IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 3, 2021, accepted August 30, 2021, date of publication September 7, 2021, date of current version September 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111101

Horizontal Nonlinear Path Following Guidance Law for a Small UAV With Parameter Optimized by NMPC

YANG CHEN[1], CHAOLEI WANG[2], WEI ZENG[1], AND YONGLIANG WU[3]
[1]School of Physics and Mechatronics Engineering, Longyan University, Longyan 364012, China
[2]Science and Technology on Special System Simulation Laboratory, Beijing Simulation Center, Beijing 100192, China
[3]Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Corresponding author: Yang Chen (chenyang4117@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61503172, and in part by the Natural Science Foundation of Fujian Province, China, under Grant 2020J05197.

ABSTRACT In this study, the problem of guiding a small fixed-wing unmanned aerial vehicle (UAV) toward a predefined horizontal path is studied. A stable nonlinear guidance law, which is a function of the inertial positions and velocities of the UAV and the predefined path, is designed using Lyapunov stability arguments. The concept of the nonlinear model predictive control (NMPC) technique was applied to optimize a key parameter of the guidance law to improve the performance of the controller (PFC_NMPC), where the stability of the relative nonlinear system is maintained. The proposed method was verified in the MATLAB/Simulink environment to realize following the straight-line, square and circular paths. The path-following performance of the proposed method is compared with those of the guidance laws with parameter fixed (PFC) or tuned by fuzzy logic (PFC_FL). With the predictive ability, the proposed method can make the UAV fly more on the desired square and circular paths than the other two methods, PFC and PFC_FL. The error overshoot by using PFC_NMPC is much smaller than those by using the PFC and PFC_FL methods in the presence of wind at 8m/s.

INDEX TERMS Path following, small unmanned aerial vehicle, Lyapunov stability, parameter optimization, nonlinear model predictive control.

I. INTRODUCTION

In the past two decades, the autonomous capability of small fixed-wing UAVs has been greatly developed [1]. UAVs have been widely utilized in military, civil, and other fields, such as military reconnaissance, aerial mapping, smart city traffic management, high-voltage line inspection, and polar environment detection [1]–[3]. Currently, advanced control topics, including coordinated control [4], swarm control [5], [6], obstacle avoidance [7], and path planning [8], are still attracting a large number of researchers to carry out theoretical analysis and experimental verification of fixed-wing UAVs. To realize these advanced control technologies, it is very important that the UAVs have the ability to track a predefined trajectory or follow the desired path accurately. Unlike trajectory tracking, which requires the UAV to arrive at a particular location at a certain time, the UAV needs to follow a predefined geometric path with any feasible speed while using the path-following technique [1]. This reduces the difficulty of controlling and saving energy [9].

Many studies have addressed for the path following of small fixed-wing UAVs in the literature. The approaches to realizing the path following include the linear and nonlinear techniques. While designing the linear path-following controller, proportion-integration-differentiation (PID) and linear quadratic regulator (LQR) are two common methods. In [10], a PID-based common path-following algorithm was demonstrated. Several controller comparisons for autonomous railway following with a fixed-wing UAV have also been presented [11], where the base PID method was the worst performing controller. Although the PID structure is simple, the PID controller has little robustness to environmental disturbance. In [12], a PID controller with a feed-forward part was adopted to realize the tight path

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

following of a UAS. The traditional PD method, together with the disturbance accommodating control (DAC)-based observer, has been applied in the outer-loop controller to produce the commands for the rotational controller [13]. However, the DAC-based observer needs the actual models of the UAV. A modified genetic algorithm was applied to optimize the parameters of PID controllers to achieve better path-following performance [14]. However, the genetic algorithm requires more calculation as the generations and population size increase.

By linearizing UAV dynamics with respect to the desired heading, the LQR method was applied to generate the guidance law for the UAV to follow the straight-line and circular paths respectively [15]. While using the LQR method, the assumption that the difference between the heading angle and the desired path angle is close to zero was adopted. The effectiveness of the method cannot be guaranteed if the initial heading of the UAV is far from the desired path angle. With the combination of the UAV's lateral linear dynamics and the derivatives of the cross-track and heading errors, Lee *et al.* used the LQR technique to generate the control laws for the aileron and rudder directly to make the UAV follow the desired straight-line or circular path [16]. However, the UAV should work near the trim conditions, and the heading and cross-track errors should also be relatively small to make the system stable.

With regard to nonlinear control of path following, intelligent control techniques, including the fuzzy logic, neural networks, and reinforcement learning, have been studied. The Takagi-Sugeno fuzzy model was used to generate the desired route heading in the horizontal plane, which was realized in the Hardware-In-the-Loop (HIL) simulations of following predefined waypoints [17]. With the linear combination of the results of the convolutional neural network for head direction control and the lateral offset, the yaw rate command was determined to follow the trail path for the UAV [18]. In [19], the authors applied a deep reinforcement learning approach to generate the desired yaw angle for the inner loop controller. The good path following simulation results verified the effectiveness of the methods. However, the stability of these path-following methods has not been proven.

Most of the stable path-following control methods were designed based on the Lyapunov stability theory. The vector fields, nested saturation, backstepping, and sliding mode techniques are applied to generate the guidance law for the UAV to follow the desired planar or three-dimensional paths in the sense of Lyapunov stability. In [20], the authors used the sliding mode approach to render the sliding surface, which is constructed by the concepts of vector fields surrounding the desired path. Horizontal straight-line and circular path following were realized in the actual flight experiments. Considering the input constraints, the nested saturation technique was adopted to generate the guidance laws for the UAV to follow a predefined path [1], [21]. The adaptive vector field was designed by addressing unknown wind and unmodelled course angle dynamics [22]. In [23], the authors presented approaches with a combination of the vector field approach and input-to-state stability to realize the curved path following of a UAV. The guidance laws for following the planar and three-dimensional curved paths were designed using conservative, solenoidal, and tangent vector fields [24], [25]. While considering the dynamics of the UAV, the backstepping technique can effectively transfer the virtual control variables between the kinematic and dynamic models. Backstepping and parameter adaptation techniques were applied to generate the roll angle command for the path following of the UAV [26]. An adaptive control strategy is developed to follow the trajectory in the presence of wind, where the design of the controller is based on the backstepping technique [27]. In addition, the familiar pure pursuit with line-of-sight (PLOS) [28], nonlinear guidance law using the virtual target point (NLGL) [29], and the Lyapunov-based 3D path- following method [30], etc., have also been designed and applied in UAVs. The stability of these controllers is guaranteed by the Lyapunov stability criteria. It can be seen that the path-following guidance laws designed based on the Lyapunov stability theory are very attractive. The relative structures are simple and easy to implement in the autopilots. However, in the process of using the Lyapunov stability criterion to design the path-following controller, the parameters of the controller are usually fixed to ensure the stability of the closed-loop system. The choice of parameters directly affects the path-following performance of aircraft under different initial states or wind disturbances, such as overshoot and stability time [31].

Nonlinear model predictive control, also referred to as receding horizon control, has been applied to solve the path- following problems with input and state constraints [32]–[38]. In some applications of NMPC for high-level control of UAVs, they have satisfactory path following performance. However, the relative closed-loop stability cannot be guaranteed [32]–[34]. Yang *et al.* used the NMPC method to follow a desired linear path. Sufficient conditions were derived to guarantee the closed-loop stability of the system [35]. An adaptive NMPC that varies the control horizon according to the path curvature profile was presented to achieve accurate performance [36]. In [37], an NMPC scheme for the path-following problem was introduced with a proof of asymptotic convergence and recursive feasibility, where a suitable terminal penalty and the related terminal constraint were also presented. A path- following controller with obstacle avoidance using NMPC was designed [38]. Specific definitions and sufficient assumptions were introduced to satisfy the stability of the system controlled by the NMPC feedback law. Because the NMPC technique can consider the dynamics and kinematics model of the UAV together with the constraints of state and control variables, it can generate control laws for the UAV to realize path following. To solve the NMPC path-following problem, the conditions designed to guarantee the stability of the closed-loop system put forward higher requirements for the optimization technology. While realizing the open-loop optimization of

the NMPC problem without considering the stability of the closed-loop system, the calculation is simpler and relatively faster. However, it lacks a stability guarantee.

Motivated by the above discussions, it can be found that the nonlinear guidance control laws designed based on the Lyapunov stability theory are simple and effective. However, the parameters of these guidance law are usually fixed. The uniform optimal path-following performance cannot be realized under different initial states or wind disturbances. In our previous works [39], a PD-like nonlinear guidance was presented to guide the UAV toward the predefined straight-line path. To achieve better path-following performance, one of the parameters was tuned by fuzzy logic. However, the design of fuzzy rules is still based on experience, and the best path following performance cannot be achieved. In this study, the PD-like nonlinear guidance law is extended to follow a class of horizontal paths. Because the NMPC technique has the predictive ability via the finite horizon optimization of a kinetic or dynamic model to anticipate future events and take control actions accordingly. the open-loop optimization of the NMPC technique is used to optimize a key parameter of the controller to make it adaptive. The proposed novel nonlinear guidance law adopts both the simple structure of the basic method and the predictive optimization of the NMPC technique to improve the path-following performance. The stability of the proposed controller is guaranteed by Lyapunov stability arguments. A preliminary version of this paper is presented in [40]. However, the reference [40] did not address the nonlinear guidance law for a class of horizontal paths, the details of the parameter optimization for the straight-line and circular path guidance laws, and did not include more simulation analysis under different methods, all of which appear in this paper.

The main contributions of this study are summarized as follows: First, a Lyapunov-stability-based guidance law for following a class of horizontal paths is presented in the Cartesian coordinate system. The guidance law for following the circular path was derived from the UAV's course and track angle in the polar coordinate system. Second, a key parameter of the guidance law for following the straight-line and circular paths is optimized using the NMPC technique. The stability of the guidance system is still guaranteed, whereas the NMPC technique is only used for parameter optimization. Thus, the proposed method inherits the advantages of the Lyapunov stability of the nonlinear controller with simple structure, and the predictive ability of NMPC. To the best of our knowledge, this the first time that the NMPC technique is applied to optimize the parameter of a Lyapunov-stability-based guidance law. Third, the control results were compared with the methods when the parameter was fixed or adjusted by fuzzy logic, which illustrates the effectiveness and better performance of the proposed control strategies.

The remainder of this paper is organized as follows. In the next section, the problem description is presented. In Section III, the guidance law to guide the UAV toward a class of horizontal paths is presented. In Section IV, the details of the NMPC technique used to optimize the parameters of the guidance law for flowing straight lines and circular paths are presented. The simulation results in MATLAB/Simulink are presented in Section V, and the comparison under different methods is also discussed. Finally, the concluding remarks are summarized in Section VI.

