to predict the wind force magnitude on the robot. The GAN is employed to train a wind force network generator, subsequently integrated into the NMPC framework. This methodology enables effective robot control in a turbulent wind field, distinguishing itself from traditional nonlinear control methods.

The GAN-NMPC controller described in this paper primarily employs GAN for the offline training of the predictive model in NMPC, essentially adopting the inherent control architecture of NMPC. It also possesses closed-loop stability [30].

The GAN is trained offline, and the generator network of the GAN serves as the wind force generator within the NMPC prediction model. The training dataset, encompassing diverse wind conditions, is generated through the utilization of the robot dynamics model and turbulent wind field model as elucidated in Sections 3.2 and 3.3. Subsequently, leveraging the GAN's generator network, wind forces acting on the reconfigurable robot can be predicted based on the robot's state and control input. The NMPC, integrated with the wind force predictor derived from GAN's offline training, is proficient in real-time control of the reconfigurable robot. This efficacy has been substantiated through simulations detailed in Section 5.

4.1. Generative Adversarial Network

In this section, we present information about the dataset and experimental platform and discuss the structure and key concepts of our neural network, GAN.

4.1.1. Data Collection and Platform

To learn an effective representation of the aerodynamic effects, we have a custom-built robot follow a randomized trajectory for 2 min each in six different static wind conditions, with speeds ranging from 0 km/h to 22.0 km/h through the simulation in Matlab. However, in experiments, we used wind speeds up to 43.6 km/h (12.1 m/s) to study how our methods extrapolate to unseen wind conditions. The data is collected at 50 Hz with a total of 10,000 data points.

Our experiment was implemented based on Python 3.8.10 and successfully ran in Pycharm and Jupyter Notebook.

4.1.2. The Principle of Generative Adversarial Networks

The GAN is a powerful deep learning model, which is composed of two neural networks: generator and discriminator. The generator is responsible for generating new data samples, and the discriminator is responsible for determining whether the input data was generated by the generator. Through the interaction of these two networks, GANs are able to learn the potential distribution of the generated data to generate high-quality data samples.

The structure diagram of the GAN neural network is shown in the Figure 10. In this structure, we use the speed, the attitude of the robot and the PWM as the input of f networks, and the wind speed as the input of c networks. Wherein, the f network is also called the generator network, which is used to generate data for different wind conditions. The f network performs better in offline training, and it does not change with wind conditions; it is only related to the flight state of the robot, such as the robot's speed. The c network is also called the discriminator network, which is used to generate the probabilities of different wind speed to discriminate the output of the f network and help it obtain a better result.

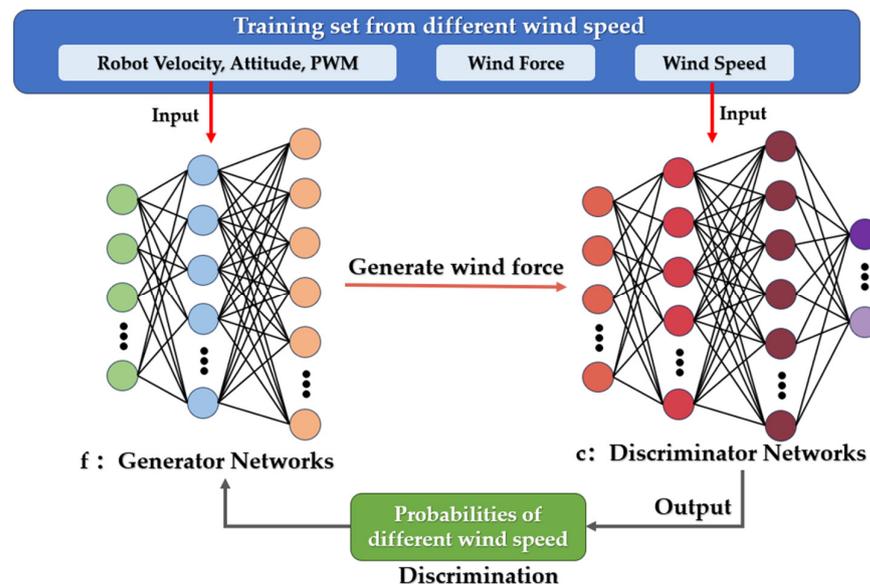


Figure 10. The structure of the GAN.

The principle of the f network is shown in the following equation:

$$\min_{f, a_1, \dots, a_k} \sum_{k=1}^K \sum_{i=1}^{N_k} \left\| y_k^{(i)} - f(x_k^{(i)}) a_k \right\|^2 \quad (30)$$

where in, y represents the label quantity, a represents the weight matrix, which is obtained by performing least squares on the f network and the data in the dataset, and x represents the input. From the formula, it can be seen that the closer the product of f and a approaches the label quantity y , the better.

The principle of the c network is shown in the following equation:

$$\max_{i \in B} \sum_{i \in B} \text{loss} \left(c \left(f \left(x_k^{(i)} \right) \right), k \right) \quad (31)$$

where in, the input of the c network is the output of the f network, k is the wind condition index, and (i) is the input-output pair index. The goal of c is to predict the index k from $f(x)$. And the output of c is a K -dimensional vector for the classification probabilities of K wind condition. The larger the output result of the c network, the better.

