



OPEN

Dual-loop control and state prediction analysis of QUAV trajectory tracking based on biological swarm intelligent optimization algorithm

Zuoming Zou^{1,3}, Shuming Yang^{1,3} & Liang Zhao^{2,3}✉

Quadrotor unmanned aerial vehicles (QUAVs) have attracted significant research focus due to their outstanding Vertical Take-Off and Landing (VTOL) capabilities. This research addresses the challenge of maintaining precise trajectory tracking in QUAV systems when faced with external disturbances by introducing a robust, two-tier control system based on sliding mode technology. For position control, this approach utilizes a virtual sliding mode control signal to enhance tracking precision and includes adaptive mechanisms to adjust for changes in mass and external disruptions. In controlling the attitude subsystem, the method employs a sliding mode control framework that secures system stability and compliance with intermediate commands, eliminating the reliance on precise models of the inertia matrix. Furthermore, this study incorporates a deep learning approach that combines Particle Swarm Optimization (PSO) with the Long Short-Term Memory (LSTM) network to foresee and mitigate trajectory tracking errors, thereby significantly enhancing the reliability and safety of mission operations. The robustness and effectiveness of this innovative control strategy are validated through comprehensive numerical simulations.

Keywords Quadrotor unmanned aerial vehicles, Vertical take-off and landing, Sliding mode control, Particle swarm optimization, Long short-term memory, Data-driven

Unmanned aerial vehicles (UAVs) play a crucial role in various fields such as military reconnaissance, geographical mapping, agricultural monitoring, and disaster relief, significantly improving operational efficiency and safety. At the same time, they also demonstrate vast potential in civilian applications, including film production and courier logistics^{1–4} and⁵. QUAVs, distinguished from fixed-wing UAVs, offer the distinct capabilities of vertical take-off and landing (VTOL) and hovering, capturing significant global scholarly interest. These attributes make QUAVs particularly versatile in various applications. Additionally, like all aircraft and dynamic systems, QUAVs are subject to inevitable external disturbances, as discussed in⁶ and⁷. The flight of UAVs are often subject to various external disturbances, including but not limited to ground effect, rotor blade oscillations, gusts, and more. These external factors can significantly diminish the stability and precision of UAV flight, and may even pose a threat to the overall stability of the UAV⁸ and⁹. Consequently, the research into trajectory tracking control for UAVs, particularly in the face of such disruptive influences, has become increasingly crucial. Developing sophisticated control strategies that can mitigate these disturbances and enhance the flight performance of UAVs is therefore of paramount importance. In addition, the need for accurate and stable UAV controller design makes the development of robust control technology an important research field. Currently, there are numerous linear and non-linear methods available for trajectory tracking in UAVs, including proportional-integral-derivative (PID) control¹⁰ and¹¹, adaptive control^{12,13} and¹⁴, active disturbance rejection control (ADRC)^{15,16} and¹⁷, sliding mode control¹⁸, Reference¹⁹ and²⁰), dynamic surface control (DSC)^{21–23} and²⁴, neural network and other intelligent control methods^{25–28} and²⁹.

¹Xi'an Jiaotong University, Xi'an 710061, Shaanxi, China. ²School of Information Engineering, Yangzhou University, Yangzhou 225009, Jiangsu, China. ³These authors contributed equally: Zuoming Zou, Shuming Yang and Liang Zhao. ✉email: silence_edu1209@163.com

In¹¹, a six-degree-of-freedom QAV system is introduced, utilizing six nonlinear controllers. The design of these controllers focuses on system stabilization and precise adherence to a set trajectory, with an emphasis on reducing energy usage and minimizing tracking discrepancies. However, the influence of external disturbance on trajectory tracking control is not considered too much in this work. In³⁰, a PID control strategy is proposed to tackle stability challenges in UAV flight when faced with disturbances. The efficacy of this method is confirmed through simulations, which show stable control of roll, pitch, and yaw angles during autonomous flight. Conversely, Reference³¹ details a robust adaptive control strategy designed to manage QAVs effectively. This method assumes that disturbances are bounded and employs Lyapunov stability analysis to ensure the stability and boundedness of all closed-loop signals. Differing from the focus of most research efforts, which predominantly concentrate on leader-follower problems, in³², the focus is on situations in which leading UAVs encounter dynamic uncertainties and unanticipated external disturbances. Consequently, this paper outlines an adaptive consensus-based control methodology designed for the coordinated flight of UAV formations. Numerical simulations further demonstrate that the proposed adaptive control strategy exhibits greater robustness compared to conventional approaches. In addressing the intrinsic nonlinearity and vulnerability to external disturbances during flight of small UAV models,³³ proposes a controller utilizing piecewise constant adaptive laws. This approach is advantageous over earlier methods, such as those discussed in³⁴ and³⁵, due to its minimalistic controller design and elimination of the need for parameter tuning. Additionally, simulations confirm that the designed controller exhibits substantial robustness. In¹³, sliding mode control technology is used to tackle the issue of pose control for QAVs amidst parameter uncertainties and external disturbances. While effective, sliding mode control has limitations, such as potential chattering effects due to high-frequency switching in control signals, which can lead to increased wear and tear on system components. In¹⁸, a distributed sliding mode controller is used to achieve consensus on altitude and heading angles among a swarm of UAVs, while also employing sliding mode control for the autopilot to manage non-consensus states. Although effective, this approach has drawbacks such as potential chattering effects, which can lead to mechanical wear and increased energy consumption in UAV systems. In³⁶, a disturbance rejection control strategy utilizing Dynamic Surface Control (DSC) is introduced to facilitate precise trajectory tracking in UAVs. This approach differs from traditional inverse control methods by offering a more streamlined design process for the dynamic controller. The DSC strategy incorporates a first-order filter to derive the derivative of the virtual control, effectively addressing the issue of dimension expansion commonly encountered during differentiation. This adaptation simplifies the overall design of the control system, making it more efficient and manageable. Numerical comparisons and simulations validate the superiority of the control strategy proposed in this paper. In³⁷, a neural network-based trajectory tracking control strategy for UAVs optimizes control inputs by solving the Hamilton-Jacobi-Bellman equation, with system stability verified via Lyapunov functions, and robustness confirmed through simulations. Meanwhile,³⁸ reports that a feedforward neural network enhances MPC accuracy for quadcopters, reducing trajectory tracking errors by 40% in simulations and real-world tests compared to PID controllers. However, the complexity of neural network models can lead to challenges in real-time implementation and require significant computational resources.

So far, the main methods for predictive analysis of existing data have been time series forecasting techniques³⁹, such as exponential smoothing⁴⁰ and grey prediction models⁴¹. Moreover, various forms of regression, including linear, logistic, and nonlinear regression, are used to predict numerical outcomes by developing mathematical models. Machine learning-based predictive methods⁴², such as decision trees, random forests, and neural networks, which leverage pattern learning and correlations in the data for prediction. Additionally, there are other predictive methods like Markov prediction, ROC curves, etc. In⁴³, the challenge of achieving effective accuracy and sufficiency in long-term wind speed prediction has been addressed in this study. This method employs a combination of Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks to reduce errors and improve prediction accuracy. The results, measured through the Root Mean Square Error (RMSE) method, indicate that the LSTM model demonstrates superior predictive capabilities with lower errors compared to other models. PSO is employed to optimize the parameters or hyper-parameters of the LSTM network^{44,45}. The goal of integrating PSO with LSTM networks, known as PSO-LSTM, is to optimize the performance of LSTM in time series prediction tasks.

Drawing inspiration from the literature cited, this study's novelty lies in its approach to solving the trajectory tracking control challenges faced by UAVs under external disturbances. Different from traditional methods, this paper proposes an innovative dual-layer structured robust sliding mode control strategy that effectively integrates a position outer loop with an attitude inner loop using robust sliding mode techniques. Additionally, this study employs a data-driven deep learning strategy, constructing an LSTM network enhanced with a bio-inspired swarm intelligence optimization algorithm to predictively analyze position tracking error data. The Key contributions of this research include:

1. In the position subsystem, a virtual sliding mode control input is implemented, augmented by adaptive estimation techniques to adjust for variations in mass loads and external disturbances. Meanwhile, in the attitude subsystem, a sliding mode controller is designed to maintain stability and ensure precise tracking of reference signals, effectively operating without requiring detailed knowledge of the inertia matrix model.
2. In this study, a differential operator with an integral chain structure is employed to effectively dampen noise. The dynamic performance of the inner loop, particularly the initial attitude angle tracking errors, plays a crucial role in the overall stability of the outer loop in the combined control systems. To improve the convergence rate of the inner loop's sliding mode control, this algorithm strategically modifies the gain coefficients in the inner loop control.

3. The integration of PSO with conventional LSTM networks, known as PSO-LSTM, provides a more efficient exploration of parameter space, which improves the global search capability and robustness of the training phase, thus demonstrating clear benefits in time series prediction tasks.
4. MATLAB/Simulink is used to perform numerical simulations, which reaffirm the effectiveness and robustness of the control strategy described in this paper. These simulations serve as further validation, reinforcing the efficiency and dependability of the proposed control method.

This paper is structured as follows: “**Dynamic model of a QUAUV and algorithms introduction**” section introduces the QUAUV model and the LSTM algorithm, and outlines the control objectives. “**Design of trajectory tracking controller for quadrotor UAVs**” section details the development of control strategies for trajectory tracking, including the design of position and attitude angle tracking controllers for the quadrotor UAV using a dual-loop control approach. Section 4 presents numerical comparative simulations conducted in MATLAB/Simulink, illustrating the robustness and disturbance resistance of the proposed control strategy. Finally, Section 5 summarizes the key findings and contributions of the research.

Dynamic model of a QUAUV and algorithms introduction

Mathematical description

A QUAUV, or Quadrotor Unmanned Aerial Vehicle, typically features four rotors, each powered by an electric motor that generates lift and controls the aircraft’s flight. The key components of a QUAUV include its rotors, electric motors, a flight controller, and various sensors. The core flight mechanics of a QUAUV involve manipulating the speeds of these rotors to create lift and direct its movement. The flight controller, which receives data from sensors and inputs from a remote control, precisely adjusts the speeds of the electric motors. This adjustment allows the QUAUV to perform fundamental maneuvers such as ascending, descending, moving forward and backward, rotating, and tilting. These functionalities are discussed in detail in References^{26,46}, and⁴⁷. A schematic representation of a QUAUV is provided in Fig. 1.

The dynamics model of the quadrotor aircraft presented in Reference⁴⁸ does not account for the coupling relationships between attitude angles. However, as referenced in Reference⁴⁹, the coupling relationships between attitude angles are considered in the dynamic model. This enhanced modeling approach in Reference⁴⁹ contributes to a more comprehensive understanding of the quadrotor’s performance and control. Following the principles of Euler-Lagrange modeling, the dynamic model of the QUAUV is established as :

$$m\ddot{P}_{pos} = U_1 Re_3 - mge_3 + d_F \quad (1)$$

$$J\ddot{\Theta} = U_2 - C\dot{\Theta} + d_\Gamma \quad (2)$$

where, Eqs. (1) and (2) respectively represent the dynamic models of the position subsystem and attitude subsystem of a QUAUV. The symbol $P_{pos} = [x \ y \ z]^T$ represents the coordinates of the center of mass position of the QUAUV in the inertial coordinate system. The vector P_{pos} is expressed as a column vector with its components being the values of x , y , and z . m stands for the total payload weight, g represents the gravitational acceleration, $\Theta = [\phi \ \theta \ \psi]^T$, denotes the attitude angles; $e_3 = [0 \ 0 \ 1]^T$ represents the unit vector in the vertical direction. U_1 represents the lift, and U_2 stands for the rotational moments in the attitude subsystem. The disturbance d_F corresponds to the QUAUV disturbance force, and d_Γ represents the disturbance moments. J denotes the inertial



Figure 1. The structure diagram of a QUAUV.

tensor, and R represents the transformation from the body coordinate system to the inertial coordinate system (see⁴⁹).

$$R = \begin{bmatrix} C_\theta C_\psi & S_\theta S_\phi C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \tag{3}$$

where, C_0 represents the cosine function, and S_0 represents the sine function. These functions are commonly used in mathematical expressions to represent angles and rotations. The matrix J represents the rigid body inertia tensor, which is defined as $I = [I_{xx}, I_{yy}, I_{zz}]^T$. Reference⁴⁹ provides the representation of this tensor in the inertial coordinate system:

$$J = \begin{bmatrix} I_{xx} & 0 & -I_{xx}S_\theta \\ 0 & I_{yy}C_\phi^2 + I_{zz}S_\phi^2 & (I_{yy} - I_{zz})S_\phi C_\phi C_\theta \\ -I_{xx}S_\theta & (I_{yy} - I_{zz})S_\phi C_\phi C_\theta & I_{xx}S_\theta^2 + I_{yy}S_\phi^2 C_\theta^2 + I_{zz}C_\phi^2 C_\theta^2 \end{bmatrix} \tag{4}$$

The term C represents the components of the coriolis and centrifugal forces, and it can be calculated using the following equation (see Reference⁵⁰):

$$C = \dot{J} - \frac{1}{2} \frac{\partial}{\partial \Theta} (\dot{\Theta}^T J) \tag{5}$$