II. PROBLEM DESCRIPTION

A small fixed-wing UAV equipped with a low-level flight controller, which can provide altitude- and airspeed-hold functions, is assumed to be used for path following. The kinematics of the UAV in the inertial horizontal frame can be used for controller design:

$$\begin{cases} \dot{x} = V_a cos(\psi) + W_x \\ \dot{y} = V_a sin(\psi) + W_y \end{cases} \tag{1}$$

where (x, y) is the position of the UAV in the inertial horizontal frame, W_x and W_y are the x and y components of the wind velocity W, and V_a, ψ denote the airspeed and heading, respectively. By introducing the groundspeed V_g and the course χ with the relationship shown in Fig. 1, equation (1) becomes

$$\begin{cases} \dot{x} = V_g cos(\chi) \\ \dot{y} = V_g sin(\chi) \end{cases} \tag{2}$$

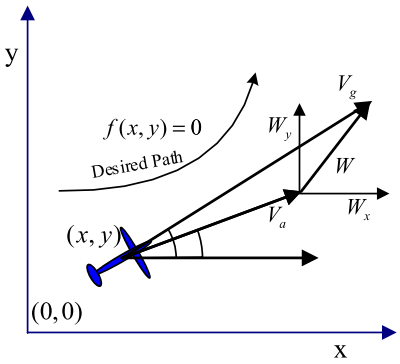

FIGURE 1. Position of the UAV with respect to the desired path and the relationship among the horizontal groundspeed V_g, airspeed V_a and the wind speed W.

Following [20], with another assumption that the autopilot can provide the course-hold function adopted, by introducing the course rate control $\dot{\chi} = u_{cmd}$, the system (2) becomes

$$\begin{cases} \dot{x} = V_g \cos(\chi) \\ \dot{y} = V_g \sin(\chi) \\ \dot{\chi} = u_{cmd} \end{cases} \tag{3}$$

Therefore, the motions of the UAV can be expressed in terms of the groundspeed and course angle. In the following section, the command design is based on model (3). Because the groundspeed contains information on the wind velocity, the wind disturbance rejection will be naturally considered [20]. If the UAV is flying on the desired path, the groundspeed should always be positive. In the following,

it is assumed that $V_g \geq V_{gp} > 0$, where V_{gp} is a given positive constant. Considering a horizontal smooth path, which can be described as a continuously differentiable function $f{:}R^2 \rightarrow R$ with

$$f(x, y) = 0 \tag{4}$$

Following [41], the curve function $f(x, y)$ instead of the Euclidean distance from the curve is used as the distance function, that is, $d(x, y) = f(x, y)$. The function $d(x, y)$ can indicate whether the UAV lies on the curve $f(x, y) = 0$. The partial derivatives and the gradient modules of $d(x, y)$ with respect to x and y can be derived as:

$$\begin{cases} d_x = \dfrac{\partial d(x, y)}{\partial x} = \dfrac{\partial f(x, y)}{\partial x} \\ d_y = \dfrac{\partial d(x, y)}{\partial y} = \dfrac{\partial f(x, y)}{\partial y} \\ \|\nabla d\| = \sqrt{d_x^2 + d_y^2} \end{cases} \tag{5}$$

For some continuously differentiable function $f{:}R^2 \rightarrow R$, the gradient modules of $f(x, y)$ can reach zero at some points (x, y). If the initial position of the vehicle is on these points, the guidance may fail due to $\|\nabla d\| = 0$. In the following, the flying area of the UAV is defined as $D_{\text{int}} = \{(x, y) \,|\, x, y \in R, \|\nabla d\| \geq \lambda\}$, where λ is a positive constant.

III. GUIDANCE LAW FOR THE HORIZONTAL PATH FOLLOWING

For a given horizontal smooth path shown as (4), if the UAV is flying on the path with the right direction, it should have the fact that the distance function $d(x, y) = 0$. Furthermore, the direction of the vehicle should equal the relative tangent direction of the path. To derive the guidance law for the horizontal path following, two new intermediate variables x_1 and χ_1 are introduced as follows:

$$\begin{cases} x_1 = d(x, y) \\ \chi_1 = \begin{cases} \chi - tan^{-1}\left(-\frac{d_x}{d_y}\right), & d_y \neq 0 \\ \chi - \frac{\pi}{2}, & d_y = 0, d_x < 0 \\ \chi + \frac{\pi}{2}, & d_y = 0, d_x > 0 \end{cases} \end{cases} \tag{6}$$

From (6), the sine value of variable χ_1 can be derived as $sin(\chi_1) = sin\left(\chi - tan^{-1}\left(-\frac{d_x}{d_y}\right)\right) = sin(\chi)\frac{d_y}{\|\nabla d\|} + cos(\chi)\frac{d_x}{\|\nabla d\|}$. Additionally, the time derivatives of x_1 and χ_1 are:

$$\begin{cases} \dot{x}_1 = \dfrac{\mathrm{d}}{\mathrm{dt}} d(x, y) = d_x\dot{x} + d_y\dot{y} \\ = d_x V_g \cos(\chi) + d_y V_g \sin(\chi) \\ = V_g \|\nabla d\| \sin(\chi_1) \\ \dot{\chi}_1 = \dot{\chi} - \dfrac{d}{dt} tan^{-1}\left(-\dfrac{f_x}{f_y}\right) \\ = u_{cmd} - \frac{d}{dt} tan^{-1}\left(-\dfrac{d_x}{d_y}\right) \end{cases} \tag{7}$$

Proposition 1: Consider the kinematic error model of the path following described in (7). Because $V_g \geq V_{gp} > 0$, then the control law,

$$u_{cmd} = -K_1 \|\nabla d\| V_g f_{sat}(x_1) - K_2 V_g \frac{dx_1}{dt} + \frac{d}{dt} tan^{-1}\left(-\frac{d_x}{d_y}\right) \tag{8}$$

where K_1 is a positive constant, $K_2 \geq K_{2p}$ with K_{2p} a given positive constant, and the saturation function

$$f_{sat}(x) = \begin{cases} x_0, & x > x_0 \\ x, -x_0 \leq & x \leq x_0 \\ -x_0, & x < x_0 \end{cases}$$

with x_0 an arbitrary given positive constant, asymptotically drives x_1 and χ_1 toward zero. The saturation function $f_{sat}(x)$ is used to ensure that the UAV will not fall into an infinite loop when its initial position is far away from the desired path.

Proof: With the control law u_{cmd} shown in (8), system (7) can be derived as

$$\begin{cases} \dot{x}_1 = V_g \|\nabla d\| \sin(\chi_1) \\ \dot{\chi}_1 = -K_1 \|\nabla d\| V_g f_{sat}(x_1) - K_2 V_g \dfrac{dx_1}{dt} \end{cases} \tag{9}$$

Define the domain $D = \{(x_1, \chi_1) | x_1 \in R, -\pi < \chi_1 < \pi\}$. It can be found that $(0, 0)$ is the equilibrium point for (9) in domain D. The candidate Lyapunov function $V(x_1, \chi_1)$ is given as:

$$V(x_1, \chi_1) = K_1 \int_0^{x_1} f_{sat}(y)dy + \int_0^{\chi_1} \sin(y)dy \tag{10}$$

It can be found that $V(x_1, \chi_1) > 0$ on $D - \{(0, 0)\}$. Furthermore,

$$\begin{cases} V(x_1, \chi_1) \geq 1 - cos(\chi_1) = W_1(x_1, \chi_1) \\ V(x_1, \chi_1) \leq \dfrac{K_1}{2} x_1^2 + 1 - cos(\chi_1) \\ \leq \dfrac{K_1}{2} x_1^2 + \dfrac{\chi_1^2}{2} = W_2(x_1, \chi_1) \end{cases} \tag{11}$$

Differentiating $V(x_1, \chi_1)$ with respect to time, it gets that

$$\begin{aligned} \dot{V}(x_1, \chi_1) &= K_1 \dot{x}_1 f_{sat}(x_1) + \dot{\chi}_1 sin(\chi_1) \\ &= K_1 f_{sat}(x_1)(\dot{x}_1 - \|\nabla d\| V_g sin(\chi_1)) \\ &\quad -K_2 V_g \dot{x}_1 sin(\chi_1) \\ &= -K_2 V_g^2 \|\nabla d\| sin^2(\chi_1) \\ &\leq -K_2 V_{gp}^2 \lambda sin^2(\chi_1) = -W_3(x_1, \chi_1) \end{aligned} \tag{12}$$

The three functions $W_1(x_1, \chi_1)$, $W_2(x_1, \chi_1)$, and $W_3(x_1, \chi_1)$ are all continuous positive definite on D. The above shows that the equilibrium point $(0, 0)$ is uniformly asymptotically stable [42], that is, the control law (8) can asymptotically drive x_1 and χ_1 toward zero. Thus, under the guidance law (8), the UAV can reach the predefined path and then follow it in the right direction. ■

For nonlinear system (9), the phase trajectory is affected by the two control parameters k_1 and k_2. The dynamics of χ_1 is a

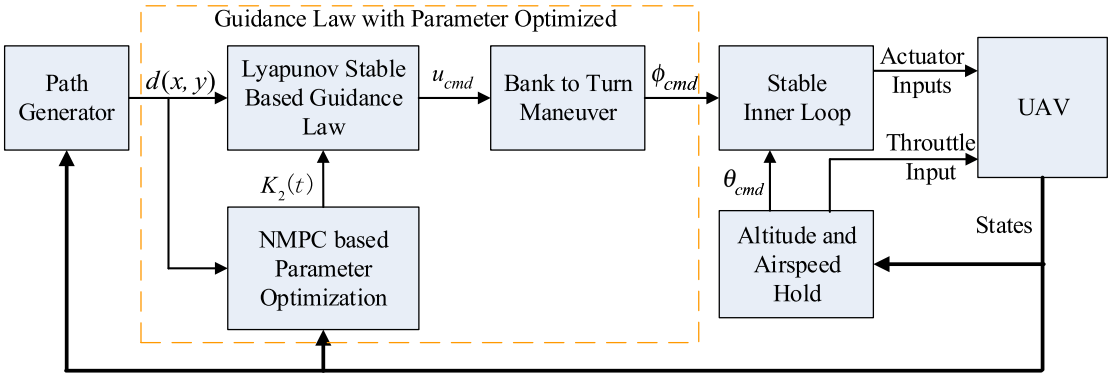

FIGURE 2. The structure of the proposed controller.

nonlinear combination of the variable x_1 and its differential, which is similar to a PD controller. The term $-K_2V_g\frac{dx_1}{dt}$ can predict the error trend and correct the error in advance, which can improve the dynamic performance of the system. However, if the parameter K_2 is too small, the prediction effect is limited. If the parameter K_2 is too large, the differential action will be too strong, which may cause oscillation. Thus, the parameter K_2 affects the guidance performance of the UAV. For a UAV flying in a windy environment, the fixed parameter K_2 cannot guarantee the same flight performance. From the above proof process, it can be found that as long as the parameter K_2 changes in a positive interval, the stability of the system (9) can still be guaranteed. Considering the limitation of the turning rate owing to the minimum turning radius of the UAV, the control input u_{cmd} has constraints $-u_m \leq u_{cmd} \leq u_m$ (where u_m is a positive constant) in this study. Accordingly, the control parameter K_2 is also limited within the range $[K_{2,\min}, K_{2,max}]$, where $K_{2,\min}$ and $K_{2,max}$ are given positive constants.

As all of the planar curves can be approximated by a series of segments, and considering the course rate constraint, this study focuses on the UAVs' straight-line and circular paths as follows. The general expression of a given straight-line path can be denoted as $ax + by + c = 0$ with $\sqrt{a^2+b^2} = 1$, where $a, b, c \in R$. The circular path can be described as $\sqrt{(x-C_x)^2+(y-C_y)^2} - C_R = 0$, where (C_x, C_y) and C_R represent the center and radius of the circle, respectively. It can be found that with the circular path function $f(x, y)$ chosen as above, the magnitude of the corresponding gradient is $\|\nabla f\| = 1$. In the following section, the parameter K_2 of the guidance law shown in (8) will be optimized to improve the straight-line and circular paths following performance.

