

Modified vector field and nonlinear guidance law for low-cost UAV path following

Mo He, Xiaogang Wang and Naigang Cui

School of Astronautics, Harbin Institute of Technology, Harbin, China

Abstract

Purpose – The purpose of this paper is to present a high accuracy path following method for low-cost fixed-wing UAVs.

Design/methodology/approach – The original vector field (VF) algorithm is condensed. A spatial integration mechanism is added to the existing VF and nonlinear guidance law, aiming to decrease steady-state cross-track-error and cope with long-term disturbance.

Findings – Numerical simulations show the proposed method could diminish steady-state cross-track-error effectively. Test flights show the proposed method is applicable on low-cost fixed-wing UAVs.

Practical implications – The path following accuracy shown in simulations and test flights indicates the proposed method could be deployed in scenarios including inflight rendezvous, formation, trafficway take-off and landing.

Originality/value – This paper provides an improved high-accuracy path following method for low-cost fixed-wing UAVs.

Keywords Path following, Flight control, UAV, Vector field, Nonlinear guidance law, Flight test

Paper type Research paper

Introduction

In the past decade, advances in micro-electronics, sensing technology and signal processing have endowed low-cost miniature unmanned aircrafts with the ability to perform increasingly diverse tasks. These new tasks include formation flight, airborne rendezvous, autonomous moving platform landing and surface target close investigating (DoD, 2003). Such tasks impose higher demands on path following accuracy. Therefore, accurate path following has become one of the key challenges in achieving full autonomy of UAVs.

UAV missions can usually be resolved into straight line path following and loiter problems. Most of previous works use one general method to fulfil these assignments. Overviews of the main known methods to solve the path following problem and a comparison of the performance of different algorithms can be found in [Xavier et al. \(2018\)](#), [Sujit et al. \(2014\)](#). Solution strategies for the path following problem are either geometric or control theoretic. Geometric techniques include pure pursuit ([Conte et al., 2004](#)), line of sight (LOS) ([Fossen et al., 2003](#); [Osborne and Rysdyk, 2005](#); [Rysdyk, 2006](#); [Ambrosino et al., 2009](#); [Zhang et al., 2018](#)) and variants ([Nelson et al., 2007](#); [Park et al., 2007](#)). A 2D path-following algorithm using Dubins curves was developed in [Dubins \(1957\)](#) and modified for 3D scenarios in [Hota and Ghose \(2009\)](#). The approach based on vector field (VF) is another alternative. An VF-based, easy-to-implement path following algorithm was developed in [Nelson et al. \(2007\)](#). Control theoretic methods constitute the other major category of path following methods. PID control, being

widely used in various engineering practices, can also be applied to solve path following problems ([Rankin et al., 1998](#); [Sun et al., 2008](#)). A nonlinear guidance law (NLGL) developed in [Park et al. \(2007\)](#) shows better performance than the standard PID controller. The NLGL uses a virtual target point (VTP) concept. It can be applied to both straight and circular trajectory, which is a prominent merit when designing the actual guidance system. A PID controller developed in [Rhee et al. \(2010\)](#) with feed-forward mechanism shows the performance comparable to that of NLGL. Other control theoretic techniques include linear quadratic regulators (LQR) ([Ratnoo et al., 2011](#)), sliding mode control ([Healey and Lienard, 1993](#)), model predictive control ([Jackson et al., 2008](#)), backstepping control ([Ahmed and Subbarao, 2010](#)), adaptive control ([Cao et al., 2007](#); [Kaminer et al., 2007](#)) and dynamic programming ([Silva and Sousa, 2011](#)).

Study in this paper is built on the works mentioned above, trying to modify existing methods to achieve an even higher path following accuracy. Combination of geometric and control theoretic approach is adopted in this paper. The merits of selected path following methods are analysed before modifications are made. Simulations are then performed to validate the proposed methods. Flight test is conducted on a low-cost UAV to check the applicability and actual performance of the proposed methods in the real-world environment.

© Mo He, Xiaogang Wang and Naigang Cui. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

The current issue and full text archive of this journal is available on Emerald Insight at: <https://www.emerald.com/insight/1748-8842.htm>



Aircraft Engineering and Aerospace Technology
94/11 (2022) 20–28
Emerald Publishing Limited [ISSN 1748-8842]
[DOI [10.1108/AEAT-03-2019-0045](https://doi.org/10.1108/AEAT-03-2019-0045)]

Received 9 March 2019
Revised 30 March 2020
27 September 2021
Accepted 5 May 2022

Algorithms description

Straight line following

According to [Xavier et al. \(2018\)](#) and [Sujit et al. \(2014\)](#), the VF algorithm ([Nelson et al., 2007](#)) gives the best overall path following performance. The basic VF algorithm is described below.

As shown in [Figure 1](#), in the east-north-up (ENU) coordinate system, we consider a 2D plane flight scenario. Let U be the position of the UAV. A start point, P_n , and an end point, P_{n+1} , define a straight line path to be followed. The variable V_g is the ground speed of the UAV and ψ_v is the trajectory deflection angle. ψ_v is defined as the angle between the north direction and the ground speed vector. The path direction is denoted as χ . In the VF algorithm, the target course angle is calculated as:

$$\psi_c = \chi - \frac{1}{\alpha} \chi^\infty \frac{2}{\pi} \frac{k}{1 + (k\varepsilon)^2} V_g \sin(\psi_v - \chi) - \frac{\kappa}{\alpha} \text{sat} \left(\frac{\chi^\infty \frac{2}{\pi} \arctan(k\varepsilon)}{\xi} \right) \quad (1)$$

$$\text{sat}(x) = \begin{cases} x, & \text{if } |x| \leq 1 \\ \text{sign}(x), & \text{otherwise} \end{cases} \quad (2)$$

where α , k , χ^∞ , κ and ξ are control parameters. Their values given in original literatures ([Sujit et al., 2014](#); [Nelson et al., 2007](#)) are listed in [Table 1](#).