The entire training process involves first calculating a through least squares and normalizing it, then fixing c and a to calculate f , and finally fixing f and a to calculate c . The two networks play games with each other to obtain the result, improving the interpretability of the network.

4.2. NMPC Formulation

A quadratic optimization problem is constructed by a multi-objective optimization scheme and the following discrete nonlinear optimal control problem is solved with time discretization by the Runge-Kutta methods of fourth order.

$$\min_u x_N^T Q x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \quad (32)$$

$$\text{subject to : } x_{k+1} = f_{RK4}(x_k, u_k, \delta t) \quad (33)$$

$$x_0 = x_{init} \quad u_{min} \leq u_k \leq u_{max}$$

We discretize the action process of the robots into N steps through time and restrict the constraints between $0 \leq u_k \leq u_{max}$. The optimal control problem is implemented using open-source software.

5. Simulation and Results

5.1. Turbulent Wind Field Environment Construction

From the turbulent velocity transfer function mentioned above, the atmospheric turbulence generator module can be designed in Simulink. The turbulence generator is utilized to carry out the simulation experiment of a turbulent wind field, which is located at a height of 6 m with a wind speed of 10 m per second. With the help of this simulation, the data set and training set required for the training of the neural network is generated. The internal structure of the atmospheric turbulent wind field modeling module built in the Simulink environment is shown in Figure 11.

By generating random signals following a Gaussian distribution, designing filters based on known spectra, and utilizing shaping filters to transform white noise signals into atmospheric turbulence signals, we accomplish the simulation and modeling of a turbulent wind field. This involves establishing a model for atmospheric turbulent wind fields. Subsequently, integrating this wind field model as a disturbance term into the dynamics equation of the robot allows us to describe the dynamic characteristics of the aircraft under the influence of wind disturbances.

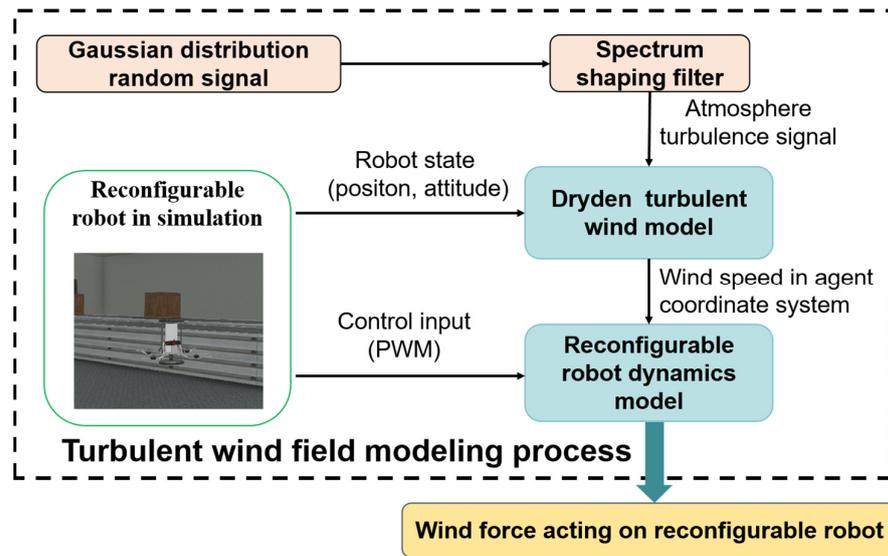


Figure 11. A control block diagram of turbulent wind field modeling.

5.2. GAN Training Experiment

The parameter settings for the simulation verification phase are shown in Table 3.

Table 3. Parameters of GAN.

Parameters	Value
Number of adaptive sampling points: K	32
Number of sampling points for network training: B	256
Adversarial loss coefficient: α	0.01
Network learning rate	5×10^{-4}
Network update probability: h	0.5
The maximum binomial γ of a	10
Epochs	1000

The convergence of f and c networks is as follows. From Figures 12 and 13, it can be seen that the loss of the f network decreases and converges; The loss of the c network has almost not decreased and has not successfully converged, which are all in line with experimental expectations.

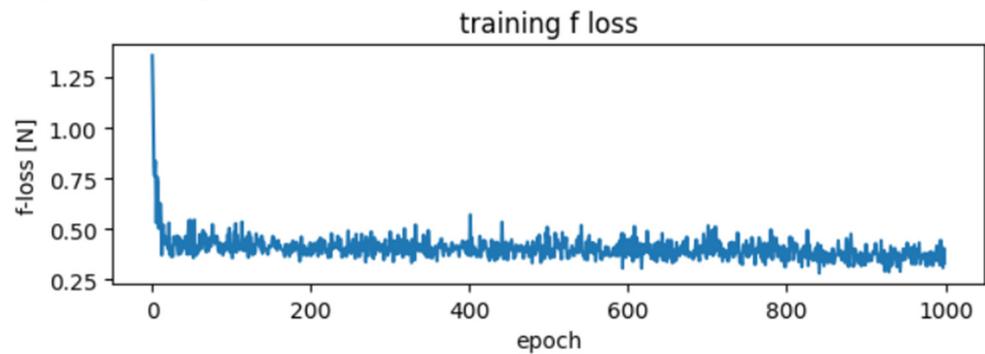


Figure 12. The loss curve of the f network.