IV. GUIDANCE LAW FOR HORIZONTAL PATH FOLLOWING WITH PARAMETER OPTIMIZED BY NMPC (PFC_NMPC)

The NMPC technique has the predictive ability via the finite horizon optimization of a kinetic model to anticipate future events and take control actions accordingly. This section shows the applications of the NMPC technique to adjust the parameter K_2 to improve the path-following performance.

The general structure of the proposed controller, which consists of three hierarchical control loops, is shown in Fig. 2. The inner-loop control block was used to stabilize the dynamics and to track the reference control signals. The outer-loop controller, including the presented guidance law and the altitude and airspeed hold block, is applied to control the position and heading of the UAV. The pitch angle command θ_{cmd} is used to realize the altitude hold through a zero climb rate. The path generator block generates the desired flight paths.

A. GUIDANCE LAW FOR STRAIGHT-LINE PATH FOLLOWING WITH PARAMETER OPTIMIZED BY NMPC

Considering that the nonlinear function (3) is continuous and Lipschitz continuous, the forward Euler method is used to approximate the solution to the differential equations with:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, u_k) = \begin{pmatrix} x_k + \Delta t V_g cos(\chi_k) \\ y_k + \Delta t V_g sin(\chi_k) \\ \chi_k + \Delta t u_{cmd,k} \end{pmatrix} \tag{13}$$

where $\boldsymbol{x}_k = (x_k, y_k, \chi_k)^T$, $u_k = u_{cmd,k}$, and Δt is the discrete time step.

The basic nonlinear model predictive control is to find a control input sequence $(u_0, \cdots, u_{N-1})$ that minimizes the cost function:

$$J = \phi(\boldsymbol{x}_N) + \sum_{i=0}^{N-1} L(\boldsymbol{x}_i, u_i) \tag{14}$$

For the straight-line path following, we have

$$\begin{aligned} \phi(\boldsymbol{x}_N) &= S_D \frac{(ax_N + by_N + c)^2}{a^2+b^2} + S_\chi (\chi_N - \chi_d)^2 \\ L(\boldsymbol{x}_i, u_i) &= Q_D \frac{(ax_i + by_i + c)^2}{a^2+b^2} + Q_\chi (\chi_i - \chi_d)^2 + Ru_i^2 \end{aligned} \tag{15}$$

where $\phi(\boldsymbol{x}_N)$ denotes the cost in the final state, $L(\boldsymbol{x}_i, \mathrm{u}_i)$ represents the cost of the state and input on the i_{th} step,

S_D, S_χ, Q_D, Q_χ, and R are weights, and N is the length of the look-ahead horizon. This optimization is performed at each sample time. The nonlinear kinematic function (13) acts as an equality constraint.

To follow the straight-line path $ax + by + c = 0$ with $\sqrt{a^2+b^2} = 1$, the control law u_{cmd} can be derived from (8) as $u_{cmd} = -K_1 V_g f_{sat}(ax + by + c) - K_2 V_g(aV_g cos(\chi) + bV_g sin(\chi))$. To optimize the parameter K_2, we substitute the control law $u(K_{2,i})$ as shown in (16) for u_i in (14).

$$u(K_{2,i}) = -K_1 V_g f_{sat}(ax_i + by_i + c) - K_{2,i} V_g(aV_g cos(\chi_i) + bV_g sin(\chi_i)) \quad (16)$$

Then, $L(\boldsymbol{x}_i, u_i)(i = 0, \cdots, N-1)$ becomes as

$$\begin{aligned} L(\boldsymbol{x}_i, u_i) &= L(\boldsymbol{x}_i, K_{2,i}) \\ &= Q_D(ax_i + by_i + c)^2 + Q_\chi(\chi_i - \chi_d)^2 \\ &\quad + Ru^2(K_{2,i}) \end{aligned} \quad (17)$$

The relative equality constrain (13) is changed as

$$\begin{aligned} &\boldsymbol{x}_{k+1} \\ &= f(\boldsymbol{x}_k, K_{2,k}) \\ &= \begin{pmatrix} x_k + \Delta t V_g cos(\chi_k) \\ y_k + \Delta t V_g sin(\chi_k) \\ \chi_k + \Delta t V_g \begin{pmatrix} -K_1 f_{sat}(ax_k + by_k + c) \\ -K_{2,k}(aV_g cos(\chi_k) + bV_g sin(\chi_k)) \end{pmatrix} \end{pmatrix} \end{aligned} \quad (18)$$

To minimize the cost function J with the components shown as (14) and (15), similar to the processing of the open-loop optimization of NMPC problem [43], the indirect method of Lagrange Multipliers is applied. The cost function J is rewritten by adding a sequence of value 0 multiplied by the vector Lagrange multiplier sequence $\{\boldsymbol{\lambda}_i : i = 0, \cdots, N\}$ as:

$$J = \phi(\boldsymbol{x}_N) + \sum_{i=0}^{N-1} L(\boldsymbol{x}_i, K_{2,i}) + \boldsymbol{\lambda}_{i+1}^T \left(\boldsymbol{f}(\boldsymbol{x}_i, K_{2,i}) - \boldsymbol{x}_{i+1}\right) \quad (19)$$

Here, the Hamiltonian and the derivative of J can be derived as:

$$\begin{aligned} &H_i(\boldsymbol{x}_i, K_{2,i}) \\ &= L(\boldsymbol{x}_i, K_{2,i}) + \boldsymbol{\lambda}_{i+1}^T \boldsymbol{f}(\boldsymbol{x}_i, K_{2,i}) \end{aligned} \quad (20)$$

$$\begin{aligned} dJ &= \left(\frac{\partial \phi(\boldsymbol{x}_N)}{\partial \boldsymbol{x}_N} - \boldsymbol{\lambda}_N^T\right) d\boldsymbol{x}_N \\ &\quad + \sum_{i=1}^{N-1} \left(\left(\frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial \boldsymbol{x}_i} - \boldsymbol{\lambda}_i^T \right) d\boldsymbol{x}_i \right. \\ &\quad \left. + \frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} dK_{2,i} \right) \\ &\quad + \frac{\partial H_0(\boldsymbol{x}_0, K_{2,0})}{\partial \boldsymbol{x}_0} d\boldsymbol{x}_0 + \frac{\partial H_0(\boldsymbol{x}_0, K_{2,0})}{\partial K_{2,0}} dK_{2,0} \end{aligned} \quad (21)$$

Because the vector multiplier sequence $\{\boldsymbol{\lambda}_i : i = 1, \cdots, N\}$ is multiplied by zero terms as shown in (19), and then they can be arbitrarily chosen as:

$$\boldsymbol{\lambda}_N = \left(\frac{\partial \phi(\boldsymbol{x}_N)}{\partial \boldsymbol{x}_N}\right)^T \quad (22)$$

$$\begin{aligned} \boldsymbol{\lambda}_i &= \left(\frac{\partial H_i(\boldsymbol{x}_i, K_{2,i}}{\partial \boldsymbol{x}_i}\right)^T \\ &= \left(\frac{\partial L(\boldsymbol{x}_i, K_{2,i}}{\partial \boldsymbol{x}_i}\right)^T + \boldsymbol{\lambda}_{i+1}^T \frac{\partial \boldsymbol{f}(\boldsymbol{x}_i, K_{2,i})}{\partial \boldsymbol{x}_i} \\ &\quad i = 0, \cdots, N-1 \end{aligned} \quad (23)$$

The details of $\boldsymbol{\lambda}_i$ in (23) are listed in equations (A.1)-(A.5), as shown in the Appendix. Then dJ can be simplified as:

$$dJ = \sum_{i=0}^{N-1} \frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} dK_{2,i} + \boldsymbol{\lambda}_0^T d\boldsymbol{x}_0 \quad (24)$$

Here, $\frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}}$ is the gradient of dJ with respect to the control parameters $K_{2,i}$ while holding $\boldsymbol{x}_0$ and $K_{2,j}$, $j \neq i$ constant. Thus, if $\boldsymbol{x}_0$ is constant, an N dimensional search direction from the control parameter sequence $(K_{2,0}, K_{2,1}, \cdots, K_{2,N-1})$ can be obtained which will lead to a lower cost. At a local minimum of J, we can conclude that $\frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} = 0$.

With the multiplier sequence $\{\boldsymbol{\lambda}_i : i = 0, \cdots, N\}$ chosen above, the gradient of the Hamiltonian with respect to the control parameter K_2 becomes

$$\frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} = \frac{\partial L(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} + \boldsymbol{\lambda}_{i+1}^T \frac{\partial \boldsymbol{f}(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} \quad (25)$$

The details of $\frac{\partial H_i(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}}$ in (25) can be seen in equations (A.6)-(A.7) shown in the Appendix. As described in Section III, the control input u_{cmd} and the parameter K_2 are bounded with $-u_m \leq u_{cmd} \leq u_m$ and $K_{2,min} \leq K_2 \leq K_{2,max}$ respectively. During the optimization, the corresponding control input u_{cmd} and parameter K_2 should satisfy the constraints described above.

With a candidate parameter sequence and a given $\{K_{2,i}\}_{i=0}^{N-1}$, the parameter optimization process at each given time instant k is performed as

The parameters β, δ and $Iter_{max}$ in **Algorithm 1** are predefined positive values.

B. GUIDANCE LAW FOR CIRCULAR PATH FOLLOWING WITH PARAMETER OPTIMIZED BY NMPC

While optimizing the parameter K_2 of the guidance law for the circular path following, the cost function includes the distance deviation and the difference between the course and azimuth angles. In the process of optimization calculation, it is very inconvenient to use the difference between the course and azimuth angles described in the Cartesian coordinate system. The description of the UAV's course and track angle in polar coordinate system is simple and intuitive. For circular path following, the polar coordinate system was

Algorithm 1 Parameter K_2 Optimized by NMPC for Straight-Line Path Guidance Law

1: **while** $\left|\Delta J_k \triangleq J_k - J_{k-1}\right| > \delta$ and $Iter\_times < Iter_{max}$ do
2: **for** $j = 0, \ldots, N-1$
3: compute $u(K_{2,j})$ using (16) with the parameter sequence $(K_{2,0}, \ldots, K_{2,N-1})$, and limit $u(K_{2,j})$ within $[-u_m, u_m]$.
4: compute $\boldsymbol{x}_{j+1}$ using (13)
5: **end for**
6: **for** $j = 0, \ldots, N$
7: compute $\boldsymbol{\lambda}_j$ using (22) and (23).
8: **end for**
9: **for** $j = 0, \ldots, N-1$
10: compute $\frac{\partial H_j(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,j}}$ using (25).
11: **end for**
12: **if** $\Delta J_k \leq 0$
13: update $(K_{2,0}K_{2,1}, \ldots, K_{2,N-1})$ via $K_{2,j}^{k+1} \triangleq K_{2,j}^{k} + h\frac{\partial H_j(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,j}^{k}}$ and limit $K_{2,j}^{k+1}$ within the range $[K_{2,min}K_{2,max}]$ for $j = 0, \ldots, N-1$
14: **else**
15: reduce h by $h = \beta \cdot h$
16: **end if**
17: update $Iter\_times = Iter\_times + 1$
18: **end while**

chosen to describe the dynamics. From Fig. 3, we have

$$\begin{cases} x = C_x + dcos(\gamma) \\ y = C_y + dsin(\gamma) \end{cases} \tag{26}$$

$$\begin{cases} \dot{x} = V_g cos(\chi_c) \\ \dot{y} = V_g sin(\chi_c) \end{cases} \tag{27}$$

where (C_x, C_y) is the center of the circle with respect to the Cartesian coordinate system, C_R is the radius of the circle, γ is the angular position of the UAV with respect to the circle center.