Though the VF algorithm is usually considered to be a geometric approach ([Sujit et al., 2014](#)), it is theoretically a feedback controller. In (1), the first term is the predetermined path heading. This could be regarded as a feedforward term, telling the information about the overall control objective. The

Table 1 Parameters used for original VF

Parameter	Value
α	1.65 rad/s
k	0.02 m ⁻¹
χ^∞	$\pi/2$ rad
κ	$\pi/2$ rad ² /s
ξ	1.0

second term includes velocity information about the vehicle. Let ε be the cross-track error, $\dot{\varepsilon}$ be the cross-track error changing rate, then:

$$\dot{\varepsilon} = V_g \sin(\psi_v - \chi) \quad (3)$$

Because $k/[1 + (k\varepsilon)^2]$ is almost invariant near the target path when using parameters stated in [Xavier et al. \(2018\)](#), [Nelson et al. \(2007\)](#), let:

$$K_d = \frac{1}{\alpha} \chi^\infty \frac{2}{\pi} \frac{k}{1 + (k\varepsilon)^2} \quad (4)$$

The second term in (1) then acts as a derivative part in a PID controller.

The third term is a variation of proportional cross track error feedback. Using the parameters above, we get:

$$\begin{aligned} \frac{\kappa}{\alpha} \text{sat} \left(\frac{\chi^\infty \frac{2}{\pi} \arctan(k\varepsilon)}{\xi} \right) &= \frac{\pi/2}{1.65} \text{sat} \left(\frac{\arctan(0.02\varepsilon)}{1} \right) \\ &= \frac{\pi}{3.3} \text{sat}(\arctan(0.02\varepsilon)) \\ &\approx \text{sat}(\arctan(0.02\varepsilon)) \end{aligned} \quad (5)$$

The third term now acts as look ahead steering formula. The proportional gain k is the reciprocal of the look ahead distance ([Breivik and Fossen, 2008](#)). When $k = 0.02 \text{ m}^{-1}$, the look ahead distance is 50 m.

An PD controller could be derived to approximate (1):

$$\psi_c = \chi - K_d \dot{\varepsilon} - \arctan(K_p \varepsilon) \quad (6)$$

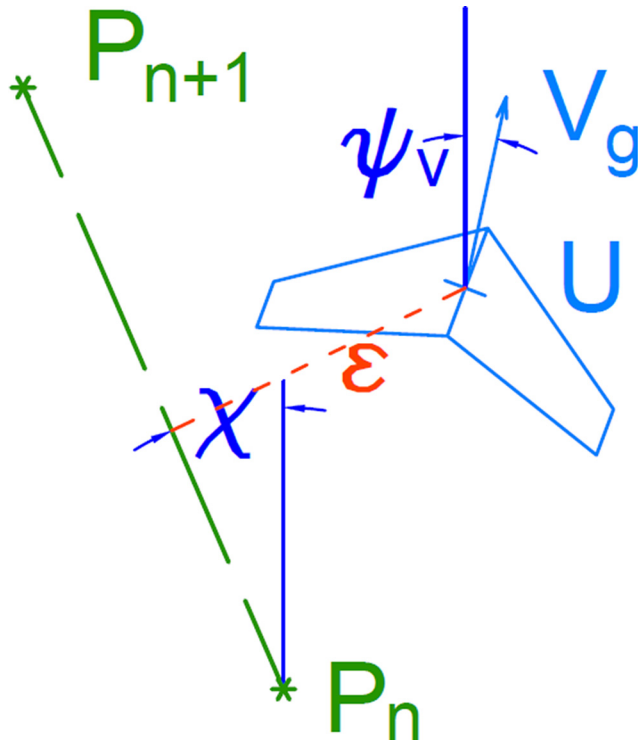
One important reason for VF algorithm outperforming other guidance laws is the algorithm containing a damping term which explicitly tells the changing rate of cross-track error. This damping mechanism ensures a higher bandwidth for the controller and keeps the controller stable when high error feedback gain is posed.

To diminish long term disturbance and decrease cross-track error, a spatial integration ([Davidson et al., 2002](#); [Davidson and Bahl, 2001](#)) on tracking error is added. In this paper, this new algorithm is named modified vector field (MVF) algorithm:

$$\psi^c = \chi - K_d \dot{\varepsilon} - \arctan(K_p \varepsilon) - K_i \int_0^t \varepsilon(\tau) V_g \cos(\psi_v - \chi) d\tau \quad (7)$$

Suppose (7) is deployed on a bank-to-turn UAV, the roll command φ_r is then proposed as:

Figure 1 Straight line following illustration



$$\begin{aligned}\varphi_r &= K_{h2r}(\psi^c - \psi_v) \\ &= K_{h2r} \left(\chi - K_d \dot{\varepsilon} - \arctan(K_p \varepsilon) - K_i \int_0^t \varepsilon(\tau) V_g \cos(\psi_v - \chi) d\tau - \psi_v \right)\end{aligned}\quad (8)$$

Arc following

The NLGL method (Park *et al.*, 2007) naturally follows any circular path when appropriate parameters are given. Analyses in Xavier *et al.* (2018), Sujit *et al.* (2014) show NLGL has lower cross-track error for arc path compared with that of the VF algorithm.

An arc path is given by a centre point P_n and a radius R . The variable L denotes the leading line and η is the angle between track made good and the leading line (Figure 2). The demanding side acceleration is calculated as (Park *et al.*, 2007):

$$a_s = 2 \frac{V_g^2}{L} \sin \eta \quad (9)$$

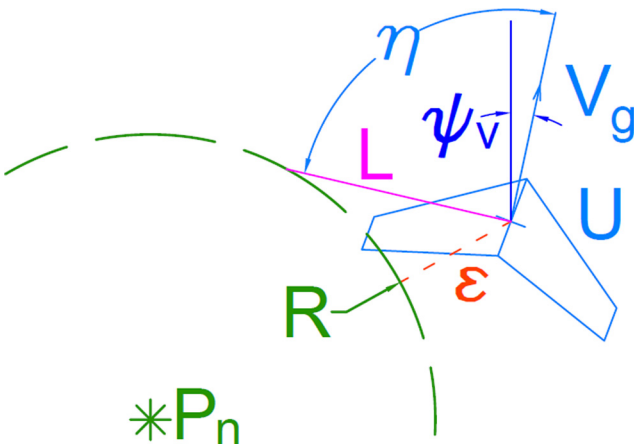
The angle η contains both information about the cross-track error and its trend of changing. Thus NLGL could achieve similar performance of a feedforward PID controller with only one parameter to tune. Though loiter curvature compensation and side movement damping are already included in (9), long term disturbance feedback is yet to be added. Same as the MVF algorithm shown in (7), spatial integration on cross-track error is applied. The new algorithm, (10), is named modified nonlinear guidance law (MNLGL) in this paper:

$$a_s = 2 \frac{V_g^2}{L} \sin \eta - K_{ai} \int_0^t \varepsilon(\tau) V_g \cos(\psi_v - \chi) d\tau \quad (10)$$