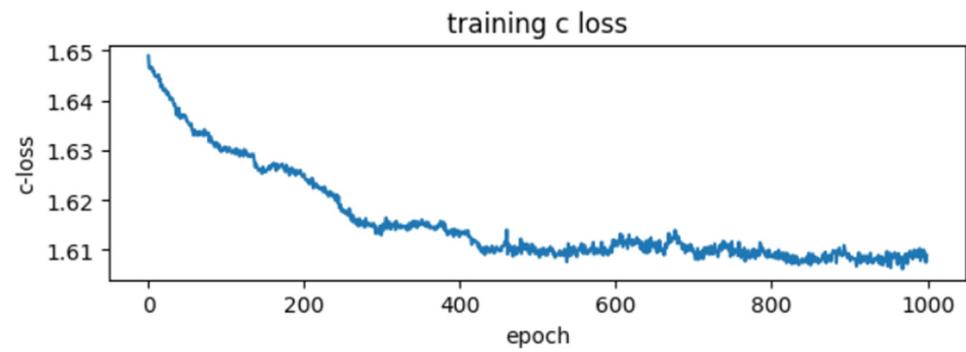


Figure 13. The loss curve of the c network.

The prediction of aerodynamic effects under five wind conditions is shown in the following Figure 14. The experiment uses trained models to predict specific aerodynamic interference. It can be seen that the experimental results of the x-axis and z-axis are generally in line with expectations.

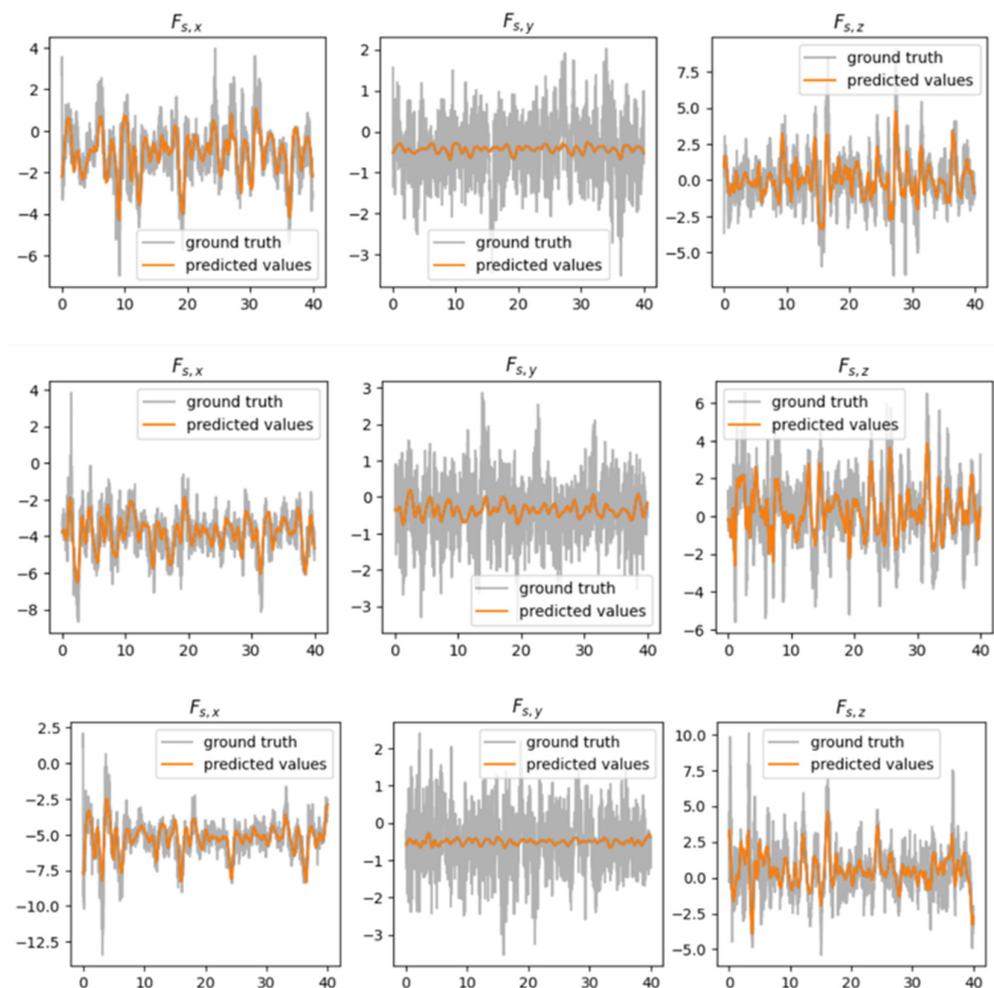


Figure 14. Wind disturbance dynamics prediction effect of the Generative Adversarial Network in a wind speed environment with a true value of 2.5 m/s, 4.9 m/s and 6.1 m/s. Wherein, ground truth represents the true value, which is the label quantity, predicted values represents values in the generated data that meet the true condition.