If the UAV follows the circle clockwise, then $d = C_R$ and $\chi_c = \gamma - \frac{\pi}{2}$. Taking the derivative of (26) and considering (27), it is clear that $\dot{d} = V_g cos(\chi_c - \gamma)$ and $\dot{\gamma} = V_g sin(\chi_c - \gamma)/d$. In the polar coordinate system, the nonlinear equation (3) in the Euclidean coordinate system can be

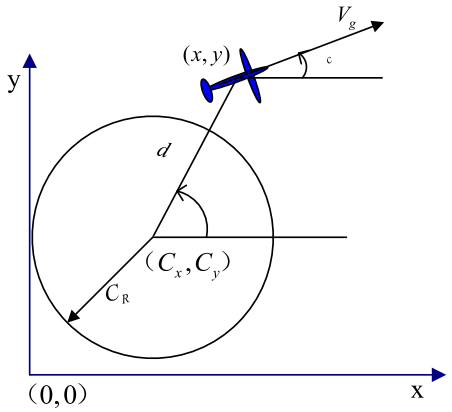

FIGURE 3. The relative position of the UAV in the polar coordinates.

expressed as

$$\begin{cases} \dot{d} = V_g cos(\chi_c - \gamma) \\ \dot{\gamma} = V_g sin(\chi_c - \gamma)/d \\ \dot{\chi}_c = u_{c,cmd} \end{cases} \tag{28}$$

For the circular path $\sqrt{(x-C_x)^2 + (y-C_y)^2} - C_R = 0$, it has the fact that $tan^{-1}\left(\frac{-d_x}{d_y}\right) = tan^{-1}\left(-\frac{x-C_x}{y-C_y}\right) = \gamma$, the corresponding control law can be derived from (8) as:

$$u_{c,cmd} = -K_{c1}V_g f_{sat}(d - C_R) - K_{c2}V_g\dot{d} + \dot{\gamma} \tag{29}$$

where the parameters K_{c1}, K_{c2} are positive. Similar to the subsection IV.A for the straight-line path following, the equality constraints used for optimization can be derived as (30), shown at the bottom of the page, where $\boldsymbol{x}_{c,k} = (d_k, \gamma_k, \chi_{c,k})^T$ and the subscript c represents the circle.

The relative cost function J_c used for optimization is defined as:

$$\begin{aligned} J_c = \phi_c(\boldsymbol{x}_{c,N}) + \sum_{i=0}^{N-1} L_c(\boldsymbol{x}_{c,i}, K_{c2,i}) \\ + \boldsymbol{\lambda}_{c,i+1}^T \left(\boldsymbol{f}_c(\boldsymbol{x}_{c,i}, K_{c2,i}) - \boldsymbol{x}_{c,i+1}\right) \end{aligned} \tag{31}$$

The cost of the final state $\phi_c(\boldsymbol{x}_{c,N})$ and the cost of the state and input on the i_{th} step $L_c(\boldsymbol{x}_{c,i}, K_{c2,i})$ are defined as

$$\begin{aligned} \phi_c(\boldsymbol{x}_{c,N}) &= S_{c,D}(d_N - C_R)^2 + S_{c,\chi}(\chi_{c,N} - \gamma_N + \pi/2)^2 \\ L_c(\boldsymbol{x}_{c,i}, K_{c2,i}) &= Q_{c,D}(d_i - C_R)^2 + Q_{c,\chi}(\chi_{c,i} - \gamma_i + \pi/2)^2 \\ &\quad + R_c u_{c,cmd,i}^2 \end{aligned} \tag{32}$$

$$\begin{aligned} \boldsymbol{x}_{c,k+1} = \boldsymbol{f}_c(\boldsymbol{x}_{c,k}, K_{c2,k}) &= \begin{pmatrix} d_k + \Delta t V_g cos(\chi_{c,k} - \gamma_k) \\ \gamma_k + \Delta t \frac{V_g sin(\chi_{c,k} - \gamma_k)}{d_k} \\ \chi_{c,k} + \Delta t u_{c,cmd,k} \end{pmatrix} \\ &= \begin{pmatrix} d_k + \Delta t V_g cos(\chi_{c,k} - \gamma_k) \\ \gamma_k + \Delta t V_g sin(\chi_{c,k} - \gamma_k)/d_k \\ \chi_{c,k} + \Delta t \begin{pmatrix} -K_{c1}V_g f_{sat}(d_k - C_R) - \\ -K_{c2,k}V_g^2 cos(\chi_{c,k} - \gamma_k) + \frac{V_g sin(\chi_{c,k} - \gamma_k)}{d_k} \end{pmatrix} \end{pmatrix} \end{aligned} \tag{30}$$

where $S_{c,D}$, $S_{c,\chi}$, $Q_{c,D}$, $Q_{c,\chi}$, and R_c are weights, $u_{c,cmd,i} = -K_{c1}V_g f_{sat}(d_i - C_R) - K_{c2,i}V_g^2 cos(\chi_{c,i} - \gamma_i) + V_g \frac{V_g sin(\chi_{c,i}-\gamma_i)}{d_i}$. The relative Hamiltonian is defined as:

$$H_{c,i}(\boldsymbol{x}_{c,i}, K_{c2,i}) = L_c(\boldsymbol{x}_{c,i}, K_{c2,i}) + \boldsymbol{\lambda}_{c,i+1}^T f_c\left(\boldsymbol{x}_{c,i}, K_{c2,i}\right) \quad (33)$$

The vector multiplier sequence $\{\boldsymbol{\lambda}_{c,i}: i = 0, \cdots, N\}$ is chosen as:

$$\boldsymbol{\lambda}_{c,N} = \left(\frac{\partial \phi_c(\mathbf{x}_{c,N})}{\partial \boldsymbol{x}_{c,N}}\right)^T \quad (34)$$

$$\begin{aligned}\boldsymbol{\lambda}_{c,i} &= \left(\frac{\partial H_{c,i}(\mathbf{x}_{c,i}, K_{c2,i})}{\partial \boldsymbol{x}_{c,i}}\right)^T \\ &= \left(\frac{\partial L_c(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial \boldsymbol{x}_{c,i}}\right)^T + \boldsymbol{\lambda}_{c,i+1}^T \frac{\partial f_c\left(\boldsymbol{x}_{c,i}, K_{c2,i}\right)}{\partial \boldsymbol{x}_{c,i}} \\ & i = 0, \cdots, N-1\end{aligned} \quad (35)$$

The details of $\boldsymbol{\lambda}_{c,i}$ in (35) are listed in equations (A.8)-(A.11), as shown in the Appendix. The derivative of J_c is:

$$dJ_c = \sum_{i=0}^{N-1} \frac{\partial H_{c,i}(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial K_{c2,i}} dK_{c2,i} + \lambda_{c,0}^T d\boldsymbol{x}_{c,0} \quad (36)$$

$$\begin{aligned}&\frac{\partial H_{c,i}(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial K_{c2,i}} \\ &= \frac{\partial L_c(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial K_{c2,i}} + \boldsymbol{\lambda}_{c,i+1}^T \frac{\partial \boldsymbol{f}_c(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial K_{c,i}}\end{aligned} \quad (37)$$

The details of the terms of $\frac{\partial H_{c,i}(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial K_{c2,i}}$ can be seen as equations (A.12)-(A.13) shown in the Appendix. The parameter K_{c2} is also limited within the range $[K_{c2,min}, K_{c2,max}]$, where $K_{c2,min}$ and $K_{c2,max}$ are given positive constants. With a candidate parameter sequence and a given $\{K_{c2,i}\}_{i=0}^{N-1}$, the parameter optimization process at each given time instant k is performed as

Algorithm 2 Parameter K_{c2} Optimized by NMPC for Circular Path Guidance Law

while $\left|\Delta J_{c,k} \triangleq J_{c,k} - J_{c,k-1}\right| > \delta$ and $Iter\_times < Iter_{max}$ **do**
  for $j = 0, \ldots, N-1$
    compute $u_{c,cmd,j}$ with the parameter sequence $(K_{c2,0}K_{c2,1}, \ldots, K_{c2,N-1})$, and limit $u_{c,cmd,j}$ within $[-u_m, u_m]$.
    compute $\boldsymbol{x}_{c,j+1}$ using (30)
  end for
  for $j = 0, \ldots, N$
    compute $\boldsymbol{\lambda}_{c,j}$ using (34) and (35).
  end for
  for $j = 0, \ldots, N-1$
    compute $\frac{\partial H_{c,j}(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial K_{c2,j}}$ using (37).
  end for
  if $\Delta J_k \leq 0$
    update $(K_{c2,0}K_{c2,1}, \ldots, K_{c2,N-1})$ via $K_{c2,j}^{k+1} \triangleq K_{c2,j}^k + h\frac{\partial H_{c,j}\left(\boldsymbol{x}_{c,j}, K_{c2,j}^k\right)}{\partial K_{c2,j}^k}$ and limit $K_{c2,j}^{k+1}$ within the range $[K_{c2,min}K_{c2,max}]$ for $j = 0, \ldots, N-1$
  else
    reduce h by $h = \beta \cdot h$
  end if
  update $Iter\_times = Iter\_times + 1$
end while

V. SIMULATIONS

For the horizontal path following, the bank-to-turn maneuver is used to realize the UAV's level turn. The course rate $\dot{\chi}$ is induced by the roll angle ϕ with the groundspeed V_g as

$$\dot{\chi} = \frac{g}{V_g} tan(\phi) \quad (38)$$

where g is the magnitude of gravity at the sea level. The roll dynamics is assumed to be much faster than the heading and altitude dynamics, which implies that the roll angle can be considered as the control variable. Therefore, (38) becomes

$$\dot{\chi} = \frac{g}{V_g} tan(\phi_{cmd}) \quad (39)$$

For the given course rate command $\dot{\chi} = u_{cmd}$ shown in (8), it can be translated into a bank angle command ϕ_{cmd} as

$$\phi_{cmd} = tan^{-1}\left(\frac{V_g u_{cmd}}{g}\right) \quad (40)$$

In the following simulation, the guidance law (8) with the parameter K_2 tuned by fuzzy logic (PFC_FL) was also applied for path following to show the performance of the proposed method. The parameter K_2 is adjusted around a given nominal value K_{20} using a fuzzy logic unit as follows:

$$K_2 = K_{20} + \Delta K_2 \quad (41)$$

where ΔK_2 is the output of the fuzzy-logic unit for adjusting the parameter K_2. The details can be found in [39].

Simulation studies on the proposed method were carried out in the MATLAB/Simulink environment with the help of the Aeronautical Simulation (Aerosim) Block Set [39].