On a bank-to-turn UAV, the roll command φ_r is then calculated as:

$$\varphi_r = \arctan(a_s/9.8) = \arctan \left(\frac{1}{9.8} \left(2 \frac{V_g^2}{L} \sin \eta - K_{ai} \int_0^t \varepsilon(\tau) V_g \cos(\psi_v - \chi) d\tau \right) \right) \quad (11)$$

Figure 2 Arc following illustration



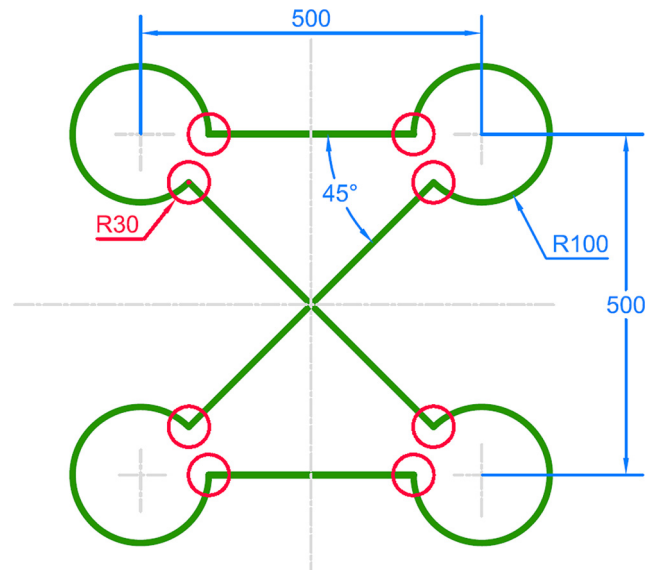
The stability of both (1) and (9) have been proved in their original literatures. In this paper, the main modification is adding a spatial integration term. Assuming this newly added term be a disturbance to the original system, the stability will still hold if this term is bounded and this bound is small enough, which is always true in practice.

Simulation

6-DOF numerical simulation is conducted in Simulink to verify the proposed path following method. Mathematical model of UAVs presented in Beard and McLain (2010) is adopted in this paper. An S-Function block programmed in C language is used to render proposed algorithm. These would simplify the code migration for successive flight experiment and data comparison. The target vehicle is a 1.5 m-wingspan delta wing flying with a 13 m/s ground speed. The reference path is shown in Figure 3, which is of the same size and pattern with the one used in Sujit *et al.* (2014). Distances in Figure 3 are measured in metres. A “switching circle” mechanism is used to update reference path segments for simplicity. Circles with specific radius are placed at the end of every path segment (see red circles in Figure 3). Centres of these switching circles coincide with the endpoints of path segments. Once the UAV enters a switching circle, the corresponding path segment is considered completed and the next path segment will be loaded to the guidance system. Radius of the switching circles is set to 30 m in this simulation.

The simulation propagates with a time step of 0.01 s, lasting for 290 s. Attitude estimating algorithm and attitude controller are updated every step (100 Hz). Positioning information for guidance loop is updated at 20 Hz. These simulation parameters are set based on average modern low-cost hardware performance (Year 2014–2019). The flight vehicle is launched at the centre point with an initial ground speed of 13 m/s heading north (+Y). A 5 m/s wind alternating its direction between the east and the west every 25 s is applied. GNSS time lag is approximated as a first order inertia link with a time constant of 0.1 s. Other lag factors including actuator lag and sensor delay are omitted

Figure 3 Reference path illustration



because their transient process is much faster than the guidance loop bandwidth. Guidance parameters used for simulation are shown in Table 2.

1-norm and 2-norm are chosen to summarize the cross-track error (ε) of the whole session. They are calculated as:

$$\|\varepsilon_1\| = \sum_{i=1}^N |\varepsilon_i| \quad (12)$$

$$\|\varepsilon_2\| = \sqrt{\sum_{i=1}^N \varepsilon_i^2} \quad (13)$$

where N is the total number of samples. For the simulation specified above, $N = 29,000$.

Flight trajectory of the simulating scenario is shown in Figure 4, where +X is aligned with east direction, +Y is aligned with north, and the origin is located at the geometric centre of the reference path. Pink arrows depict the flying direction. Detail views of straight segment and loiter segment show that the proposed path

Table 2 Parameters used for simulation

Straight line following		Arc following	
Parameter	Value	Parameter	Value
K_{h2r}	0.8	L	33.3 m
K_p	5.2×10^{-3} rad/m	K_{ai}	$0.03 \text{ m}^{-1} \text{ s}^{-2}$
K_i	5.2×10^{-5} rad/m ²		
K_d	1.7×10^{-2} rad/(m s ⁻¹)		

following method could effectively diminish the steady-state cross-track error in both straight line and arc following. For the combination of original VF and NLGL, the 1-norm of cross-track error is 7.90×10^4 and the 2-norm is 8.62×10^2 . For the propose path following method, the 1-norm of cross-track error is 6.04×10^4 and the 2-norm is 8.21×10^2 . The absolute value of cross-track error is plotted in Figure 5. Because of the switching circle mechanism, the cross-track error after loading a new reference path segment is always about 30 m, which constitutes those cusps in Figure 5.

One representative transient segment (samples 12,376–14,874, corresponding to the fourth cusp) is shown in Figure 6. With the proposed path following method, the cross-track error decreases faster after segment switching. This is mainly because of the spatial integration mechanism added in (8) and (11). When flying along the reference path after the approach process, newly added spatial integration mechanism helps the proposed path following method achieve a higher steady state accuracy than the original method.