By combining Eqs. (4) and (5), it can be obtained

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{6}$$

where

$$\begin{cases} c_{11} = 0 \\ c_{12} = -I_{xx}\dot{\psi}C_\theta + (I_{yy} - I_{zz})\left(\dot{\theta}S_\phi C_\phi + \psi S_\phi^2 C_\theta - \dot{\psi}C_\phi^2 C_\theta\right) \\ c_{13} = (I_{zz} - I_{yy})\dot{\psi}S_\phi C_\phi C_\theta^2 \\ c_{21} = -c_2 \\ c_{22} = (I_{zz} - I_{yy})\dot{\phi}S_\phi C_\phi \\ c_{23} = -I_{xx}\dot{\psi}S_\theta C_\theta + I_{yy}\dot{\psi}S_\phi^2 S_\theta C_\theta + I_{zz}\dot{\psi}C_\phi^2 S_\theta C_\theta \\ c_{31} = -I_{xx}\dot{\theta}C_\theta + (I_{yy} - I_{zz})\dot{\psi}C_\theta^2 S_\phi C_\phi \\ c_{32} = (I_{zz} - I_{yy})\left(\dot{\theta}S_\phi C_\phi S_\theta + \dot{\phi}S_\phi^2 C_\theta - \dot{\phi}C_\phi^2 C_\theta\right) + I_{xx}\dot{\psi}S_\theta C_\theta - I_{yy}\dot{\psi}S_\phi^2 S_\theta C_\theta - I_{zz}\dot{\psi}C_\phi^2 S_\theta C_\theta \\ c_{33} = I_{xx}\dot{\theta}S_\theta C_\theta + I_{yy}\left(\dot{\phi}C_\phi S_\phi C_\theta^2 - \dot{\theta}S_\phi^2 C_\theta S_\theta\right) - I_{zz}\left(\dot{\phi}C_\phi S_\phi C_\theta^2 + \dot{\theta}C_\phi^2 C_\theta S_\theta\right) \end{cases} \tag{7}$$

In the model, d_F represents the vector of disturbance forces, which can be detailed as $d_F = [d_x, d_y, d_z]^T$. Correspondingly, d_Γ represents the vector of disturbance moments, expressed as $d_\Gamma = [d_\phi, d_\theta, d_\psi]^T$. These disturbances primarily arise from the airflow effects experienced by the QAVs. Within the equations Eqs. (1) and (2), the terms J and C illustrate the interdependencies among the attitude angles.

Remark 1 It's important to highlight that there is a non-linear relationship between the disturbance d_F and the control input U_1 , stability can not be ensured through sliding mode switching robust terms. Hence, an adaptive control law needs to be designed specifically for d_F . Moreover, given that the attitude subsystem does not include position variables, the control law design can be simplified by appropriately decomposing the system structure. This approach helps in managing the nonlinear relationship between the disturbance d_F and the control input U_1 through a targeted adaptive control strategy. Additionally, the absence of position variables in the attitude subsystem allows for a more streamlined control law design, facilitated by an effective system structure decomposition. This can significantly enhance the efficiency of the control process.

Direct current motor model

The relationship between the applied voltage u , representing the actual control input signal, and the rotational speed ω of the propellers on the QAVs can be approximated by the following first-order differential equation:

$$J_r \dot{\omega} = -\frac{k_r^2}{\rho} \omega - \tau_d + \frac{k_r}{\rho} u \tag{8}$$

where, J_r represents the motor's inertia constant, ρ represents the impedance of the motor, k_r denotes the torque constant, and τ_d represents the motor load. The relationship between the rotational speed ω of the propellers and the thrust generated F_i is a quadratic relationship, described by:

$$F_i = b\omega_i^2, \quad (i = 1, 2, 3, 4) \tag{9}$$

The total thrust generated by all four propellers is expressed by:

$$U_1 = \sum_{i=1}^4 F_i = \sum_{i=1}^4 b\omega_i^2 \tag{10}$$

Using the reverse torque W_i the yaw moment U_ϕ can be obtained. The control inputs for the attitude control of the QUAUVs are then represented by the vector:

$$U_2 = \begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} = \begin{bmatrix} 0 & bl & 0 & -bl \\ -bl & 0 & bl & 0 \\ -c & c & -c & c \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{11}$$

where, the variables b and c denote constants associated with aerodynamics, while l represents the radius of the QUAUV configuration. In the design of control laws, it is possible to use the lift U_1 and rotational moments U_2 as control inputs and transform them into control voltages using the relationships described above. These control voltages can then be applied in the actual flight control system.

Long short-term memory network, LSTM

Remark 2 The RNN is a specific type of neural network designed for processing sequential data. It distinguishes itself from traditional feedforward neural networks by incorporating cyclic connections, which allow it to maintain a memory state while processing sequences (refer to Fig. 2). Within the RNN family, the LSTM network stands out as a variant of deep learning neural networks and falls under the category of RNNs. Its primary purpose is to handle sequential data and is particularly effective in capturing and retaining long-term dependencies in the data. This capability addresses the issue of gradient vanishing that is encountered in conventional RNNs.

LSTM networks feature a memory cell, an essential element designed to store and retrieve information over different time intervals. This memory cell regulates information flow through a series of gating mechanisms: the input gate, the forget gate, and the output gate. These gates control the entry and exit of information, thus preserving the memory within the cell. The structure of an LSTM is illustrated in the schematic diagram referred to as Fig. 3.

In an LSTM network, the decision regarding how much information from the previous cell state should be forgotten is crucial. This decision is determined by processing the current input together with the hidden state from the prior timestep. These data points are fed into a fully connected layer, where they undergo transformation by a sigmoid activation function, resulting in the output for the forget gate. The output value of the forget gate ranges between 0 and 1, where a value of 0 indicates that all previous cell state information is forgotten, and a value of 1 indicates that all previous cell state information is retained. The equation governing the forget gate is typically given by:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{12}$$

where, W_f is the weight matrix, b_f is the bias term, h_{t-1} is the previous time step's hidden state, x_t is the current input, and σ is the sigmoid function.

In an LSTM network, determining which new information to store in the memory cell is pivotal. This decision-making process involves two key components: the input gate, which decides which portions of the memory cell are updated, and a hyperbolic tangent (tanh) layer, which creates a new candidate values vector that may be added to the memory cell. The values for both the input gate and the candidate value vector are computed based on the current input and the hidden state from the preceding time step. The equation for the input gate is as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{13}$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{14}$$

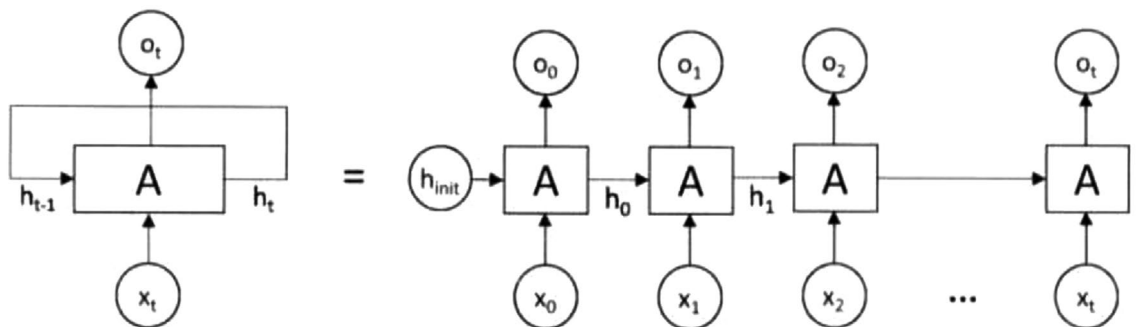


Figure 2. The schematic diagram of RNN structure.

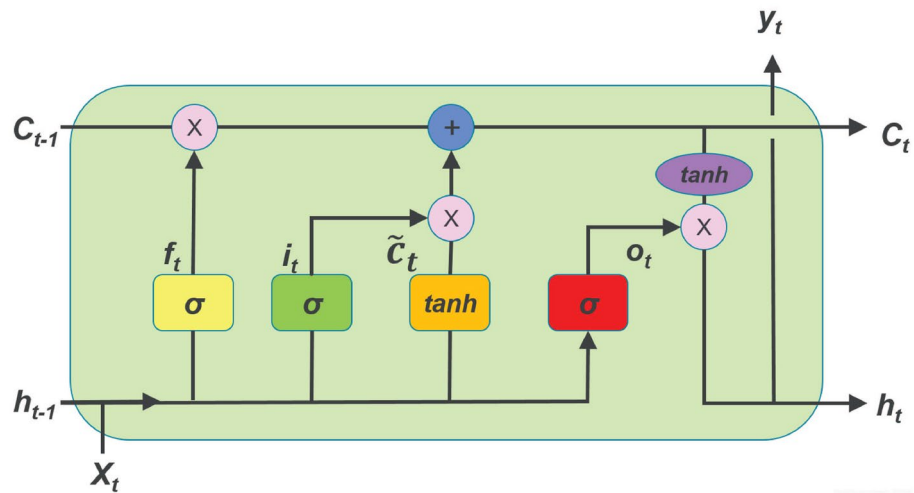


Figure 3. The structure diagram of LSTM.

where, W_C and W_i represent weight matrices, while b_C and b_i are bias terms. h_{t-1} denotes the hidden state from the previous time step, and x_t corresponds to the current input. The symbol σ represents the sigmoid function, and \tanh stands for the hyperbolic tangent function.

The memory cell in an LSTM network is updated based on the outputs of the forget gate and the input gate. The process involves multiplying the current cell state by the forget gate's output to discard parts of the existing state information deemed irrelevant. Subsequently, the product of the input gate's output and the candidate value is added to the cell state, indicating the incorporation of new relevant information into the state. The formula for updating the cell state can be expressed as:

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (15)$$

where, f_t represents the output of the forget gate, while c_{t-1} corresponds to the previous time step's cell state. i_t denotes the input gate value, and \tilde{c}_t is the candidate value.

Determining what to output based on the cell state. It takes the current input and the previous time step's hidden state, passes them through a fully connected layer, and applies a sigmoid function to obtain the output gate's value. This value ranges from 0 to 1, where 0 means no output and 1 means full output. Next, the cell state is passed through a hyperbolic tangent (\tanh) function to transform it into a value within the range of -1 to 1. This transformed value is then multiplied by the output gate's value to compute the final hidden state. The formula for the output gate can be expressed as:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (16)$$

$$h_t = o_t \times \tanh(c_t) \quad (17)$$

where, W_o represents the weight matrix, b_o corresponds to the bias term, and h_{t-1} denotes the hidden state from the previous time step.

Particle Swarm Optimization, PSO

The Particle Swarm Optimization (PSO) algorithm models the dynamics of bird flocking by employing massless particles, each characterized by two key attributes: velocity, represented by V , and position, represented by X . Velocity signifies the rate of movement, while position indicates the travel direction. Each particle autonomously explores the search space to find the optimum solution, which it records as its current personal best, denoted as P_{best} . These personal best values are exchanged among the particles across the entire swarm. The optimal personal best across the swarm is identified as the global best, represented as G_{best} . Based on these benchmarks, all particles within the swarm recalibrate their velocities and positions in relation to both their own personal best (P_{best}) and the collectively determined global best (G_{best}). Below is a flowchart depicting the steps involved in the PSO process (refer to Fig. 4).

The PSO algorithm is relatively straightforward and can be summarized into the following steps:

1. Initialize the particle swarm.
2. Evaluate particles by computing their fitness values.
3. Search for individual best (P_{best}).
4. Search for the global best (G_{best}).
5. Modify particle velocities and positions.

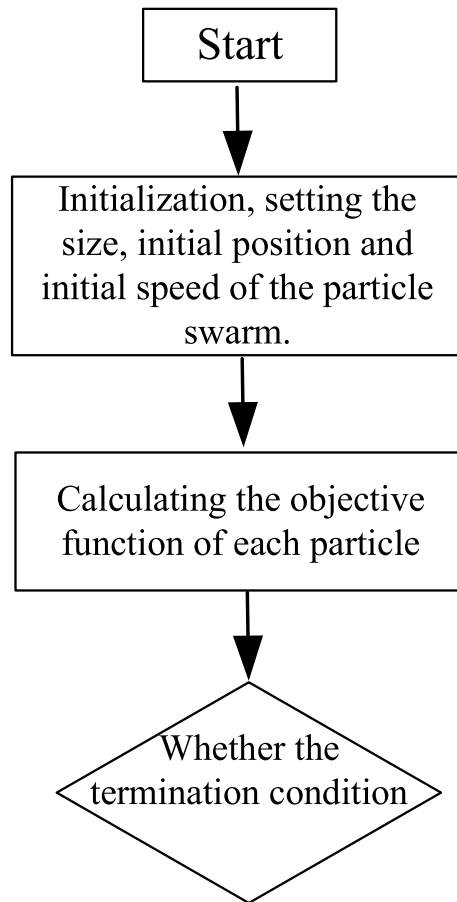


Figure 4. The flow chart of PSO algorithm.

The core idea behind PSO is the collaborative movement of particles towards better solutions. Each particle adjusts its position based on its own experience (personal best) and the collective experience of the entire swarm (global best). Through iterations, PSO aims to converge towards the optimal solution by updating particle positions and velocities according to certain rules.

The velocity and position of each particle are updated according to the following Speed update:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (Pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (Gbest - x_i(t)) \quad (18)$$

Location update:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (19)$$

where, $v_i(t)$ represents the velocity of particle i at time t , and $x_i(t)$ denotes the position of particle i at time t . The symbol w corresponds to the inertia weight, while c_1 and c_2 are learning factors. Additionally, r_1 and r_2 are random numbers.