The Aerosonde UAV, which has a wingspan of 2.9m was prepared as a test vehicle [40]. The detailed 6 degree-of-freedom nonlinear model of the UAV was built using the means of Aerosim Block Set. Fig. 4 presents the Simulink model, which contains the proposed guidance law, static inner-loop controller, and the UAV model used for the simulation studies.

Table 1 lists the initial setup and the parameters used in the simulation. A constant wind speed of 8m/s from 270deg west was added to the model of the UAV. The simulation period was set to 50ms.

A. STRAIGHT-LINE PATH FOLLOWING

The response comparison of the three proposed methods while following the four directional paths is shown in Fig. 5.

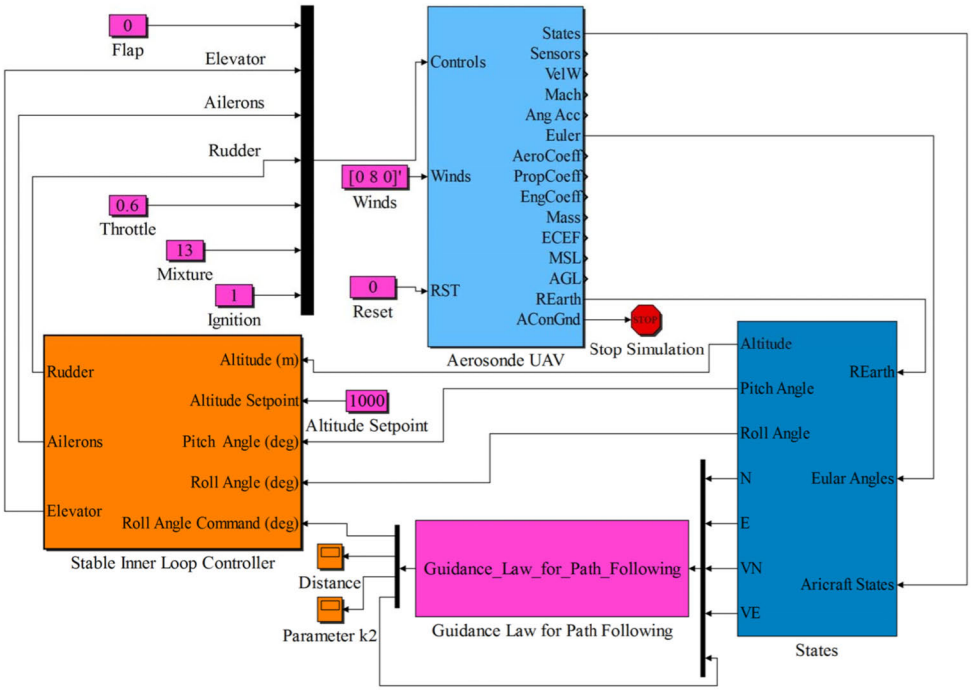

FIGURE 4. The Simulink block diagram for the simulation studies.

TABLE 1. The initial setup and parameters for simulation.

Parameter	Value	Parameter	Value
UAV's initial inertial horizontal speed	25m/s	Wind speed	8m/s
Simulation period	50ms	Wind direction	270deg West
Δt	50ms	$S_\chi, S_{c,\chi}$	2.4×10^{-3}
x_0	25m	$Q_\chi, Q_{c,\chi}$	2.4×10^{-3}
K_1, K_{c1}	2.0×10^{-4}	R, R_C	1.0×10^{-6}
$K_{2,min}, K_{c2,min}$	8.0×10^{-5}	N	100
$K_{2,max}, K_{c2,max}$	2.5×10^{-3}	δ	0.005
u_m	0.25rad/s	h	2.0×10^{-5}
$S_D, S_{C,D}$	2.4×10^{-6}	β	0.7
$Q_D, Q_{C,D}$	2.4×10^{-6}	$Iter_{max}$	20

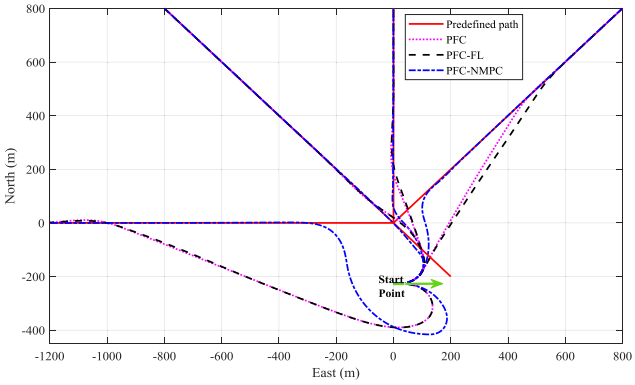

FIGURE 5. Comparison of the four directional paths following with the methods PFC, PFC_FL, and PFC_NMPC.

The initial position and course of the UAV were set as $(-160\sqrt{2}, 0)$ and 90deg East respectively. Fig. 6 presents a comparison of the distances $d(x, y)$.

The response to directional paths can be evaluated by the rise time, convergence distance, and error overshoot. The rise time is the time the UAV travels along the straight line before the absolute value of distance $|d(x, y)|$ the first time becomes less than 5m. The convergence time is the time the UAV travels along the straight line before the absolute value of distance $|d(x, y)|$ becomes less than 1.45m (defined as half of the aircraft's wingspan). The error overshoot is the maximum deviation from the straight reference line.

The performance of the responses to the four directional paths using the three methods is presented in Table 2. When following the four directional paths at 45°, 0° and −90°, the rise time and the convergence time by using the PFC_NMPC method are both shortest. When following the directional path at −45°, the corresponding rise times by using the three methods PFC, PFC_FL, and PFC_NMPC are 9.00, 9.20, and 15.70 seconds respectively. However, the corresponding convergence times are 37.35, 36.15, and 34.40 seconds respectively. Moreover, the error overshoot by using the method PFC_NMPC is only 1.65m, while the error overshoots by using the methods PFC and PFC_FL are 22.16m and 18.34m respectively. This can also be observed in Fig. 6. The above indicates that the PFC_NMPC method can make the UAV fly toward the desired straight-line path faster than the other two methods PFC and PFC_FL.

From Table 2, it can be seen that the guidance law can make the UAV have a smaller error overshoot with the parameter K_2 optimized by fuzzy logic or NMPC.

TABLE 2. The performance of the response comparison with the three methods.

Angular Discontinuity (°)	Rise Time (s)			Convergence Time (s)			Error Overshoot (m)		
	PFC	PFC_FL	PFC_NMPC	PFC	PFC_FL	PFC_NMPC	PFC	PFC_FL	PFC_NMPC
45	35.05	39.30	20.80	40.05	40.80	26.50	1.81	0.67	0.93
0	26.40	27.70	21.15	37.90	39.10	28.85	8.60	6.30	3.79
-45	9.00	9.20	15.70	37.35	36.15	34.40	22.16	18.34	1.65
-90	89.40	89.50	55.50	114.25	111.95	69.15	10.56	8.06	1.93

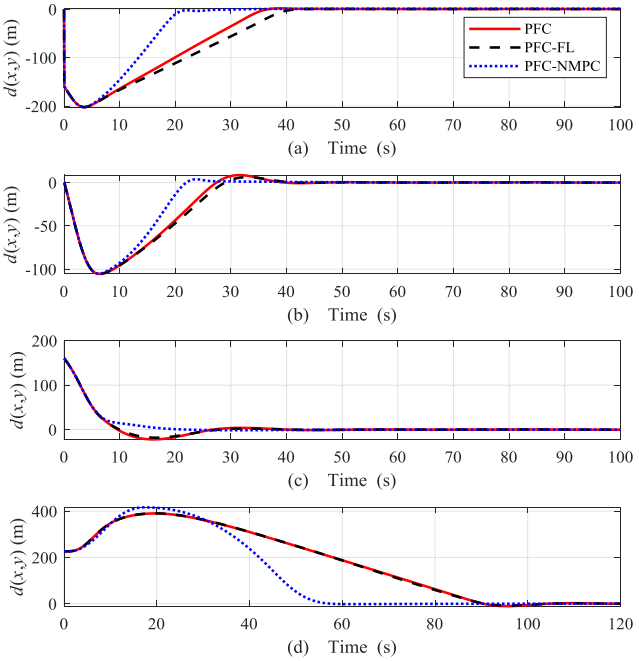

FIGURE 6. Comparison of the distance $d(x, y)$ for the directional path following with the methods PFC, PFC_FL, and PFC_NMPC. (a) straight-line heading at 45°, (b) straight-line heading at 0°, (c) straight-line heading at −45°, (d) straight-line heading at −90°.

In particular, when following the four directional paths at 0°, −45° and −90°, the corresponding error overshoots by using PFC_NMPC are 3.79m, 1.65m, and 1.93m respectively. The error overshoots derived using the methods PFC and PFC_FL are all larger than 6.30m. When following the directional path at 45°, the error overshoots by using the three methods PFC, PFC_FL, and PFC_NMPC are 1.81m, 0.67m, and 0.93m respectively. This shows that the smallest error overshoot is derived by using the PFC_FL method. However, the error overshoot by using the method PFC_NMPC is less than 1.45m. From the perspective of convergence, when the subsequent errors are less than 1.45m, the vehicle is flying along the desired path.

The above comparative analysis shows that the PFC_NMPC method can make the UAV fly to and along the desired path more smoothly and faster.

B. SQUARE PATH FOLLOWING

The desired horizontal square path was defined with four waypoints A (0, 0), B (2000, 2000), C (0, 4000), and D (2000, −2000), as shown in Fig. 8(a). The initial position and course of the UAV were set as $(0, 0)$ and 0deg north respectively. A comparison of the path following with the methods PFC, PFC_FL, and PFC_NMPC can also be seen in Fig. 8(a). Fig. 8(b) shows a comparison of the distance while following the predefined square path using the three methods. While executing the square path following, the predefined behavior that the UAV will switch segment in advance with 160 m was adopted.

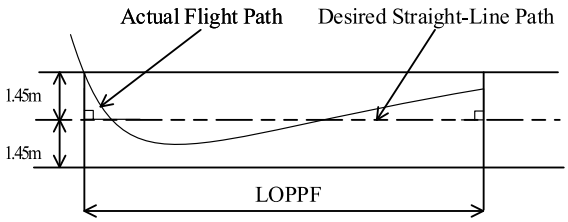

FIGURE 7. Illustration of LOPPF.

Table 3 lists the performance comparison of the square path following using the three methods PFC, PFC_FL, and PFC_NMPC. For a given predefined segment path, from the time when the error converges to less than 1.45 meters until the aircraft switches to the next path, it is called that the aircraft flies along the set path in this period of time. The length of the predefined path followed (LOPPF) is defined as the projection length of the actual flight path on a predefined segment. An illustration of LOPPF is shown in Fig. 7. It can be seen from Table 3 that when the UAV follows segments AB, BC, and DA with the PFC_NMPC method, the LOPPF is the longest compared with those by using the other two methods.

While following the segment CD, the LOPPF caused by the method PFC_NMPC is 2173.36m, which is smaller than those by using the other two methods PFC and PFC_FL. It also takes more convergence time before the UAV flies along the segment path with the PFC_NMPC method. However, the rise time caused by using the PFC_NMPC is 9.20 seconds, which is shorter than those obtained using the PFC and PFC_FL methods. Furthermore, the relative rise times obtained using the PFC_NMPC method are all shorter than those obtained using the other two methods while following the predefined segment.