With cusps corresponding to reference path segment switching trunked, Figure 7 shows the absolute value of cross-track error for steady flight periods. 1-norm and 2-norm of cross-track error for the original method are 4.84×10^4 and 3.38×10^2 ; for the proposed method are 2.56×10^4 and 2.34×10^2 , respectively (Table 3).

Average cross-track error could be calculated dividing ε_1 by number of samples (Table 4). It can be seen from Tables 3 and 4 that the path following accuracy of the proposed method has been evidently improved.

Flight test

Flight test is conducted on a carbon and foam version of Zagi fly wing (Beard and McLain, 2010; Nelson et al., 2007) (Figure 8).

Figure 4 Simulation flight path

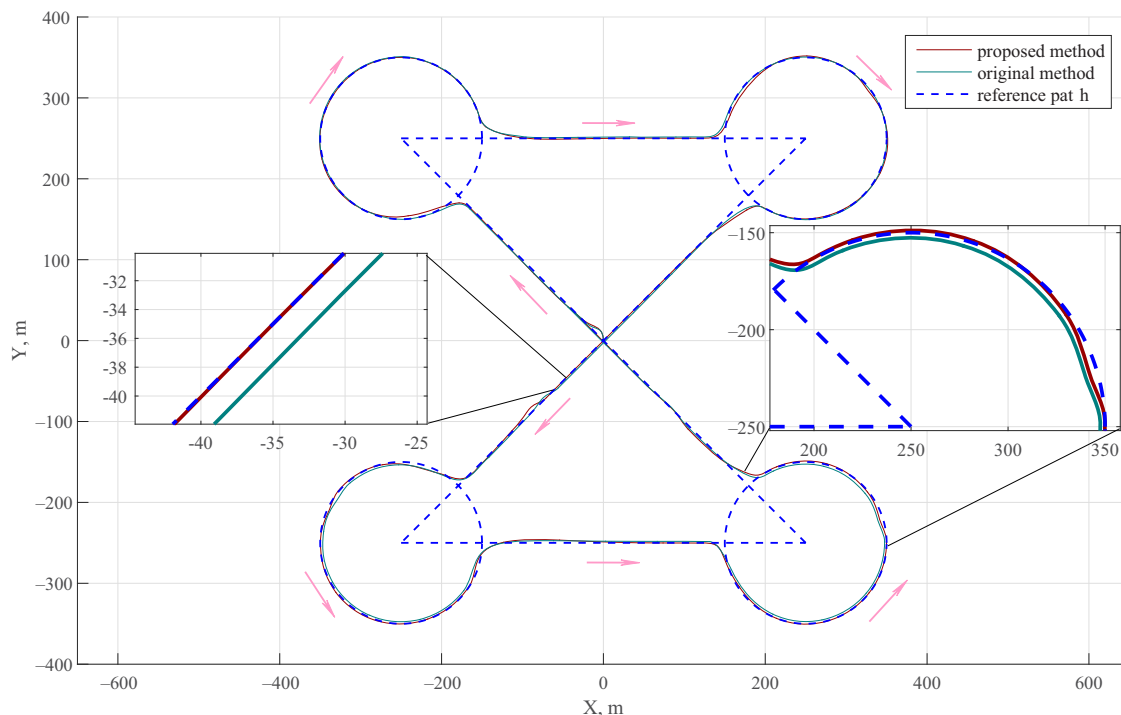


Figure 5 Whole session cross-track error

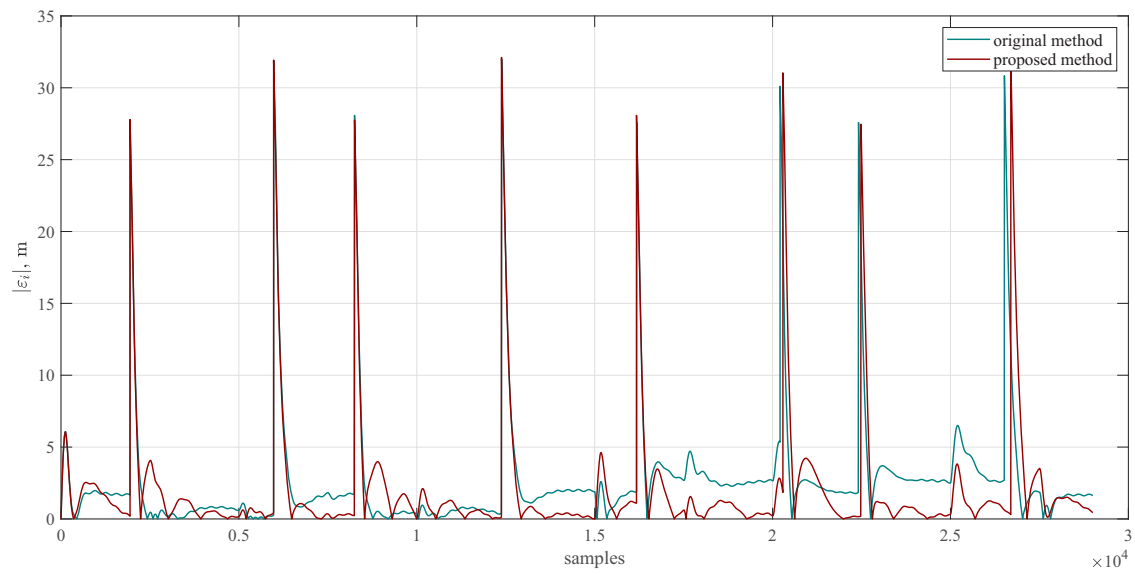


Figure 6 Transient illustration

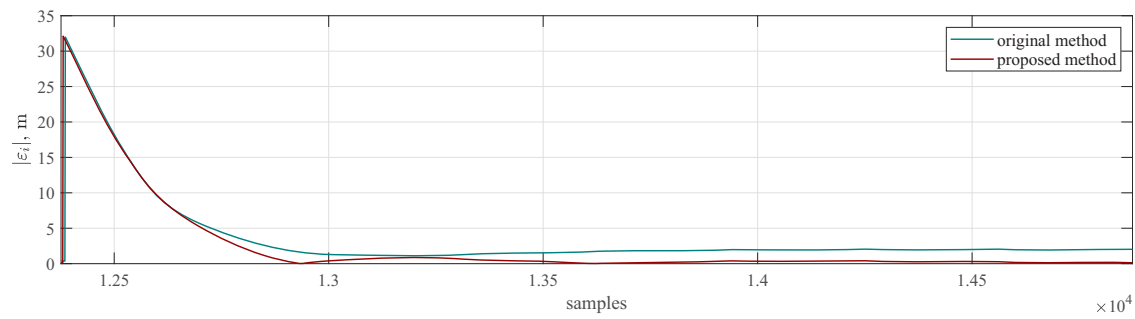


Figure 7 Steady state cross-track error

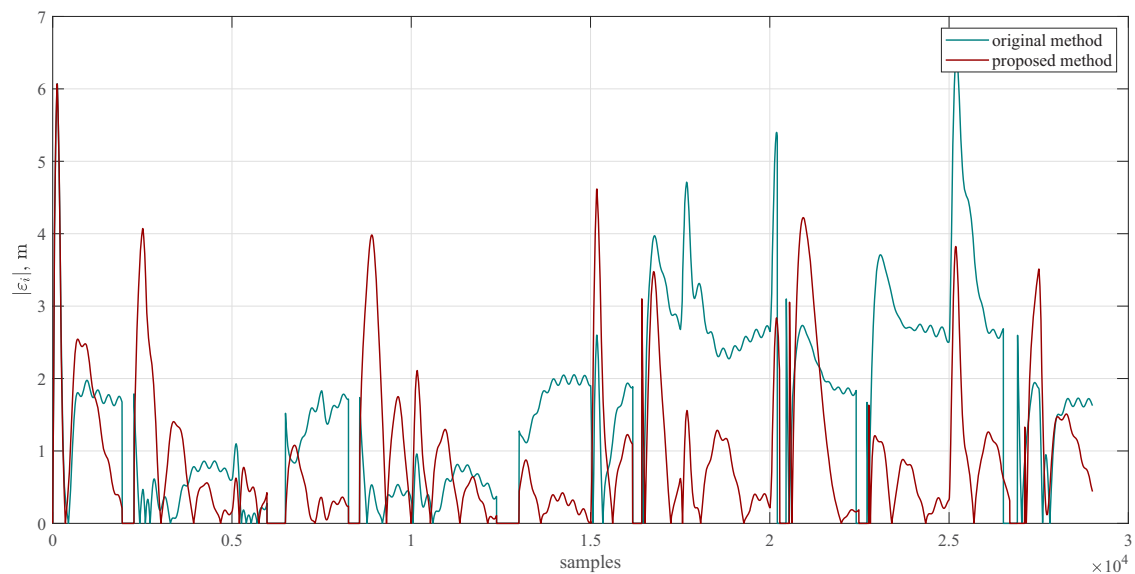


Table 3 Statistics on cross-track error of the simulation

	Whole session		Steady state	
	$\ e\ _1, \text{ m}$	$\ e\ _2, \text{ m}$	$\ e\ _1, \text{ m}$	$\ e\ _2, \text{ m}$
Original method	7.90×10^4	8.45×10^4	4.84×10^4	3.38×10^2
Proposed method	6.04×10^4	8.21×10^2	2.56×10^4	2.34×10^2

Table 4 Average cross-track error

	Whole session	Steady state
Original method	2.72 m	1.30 m
Proposed method	2.08 m	0.90 m

The test vehicle has a 1.5 m wingspan 0.53 m^2 wing area. Take-off weight of the test vehicle is about 2.1 kg. A 11*8 folding propeller is driven by a Cobra CP2814(KV560) BLDC motor. In total, 12 pieces of Sony VCT6 18,650 batteries form a 6S2P battery-pack to power the propelling system.