Remark 3 PSO-LSTM (Particle Swarm Optimization-based Long Short-Term Memory) leverages the PSO algorithm to optimize LSTM network parameters for time-series prediction tasks^{44,51}. Unlike traditional LSTMs, which typically rely on optimization algorithms like gradient descent for parameter updates⁵², PSO-LSTM utilizes the global search capabilities of the PSO algorithm to find optimal parameters. This approach helps avoid the local optima pitfalls associated with gradient descent, allowing for more efficient parameter space exploration.

Control objectives

Given the underactuated nature of QUAVs, it is impractical to track and control all states directly. Consequently, in this study, we segment the entire QUAV system into inner and outer loops, adopting a dual-loop control strategy for designing the control laws. The position subsystem serves as the outer loop, while the attitude subsystem functions as the inner loop. The outer loop generates two intermediate command signals, θ_d and ϕ_d , which are then relayed to the inner loop system. The inner loop is responsible for tracking these command signals using a dedicated sliding mode control law tailored for the inner loop. Figure 5 depicts the schematic representation of the proposed closed-loop control system, demonstrating the interaction between the inner and outer loops.

Design of trajectory tracking controller for quadrotor UAVs

Position tracking controller design

Remark 4 A quadcopter, also known as a four-rotor UAV, exhibits six degrees of freedom (DOF). However, it is classified as an underactuated system because it relies on only four independent control inputs to govern its motion. This limitation prevents the comprehensive control of all state variables. In this section, we will employ a cascaded control approach, dividing the entire trajectory tracking closed-loop system into an inner-loop attitude control system and an outer-loop position subsystem. The outer loop produces two intermediate command signals, denoted as θ_d and ϕ_d . These signals are then conveyed to the inner-loop system. The inner loop, in turn, tracks these intermediate command signals by employing an inner-loop sliding mode control law, thereby effectively mitigating errors introduced by the outer loop control. This cascaded control strategy allows for precise control of the quadcopter’s attitude and position, overcoming the inherent underactuation challenge.

Assuming the desired reference position is denoted as P_d , we define the tracking error as

$$e_p = P_{pos} - P_{posd} \tag{20}$$

Using Eq. (1), we can derive the error equation for the position subsystem as follows:

$$\ddot{e}_p = \frac{1}{m}(U_p + d_F) - ge_3 - \ddot{P}_{posd} \tag{21}$$

where, $U_p = U_1 Re_3$ represents the virtual control input that needs to be designed.

We define the sliding mode function as:

$$\sigma_1 = \dot{e}_p + \lambda_1 e_p, \lambda_1 > 0 \tag{22}$$

To achieve precise tracking of the desired reference position signal, we design the virtual control law U_p for the position subsystem as follows:

$$U_p = \hat{m}\bar{U}_p - \hat{d}_F \tag{23}$$

where,

$$\bar{U}_p = ge_3 + \ddot{P}_{posd} - c_1\sigma_1 - \lambda_1\dot{e}_p \tag{24}$$

where, \hat{m} represents the estimated value of mass, \hat{d}_F represents the estimated value of external disturbance forces, and $c_1 > 0$.

Then

$$\begin{aligned} \dot{\sigma}_1 &= \ddot{e}_p + \lambda_1\dot{e}_p \\ &= \frac{1}{m}(U_p + d_F) - ge_3 - \ddot{P}_{posd} + \lambda_1\dot{e}_p \\ &= \frac{1}{m}(\hat{m}\bar{U}_p - \hat{d}_F + d_F) - ge_3 - \ddot{P}_{posd} + \lambda_1\dot{e}_p \end{aligned} \tag{25}$$

The adaptive law for designing external disturbance and quality is:

$$\begin{cases} \dot{\hat{d}}_F = \gamma_1\sigma_1 \\ \dot{\hat{m}} = -\gamma_2\sigma_1^T \bar{U}_p \end{cases} \tag{26}$$

Defining disturbance estimation error as

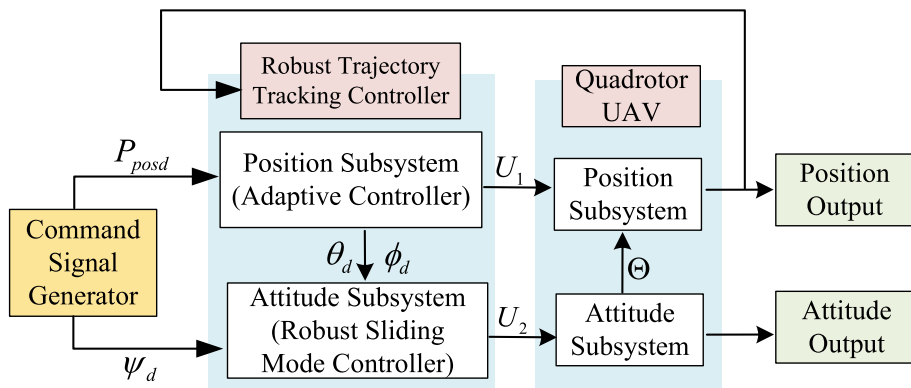


Figure 5. The response diagram of closed-loop system of a QUAV.

$$\tilde{d}_F = d_F - \hat{d}_F \tag{27}$$

and mass estimation error as

$$\tilde{m} = m - \hat{m} \tag{28}$$

For the stability analysis of the system, we select the following Lyapunov function:

$$V_1 = \frac{1}{2}\sigma_1^T\sigma_1 + \frac{1}{2m\gamma_1}\tilde{d}_F^T\tilde{d}_F + \frac{1}{2m\gamma_2}\tilde{m}^2 \tag{29}$$

As can be seen from Eq. (24):

$$-ge_3 - \ddot{P}_{posd} + \lambda_1\dot{e}_p + \bar{U}_p - c_1\sigma_1 = 0 \tag{30}$$

Then, the derivative of σ_1 can be obtained as follows:

$$\dot{\sigma}_1 = \frac{\hat{m}\bar{U}_p - \hat{d}_F + d_F}{m} - \bar{U}_p - c_1\sigma_1 \tag{31}$$

Therefore

$$\dot{V}_1 = \sigma_1^T \left(\frac{\hat{m}\bar{U}_p - \hat{d}_F + d_F}{m} - \bar{U}_p - c_1\sigma_1 \right) + \frac{1}{m\gamma_1}\tilde{d}_F^T\dot{\tilde{d}}_F + \frac{1}{m\gamma_2}\tilde{m}\dot{\tilde{m}} \tag{32}$$

Substituting the adaptive law from Eqs. (25) and (26) into Eq. (32), it can be obtained:

$$\begin{aligned} \dot{V}_1 &= -c_1\sigma_1^T\sigma_1 - \frac{\tilde{d}_F^T}{m\gamma_1}(\dot{\hat{d}}_F - \gamma_1\sigma_1) - \frac{\tilde{m}}{m\gamma_2}(\dot{\hat{m}} + \gamma_2\sigma_1^T\bar{U}_p) \\ &= -c_1\sigma_1^T\sigma_1 \leq 0 \end{aligned} \tag{33}$$

Based on the analysis provided above, it can be concluded that when $\sigma_1 \neq 0$, $\dot{V}_1 < 0$, which means that V_1 gradually decreases. In other words, both σ_1 , \tilde{d}_F , and \tilde{m} decrease over time. Only when $\sigma_1 = 0$ is reached, $\dot{V}_1 = 0$. Therefore, the sliding mode function σ_1 asymptotically converges to zero, which implies that as time approaches infinity ($t \rightarrow \infty$), σ_1 tends to zero. In the presence of a non-zero σ_1 , the system is moving towards a desired state, gradually reducing the error represented by σ_1 , \tilde{d}_F , and \tilde{m} . Only when σ_1 reaches zero, the system reaches its desired equilibrium.

Remark 5 When $\sigma_1 = 0$, \dot{V}_1 becomes zero, indicating that V_1 is stable. However, both \tilde{d}_F and \tilde{m} remain bounded but do not necessarily converge to zero as they continuously fluctuate within defined limits^{53,54}. To manage these fluctuations and prevent excessive values of \hat{m} , potentially leading to overly high thrust inputs, we propose modifying the adaptive law (Eq. (26)) and implementing discontinuous projection mapping to constrain parameter estimates within safe operational bounds.

In accordance with the adaptive law (specified in Eq. (26)), the following method of utilizing a discontinuous projection mapping is implemented

$$\dot{\hat{m}} = \text{Proj}_{\hat{m}}(-\gamma_2\sigma_1^T\bar{U}_p) \tag{34}$$

where

$$\text{Proj}_{\hat{m}}(\cdot) = \begin{cases} 0, & \text{if } \hat{m} \geq m_{\max} \text{ and } \cdot > 0 \\ 0, & \text{if } \hat{m} \leq -m_{\max} \text{ and } \cdot < 0 \\ \text{others} & \end{cases} \tag{35}$$

After obtaining the virtual control input U_p from Eq. (24), it is necessary to compute the actual thrust U_1 and the intermediate command signals for the attitude subsystem, denoted as Θ_d . We can express the virtual control input obtained from Eq. (24) in vector form as $U_p = [U_x, U_y, U_z]^T$, and define the intermediate command signals for the attitude subsystem as $\Theta_d = [\phi_d, \theta_d, \psi_d]^T$.

From the provided equations, we can deduce the relationship between the virtual control input vector U_p , the actual thrust U_1 , and the unit vector e_3 :

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = U_1 \begin{bmatrix} C_\theta C_\psi & S_\theta S_\phi C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{36}$$

where

$$U_x = U_1(C_\phi S_\theta C_\psi + S_\phi S_\psi) \tag{37}$$

$$U_y = U_1 (C_\phi S_\theta S_\psi - S_\phi C_\psi) \tag{38}$$

$$U_z = U_1 C_\phi C_\theta \tag{39}$$

Next, by substituting $U_1 = \frac{U_z}{C_\phi C_\theta}$, we can derive the following relationship

$$\begin{aligned} U_x &= \frac{U_z}{C_\phi C_\theta} (C_\phi S_\theta C_\psi + S_\phi S_\psi) = U_z (\tan \theta C_\psi + S_\psi \tan \phi \sec \theta) \\ &= U_z [C_\phi \ S_\psi] \begin{bmatrix} \tan \theta \\ \tan \phi \sec \theta \end{bmatrix} \end{aligned} \tag{40}$$

$$\begin{aligned} U_y &= \frac{U_z}{C_\phi C_\theta} (C_\phi S_\theta S_\phi - S_\phi C_\psi) = U_z (\tan \theta S_\psi - C_\psi \tan \phi \sec \theta) \\ &= U_z [S_\phi \ -C_\phi] \begin{bmatrix} \tan \theta \\ \tan \phi \sec \theta \end{bmatrix} \end{aligned} \tag{41}$$

Then

$$\begin{bmatrix} U_x \\ U_y \end{bmatrix} = U_z \begin{bmatrix} C_\psi & S_\psi \\ S_\psi & -C_\psi \end{bmatrix} \begin{bmatrix} \tan \theta \\ \tan \phi \sec \theta \end{bmatrix} \tag{42}$$

If we rewrite the intermediate command signals for the attitude subsystem Θ_d as simply Θ , we can express the virtual control input as follows:

$$\begin{bmatrix} U_x \\ U_y \end{bmatrix} = U_z \begin{bmatrix} C_{\psi_d} & S_{\psi_d} \\ S_{\psi_d} & -C_{\psi_d} \end{bmatrix} \begin{bmatrix} \tan(\theta_d) \\ \tan(\phi_d) \sec(\theta_d) \end{bmatrix} \tag{43}$$

Multiplying both sides of Eq. (43) by the matrix $\begin{bmatrix} C_{\phi_d} & S_{\phi_d} \\ S_{\phi_d} & -C_{\phi_d} \end{bmatrix}$, we obtain the relationship $U_x C_{\psi_d} + U_y S_{\psi_d} = U_z \tan(\theta_d)$. In this research, it is assumed that the ranges of θ_d and ϕ_d are within $(-\pi/2, \pi/2)$. This allows for the use of the arctangent function to compute θ_d and ϕ_d . Multiplying both sides of (Eq. (43)) by the matrix $\begin{bmatrix} C_{\psi_d} & S_{\psi_d} \end{bmatrix}$ yields the pitch angle command as

$$\theta_d = \arctan \left(\frac{U_x C_{\psi_d} + U_y S_{\psi_d}}{U_z} \right) \tag{44}$$

Similarly, multiplying both sides of Eq. (43) by the matrix $\begin{bmatrix} S_{\psi_d} & -C_{\psi_d} \end{bmatrix}$ yields the roll angle command signal

$$\phi_d = \arctan \left(C_{\theta_d} \frac{U_x S_{\psi_d} - U_y C_{\psi_d}}{U_z} \right) \tag{45}$$

Then, an actual position controller is designed as

$$U_1 = \frac{U_z}{C_{\phi_d} C_{\theta_d}} \tag{46}$$

Attitude tracking controller design

The attitude control subsystem, functioning as the inner loop, is designed to manage attitude stabilization via the internal control law. It is tasked with following the angular commands θ_d and ϕ_d , which are produced by the outer loop. The dynamics of the attitude subsystem are governed by Eq. (2). To accurately track the intermediary command signals Θ_d , the design of the control input torque vector U_2 is crucial. Additionally, the equation must incorporate the complexities of model uncertainties and external non-structured disturbance torques, which results in Eq. (2) being reformulated as follows:

$$J_0 \ddot{\Theta} + C_0 \dot{\Theta} + J_\Delta \ddot{\Theta} + C_\Delta \dot{\Theta} - U_2 - d_\Gamma = 0 \tag{47}$$

where

$$J = J_0 + J_\Delta, \quad C = C_0 + C_\Delta$$

Then

$$d_1 = d_\Gamma - J_\Delta \ddot{\Theta} - C_\Delta \dot{\Theta} \tag{48}$$

where, $\|d_1\| \leq D_1$.