We also collected MSE values before and after training under different wind speeds, as shown in the Table 4. It can be observed that as the wind speed increases, the MSE decreases more, and the improvement effect becomes more apparent. Figure 15 displays the confusion matrix for the GAN, which is mainly used to compare the predicted values and true values under five different wind conditions of 1.3 m/s, 2.5 m/s, 3.7 m/s, 4.9 m/s and 6.1 m/s. Among them, the abscissa represents the predicted value, the ordinate represents the true value, the sum of each row represents the real number of samples for this wind condition, and the sum of each column represents the number of samples predicted for this category. The numbers in the picture, such as “815” in the first row and column, represent the number of predicted values when the true value is also 1.3 m/s. Both probability and color depth represent accuracy. The higher the probability, the darker the color, and the higher the accuracy. It can be seen from the figure that the probability of correct judgment decreases from 69.39% to 81.58% as the wind force increases, which shows that the wind force has an impact on the predicted value.

Table 4. Error values after neural network training under six wind speed conditions.

Situation	Before Learning (N)	After Learning Error (N)
1.3 m/s	1.20	0.54
2.5 m/s	2.17	0.85
3.7 m/s	3.58	0.89
4.9 m/s	6.69	1.00
6.1 m/s	11.41	1.08

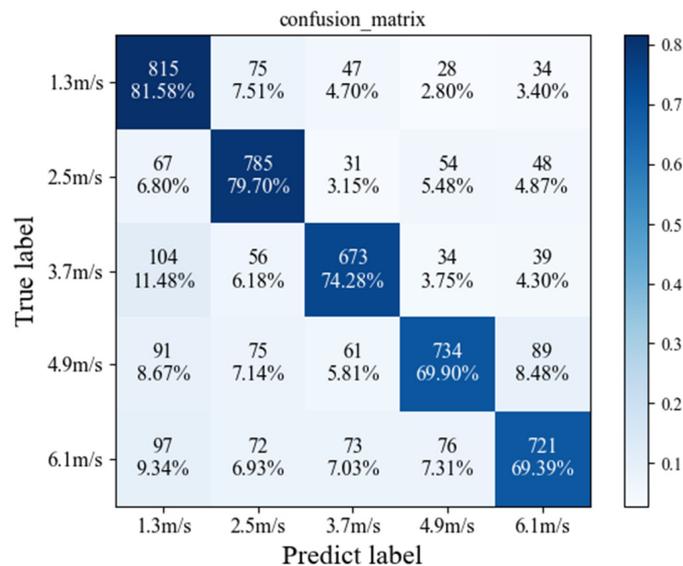


Figure 15. Confusion matrix for GAN training on wind prediction.

5.3. Simulation Experiment

We begin by implementing several state-of-the-art controllers in simulation including PID, NMPC, and the GAN-NMPC controller that we used for the above configuration robot. As shown in Figure 16, the simulation is performed using the physical model of Webots and controlled by Simulink, and the physical parameters of the robots are known in the simulation experiments.

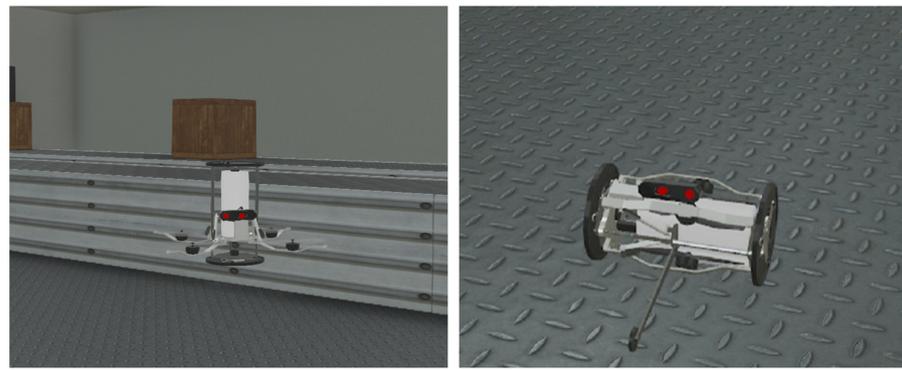


Figure 16. The physical model of the hybrid robot in Webots.

In this section, a simulation case of a robot flying around a spatial Figure 8 in a windy environment is used to verify that the use of the GAN-NMPC controller has good trajectory tracking performance and robustness, which can inhibit the effect of the interference of the wind field on the flight of the robot as well as overcome the flip phenomenon caused by the center of gravity offset to satisfy the expectation that the flight trajectory is closer to the reference trajectory. In the simulation case, the wind field modeling presented in Section 3 was used as the wind field environment, and the ability of each controller to resist wind interference under the state of the robot’s center of gravity shift is judged by comparing the position error with the reference trajectory under the action of each controller, and the comparison concludes that the tracking performance and wind interference resistance of the GAN-NMPC controller applicable in this paper are optimal and have better stability.

To verify the effectiveness of the proposed GAN-NMPC controller, the classical PID Controller and NMPC controller which have been widely used in the field of stand quadrotors had been used as the comparative simulation. The control flowchart of the three controllers in the simulations are shown in Figures 17–19.