A customized low-cost general-purpose controller LC211 is used as the onboard controller, as shown in Figure 9. The flight control program runs on a STM32F405 microcontroller. Two ICM-20689 6-axis inertial sensors from TDK serve as IMU for the LC211 controller. Angular velocities and accelerations are sampled at a frequency of 400 Hz. Attitude estimating algorithm is updated at the same frequency.

Positioning information is acquired at 20 Hz through the on-board Trimble BD910 GNSS module with an RTK base station. One-sigma (1σ) of the positioning measurement for the whole flight session is no more than 0.01 m in both horizontal and vertical directions. Flight data is transmitted to the ground station at 10 Hz by a 900 MHz radio.

The hardware cost of LC211 controller is about US\$500 and the total hardware cost of the airborne unit is no more than US\$1,000.

The same C code of path following algorithm being used in previous simulation is deployed on the controller.

Flight test was conducted in south-east wind with an average wind speed of about 3.5 m/s. Target ground speed was 17 m/s. Reference path and actual flight path of the original method and the

proposed method are shown in Figure 10. Flights corresponding to the two methods were run for three times consecutively in the test. Because of battery consumption and wind condition, the flight speed fluctuated in the test, which made the timing of path segment switching different. For the convenience of comparison, error data is partitioned and shifted. Absolute value of cross-track error of each lap is plotted in Figures 11–13. Compared with the original method, the proposed method could effectively reduce cross-track error after path segment switching. 1-norm and 2-norm of the cross-track error of all three laps are shown in Table 5. Average cross-track error is shown in Table 6.