The dynamic model of attitude subsystem can be further written as

$$J_0 \ddot{\Theta} + C_0 \dot{\Theta} = \Gamma + d_1 \tag{49}$$

The tracking error of the attitude subsystem is defined as follows

$$\Theta_e = \Theta - \Theta_d \quad (50)$$

$$\dot{\Theta}_r = \dot{\Theta}_d - \lambda_2 \Theta_e \quad (51)$$

where, $\Theta = [\phi \ \theta \ \psi]^T$, $\Theta_d = [\phi_d \ \theta_d \ \psi_d]^T$.

Then, a sliding mode function is introduced as

$$\sigma_2 = \dot{\Theta} - \dot{\Theta}_r = \dot{\Theta}_e + \lambda_2 \Theta_e, \lambda_2 > 0 \quad (52)$$

Therefore, the controller for the attitude subsystem is designed as

$$U_2 = J_0 \ddot{\Theta}_r + C_0 \dot{\Theta} - c_2 \sigma_2 - \eta_2 \text{sgn}(\sigma_2) \quad (53)$$

Therefore, under the attitude control law design given in Eq. (53), it is possible to achieve the tracking of the angle commands θ_d and ϕ_d generated by the outer loop.

Proof Select the following Lyapunov function

$$V_2 = \frac{1}{2} \sigma_2^T J_0 \sigma_2 \quad (54)$$

If we take the derivative of Eq. (54) with respect to time, we get:

$$\dot{V}_2 = \sigma_2^T J_0 \dot{\sigma}_2 = \sigma_2^T (U_2 + d_1 - C_0 \dot{\Theta} - J_0 \ddot{\Theta}_r) \quad (55)$$

Substituting Eq. (53) into Eq. (54), it can be obtained

$$\dot{V}_2 = -c_2 \sigma_2^T \sigma_2 - \eta_2 \|\sigma_2\| + \sigma_2^T d_1 \leq -c_2 \sigma_2^T \sigma_2 = -2c_2 J_0^{-1} V_2 \quad (56)$$

This implies that the attitude error subsystem is exponentially stable, meaning that Θ_c exponentially converges.

When considering the complete closed-loop system, which encompasses both the attitude subsystem and the position subsystem, we select the Lyapunov function for the entire closed-loop system to be

$$V_{total} = V_1 + V_2 \quad (57)$$

Then

$$\dot{V}_{total} \leq -c_1 \sigma_1^T \sigma_1 - c'_2 V_2 \leq 0 \quad (58)$$

Proof complete. \square

Remark 6 In the formulation of the control law Eq. (53), it is crucial to compute the first and second derivatives of the two intermediate command signals, θ_d and ϕ_d . To accomplish this, one can utilize a third-order finite-time sliding-mode differentiator. This method ensures accurate and robust differentiation even in the presence of system noise and disturbances, as detailed in Reference⁵⁵ and Reference⁵⁶.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \varepsilon^3 \dot{x}_3 = -2^{3/5} \cdot 4(x_1 - v(t) + (\varepsilon x_2)^{9/7})^{1/3} - 4(\varepsilon^2 x_3)^{3/5} \\ y_1 = x_2, \quad y_2 = x_3 \end{cases} \quad (59)$$

where, the input signal that requires differentiation is represented as $v(t)$, $\varepsilon = 0.04$, and x_1 is used for signal tracking, x_2 estimates the first derivative of the signal. Additionally, x_3 estimates the second derivative of the signal, with the initial values for the differentiator set as $x_1(0) = 0$, $x_2(0) = 0$, and $x_3(0) = 0$. Since this differentiator employs an integral chain structure, in practical applications for signal differentiation with noise, the presence of noise is limited to the final layer of the differentiator. This allows for more effective noise suppression through the integral action on the first derivative of the signal.

Simulation examples

A comparison of two different disturbance situations

This manuscript addresses the trajectory tracking control issue for an underactuated QUAUV by introducing a robust dual-loop sliding mode tracking control method. To demonstrate the effectiveness and capability of the proposed control method to reject disturbances, this section provides a comparative analysis across different types of disturbances, including constant and time-varying disturbances. Additionally, the performance of the open-loop system and the robustness of the proposed control strategy are validated through MATLAB/Simulink simulations. The complete diagram of the QUAUV closed-loop simulation utilizing MATLAB's S-function is illustrated in Fig. 6.

Time-varying disturbance situation

In this section, MATLAB/Simulink is utilized to perform numerical simulations to verify the efficacy of our control approach under conditions of time-varying disturbances. Throughout the simulation, the QUAUV is instructed to perform complex spiral ascent maneuvers. We set the target reference position for tracking as

$$P_d = \left[0.5 \cos\left(\frac{t}{2}\right), 0.5 \sin\left(\frac{t}{2}\right), 2 + \frac{t}{10} \right] \tag{60}$$

During the simulation, the reference yaw angle is maintained constant at $\psi_d = \frac{\pi}{3}$. The actual inertia matrix is specified as $I = \text{diag}(0.0038, 0.0038, 0.008)$, and the distance between rotors is set at $l = 0.5m$. The simulation is conducted over a period of 60s, with the mass m being adjusted at intervals of every 20s. The dynamic response of the mass over time is detailed as follows:

$$m = \begin{cases} 3.5 \text{ kg}, & 0 \leq t < 20 \text{ s} \\ 1.8 \text{ kg}, & 20 \text{ s} \leq t < 40 \text{ s} \\ 0.9 \text{ kg}, & 40 \text{ s} \leq t \leq 60 \text{ s} \end{cases} \tag{61}$$

During the flight of the QUAUV, it encounters slowly varying external aerodynamic disturbance forces and time-varying disturbance torques. These disturbances are characterized as follows:

$$d_F = [0.15 \sin(0.15\pi t), 0.15 \cos(0.15\pi t), 0.15 \cos(0.15\pi t)] \tag{62}$$

$$d_\Gamma = [0.25 \sin(0.15\pi t) + 0.15, 0.35 \cos(0.25\pi t) + 0.255, 0.55 \sin(0.15\pi t) + 0.2] \tag{63}$$

The tracking controller parameters are set as $c_1 = 4, \lambda_1 = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \gamma_1 = 0.55, \gamma_2 = 0.15$. The attitude subsystem controller parameters are defined as $c_2 = 15, \eta_2 = 0.25, \lambda_1 = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 30 \end{bmatrix}$. Besides, in the switching control,

instead of the sign function $\text{sgn}(\sigma_2)$, we utilize a saturation function $\text{sat}(\sigma_2)$, and the boundary layer thickness is configured to be 0.30. The dynamic model in Eq. (1) reveals a direct relationship between the mass, m , and the dynamic changes in position. Since the mass, m , undergoes a change every 10 seconds, this results in a discontinuity in position tracking error every 10 seconds during the simulation. In the presence of time-varying disturbance, Figs. 7, 8, 9, 10, 11, 12 display the three-dimensional position tracking and its tracking error, attitude angle tracking, control input, mass m , and its adaptive estimation.

PSO-LSTM experiment and result analysis

Simulation test and training data

In this paper, we employ a network model based on the PSO-LSTM framework. The model utilizes data derived from the position error of a VTOL system, which is controlled using the method described in this paper. The error data is computed as the square root of the difference between the ideal and actual positions. The initial seventy percent of this error dataset is allocated for training the model, while the remaining thirty percent serves as the test sample for predictive comparison and model validation.

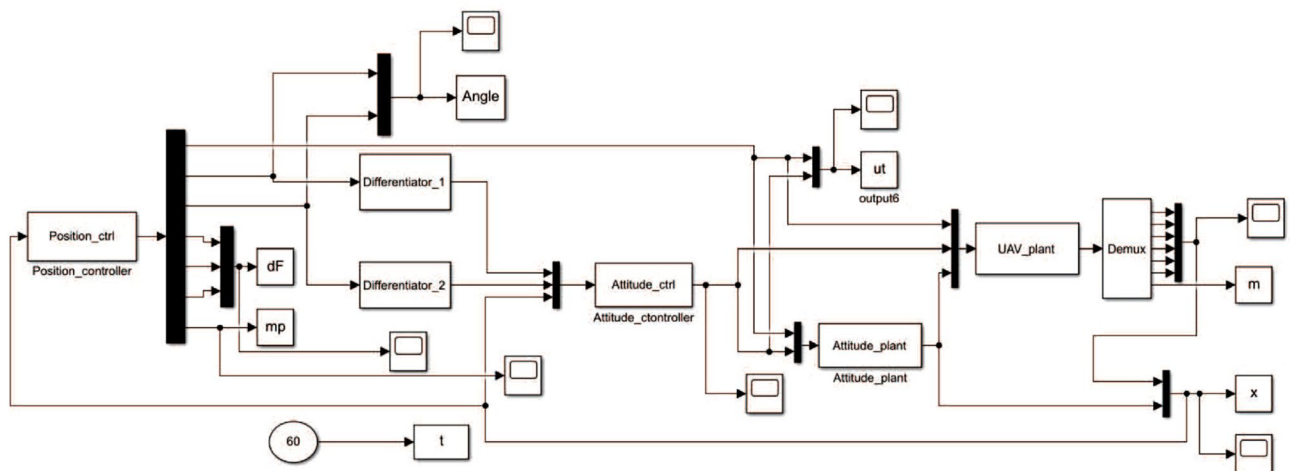


Figure 6. The diagram of closed-loop simulation of QUAUV based on MATLAB S-function.

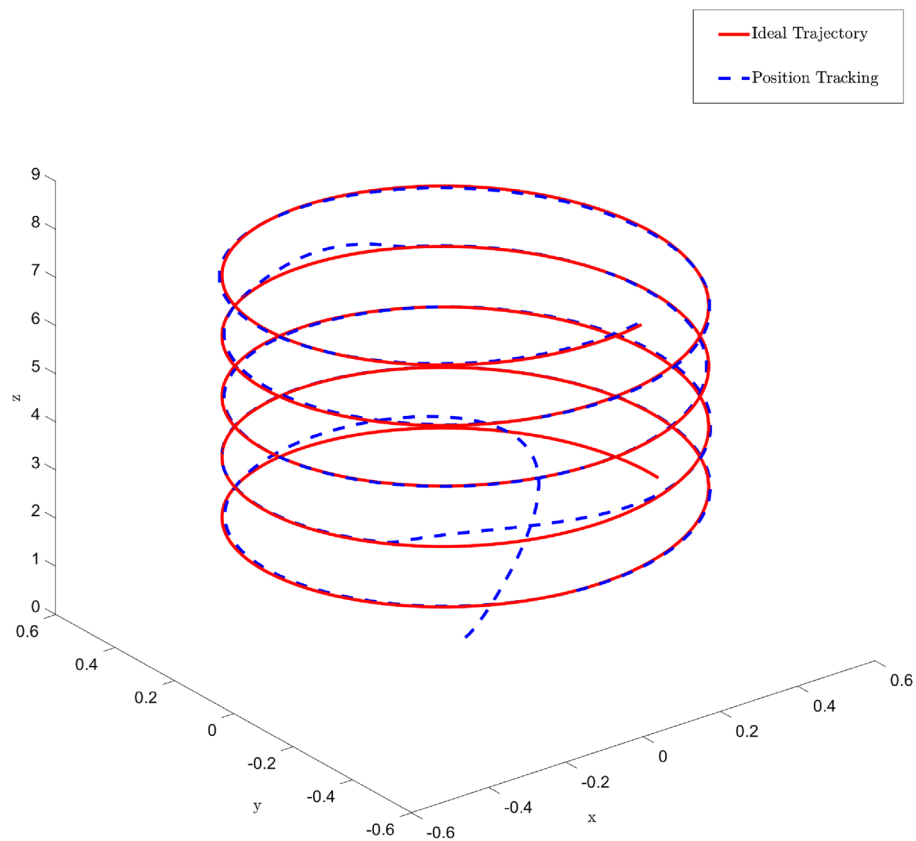


Figure 7. The three-dimensional effect diagram of position tracking under time-varying disturbance.