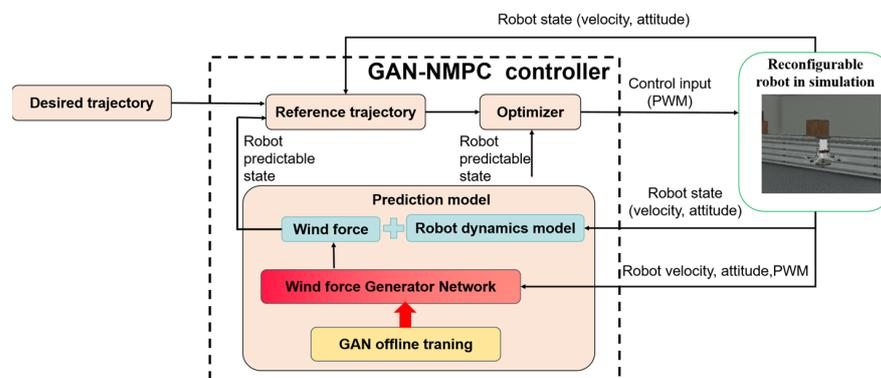


Figure 17. Control flowchart of GAN-NMPC controller in simulation.

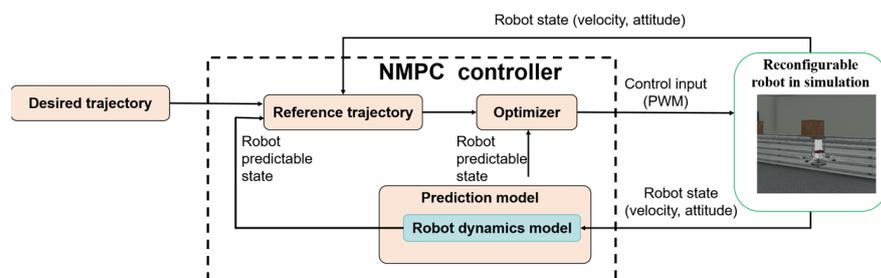


Figure 18. Control flowchart of NMPC controller in simulation.

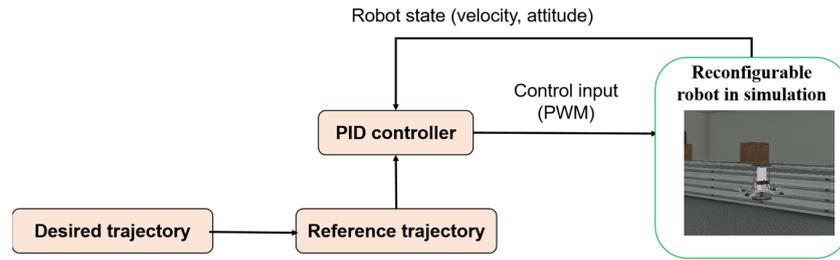


Figure 19. Control flowchart of PID controller in simulation.

The purpose of this simulation experiment is to test the tracking performance and wind interference rejection performance of the robot when flying a figure8 trajectory in x-y-z three-dimensional space. The flight trajectory in three-dimensional space can effectively improve the difficulty of robot trajectory tracking, so as to test whether the controller proposed in this paper can realize excellent trajectory tracking effect and maintain good tracking performance and interference suppression performance. The reference trajectory is a Figure 8 located in a $10\text{ m} \times 10\text{ m} \times 10\text{ m}$ space. The average speed range during flight is 3–6 m/s. The initial position of the robot is set to $[0\ 0\ 0]$, and the robot is specified to fly continuously within 6.2 s. The control effect expected to be achieved in this experiment is to achieve flight stabilization of the quadrotor during straight ahead and steering and accurate tracking of the reference trajectory in the unstable situation of the center of gravity being upward.

The results from Figures 20 and 21 indicate that, the presented quadcopter is unable to maintain stable trajectory tracking under PID or NMPC control when affected by external wind disturbances. It significantly deviates from the intended figure8 trajectory.

Figure 22 shows the effectiveness of the NMPC controller enhanced by wind estimation through the neural network. It adeptly manages the robot amid turbulent wind disturbances, avoiding sudden velocity changes and notably minimizing tracking errors.

Table 5 compares the deviation values of tracking trajectories under the influence of three different controllers. It can be concluded that the PID control, which only relies on the feedback device of the control loop to regulate all the process variables, can neither adjust the model uncertain parameters nor add the constraints, will expose great control defects when subjected to uncertain disturbances, and is far worse than the other two controllers in trajectory tracking performance. The NMPC control, due to its own capacity limitations, cannot correct the disturbances brought by the model, and maintains a certain steady state error in the z-axis. At the same time, from the displacement deviation in the x-y axis, it can be seen that this controller is not as effective as the GAN-NMPC in resisting the turbulent wind.

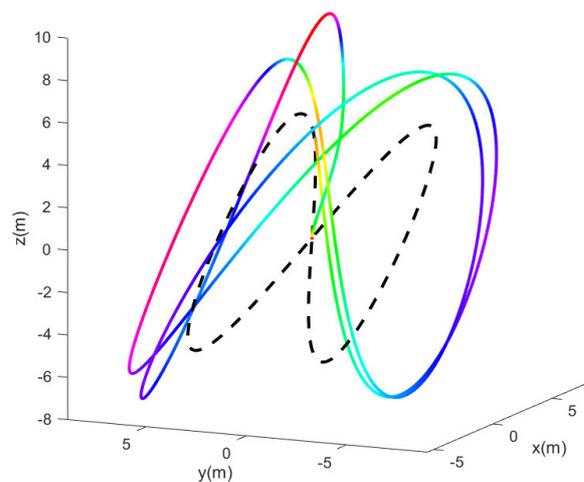


Figure 20. Simulation results of PID algorithm-based experiment under windy conditions. Validity of tracking control algorithm for quadcopter in balanced state.