Discussion

Cusps in Figure 5 show that a majority part of cross-track error was produced when reference path segments were switched. Eliminating this part of cross-track error relies on proper path segment planning and switching mechanism. That is beyond the scope of this study. In Figures 6 and 7, we can see the cross-track error attenuate apparently after reference path switching for the proposed path following method (Figures 5 and 7, samples 360–1,940, 3,020–5,200, 6,500–8,250, 10,000–12,000, 13,000–14,900, 16,500–20,000, 20,600–22,400, 23,100–24,900, 27,800–29,000). The cross-track error for the original VF and NLGL shows no such character because there is no specific mechanism coping with long term disturbance and steady-state error in either (1) or (9). Table 4 shows the proposed path following method gives a 0.7 m improvement in overall average cross-track error and 0.4 m improvement in steady state average cross-track error. For the proposed method, the vehicle was almost flying “on” the reference path after the transient process (see Figures 4, 7 and 10). The cross-track error for those periods is about 0.5 m or less. In the flight test,

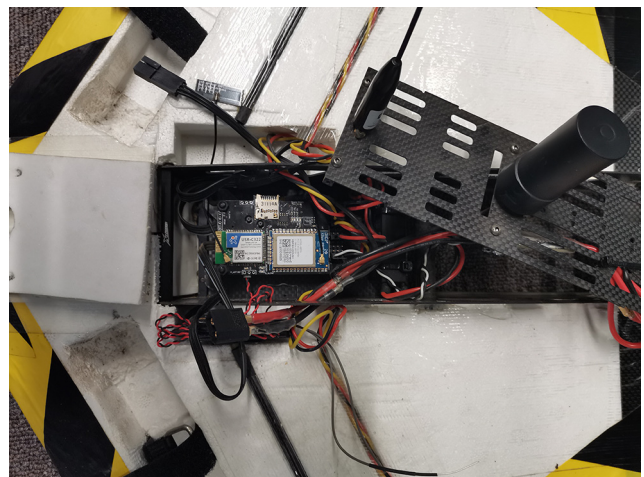
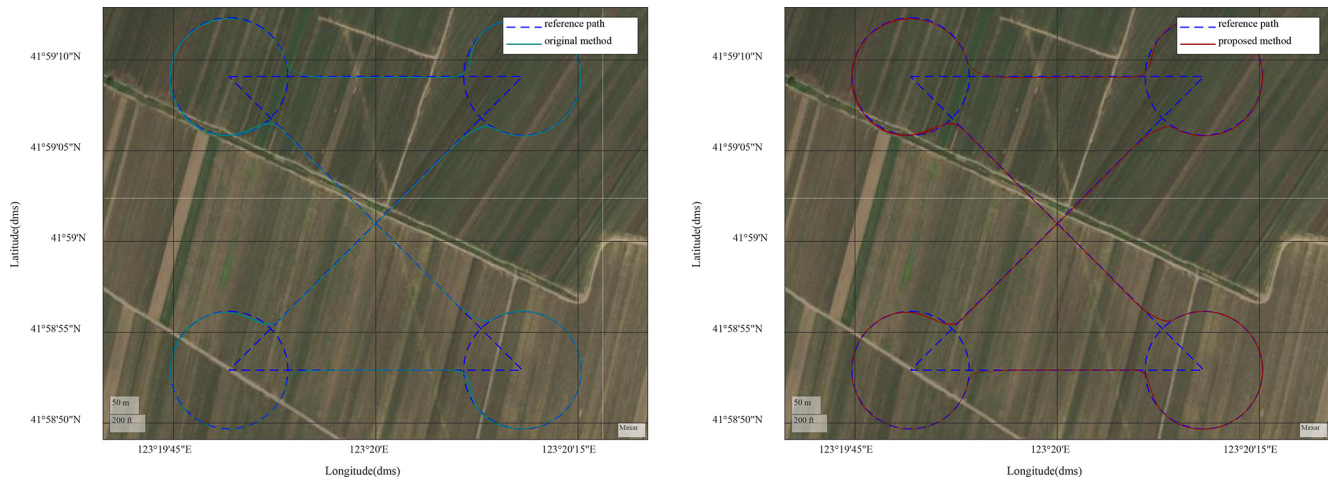
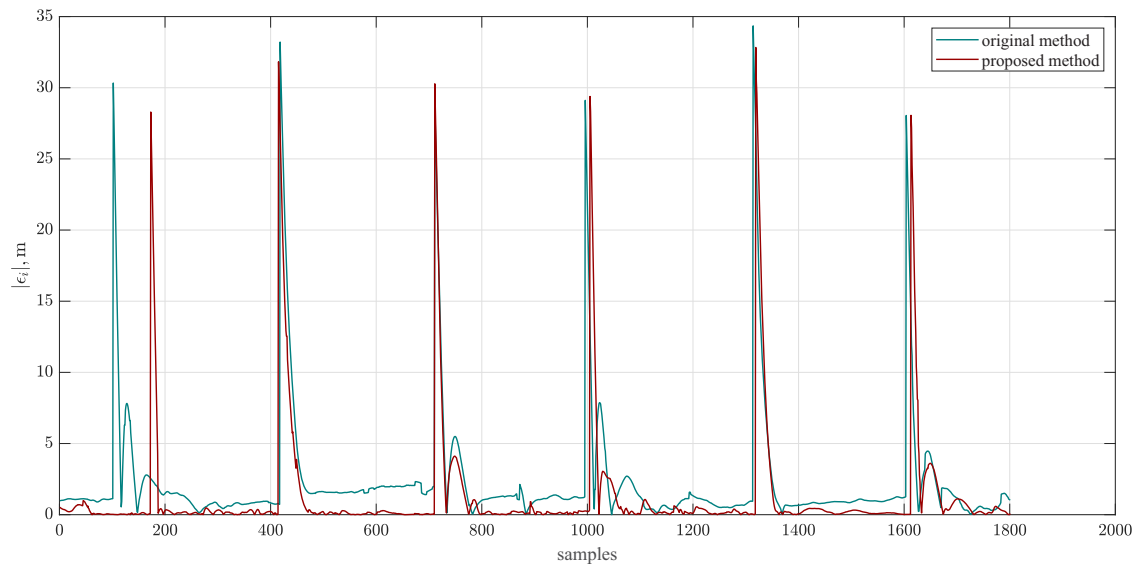
Figure 8 Test vehicle**Figure 9** LC211 controller installed in the test vehicle

Figure 10 Test flight path**Figure 11** Test flight cross-track error (Lap 1)

the proposed method shows less cross-track error in most segments (Figures 11–13). The statistics on flight data are similar to simulation results (Table 5). The proposed path following method gives a 0.7 m improvement in overall average cross-track error and 0.6 m improvement in steady state average cross-track error (Table 6). For the 1.5 m-wing-span test vehicle, this means centre of the reference path will be under the wing coverage eventually (straight line or arc path, crosswind presented). With such path following accuracy, small fixed wing UAVs will be able to follow general urban roads or highways and landing on them autonomously.