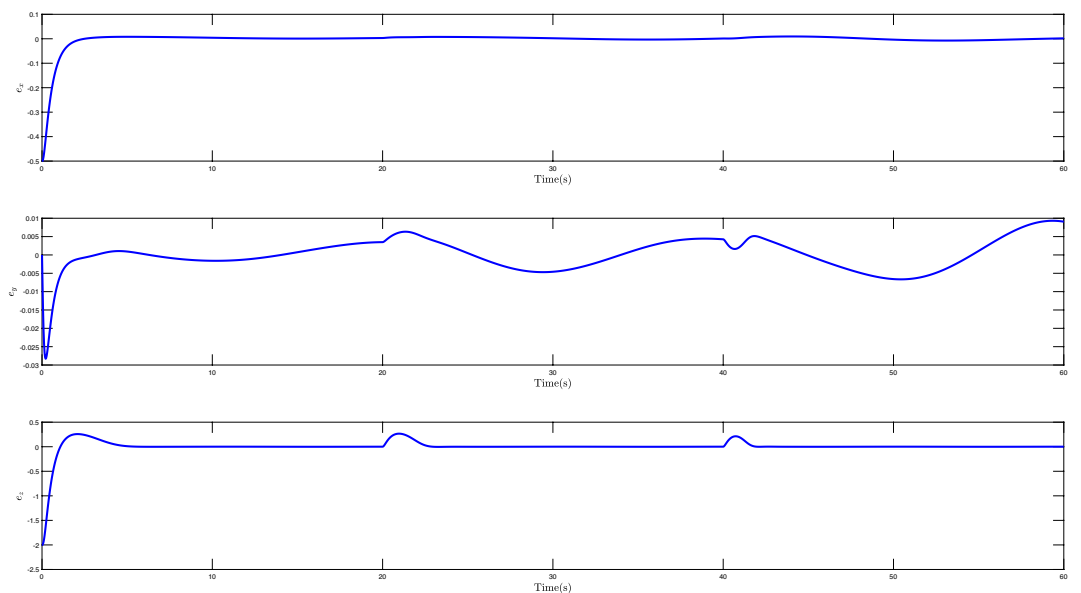


Figure 8. The response diagram of position tracking error under time-varying disturbance.

Predictive performance evaluation

Mean absolute error MAE, also called mean absolute deviation; In the calculation, the actual value and the predicted value are summed first, and then the average value is taken. The corresponding formula of MAE is as follows

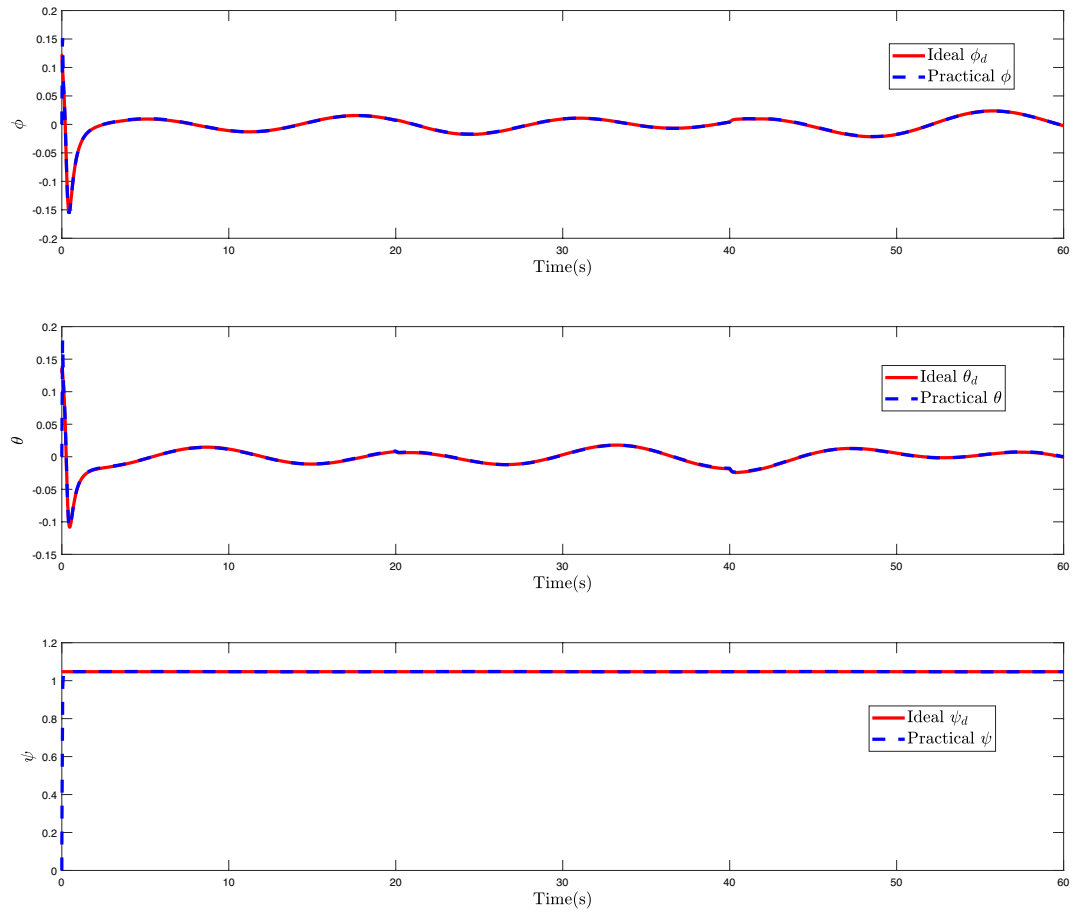


Figure 9. The response diagram of attitude angle tracking under time-varying disturbance.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \tag{64}$$

The average absolute percentage error is an improvement of MAE, which avoids the influence of data range by calculating the error percentage between the real value and the prediction. MAPE calculation formula is

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \tag{65}$$

The root mean square error is the square root of the ratio of the deviation between the predicted value and the real value to the observation times n , RMSE calculation formula is

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \tag{66}$$

where, y_i and \hat{y}_i represent the actual and predicted values of the VTOL position error.

Constant disturbance situation

In this section, we modify the disturbance scenario by assuming that the disturbance remains constant. Under this assumption, and while keeping other simulation conditions the same, we further validate the anti-disturbance capabilities of the proposed method. The specific constant disturbance applied in this section is detailed as follows:

$$d_F = [0.35, 0.35, 45], d_\Gamma = [0.25, 0.35, 0.15]$$

Under constant disturbance, position and attitude trajectory tracking and its tracking error, mass m and its adaptive estimation are shown in Figs. 13, 14, 15, 16.

The three-dimensional diagram of position tracking of QUAUV shown in Fig. 7. Figure 8 shows the position tracking error of QUAUV under time-varying disturbance. From Fig. 7, the red solid line represents the expected

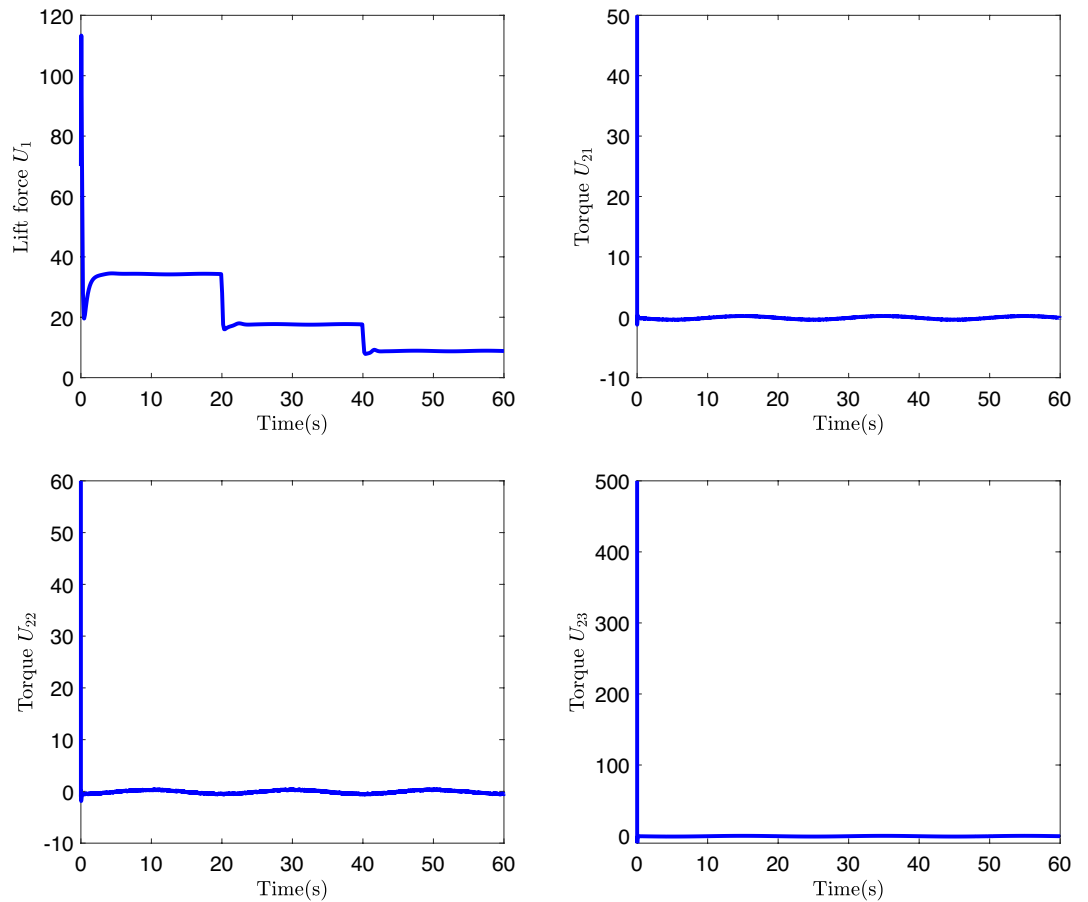


Figure 10. The response of control input under time-varying disturbance.

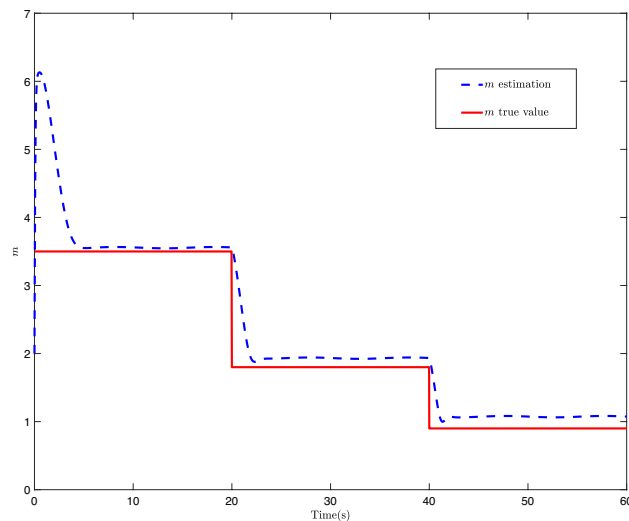


Figure 11. The response of mass m and its adaptive estimation under time-varying disturbance.

trajectory position, and the blue dashed line represents the actual position. From Figs. 7 and 8, it is evident that, even in the presence of time-varying disturbances and the control method described in this paper, precise tracking of the position trajectory can be achieved. Attitude angle tracking of QUAV under time-varying disturbance is shown in Fig. 9. It can be seen from Fig. 9 that the attitude angle of QUAV can accurately track the ideal signal under time-varying disturbance. The response of control input under time-varying disturbance as shown in Fig. 10. Figure 10 demonstrates that the system control input remains stable when utilizing the control method

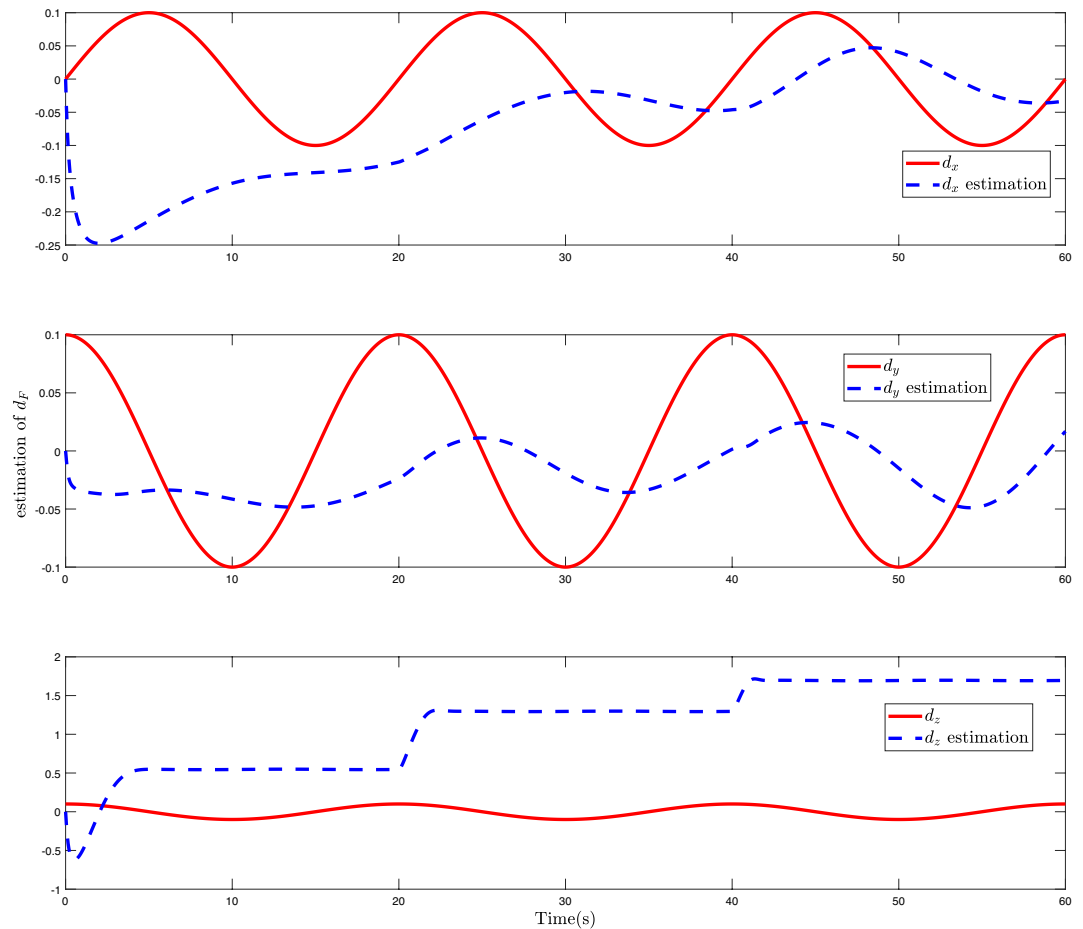


Figure 12. The response of disturbance torque and its adaptive estimation under time-varying disturbance.

outlined in this paper. The response of mass m and its adaptive estimation under time-varying disturbance can be seen in Fig. 11, as shown in Fig. 11, it can be seen from the dynamic model Eq. (1) that the mass m is directly related to the dynamic change of position. Because the mass m changes every 20 seconds with time, the position tracking error jumps every 20 seconds in the simulation. Furthermore, the response of disturbance torque and its adaptive estimation under time-varying disturbance is depicted in Fig. 12. From Figs. 11 and 12, it is evident that σ_1 converges to zero, but \tilde{d}_F and \tilde{m} do not converge to zero, which aligns with the analysis provided earlier.