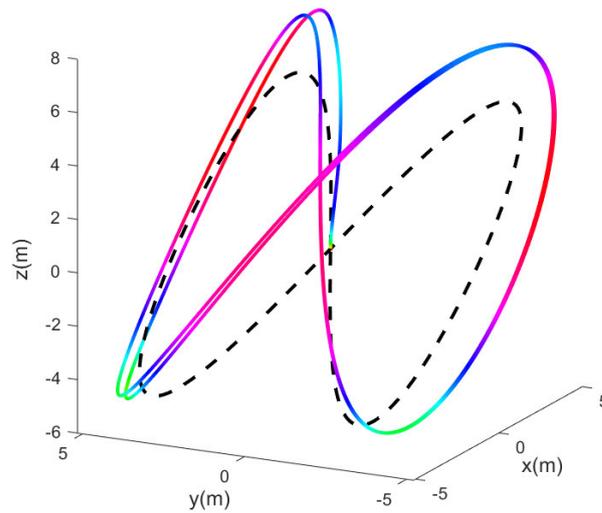


Figure 21. Simulation results of NMPC algorithm-based experiment under windy conditions. Validity of tracking control algorithm for quadcopter in balanced state.

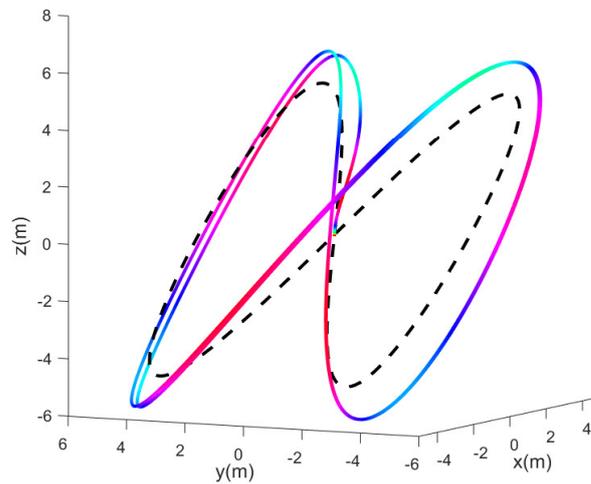


Figure 22. Simulation results of GAN-NMPC algorithm-based experiment under windy conditions. Validity of tracking control algorithm for quadcopter in balanced state.

Table 5. Tracking error statistics in cm for different wind conditions. Two metrics are considered: root mean square (RMS) and mean.

Wind	Model	PID		NMPC		GAN-NMPC	
		RMS	MEAN	RMS	MEAN	RMS	MEAN
12.1 m/s		63.7	59.4	31.4	28.7	13.9	11.2
8.5 m/s		31.6	27.2	16.3	13.9	7.3	6.3
4.2 m/s		16.2	14.6	10.7	9.9	3.7	2.9

As can be seen from Figure 23, the motor speed of the robot flying the preset trajectory is within the normal range when it is controlled by the GAN-NMPC, and there is no uncontrolled state.

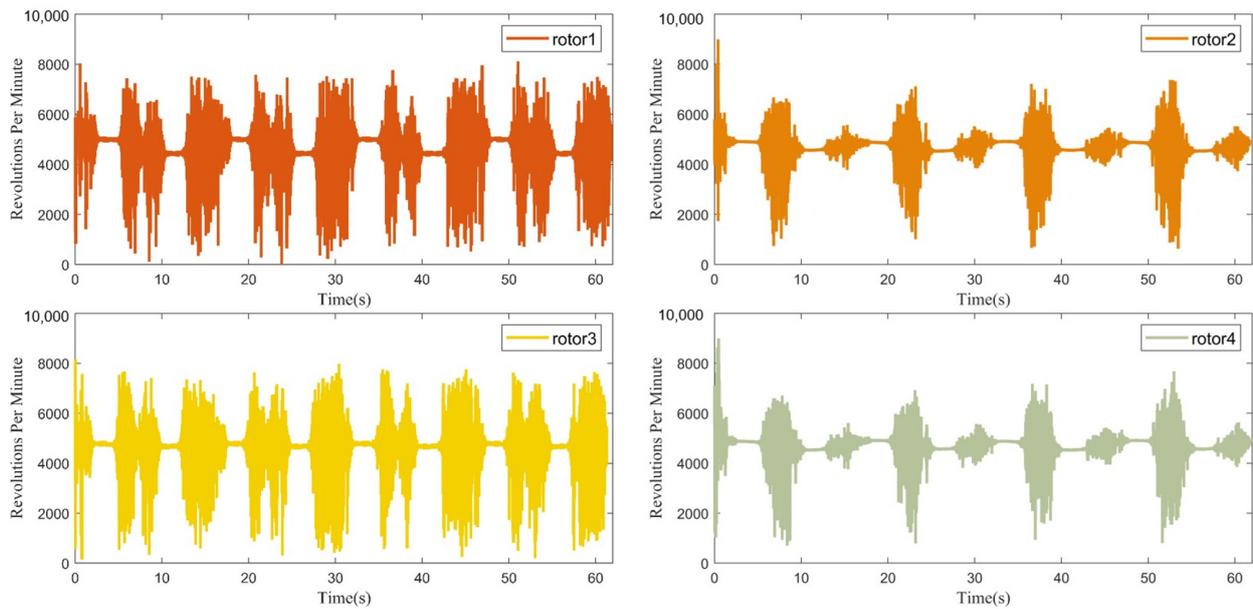


Figure 23. Motor speed in simulation.