Conclusion

Modified path following method based on vector field and nonlinear guidance law has been presented in this paper. The original vector field method has been simplified.

Spatial integrating mechanism was introduced to deal with steady-state interference and improve path following accuracy. Simulation results showed that the proposed method has lower cross-track error than the original method. Flight test was conducted on the platform built with up-to-date ICs and modules. Practicability of the proposed method is validated. Results of the flight test matched the simulation results.

In this paper, only planar path following method was studied. Height control or three-dimensional path following is necessary for UAVs to accomplish actual tasks. Besides, time constraint, moving path and irregular path configuration were not covered in this paper, either. These features are indispensable in high maneuver and complex trajectory implementation. Further study will focus on these aspects.

Figure 12 Test flight cross-track error (Lap 2)

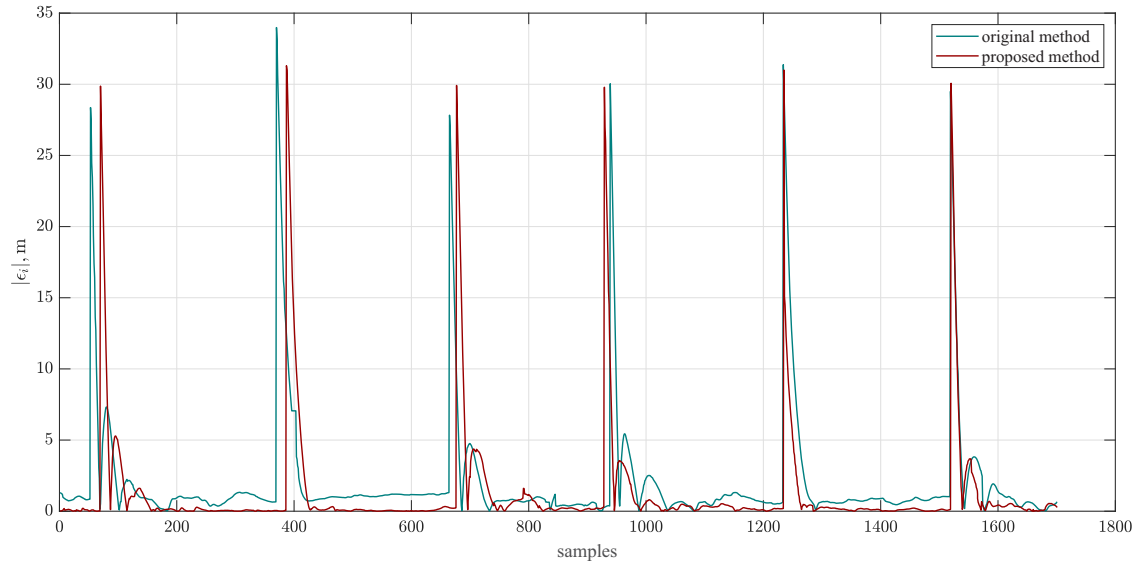


Figure 13 Test flight cross-track error (Lap 3)

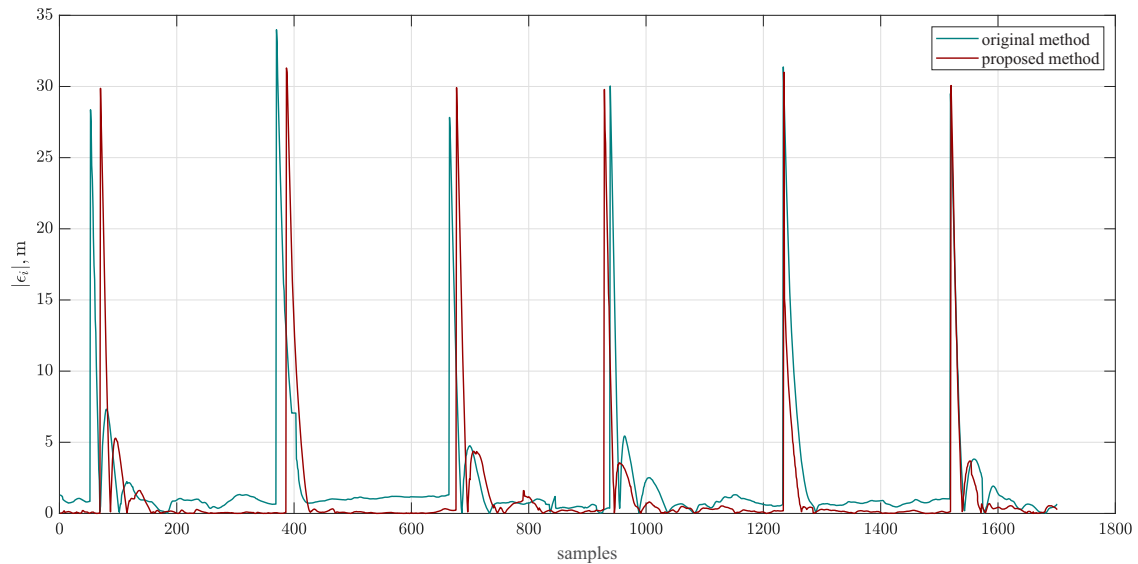


Table 5 Statistics on cross-track error of the simulation

	Whole session		Steady state	
	$\ \varepsilon\ _1, \text{m}$	$\ \varepsilon\ _2, \text{m}$	$\ \varepsilon\ _1, \text{m}$	$\ \varepsilon\ _2, \text{m}$
Original method	1.14×10^4	6.26×10^2	4.01×10^3	1.20×10^2
Proposed method	7.78×10^3	5.82×10^2	1.28×10^3	5.93×10^1

Table 6 Average cross-track error

	Whole session	Steady state
Original method	2.43 m	0.86 m
Proposed method	1.66 m	0.27 m