Just like Figs. 7, 8, 9 and 11, the three-dimensional position tracking and its error under a constant disturbance are depicted in Figs. 13 and 14. From these two figures, it is evident that, under the control method described in this paper, the QUAUV's position can be accurately tracked, irrespective of whether the disturbance is time-varying or constant. The response diagram in Fig. 15 and the adaptive estimation diagram in Fig. 16 depict the tracking of the attitude angle in the presence of a constant disturbance for a system with mass m observing these figures, it is evident that the control method proposed in this paper effectively tracks the attitude angle even in the presence of a constant disturbance. In addition, the mass m can be estimated adaptively under the step jump every 20 seconds.

Figure 17 shows a sample fitting graph for the overall QUAUV position error data. The root mean square error (RMSE) for the linear predictive fitting of the entire dataset is 0.72696, as observed in Fig. 17. Figures 18 and 19 depict the data fitting graphs for two different network models (LSTM and PSO-LSTM), showcasing the training set error prediction and corresponding error graphs. Figure 18 highlights that the RMSE for the PSO-LSTM model is significantly lower, indicating superior predictive performance compared to the traditional LSTM model. Additionally, Fig. 19 demonstrates that although both models are capable of performing the prediction task, the PSO-LSTM model exhibits markedly better predictive performance and effectively addresses the issue of data overfitting. Furthermore, Figs. 20 and 21 illustrate the fitting effect of the testing set. Figure 20 shows that the RMSE of the test set data using the PSO-LSTM model is lower than that obtained with the LSTM model, indicating better performance of the PSO-LSTM model on the test set. Figure 21 presents the prediction display for the QUAUV position error test set under both network models (PSO-LSTM and LSTM). It is evident from Fig. 21 that the PSO-LSTM-based network model demonstrates better prediction performance and greater robustness compared to the traditional LSTM model, making it more effective for this prediction task.

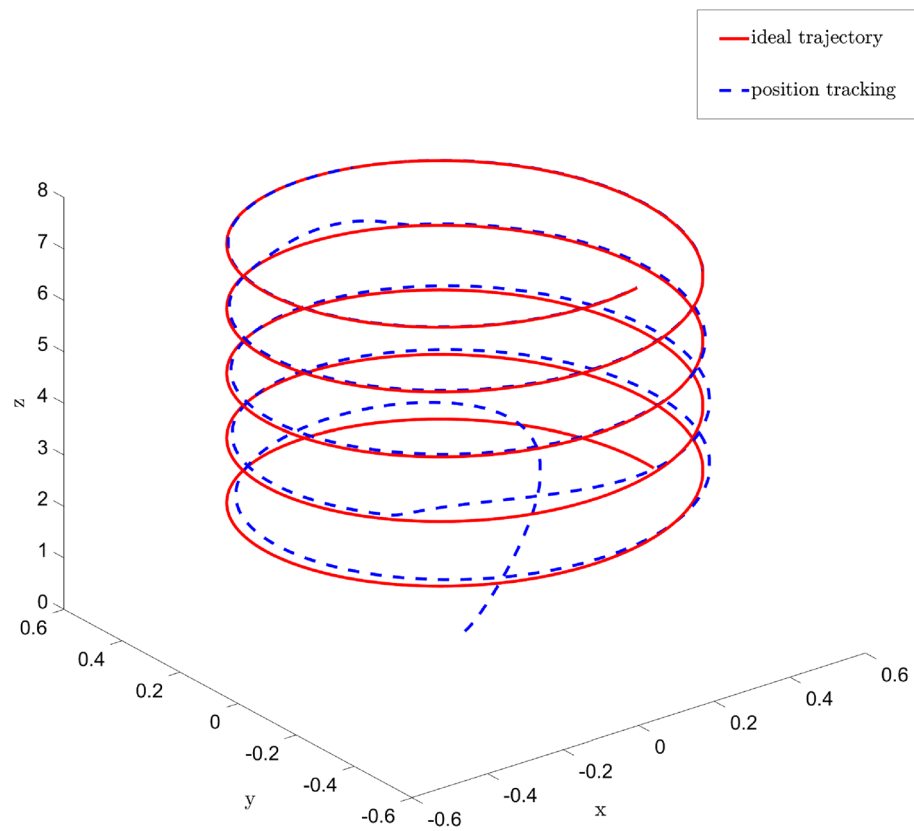


Figure 13. The three-dimensional effect diagram of position tracking under constant disturbance.

Conclusion

This paper addresses the high-precision tracking control problem of QUAVs under external disturbances using a dual-layer robust sliding mode control strategy. Additionally, a data-driven approach based on deep learning is employed to construct a PSO-LSTM neural network model for the analysis and prediction of VTOL position error data. The incorporation of PSO enables global optimization in the search space, facilitating the acceleration of the network training convergence process. This capability aids in better tuning network parameters, enhancing the overall performance of the model. Furthermore, the PSO-LSTM model may exhibit superior generalization capabilities, improving its predictive performance on new data. The numerical simulation based on MATLAB further verifies the effectiveness and robustness of the proposed control method and algorithm model.

Future research endeavors should continue building upon this foundation to further enhance their capabilities, versatility, and safety across various applications, fostering continuous innovation in the field of QUAV systems.

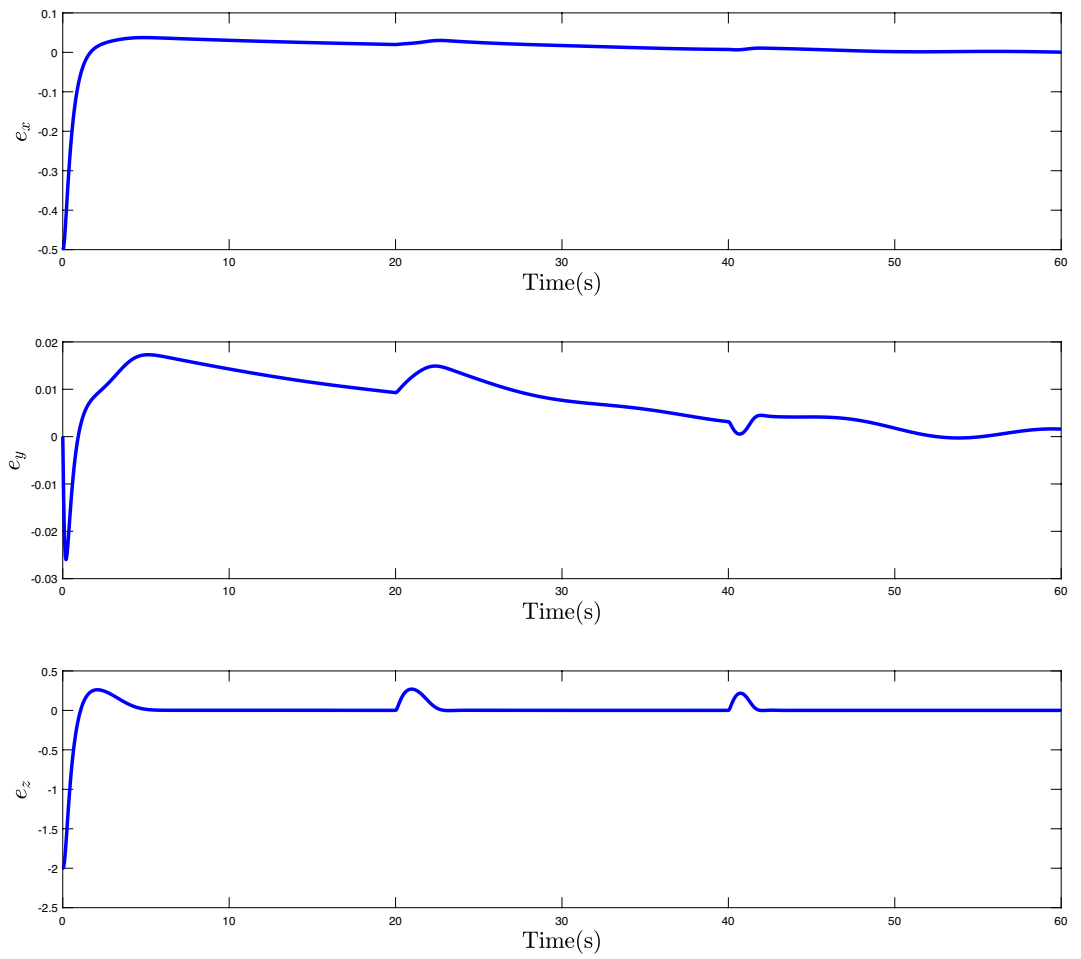


Figure 14. The response diagram of position tracking error under constant disturbance.

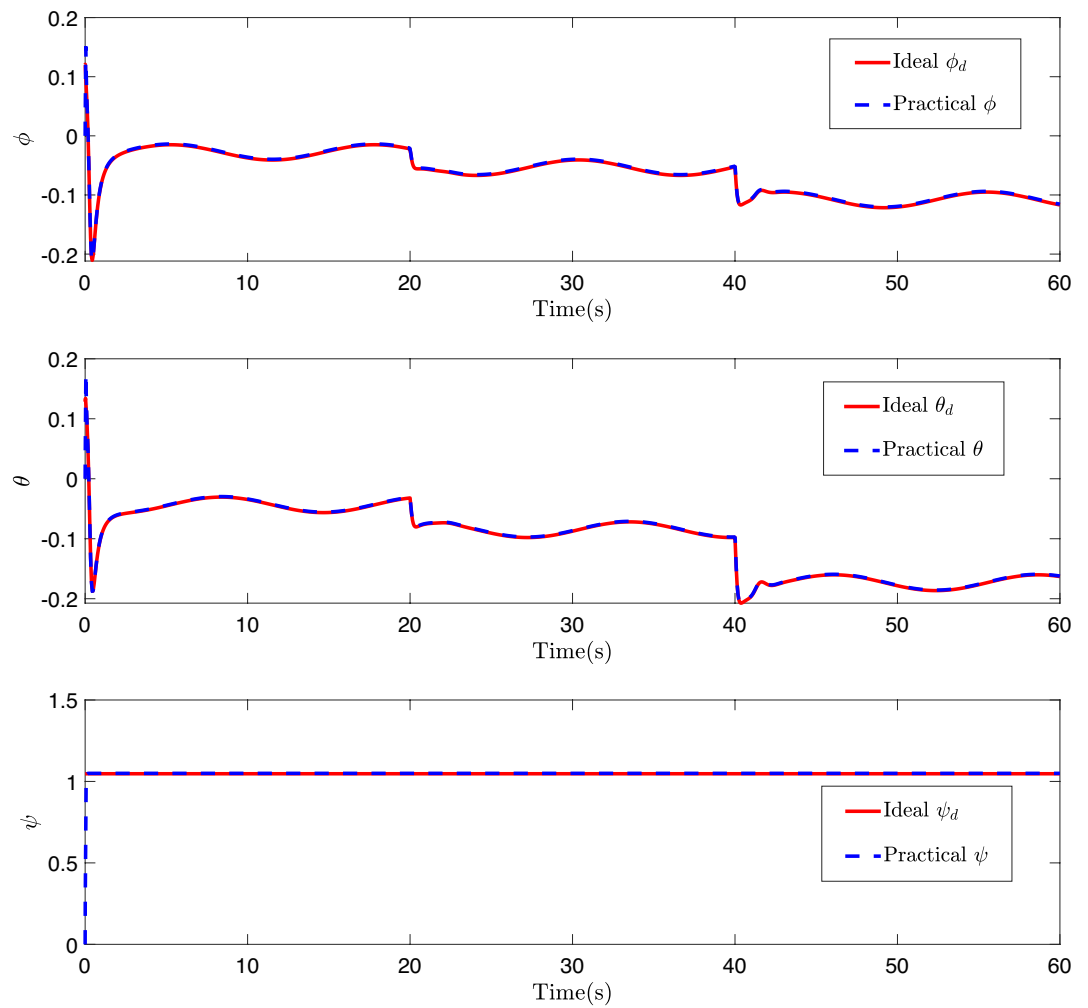


Figure 15. The response diagram of attitude angle tracking under constant disturbance.