The GAN-NMPC control is able to compensate for the internal model uncertainty parameter interference in real time, and at the same time, reduce the tracking error when wind exogenous interference is introduced, and it always maintains a good tracking performance. All these results show a clear trend: the GAN-NMPC control is significantly better than the PID control and the NMPC control.

6. Conclusions

In this paper, we first conducted a comparative analysis of existing reconfigurable quadrotors with land-air motion capabilities. The multimodal design enhances the robot's obstacle-crossing ability and expands its range of motion. Building upon this, we proposed a novel umbrella retracting mechanism that simplifies the deformation structure and incorporates a lightweight design, demonstrating a high degree of integration. The robot, weighing only 1.8 kg, achieves a flight endurance of 15 min. Additionally, we described the mechanical structure of the robot and modeled the kinematics and dynamics of the quadrotor using Newton-Euler equations. Subsequently, we conducted controller simulation experiments on the robot. By comparing the control effects of PID/NMPC/GAN-NMPC, we verified the effectiveness of the controller in stabilizing attitude and position control under the influence of wind field disturbances. In the simulated wind field environment with a speed of 12.1 m/s, the GAN-NMPC control strategy effectively maintained the error in the robot's flight trajectory within the range of 10–15 cm. The results indicate that GAN-NMPC has significant advantages over PID or NMPC, enabling the robot to perform stable high-maneuver operations in windy conditions while meeting tracking error expectations. It boasts the advantages of fast adjustment speed, minimal overshooting, and strong robustness. In future work, we plan to integrate computer vision technology with the robot, enabling it to autonomously select modes and seamlessly transition between them when navigating complex terrains. Improving the physical prototype of the reconfigurable robot and verifying the proposed algorithm on the prototype are our next main focus of research. At the same time, to ensure the robot maintains its mobility in challenging environments, such as high temperatures and humidity, we intend to incorporate a protective shell made of 3D printing material between the main components of the robot.