References

- Ahmed, M. and Subbarao, K. (2010), "Nonlinear 3-D trajectory guidance for unmanned aerial vehicles", *2010 11th International Conference on Control Automation Robotics and Vision*, pp. 1923-1927.
- Ambrosino, G., Ariola, M., Ciniglio, U., Corrado, F., De Lellis, E. and Pironti, A. (2009), "Path generation and tracking in 3-D for UAVs", *IEEE Transactions on Control Systems Technology*, Vol. 17 No. 4, pp. 980-988.
- Beard, R. and McLain, T. (2010), *Navigation, Guidance, and Control of Small and Miniature Air Vehicles*, Brigham Young University, Provo, p. 12.
- Breivik, M. and Fossen, T.I. (2008), "Guidance laws for planar motion control", *2008 47th IEEE Conference on Decision and Control*, pp. 570-577.
- Cao, C., Hovakimyan, N., Patel, V.V., Kaminer, I. and Dobrokhodov, V. (2007), "Stabilization of cascaded systems via L1 adaptive controller with application to a UAV path following problem and flight test results", *2007 American Control Conference*, pp. 1787-1792.
- Conte, G., Duranti, S. and Merz, T. (2004), "Dynamic 3D path following for an autonomous helicopter", *IFAC Proceedings Volumes*, Vol. 37 No. 8, pp. 472-477.
- Davidson, M. and Bahl, V. (2001), "The scalar-e-controller: a spatial path tracking approach for ODV, Ackerman, and differentially-steered autonomous wheeled mobile robots", *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 175-180.
- Davidson, M., Bahl, V. and Moore, K.L. (2002), "Spatial integration for a nonlinear path tracking control law", *Proceedings of the 2002 American Control Conference*, Vol. 5, pp. 4291-4296.
- DoD (2003), "Unmanned aerial vehicles roadmap: 2002-2027", available at: www.nasa.gov/centers/dryden/pdf/111759main_DoD_UAV_Roadmap_2003.pdf (accessed 12 November 2017).
- Dubins, L.E. (1957), "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", *American Journal of Mathematics*, Vol. 79 No. 3, pp. 497-516.
- Fossen, T.I., Breivik, M. and Skjetne, R. (2003), "Line-of-sight path following of under actuated marine craft", *IFAC Proceedings Volumes*, Vol. 36 No. 21, pp. 211-216.
- Healey, A. and Lienard, D. (1993), "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles", *IEEE Journal of Oceanic Engineering*, Vol. 18 No. 3, pp. 327-339.
- Hota, S. and Ghose, D. (2009), "A modified Dubins method for optimal path planning of a miniature air vehicle converging to a straight line path", *2009 American Control Conference*, pp. 2397-2402.
- Jackson, S., Tisdale, J., Kamgarpour, M., Basso, B. and Hedrick, J.K. (2008), "Tracking controllers for small UAVs with wind disturbances: theory and flight results", *2008 47th IEEE Conference on Decision and Control*, pp. 564-569.
- Kaminer, I., Hovakimyan, N., Patel, V., Cao, C., Young, A., Pascoal, A. and Dobrokhodov, V. (2007), "Coordinated path following for time-critical missions of multiple UAVs via L1 adaptive output feedback controllers", *2007 AIAA Guidance, Navigation and Control Conference and Exhibit*.
- Nelson, D.R., Barber, D.B., McLain, T.W. and Beard, R.W. (2007), "Vector field path following for miniature air vehicles", *IEEE Transactions on Robotics*, Vol. 23 No. 3, pp. 519-529.
- Osborne, J. and Rysdyk, R. (2005), "Waypoint guidance for small UAVs in wind", *Infotech@Aerospace*.
- Park, S., Deyst, J. and How, J.P. (2007), "Performance and Lyapunov stability of a nonlinear path following guidance method", *Journal of Guidance, Control, and Dynamics*, Vol. 30 No. 6, pp. 1718-1728.
- Rankin, A.L., Iii, C.D.C. and Ii, D.G.A. (1998), "Evaluating a PID, pure pursuit, and weighted steering controller for an autonomous land vehicle", *Mobile Robots XII*, Vol. 3210, pp. 1-12.
- Ratnoo, A., Sujit, P.B. and Kothari, M. (2011), "Adaptive optimal path following for high wind flights", *IFAC Proceedings Volumes*, Vol. 44 No. 1, pp. 12985-12990.
- Rhee, I., Park, S. and Ryoo, C.K. (2010), "A tight path following algorithm of an UAS based on PID control", *Proceedings of SICE Annual Conference*, 2010, pp. 1270-1273.
- Rysdyk, R. (2006), "Unmanned aerial vehicle path following for target observation in wind", *Journal of Guidance, Control, and Dynamics*, Vol. 29 No. 5, pp. 1092-1100.
- Silva, J.E.D. and Sousa, J.B.D. (2011), "A dynamic programming based Path-Following controller for autonomous vehicles", *Control and Intelligent Systems*, Vol. 39 No. 4.
- Sujit, P.B., Saripalli, S. and Sousa, J.B. (2014), "Unmanned aerial vehicle path following: a survey and analysis of algorithms for fixed-wing unmanned aerial vehicles", *IEEE Control Systems Magazine*, Vol. 34 No. 1, pp. 42-59.
- Sun, M., Zhu, R. and Yang, X. (2008), "UAV path generation, path following and gimbal control", *2008 IEEE International Conference on Networking, Sensing and Control*, pp. 870-873.
- Xavier, D.M., Silva, B.N. and Branco, K.R. (2018), "Comparison of path-following algorithms for loiter paths of unmanned aerial vehicles", *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1243-1248.
- Zhang, Y., Zhang, Y., Liu, Z., Yu, Z. and Qu, Y. (2018), "Line-of-sight path following control on UAV with sideslip estimation and compensation", *2018 37th Chinese Control Conference*.

Further reading

- Oliveira, T., Aguiar, A.P. and Encarnacao, P. (2016), "Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems", *IEEE Transactions on Robotics*, Vol. 32 No. 5, pp. 1062-1078.

Corresponding author

Xiaogang Wang can be contacted at: wangxiaogang@hit.edu.cn