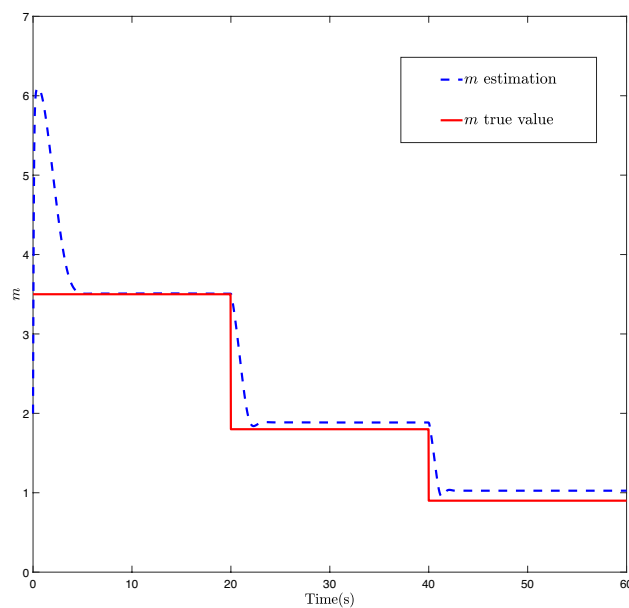


Figure 16. The response of mass m and its adaptive estimation under constant disturbance.

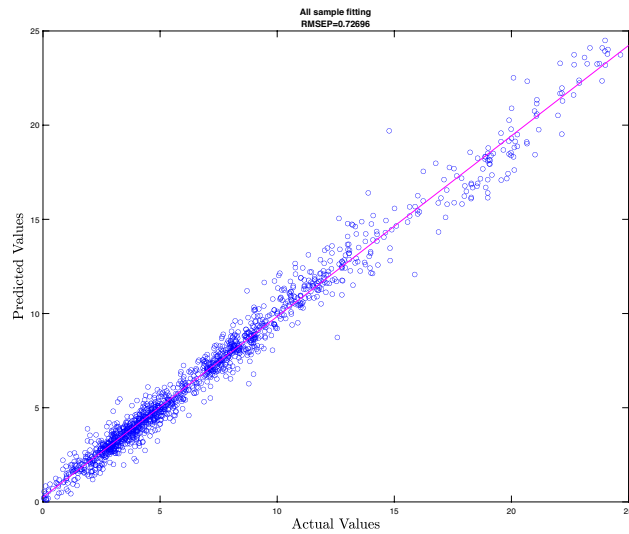


Figure 17. A fitting forecast display chart of all samples.

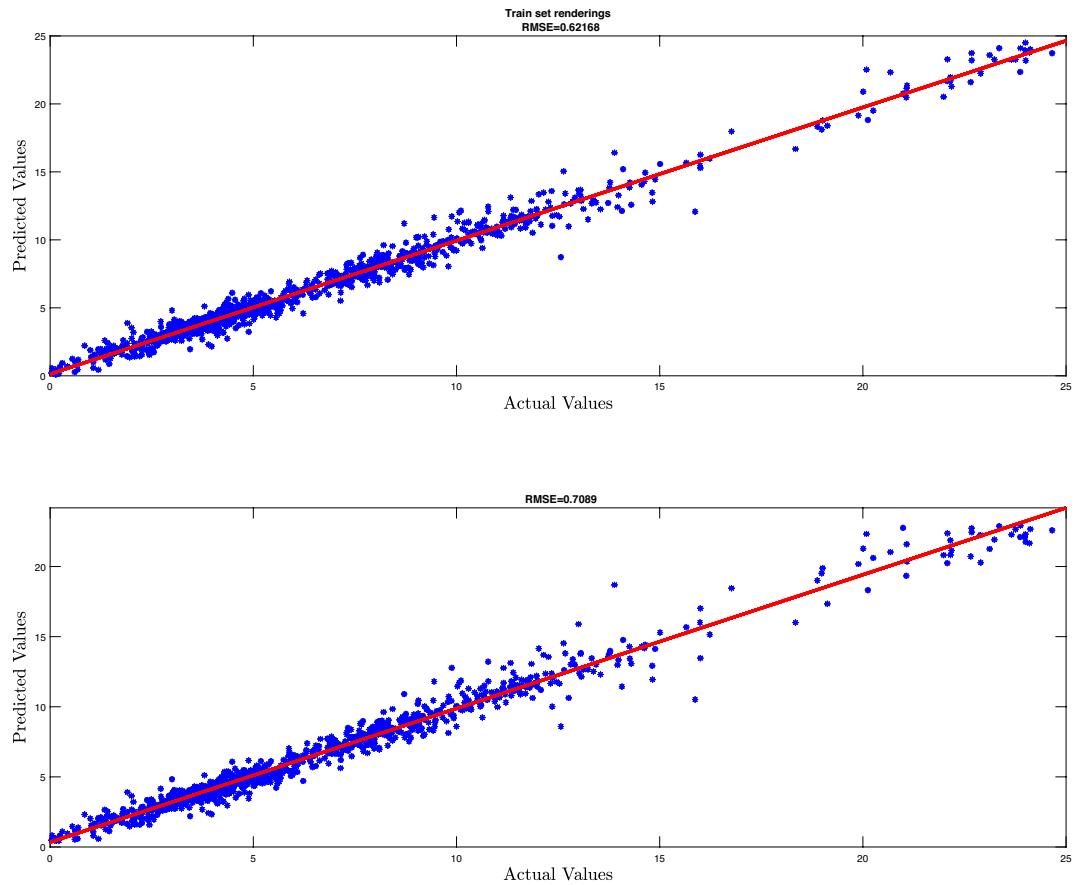


Figure 18. The diagram of training set fitting effect.

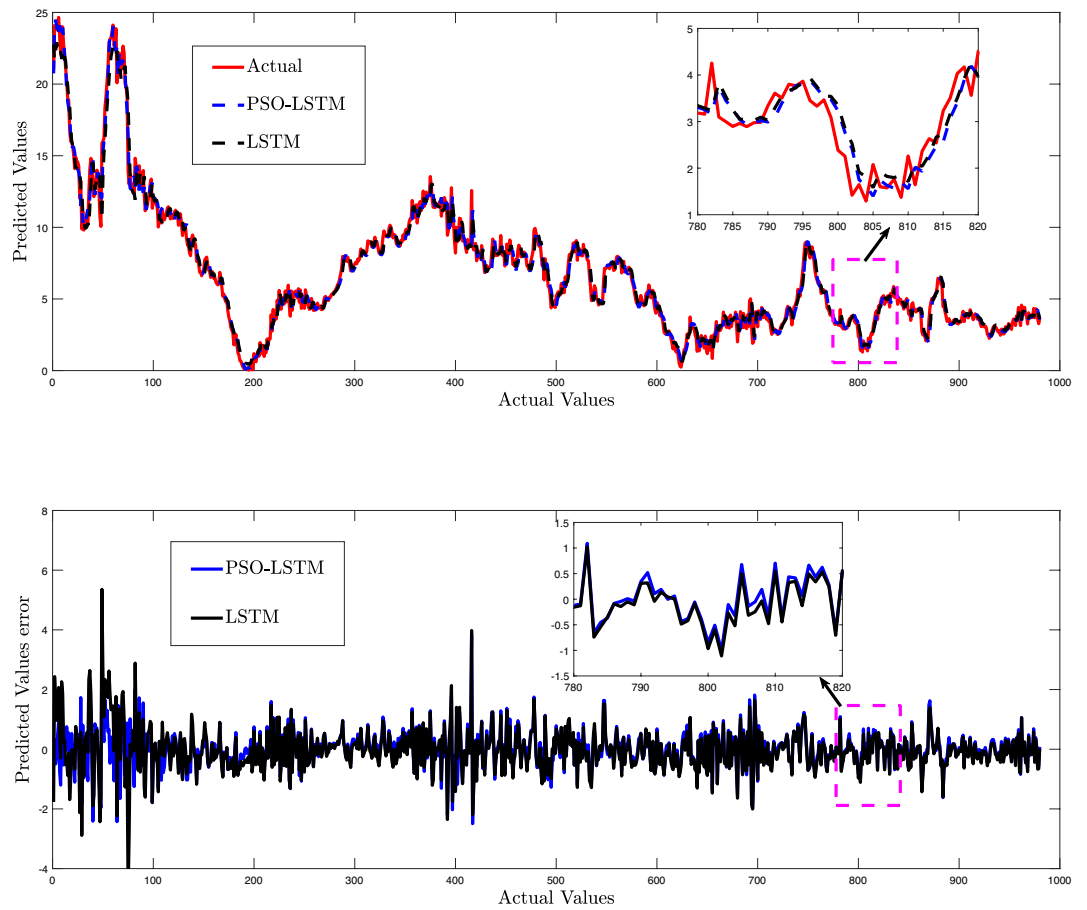


Figure 19. The prediction of training set of position error data and its error.

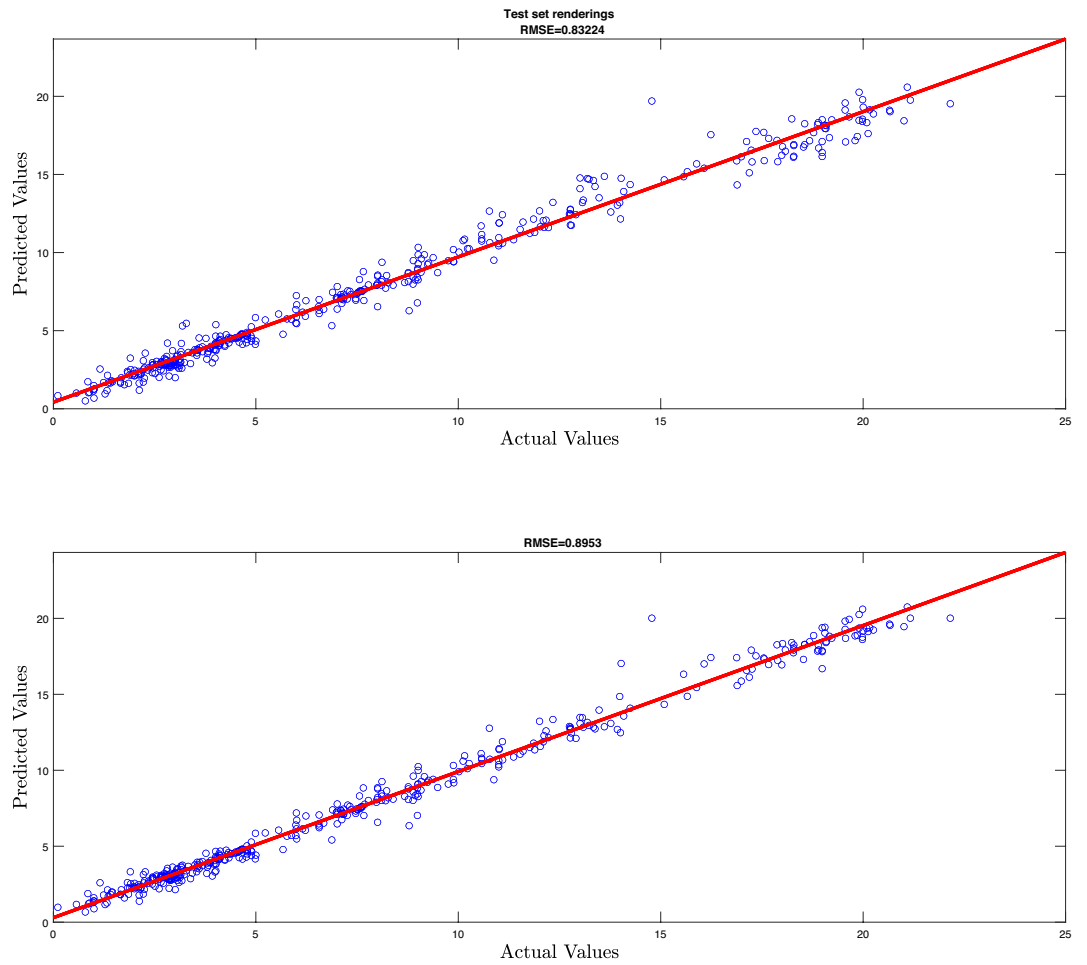


Figure 20. The diagram of testing set fitting effect.