After the UAV has filed a lap along the square path under the action of the three controllers with the same initial

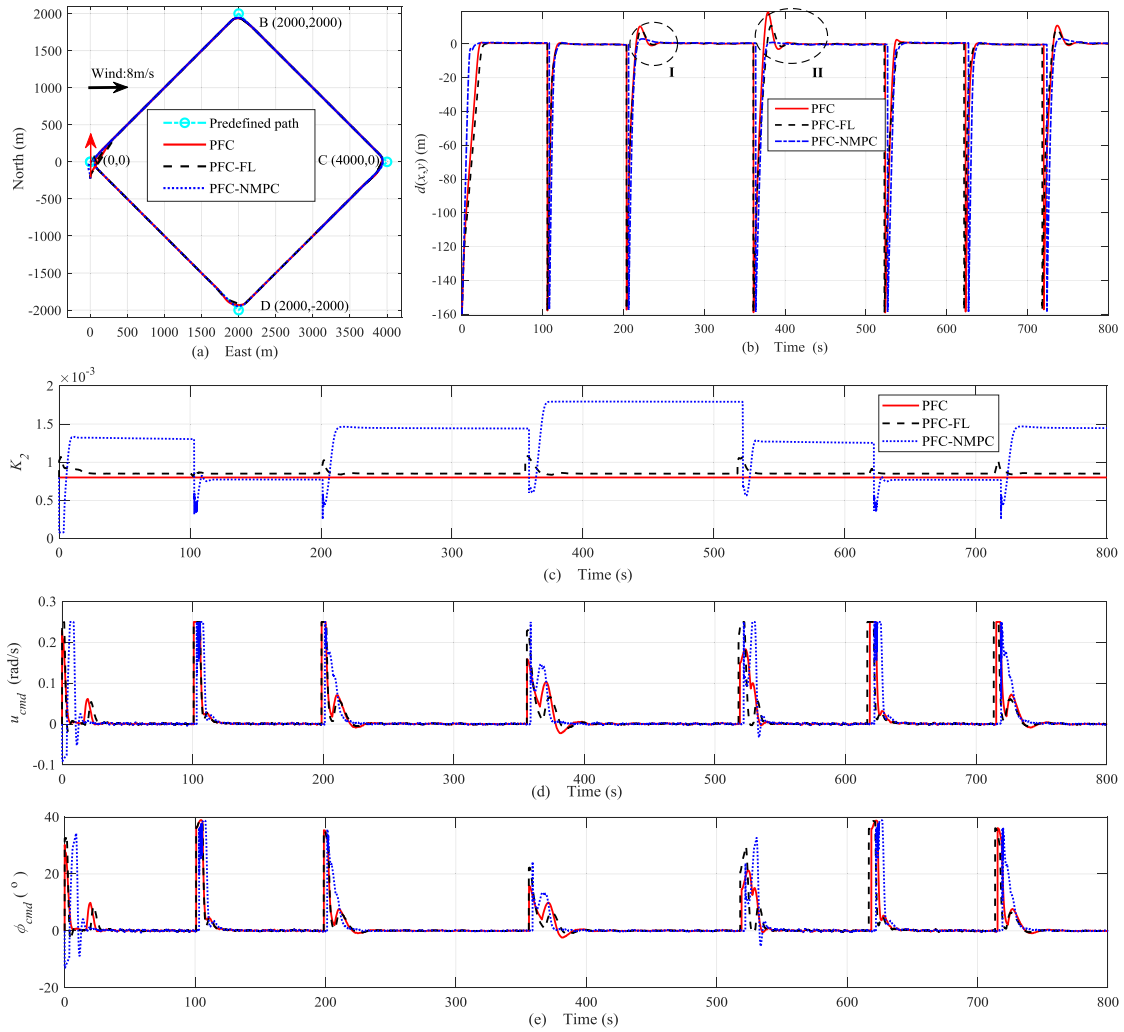

FIGURE 8. Comparison of the square path following with the methods PFC, PFC_FL, and PFC_NMPC. (a) the position, (b) the distance $d(x, y)$, (c) the parameter K_2, (d) the courserate command u_{cmd}, (e) the bank angle command ϕ_{cmd}.

TABLE 3. The performance comparison of the square path following with the three methods.

Segment (@ which lap)	LOPPF (m)			Convergence Time (s)			Rise Time (s)		
	PFC	PFC_FL	PFC_NMPC	PFC	PFC_FL	PFC_NMPC	PFC	PFC_FL	PFC_NMPC
AB (1^{st})	2274.42	2148.68	2529.12	21.40	25.65	14.75	20.25	23.75	9.55
BC (1^{st})	2423.82	2396.90	2440.14	11.40	12.35	11.00	9.40	10.05	8.50
CD (1^{st})	2268.62	2239.72	2173.36	25.55	27.20	31.55	11.05	12.30	9.20
DA (1^{st})	2162.33	2249.45	2549.37	35.40	29.50	13.30	12.15	15.90	12.00
AB (2^{nd})	2324.88	2324.03	2420.33	17.75	16.85	14.95	10.70	15.40	9.90
BC (2^{nd})	2418.05	2392.57	2444.43	11.60	12.55	10.85	9.65	10.35	8.40

conditions, the total lengths of the path followed by the UAV with the three methods PFC, PFC_FL, and PFC_NMPC are 9129.19m, 9034.75m and 9691.99m respectively. For the purpose of path following, the PFC_NMPC method can make the UAV fly more on the target path.

From Fig. 8(b), the error overshoots displayed in the elliptic marks I and II indicate that the wind will influence the path following performance if the UAV flies against the wind. Under the methods PFC, PFC_FL, and PFC_NMPC, the error overshoots in the area I are 10.48m, 8.37m, and 0.85m respectively. In area II, the corresponding error overshoots are 18.99m, 10.74m, and 1.93m respectively. This shows that the error overshoots by using the PFC_NMPC method are both much smaller than those using the PFC and PFC_FL methods.

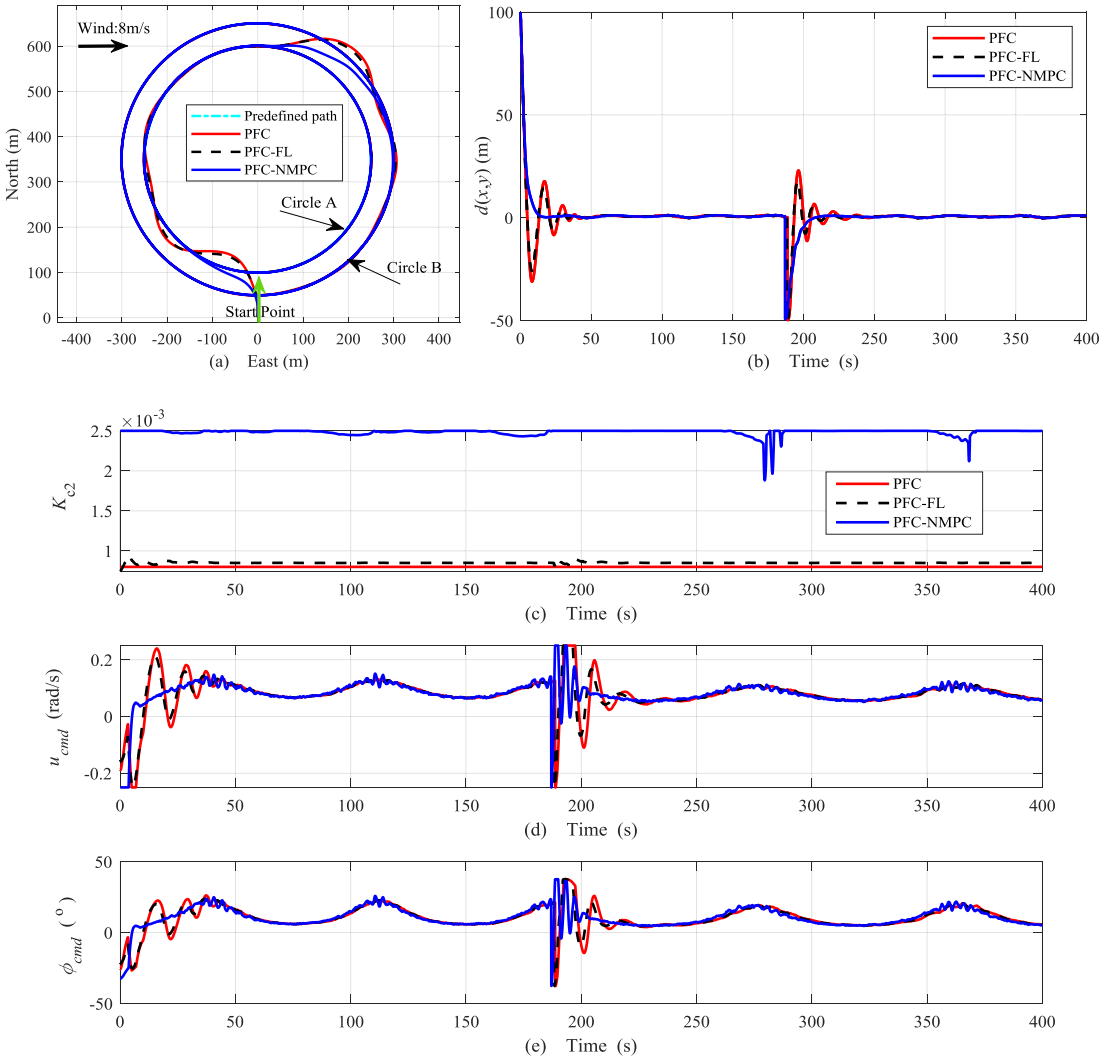

FIGURE 9. Comparison of the circular path following with the methods PFC, PFC_FL and PFC_NMPC. (a) the position, (b) the distance $d(x, y)$, (c) the parameter K_{c2}, (d) the courserate command u_{cmd}, (e) the bank angle command ϕ_{cmd}.

TABLE 4. The performance comparison of the circular path following with the three methods.

Circle	LOPCPF (m)			Convergence Time (s)		
	PFC	PFC_FL	PFC_NMPC	PFC	PFC_FL	PFC_NMPC
A	3167.98	3398.09	3769.59	40.00	31.75	12.95
B	3777.52	3818.63	4147.30	35.60	33.50	11.70

The results demonstrate the effectiveness of the PFC_NMPC method.

Fig. 8(c) presents a comparison of the parameter K_2 while following the predefined square path with the three methods PFC, PFC_FL, and PFC_NMPC. A comparison of the course rate command u_{cmd} and bank angle command ϕ_{cmd} is shown in Fig. 8(d) and Fig. 8(e), respectively. While turning near the switched points, the parameter K_2 is adjusted automatically by NMPC to adapt to the new desired segment path. The corresponding bank angle command ϕ_{cmd} will make the UAV fly toward the new desired path. After the UAV has followed the new path stably, the parameter K_2 remains relatively unchanged to ensure stable path following. It should be noted that the fuzzy logic method makes the parameter K_2 tuned based on the distance $d\ (x, y)$ and its time derivative. The NMPC method has the ability to minimize the cost function of the distance $d\ (x, y)$ and the heading error in the optimization window. The optimized parameter K_2 by NMPC is more predictable than that by fuzzy logic. The comparison of rise time shown in Table 3 also demonstrates that the predictive ability of the PFC_NMPC method can make the UAV fly toward the new segment path faster.