Author Contributions: Conceptualization, Q.C. and B.Y.; methodology, Q.C. and H.L.; software, Q.C.; validation, B.Y. and T.Y.; formal analysis, H.J. and J.Z.; investigation, X.Z.; resources, Q.C.; data curation, B.Y.; writing—original draft preparation, B.Y.; writing—review and editing, Q.C. and H.R.; supervision, H.J.; project administration, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Chunhui Project Foundation of the Education Department of China, grant number HZKY20220599, and Tianjin Research Innovation Project for Postgraduate Students, grant number: 2022SKYZ025.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fu, J.; Rota, A.; Li, S.; Zhao, J.; Liu, Q.; Iovene, E.; Ferrigno, G.; De Momi, E. Recent Advancements in Augmented Reality for Robotic Applications: A Survey. *Actuators* **2023**, *12*, 323. [\[CrossRef\]](#)
2. Dinelli, C.; Racette, J.; Escarcega, M.; Lotero, S.; Gordon, J.; Montoyaet, J.; Dunaway, C.; Androulakis, V.; Khaniani, H.; Shao, S.; et al. Configurations and applications of multi-agent hybrid drone/unmanned ground vehicle for underground environments: A review. *Drones* **2023**, *7*, 136. [\[CrossRef\]](#)
3. Savonitto, G.; Paganini, P.; Pavan, A.; Buseti, M.; Giustiniani, M.; Dal Cin, M.; Comici, C.; Kuchler, S.; Gerin, R. Aerial Drone Imaging in Alongshore Marine Ecosystems: Small-Scale Detection of a Coastal Spring System in the North-Eastern Adriatic Sea. *Remote Sens.* **2023**, *15*, 4864. [\[CrossRef\]](#)
4. Thomasberger, A.; Nielsen, M.M. UAV-Based Subsurface Data Collection Using a Low-Tech Ground-Truthing Payload System Enhances Shallow-Water Monitoring. *Drones* **2023**, *7*, 647. [\[CrossRef\]](#)
5. Liu, H.; Ge, J.; Wang, Y.; Li, J.; Ding, K.; Zhang, Z.; Guo, Z.; Li, W.; Lan, J. Multi-UAV Optimal Mission Assignment and Path Planning for Disaster Rescue Using Adaptive Genetic Algorithm and Improved Artificial Bee Colony Method. *Actuators* **2022**, *11*, 4. [\[CrossRef\]](#)
6. Choutri, K.; Lagha, M.; Meshoul, S.; Batouche, M.; Bouzidi, F.; Charef, W. Fire Detection and Geo-Localization Using UAV's Aerial Images and Yolo-Based Models. *Appl. Sci.* **2023**, *13*, 11548. [\[CrossRef\]](#)
7. Huang, Z.; Chen, G.; Shen, Y.; Wang, R.; Liu, C.; Zhang, L. An Obstacle-Avoidance Motion Planning Method for Redundant Space Robot via Reinforcement Learning. *Actuators* **2023**, *12*, 69. [\[CrossRef\]](#)
8. Naghdi, M.; Hassanalian, M. Computational Analysis of Dandelion Seeds: A Novel Flight Mechanism for Design of Efficient Bioinspired Micro Drones. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022.
9. Tu, G.-T.; Juang, J.-G. UAV Path Planning and Obstacle Avoidance Based on Reinforcement Learning in 3D Environments. *Actuators* **2023**, *12*, 57. [\[CrossRef\]](#)
10. Pan, N.; Jiang, J.; Zhang, R.; Xu, C.; Gao, F. Skywalker: A Compact and Agile Air-Ground Omnidirectional Vehicle. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2534–2541. [\[CrossRef\]](#)
11. Liu, Z.; Song, M.; Liu, Y.; Bai, B. Design, Modeling and Simulation of a Reconfigurable Land-Air Amphibious Robot. In Proceedings of the 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Jiaying, China, 27–31 July 2021.
12. Khalil, A.; Jaradat, M.A.; Mukhopadhyay, S.; Abdel-Hafez, M.F. Autonomous Control of a Hybrid Rolling and Flying Caged Drone for Leak Detection in HVAC Ducts. *IEEE/ASME Trans. Mechatron.* **2023**, *1*, 1–13. [\[CrossRef\]](#)
13. Choi, H.C.; Wee, I.; Corah, M.; Sabet, S.; Kim, T.; Touma, T.; Agha-mohammadi, A.A. Baxter: Bi-modal aerial-terrestrial hybrid vehicle for long-endurance versatile mobility. In *Experimental Robotics: The 17th International Symposium*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; Volume 19, pp. 60–72.
14. Premachandra, C.; Otsuka, M.; Gohara, R.; Ninomiya, T.; Kato, K. A study on development of a hybrid aerial/terrestrial robot system for avoiding ground obstacles by flight. *IEEE/CAA J. Autom. Sin.* **2018**, *6*, 327–336. [\[CrossRef\]](#)
15. Kim, K.; Spieler, P.; Lupu, E.S.; Ramezani, A.; Chung, S.J. A bipedal walking robot that can fly, slackline, and skateboard. *Sci. Robot.* **2021**, *6*, 59. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Jia, H.; Bai, S.; Ding, R.; Shu, J.; Deng, Y.; Khoo, B.L.; Chirarattananon, P. A quadrotor with a passively reconfigurable airframe for hybrid terrestrial locomotion. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4741–4751. [\[CrossRef\]](#)
17. David, N.B.; Zarrouk, D. Design and analysis of FCSTAR, a hybrid flying and climbing sprawl tuned robot. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6188–6195. [\[CrossRef\]](#)
18. Beck, A.T. Optimal design of redundant structural systems: Fundamentals. *Eng. Struct.* **2020**, *219*, 110542. [\[CrossRef\]](#)
19. He, Y.; Wang, W.; Wang, Q.; Wang, Y. Novel Design of the Wheel-footed obstacle-surmounting Robot. *J. Phys. Conf. Ser.* **2020**, *1550*, 022019. [\[CrossRef\]](#)
20. Tajima, Y.; Hiraguri, T.; Matsuda, T.; Imai, T.; Hirokawa, J.; Shimizu, H.; Kimura, T.; Maruta, K. Analysis of Wind Effect on Drone Relay Communications. *Drones* **2023**, *7*, 182. [\[CrossRef\]](#)

21. Sorbelli, F.B.; Corò, F.; Das, S.K.; Pinotti, C.M. Energy-constrained delivery of goods with drones under varying wind conditions. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 6048–6060. [[CrossRef](#)]
22. Wen, S.; Han, J.; Ning, Z.; Lan, Y.; Yin, X.; Zhang, J.; Ge, Y. Numerical analysis and validation of spray distributions disturbed by quad-rotor drone wake at different flight speeds. *Comput. Electron. Agric.* **2019**, *166*, 105036. [[CrossRef](#)]
23. Santoso, F.; Garratt, M.A.; Anavatti, S.G. Hybrid PD-fuzzy and PD controllers for trajectory tracking of a quadrotor unmanned aerial vehicle: Autopilot designs and real-time flight tests. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 1817–1829. [[CrossRef](#)]
24. Okyere, E.; Bousbaine, A.; Poyi, G.T.; Joseph, A.K.; Andrade, J.M. LQR controller design for quad-rotor helicopters. *J. Eng.* **2019**, *17*, 4003–4007. [[CrossRef](#)]
25. Islam, M.; Okasha, M.; Sulaeman, E. A model predictive control (MPC) approach on unit quaternion orientation based quadrotor for trajectory tracking. *Int. J. Control Autom. Syst.* **2019**, *17*, 2819–2832. [[CrossRef](#)]
26. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [[CrossRef](#)]
27. Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; Aila, T. Training generative adversarial networks with limited data. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12104–12114.
28. O’Connell, M.; Shi, G.; Shi, X.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; Chung, S.J. Neural-fly enables rapid learning for agile flight in strong winds. *Sci. Robot.* **2022**, *7*, 66. [[CrossRef](#)]
29. Li, J.; Li, Y. Dynamic analysis and PID control for a quadrotor. In Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 7–10 August 2011; pp. 573–578.
30. Zavala, V.M.; Biegler, L.T. The advanced-step NMPC controller: Optimality, stability, and robustness. *Automatica* **2009**, *45*, 86–93. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.