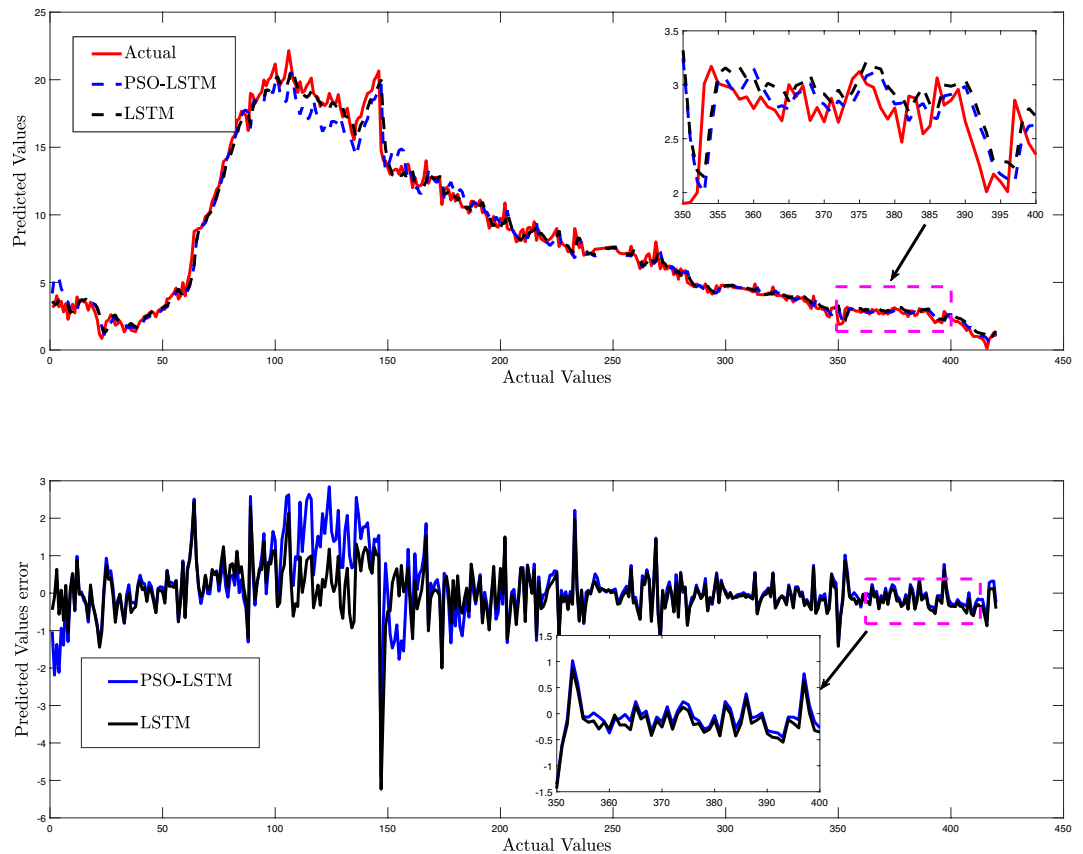


Figure 21. The prediction of testing set of position error data and its error.

Data availability

The data set needed in this paper is supplemented (Supplementary Information) in the attachment, and other simulation programs can be provided upon reasonable request by contacting the correspondent.

Received: 25 December 2023; Accepted: 9 August 2024

Published online: 17 August 2024

References

- Fahlstrom, P. G., Gleason, T. J. & Sadraey, M. H. *Introduction to UAV Systems* (Wiley, 2022).
- Fan, B., Li, Y., Zhang, R. & Fu, Q. Review on the technological development and application of UAV systems. *Chin. J. Electron.* **29**, 199–207 (2020).
- Geng, R., Ji, R. & Zi, S. Research on task allocation of UAV cluster based on particle swarm quantization algorithm. *Math. Biosci. Eng.* **20**, 18–33 (2022).
- Gupta, L., Jain, R. & Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutorials* **18**, 1123–1152 (2015).
- Radzki, G., Bocewicz, G., Wikarek, J., Nielsen, P. & Banaszak, Z. Comparison of exact and approximate approaches to UAVs mission contingency planning in dynamic environments. *Math. Biosci. Eng.* **19**, 7091–7121 (2022).
- Wei, Y., Zhang, Y. & Hang, B. Construction and management of smart campus: Anti-disturbance control of flexible manipulator based on PDE modeling. *Math. Biosci. Eng.* **20**, 14327–14352 (2023).
- Hang, B., Su, B. & Deng, W. Adaptive sliding mode fault-tolerant attitude control for flexible satellites based on ts fuzzy disturbance modeling. *Math. Biosci. Eng.* **20**, 12700–12717 (2023).
- Jiang, G. & Voyles, R. A nonparallel hexrotor UAV with faster response to disturbances for precision position keeping. In *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, 1–5 (IEEE, 2014).
- Altan, A. & Hacıoğlu, R. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mech. Syst. Signal Process.* **138**, 106548 (2020).
- Kada, B. & Ghazzawi, Y. Robust PID controller design for an UAV flight control system. In *Proceedings of the World Congress on Engineering and Computer Science Vol. 2*, 1–6 (2011).
- Najm, A. A. & Ibraheem, I. K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Eng. Sci. Technol. Int. J.* **22**, 1087–1097 (2019).
- Rosales, C., Soria, C. M. & Rossomando, F. G. Identification and adaptive PID control of a hexacopter UAV based on neural networks. *Int. J. Adapt. Control Signal Process.* **33**, 74–91 (2019).
- Noordin, A., Mohd Basri, M. A., Mohamed, Z. & Mat Lazim, I. Adaptive PID controller using sliding mode control approaches for quadrotor UAV attitude and position stabilization. *Arab. J. Sci. Eng.* **46**, 963–981 (2021).
- Yadav, A. K. & Gaur, P. Ai-based adaptive control and design of autopilot system for nonlinear UAV. *Sadhana* **39**, 765–783 (2014).
- Lotufo, M. A., Colangelo, L., Perez-Montenegro, C., Canuto, E. & Novara, C. UAV quadrotor attitude control: An ADRC-EMC combined approach. *Control. Eng. Pract.* **84**, 13–22 (2019).

16. Niu, T., Xiong, H. & Zhao, S. Based on ADRC UAV longitudinal pitching angle control research. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, 21–25 (IEEE, 2016).
17. Gao, T. Y., Wang, D. D., Tao, F. & Ge, H. L. Control of small unconventional UAV based on an on-line adaptive ADRC system. *Appl. Mech. Mater.* **494**, 1206–1211 (2014).
18. Rao, S. & Ghose, D. Sliding mode control-based autopilots for leaderless consensus of unmanned aerial vehicles. *IEEE Trans. Control Syst. Technol.* **22**, 1964–1972 (2013).
19. Wang, J., Han, L., Dong, X., Li, Q. & Ren, Z. Distributed sliding mode control for time-varying formation tracking of multi-UAV system with a dynamic leader. *Aerosp. Sci. Technol.* **111**, 106549 (2021).
20. Razmi, H. & Afshinfar, S. Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV. *Aerosp. Sci. Technol.* **91**, 12–27 (2019).
21. Qian, M., Jiang, B. & Xu, D. Fault tolerant tracking control scheme for UAV using dynamic surface control technique. *Circuits Syst. Signal Process.* **31**, 1713–1729 (2012).
22. Shen, Z., Li, F., Cao, X. & Guo, C. Prescribed performance dynamic surface control for trajectory tracking of quadrotor UAV with uncertainties and input constraints. *Int. J. Control* **94**, 2945–2955 (2021).
23. Shao, X., Liu, J., Cao, H., Shen, C. & Wang, H. Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer. *Int. J. Robust Nonlinear Control* **28**, 2700–2719 (2018).
24. Lin, X., Yu, Y. & Sun, C.-Y. A decoupling control for quadrotor UAV using dynamic surface control and sliding mode disturbance observer. *Nonlinear Dyn.* **97**, 781–795 (2019).
25. Gu, W., Valavanis, K. P., Rutherford, M. J. & Rizzo, A. UAV model-based flight control with artificial neural networks: A survey. *J. Intell. Robot. Syst.* **100**, 1469–1491 (2020).
26. Dierks, T. & Jagannathan, S. Output feedback control of a quadrotor UAV using neural networks. *IEEE Trans. Neural Netw.* **21**, 50–66 (2009).
27. Kyrkou, C., Plastiras, G., Theocharides, T., Venieris, S. I. & Bouganis, C.-S. Dronet: Efficient convolutional neural network detector for real-time UAV applications. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 967–972 (IEEE, 2018).
28. Iannace, G., Ciaburro, G. & Trematerra, A. Fault diagnosis for UAV blades using artificial neural network. *Robotics* **8**, 59 (2019).
29. Jiang, F., Pourpanah, F. & Hao, Q. Design, implementation, and evaluation of a neural-network-based quadcopter UAV system. *IEEE Trans. Ind. Electron.* **67**, 2076–2085 (2019).
30. Susanto, T. *et al.* Application of unmanned aircraft PID control system for roll, pitch and yaw stability on fixed wings. In *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITTE)*, 186–190 (IEEE, 2021).
31. Bialy, B. J., Klotz, J., Brink, K. & Dixon, W. E. Lyapunov-based robust adaptive control of a quadrotor UAV in the presence of modeling uncertainties. In *2013 American Control Conference*, 13–18 (IEEE, 2013).
32. Zhen, Z., Tao, G., Xu, Y. & Song, G. Multivariable adaptive control based consensus flight control system for UAVs formation. *Aerosp. Sci. Technol.* **93**, 105336 (2019).
33. Capello, E., Guglieri, G., Quagliotti, F. & Sartori, D. Design and validation of an adaptive controller for mini-UAV autopilot. *J. Intell. Robot. Syst.* **69**, 109–118 (2013).
34. Rothe, J., Zevering, J., Strohmeier, M. & Montenegro, S. A modified model reference adaptive controller (M-MRAC) using an updated MIT-rule for the altitude of a UAV. *Electronics* **9**, 1104 (2020).
35. Mo, H. & Farid, G. Nonlinear and adaptive intelligent control techniques for quadrotor UAV—a survey. *Asian J. Control* **21**, 989–1008 (2019).
36. Zhang, Y., Chen, Z. & Sun, M. Trajectory tracking control for a quadrotor unmanned aerial vehicle based on dynamic surface active disturbance rejection control. *Trans. Inst. Meas. Control.* **42**, 2198–2205 (2020).
37. Nodland, D., Zargarzadeh, H. & Jagannathan, S. Neural network-based optimal adaptive output feedback control of a helicopter UAV. *IEEE Trans. Neural Netw. Learn. Syst.* **24**, 1061–1073 (2013).
38. Jiang, B. *et al.* Neural network based model predictive control for a quadrotor UAV. *Aerospace* **9**, 460 (2022).
39. Lim, B. & Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A* **379**, 20200209 (2021).
40. Hyndman, R., Koehler, A. B., Ord, J. K. & Snyder, R. D. *Forecasting with Exponential Smoothing: The State Space Approach* (Springer Science & Business Media, 2008).
41. Hsu, C.-C. & Chen, C.-Y. Applications of improved grey prediction model for power demand forecasting. *Energy Convers. Manag.* **44**, 2241–2249 (2003).
42. Tercan, H. & Meisen, T. Machine learning and deep learning based predictive quality in manufacturing: A systematic review. *J. Intell. Manuf.* **33**, 1879–1905 (2022).
43. Elsaraiti, M. & Merabet, A. A comparative analysis of the ARIMA and LSTM predictive models and their effectiveness for predicting wind speed. *Energies* **14**, 6782 (2021).
44. Gundu, V. & Simon, S. P. PSO-LSTM for short term forecast of heterogeneous time series electricity price signals. *J. Ambient. Intell. Humaniz. Comput.* **12**, 2375–2385 (2021).
45. Jiao, X., Song, Y., Kong, Y. & Tang, X. Volatility forecasting for crude oil based on text information and deep learning PSO-LSTM model. *J. Forecast.* **41**, 933–944 (2022).
46. Ryll, M., Bühlhoff, H. H. & Giordano, P. R. A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Trans. Control Syst. Technol.* **23**, 540–556 (2014).
47. Idrissi, M., Salami, M. & Annaz, F. A review of quadrotor unmanned aerial vehicles: Applications, architectural design and control algorithms. *J. Intell. Robot. Syst.* **104**, 22 (2022).
48. Lee, D., Nataraj, C., Burg, T. C. & Dawson, D. M. Adaptive tracking control of an underactuated aerial vehicle. In *Proceedings of the 2011 American Control Conference*, 2326–2331 (IEEE, 2011).
49. Raffo, G. V., Ortega, M. G. & Rubio, F. R. An integral predictive/nonlinear h_∞ control structure for a quadrotor helicopter. *Automatica* **46**, 29–39 (2010).
50. Das, A., Lewis, F. & Subbarao, K. Backstepping approach for controlling a quadrotor using lagrange form dynamics. *J. Intell. Robot. Syst.* **56**, 127–151 (2009).
51. Sohane, A. & Agarwal, R. A single platform for classification and prediction using a hybrid bioinspired and deep neural network (PSO-LSTM). *Mapan* **37**, 47–58 (2022).
52. Lu, P. *et al.* A time series image prediction method combining a CNN and LSTM and its application in typhoon track prediction. *Math. Biosci. Eng.* **19**, 12260–12278 (2022).
53. Zhang, Y., Chen, Z., Zhang, X., Sun, Q. & Sun, M. A novel control scheme for quadrotor UAV based upon active disturbance rejection control. *Aerosp. Sci. Technol.* **79**, 601–609 (2018).
54. Guo, K., Jia, J., Yu, X., Guo, L. & Xie, L. Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances. *Control. Eng. Pract.* **102**, 104560 (2020).
55. Michel, L., Ghanes, M., Aoustin, Y. & Barbot, J.-P. A third order semi-implicit homogeneous differentiator: Experimental results. In *2022 16th International Workshop on Variable Structure Systems (VSS)*, 77–82 (IEEE, 2022).
56. Gupta, M., Varshney, P. & Visweswaran, G. Digital fractional-order differentiator and integrator models based on first-order and higher order operators. *Int. J. Circuit Theory Appl.* **39**, 461–474 (2011).

Acknowledgements

The author would like to thank Dr. Tao Wang and Professor Li Shuai for their insightful discussions. In addition, we would also like to thank Professor Yao for his valuable help and guidance.

Author contributions

Zuoming Zou and Shuming Yang conceived the experiment(s), Zuoming Zou and Liang Zhao conducted the experiment(s), Zuoming Zou analysed the results. Liang Zhao prepared the simulation part and completed the revised draft of the paper. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-69911-5>.

Correspondence and requests for materials should be addressed to L.Z.

Reprints and permissions information is available at www.nature.com/reprints.

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024