The above comparative analysis shows that the UAV can have the best square path following performance with the PFC_NMPC method even under horizontal wind conditions.

C. CIRCULAR PATH FOLLOWING

The initial position of the UAV was set to (0, 0). The desired path consists of two concentric circles, which are defined as $\sqrt{x^2+(y-350)^2}-250=0$ and $\sqrt{x^2+(y-350)^2}-300=0$ respectively. The UAV first flew toward circle A with a radius of 250m. Approximately 187 seconds later, the UAV started to switch from position (0, 600) and flew toward circle B with a radius of 300m.

Fig. 9(a) shows that the UAV will cross the circular path approximately four times before flying along the desired path using the PFC and PFC_FL methods. The PFC_FL method can allow the vehicle to reach the predefined path with a smaller overshoot compared with that using the PFC method. However, there is almost no overshoot when the UAV flew toward the predefined circular path with the PFC_NMPC method. The distance $d(x, y)$ presented in Fig. 9(b) can also show that with the parameter K_2 optimized by NMPC, the corresponding bank angle command ϕ_{cmd} will make the UAV fly toward the desired circular path the most smoothly. The error overshoots of $d(x, y)$ for the first circular path following using the methods PFC, PFC_FL, and PFC_NMPC are 31m, 25.80m and 0.12m respectively. When switching to fly toward a circular path with a larger radius, the error overshoots of $d(x, y)$ for the second circular path following using the methods PFC, PFC_FL, and PFC_NMPC are 23m, 17.72m and 1.16m respectively.

The length of the predefined circular path followed (LOPCPF) is defined as the length of the actual flight path after the time that the UAV has converged to the desired circular path with the absolute value of distance $|d(x, y)|$ becomes less than 1.45m (defined as half of the aircraft's wingspan). Table 4 listed the LOPCPF and the convergence time for the UAV to follow the two circles A and B. it can be found that it takes only 12.95 seconds for the UAV to converge to Circle A with the method PFC_NMPC. When using the other two methods, the convergence times were both more than 30 seconds. During the 400 seconds' flight, the LOPCPF by using the method PFC_NMPC is 7916.89m, while those with the methods PFC and PFC_FL are 6945.50m and 7216.72m respectively.

The above indicates that the PFC_NMPC method will still make the UAV have the best circular path following performance compared with those by using the PFC and PFC_FL methods.

VI. CONCLUSION

A horizontal guidance law for paths following of a small unmanned air vehicle is presented in this paper. With the combination of the Lyapunov stability and the predictive ability of NMPC via the finite horizon optimization of a kinetic model to anticipate future events, the proposed path following guidance law PFC_NMPC still has its stability guaranteed by the mechanism of the Lyapunov stability arguments and adopts the simple structure of the method PFC. With a key parameter of the controller adjusted based on the NMPC technique, the path following performance was improved. While following the straight-line path, the error overshoot by using the PFC_NMPC method is much smaller than those using the PFC and PFC_FL methods in the presence of wind at 8m/s. The predictive ability of the PFC_NMPC method can make the UAV fly toward the new segment path faster while following the desired square and circular paths. For path following, the PFC_NMPC method can make the UAV fly more on the target path than the other two methods, PFC and PFC_FL.

In the future, the performance of the controller will be tested in a Hardware-In-the-Loop simulation and the real flight. In addition, the closed-loop roll dynamics will also be considered for more precise path-following. Since the dynamics has the model uncertainties and control constraint, the adaptive fuzzy control [44], [45] and nested saturation method [1] will be applied to handle these problems. Moreover, the relationship between the curve represented by parameters and the implicit function will also be studied so that the method can adapt to more types of paths. The 3D space path-following controller with optimized parameters will be considered for further study.

APPENDIX

During optimization of the parameter K_2 with NMPC for the straight-line path guidance law, the details of the terms of the vector Lagrange multiplier sequence $\boldsymbol{\lambda}_i$ are derived as

$$\boldsymbol{\lambda}_N = \left(\frac{\partial \phi(\boldsymbol{x}_N)}{\partial \boldsymbol{x}_N}\right)^T = \begin{bmatrix} 2aS_D(ax_N+by_N+c) \\ 2bS_D(ax_N+by_N+c) \\ 2S_\chi(\chi_i-\chi_d) \end{bmatrix}^T \tag{A.1}$$

$$\frac{\partial L(\boldsymbol{x}_i, K_{2,i})}{\partial \boldsymbol{x}_i} = \begin{bmatrix} 2aQ_D(ax_i+by_i+c)+2Ru_i\frac{\partial u_i}{\partial x_i} \\ 2bQ_D(ax_i+by_i+c)+2Ru_i\frac{\partial u_i}{\partial y_i} \\ 2Q_\chi(\chi_i-\chi_d)+2Ru_i\frac{\partial u_i}{\partial \chi_i} \end{bmatrix}^T \tag{A.2}$$

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x}_i, K_{2,i})}{\partial \boldsymbol{x}_i} = \begin{bmatrix} 1 & 0 & -\Delta t V_g sin(\chi_i) \\ 0 & 1 & \Delta t V_g cos(\chi_i) \\ \Delta t\frac{\partial u_i}{\partial x_i} & \Delta t\frac{\partial u_i}{\partial y_i} & 1+\Delta t\frac{\partial u_i}{\partial \chi_i} \end{bmatrix} \tag{A.3}$$

$$\begin{bmatrix} \frac{\partial u_i}{\partial x_i} \\ \frac{\partial u_i}{\partial y_i} \\ \frac{\partial u_i}{\partial \chi_i} \end{bmatrix} = V_g \begin{bmatrix} -K_1 a f'_{sat}(ax_i+by_i+c) \\ -K_1 b f'_{sat}(ax_i+by_i+c) \\ K_{2,i}(aV_g sin(\chi_i)-bV_g cos(\chi_i)) \end{bmatrix} \tag{A.4}$$

Because the saturation function $f_{sat}(x)$ is continuous, but it is non-differentiable when $x = \pm x_0$, the function $f'_{sat}(x)$ above is arbitrarily defined as

$$f'_{sat}(x) = \begin{cases} 1, & if \ -x_0 \le x \le x_0 \\ 0, & else \end{cases} \tag{A.5}$$

$$\boldsymbol{\lambda}_{c,N} = \left(\frac{\partial \phi_c(\boldsymbol{x}_{c,N})}{\partial \boldsymbol{x}_{c,N}}\right)^T = \begin{bmatrix} 2S_{c,D}(d_N - C_R) \\ -2S_{c,\chi}(\chi_{c,N} - \gamma_N + \pi/2) \\ 2S_{c,\chi}(\chi_{c,N} - \gamma_N + \pi/2) \end{bmatrix}^T \tag{A.8}$$

$$\frac{\partial L_c(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial \boldsymbol{x}_{c,i}} = \begin{bmatrix} 2Q_{c,D}(d_i - C_R) + 2R_c u_{c,i} \frac{\partial u_{c,cmd,i}}{\partial d_i} \\ -2Q_{c,\chi}(\chi_{c,i} - \gamma_i + \pi/2) + 2R_c u_{c,i} \frac{\partial u_{c,cmd,i}}{\partial \gamma_i} \\ 2Q_{c,\chi}(\chi_{c,i} - \gamma_i + \pi/2) + 2R_c u_{c,i} \frac{\partial u_{c,cmd,i}}{\partial \chi_{c,i}} \end{bmatrix}^T \tag{A.9}$$

$$\frac{\partial \boldsymbol{f}_c(\boldsymbol{x}_{c,i}, K_{c2,i})}{\partial \boldsymbol{x}_{c,i}} = \begin{bmatrix} 1 & \Delta t V_g sin(\chi_{c,i} - \gamma_i) & -\Delta t V_g sin(\chi_{c,i} - \gamma_i) \\ -\Delta t \frac{V_g sin(\chi_{c,i} - \gamma_i)}{d_i^2} & 1 - \Delta t \frac{V_g cos(\chi_{c,i} - \gamma_i)}{d_i} & \Delta t \frac{V_g cos(\chi_{c,i} - \gamma_i)}{d_i} \\ \Delta t \frac{\partial u_{c,cmd,i}}{\partial d_i} & \Delta t \frac{\partial u_{c,cmd,i}}{\partial \gamma_i} & 1 + \Delta t \frac{\partial u_{c,cmd,i}}{\partial \chi_{c,i}} \end{bmatrix} \tag{A.10}$$

$$\begin{bmatrix} \frac{\partial u_{c,cmd,i}}{\partial d_i} \\ \frac{\partial u_{c,cmd,i}}{\partial \gamma_i} \\ \frac{\partial u_{c,cmd,i}}{\partial \chi_{c,i}} \end{bmatrix} = \begin{bmatrix} -K_{c1} V_g f'_{sat}(d_i - C_R) - \frac{V_g sin(\chi_{c,i} - \gamma_i)}{d_i^2} \\ -K_{c2,i} V_g^2 sin(\chi_{c,i} - \gamma_i) - \frac{V_g cos(\chi_{c,i} - \gamma_i)}{d_i} \\ K_{c2,i} V_g^2 sin(\chi_{c,i} - \gamma_i) + \frac{V_g cos(\chi_{c,i} - \gamma_i)}{d_i} \end{bmatrix} \tag{A.11}$$

The gradient of Hamiltonian with respect to the control parameter K_2 are derived as

$$\frac{\partial L(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} = -2Ru_i V_g(aV_g cos(\chi_i) + bV_g sin(\chi_i)) \tag{A.6}$$

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x}_i, K_{2,i})}{\partial K_{2,i}} = \begin{bmatrix} 0 \\ 0 \\ -\Delta t V_g(aV_g cos(\chi_i) + bV_g sin(\chi_i)) \end{bmatrix} \tag{A.7}$$

During optimization of the parameter K_{c2} with NMPC for the circular path guidance law, the details of terms of the vector Lagrange multiplier sequence $\boldsymbol{\lambda}_{c,i}$ are derived as (A.8)–(A.11), shown at the top of the page.

The gradient of Hamiltonian with respect to the control parameter K_{c2} are derived as:

$$\frac{\partial L_c\left(\boldsymbol{x}_{c,i}, K_{c2,i}\right)}{\partial K_{c2,i}} = -2R_c u_{c,cmd,i} V_g^2 cos(\chi_{c,i} - \gamma_i) \tag{A.12}$$

$$\frac{\partial \boldsymbol{f}_c\left(\boldsymbol{x}_{c,i}, K_{c2,i}\right)}{\partial K_{c2,i}} = -\boldsymbol{\Delta} t \begin{bmatrix} 0 \\ 0 \\ V_g^2 cos(\chi_{c,i} - \gamma_i) \end{bmatrix} \tag{A.13}$$

REFERENCES

[1] S. Zhao, X. Wang, D. Zhang, and L. Shen, "Curved path following control for fixed-wing unmanned aerial vehicles with control constraint," *J. Intell. Robotic Syst.*, vol. 89, nos. 1–2, pp. 107–119, Jan. 2018.

[2] D. D. Nguyen, J. Rohacs, and D. Rohacs, "Autonomous flight trajectory control system for drones in smart city traffic management," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 5, p. 338, May 2021.

[3] A. Lucieer, D. Turner, D. H. King, and S. A. Robinson, "Using an unmanned aerial vehicle (UAV) to capture micro-topography of Antarctic moss beds," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 27, pp. 53–62, Apr. 2014.

[4] H. Chen, Y. Cong, X. Wang, X. Xu, and L. Shen, "Coordinated path-following control of fixed-wing unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jan. 25, 2021, doi: 10.1109/TSMC.2021.3049681.

[5] X. Fu, J. Pan, H. Wang, and X. Gao, "A formation maintenance and reconstruction method of UAV swarm based on distributed control," *Aerosp. Sci. Technol.*, vol. 104, Sep. 2020, Art. no. 105981.

[6] T. Z. Muslimov and R. A. Munasypov, "Adaptive decentralized flocking control of multi-UAV circular formations based on vector fields and backstepping," *ISA Trans.*, vol. 107, pp. 143–159, Dec. 2020.

[7] J. P. Wilhelm and G. Clem, "Vector field UAV guidance for path following and obstacle avoidance with minimal deviation," *J. Guid., Control, Dyn.*, vol. 42, no. 8, pp. 1848–1856, Aug. 2019.

[8] J. Keller, D. Thakur, M. Likhachev, J. Gallier, and V. Kumar, "Coordinated path planning for fixed-wing UAS conducting persistent surveillance missions," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 17–24, Jan. 2017.

[9] X. Yu, J. Yang, and S. Li, "Finite-time path following control for small-scale fixed-wing UAVs under wind disturbances," *J. Franklin Inst.*, vol. 357, no. 12, pp. 7879–7903, Aug. 2020.

[10] M. Sun, R. Zhu, and X. Yang, "UAV path generation, path following and gimbal control," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Apr. 2008, pp. 870–873.

[11] C. Jarrett, K. Perry, and K. A. Stol, "Controller comparisons for autonomous railway following with a fixed-wing UAV," in *Proc. 6th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2015, pp. 104–109.

[12] I. Rhee, S. Park, and C. K. Ryoo, "A tight path following algorithm of an UAS based on PID control," in *Proc. SICE Annu. Conf.*, Aug. 2010, pp. 1270–1273.

[13] S. Kohno and K. Uchiyama, "Design of robust controller of fixed-wing UAV for transition flight," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2014, pp. 1111–1116.

[14] B. K. Wilburn, M. G. Perhinschi, and J. N. Wilburn, "A modified genetic algorithm for UAV trajectory tracking control laws optimization," *Int. J. Intell. Unmanned Syst.*, vol. 2, no. 2, pp. 58–90, May 2014.

[15] A. Ratnoo, P. B. Sujit, and M. Kothari, "Optimal path following for high wind flights," in *Proc. 18th IFAC World Cong.*, Aug. 2011, pp. 12985–12990.

[16] S. Lee, A. Cho, and C. Kee, "Integrated waypoint path generation and following of an unmanned aerial vehicle," *Aircr. Eng. Aerosp. Technol.*, vol. 82, no. 5, pp. 296–304, Sep. 2010.

[17] L. Cancemi, M. Innocenti, and L. Pollini, "Hardware implementation of a fuzzy guidance system with prescribed waypoint approach trajectory," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 5254–5259.

[18] S. Back, G. Cho, J. Oh, X.-T. Tran, and H. Oh, "Autonomous UAV trail navigation with obstacle avoidance using deep neural networks," *J. Intell. Robotic Syst.*, vol. 100, nos. 3–4, pp. 1195–1211, Sep. 2020.

[19] Y. Zhang, Y. Zhang, and Z. Yu, "Path following control for UAV using deep reinforcement learning approach," *Guid., Navigat. Control*, vol. 1, no. 1, pp. 1–18, Mar. 2021.

[20] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 519–529, Jun. 2007.

[21] R. W. Beard, J. Ferrin, and J. Humpherys, "Fixed wing UAV path following in wind with input constraints," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2103–2117, Nov. 2014.

[22] S. Fari, X. Wang, S. Roy, and S. Baldi, "Addressing unmodeled path-following dynamics via adaptive vector field: A UAV test case," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1613–1622, Apr. 2020.
[23] S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen, "Integrating vector field approach and Input-to-State stability curved path following for unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 8, pp. 2897–2904, Aug. 2020.
[24] Y. Liang, Y. Jia, Z. Wang, and F. Matsuno, "Combined vector field approach for planar curved path following with fixed-wing UAVs," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 5980–5985.
[25] Y. Liang, Y. Jia, J. Du, and J. Zhang, "Vector field guidance for three-dimensional curved path following with fixed-wing UAVs," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 1187–1192.
[26] D. Jung and P. Tsiotras, "Bank-to-turn control for a small UAV using backstepping and parameter adaptation," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 4406–4411, 2008.
[27] A. Brezoescu, T. Espinoza, P. Castillo, and R. Lozano, "Adaptive trajectory following for a fixed-wing UAV in presence of crosswind," *J. Intell. Robotic Syst.*, vol. 69, nos. 1–4, pp. 257–271, Jan. 2013.
[28] M. Kothari, I. Postlethwaite, and D.-W. Gu, "A suboptimal path planning algorithm using rapidly-exploring random trees," *Int. J. Aerosp. Innov.*, vol. 2, nos. 1–2, pp. 93–104, Apr. 2010.
[29] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov stability of a nonlinear path following guidance method," *J. Guid., Control, Dyn.*, vol. 30, no. 6, pp. 1718–1728, Nov. 2007.
[30] I. Lugo-Cárdenas, S. Salazar, and R. Lozano, "Lyapunov based 3D path following kinematic controller for a fixed wing UAV," *IFAC PapersOnLine*, vol. 51, no. 1, pp. 15946–15951, Oct. 2017.
[31] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles," *IEEE Control Syst. Mag.*, vol. 34, no. 1, pp. 42–59, Feb. 2014.
[32] J. Sprinkle, J. M. Eklund, H. J. Kim, and S. Sastry, "Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, Dec. 2004, pp. 2609–2614.
[33] N. Slegers, J. Kyle, and M. Costello, "Nonlinear model predictive control technique for unmanned air vehicles," *J. Guid., Control Dyn.*, vol. 29, no. 5, pp. 1179–1188, 2006.
[34] Y. Hamada, T. Tsukamoto, and S. Ishimoto, "Receding horizon guidance of a small unmanned aerial vehicle for planar reference path following," *Aerosp. Sci. Technol.*, vol. 77, pp. 129–137, Jun. 2018.
[35] Y. Kang and J. K. Hedrick, "Linear tracking for a fixed-wing UAV using nonlinear model predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1202–1210, Sep. 2009.
[36] K. Yang, Y. Kang, and S. Sukkarieh, "Adaptive nonlinear model predictive path-following control for a fixed-wing unmanned aerial vehicle," *Int. J. Control, Autom. Syst.*, vol. 11, no. 1, pp. 65–74, Jan. 2013.
[37] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," *Int. J. Robust Nonlinear Control*, vol. 25, no. 8, pp. 1168–1182, 2015.
[38] I. Sánchez, A. D'Jorge, G. V. Raffo, A. H. González, and A. Ferramosca, "Nonlinear model predictive path following controller with obstacle avoidance," *J. Intell. Robotic Syst.*, vol. 102, no. 1, pp. 1–18, May 2021, doi: 10.1007/s10846-021-01373-7.
[39] Y. Chen, T. Wang, J. Liang, C. Wang, and C. Xue, "A fuzzy robust path following controller for a small unmanned air vehicle," in *Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jul. 2012, pp. 1189–1194.
[40] Y. Chen and C. Wang, "A robust path following controller for a small unmanned aerial vehicle with constrained parameters optimized," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 789–793.
[41] T. Wang, Y. Chen, J. Liang, C. Wang, and Y. Zhang, "Combined of vector field and linear quadratic Gaussian for the path following of a small unmanned helicopter," *IET Control Theory Appl.*, vol. 6, no. 17, pp. 2696–2703, Nov. 2012.
[42] H. K. Khalil, *Nonlinear Systems*. 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002, pp. 147–156.
[43] G. J. Sutton and R. R. Bitmead, "Performance and computational implementation of nonlinear model predictive control on a submarine," *Prog. Syst. Control Theory*, vol. 26, pp. 461–472, Jan. 2000.
[44] T. Yang, N. Sun, and Y. Fang, "Adaptive fuzzy control for a class of MIMO underactuated systems with plant uncertainties and actuator deadzones: Design and experiments," *IEEE Trans. Cybern.*, early access, Feb. 2, 2021, doi: 10.1109/TCYB.2021.3050475.
[45] T. Yang, N. Sun, and Y. Fang, "Adaptive fuzzy control for uncertain mechatronic systems with state estimation and input nonlinearities," *IEEE Trans. Ind. Informat.*, early access, Jun. 14, 2021, doi: 10.1109/TII.2021.3089143.

YANG CHEN received the B.S. degree in material forming and control engineering from Beihang University, Beijing, China, in 2007, and the Ph.D. degree in mechatronics engineering from the Robotics Institute, Beihang University, in 2013.

From 2013 to 2014, he was an Engineer with Shanghai Aerospace Control Technology Research Institute. Afterwards, he joined the School of Physics and Mechatronics Engineering, Longyan University, Longyan, China, where he is currently an Associate Professor. His research interests include system modeling, parameter identification, and intelligent control of small UAVs and ground mobile robots.

CHAOLEI WANG received the B.S. degree in material forming and control engineering from Beihang University, Beijing, China, in 2008, and the Ph.D. degree in mechatronics engineering from the Robotics Institute, Beihang University, in 2014.

Afterwards, he joined the Science and Technology on Special System Simulation Laboratory, Beijing, where he is currently a Senior Engineer. His research interests include information fusion, object recognition, and modeling and control of unmanned systems.

WEI ZENG received the B.S. degree in mechanical engineering and automation from the South China University of Technology, Guangzhou, China, in 2002, the M.S. degree in control engineering from Xiamen University, Xiamen, China, in 2008, and the Ph.D. degree in control theory and control engineering from the South China University of Technology, in 2012.

Since 2003, he has been with the School of Physics and Mechatronics Engineering, Longyan University, Longyan, China, where he is currently a Professor. From 2012 to 2014, he was a Postdoctoral Research Fellow with the School of Mechanical and Automotive Engineering, South China University of Technology. From 2017 to 2018, he was a Visiting Scholar with the Faculty of Health Sciences, The University of Sydney, Australia. His research interests include nonlinear adaptive system identification and control, intelligent robot control, dynamic pattern recognition, and its applications in gait recognition, gesture recognition, and biomedicine.

YONGLIANG WU received the B.S. degree in mechanical engineering and automation from Beihang University, Beijing, China, in 2005, and the Ph.D. degree in mechatronics engineering from the Robotics Institute, Beihang University, in 2011.

From 2011 to 2019, he was a Senior Engineer with Chengdu Aircraft Design and Research Institute, Chengdu, China. From 2013 to 2015, he was a Postdoctoral Research Fellow with the School of Aerospace Engineering, Tsinghua University, Beijing. Since 2019, he has been with the Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen, China, where he is currently an Associate Researcher. His research interests include information fusion, the system design and flight control of UAVs, path planning, and nonlinear optimization.