2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

An Evaluation of UAV Path Following Algorithms

P.B. Sujit, Srikanth Saripalli, J.B. Sousa

Abstract— **Path following is the simplest desired autonomous navigation capability for UAVs. Several approaches have been developed in the literature for path following. However, there is no adequate information on the comparison analysis of these algorithms. In this paper, we compare five path following algorithms that are easy to implement, take less implementation time and are robust to disturbances. The comparison of these guidance laws is carried out through simulations using the total cross-track error and control effort performance metrics.**

I. Introduction

With low cost sensors, electronics and airframes, there is a great interest in using unmanned aerial vehicles (UAVs) for several non-military applications like habitat mapping, aerial photogrammetry, agriculture, marine sciences, etc. Most of these missions require the UAV to autonomously follow predefined paths at constant height. The common paths being straight line paths and circular orbits (called as loiter). The path following algorithms must be accurate in following the path and robust to wind disturbances.

In the literature, several path following algorithms have been well developed for unmanned aerial vehicles, underwater vehicles, surface vehicles and ground vehicles. The basic functionality of path following for any type of unmanned vehicle is the same as the guidance laws developed for path following use kinematic constraints with constant velocity. Although, several types of path following algorithms are available in the literature, there is no adequate information on the comparison between different algorithms in terms of performance or the implementation time.

In this paper, we provide an overview of path following problem, describe different path following algorithms from the literature and characterize their performance based on accuracy, robustness and control effort. The selected path following algorithms are carrot following algorithm, nonlinear guidance law (NLGL) [1], vector field based path following (VF) [2], linear quadratic regulator based path following (LQR) [3], and pure pursuit and line-of-sight based path following (PLOS) [4]. These algorithms were mainly selected due to their simplicity and ease of implementation.

There are several other types of path following algorithms using guidance approaches [5], [6], and advanced control concepts [7], [8], [9], [10], [11], [12]. The PLOS and LQR based path following algorithms consider some of guidance concepts similar to [5], [6]. The NLGL and VF path following algorithms have been proven theoretically using Lyapunov based stability conditions in [1], [2] similar to [7], [8], [9], [10], [11]. Silva and Sousa [12] use dynamic programming approach which is difficult to implement. Since the selected path following algorithms [1], [2], [3], [4] address some principles used in other path following algorithms, we do not consider them for analysis. In this paper, by using a kinematic model and simulating various path following strategies we are able to demonstrate the advantages and disadvantages of each one of them and study the effect of change in parameters.

P.B. Sujit is a Research Scientist in the Department of Electrical and Computer Engineering, University of Porto, Portugal - 4200-165. Email - sujit@fe.up.pt

Srikanth Saripalli is Asst. Professor in the School of Earth and Space Exploration, University of Arizona, Tempe, AZ 85287. Email - srikanth.saripalli@asu.edu

J.B. Sousa is Asst. Professor in the Department of Electrical and Computer Engineering, University of Porto, Portugal - 4200-165. Email - jtasso@fe.up.pt

A. Path following problem

Given a path, initial location $p = (x, y)$ of the aerial vehicle along with its heading angle (ψ), the path following problem is to determine the commanded heading angle (ψ_d) for the vehicle such that by following (ψ_d), the UAV accurately tracks the path as the mission progresses. Typically, for most of the missions either straight line paths or circular orbit paths are used. The straight line paths can be defined as a set of waypoints (as shown in Fig 1(a)). The angle formed by waypoints W_i and W_{i+1} is called the line-of-sight (LOS) angle. Figure 1(b) shows the initial geometry for a circular orbit also called loiter following. Assume that the UAV is at a distance d from the path. This distance d is called as cross-track error. Apart from minimizing the cross-track error, the UAV must ensure that ψ is same as the LOS angle θ. The objective of the path following algorithms is to minimize $d \to 0$ and $|\psi - \theta| \to 0$, where $|.|$ represents the absolute value as the mission time $t \to \infty$.

B. Organization

The rest of the paper is organized as follows. In Section II, we describe different path following algorithms for straight line and loiter paths, and analyze their behavior for different parameter variations. In Section III, we compare the performance of path following algorithms and we conclude in Section IV.

II. Path following algorithms

The two most commonly used paths for UAVs are straight lines and loiter paths. Both the type of paths are usually defined on a xy plane with constant altitude and speed. Under this assumption, we can model the UAV kinematics as

$$\begin{aligned} \dot{x} &= v_a \cos\psi + v_w \cos(\psi_w) = v_g \cos\chi, \\ \dot{y} &= v_a \sin\psi + v_w \sin(\psi_w) = v_g \sin\chi, \\ \dot{\psi} &= \dot{\chi} = u, \end{aligned} \tag{1}$$

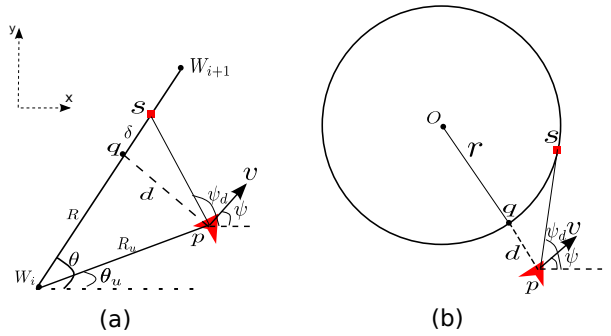


Fig. 1. (a) The UAV has to follow a straight line path (b) The UAV has to follow a loiter or orbit of radius r

where v_a m/s is the UAV airspeed, ψ is the UAV heading heading angle, $\dot{\psi}$ is the heading angle rate which is constrained by $-\omega_{\max} < \dot{\psi} < \omega_{\max}$, v_w m/s is the wind speed in the ψ_w rad direction and u is the control. When the wind component is not considered then v_a and ψ are used, while under the influence of wind, the equations with ground speed $v_g = \sqrt{(v_a \cos\psi + v_w \cos\psi_w)^2 + (v_a \sin\psi + v_w \sin\psi_w)^2}$ and course angle $\chi = atan2(\dot{y}, \dot{x})$ are considered. The kinematic model of the UAV represents a Dubins car model [18]. For all the simulations in this paper, we consider $v_a = 15\,\mathrm{m/s}$ with minimum turning radius of $45\,\mathrm{m}$. The considered loiter circle is of radius $r = 100\,\mathrm{m}$.

In order to analyze the working principle of the algorithms and performance under different parameter conditions, we considered the kinematic model without wind for simplicity. However, in Section III, we will consider the presence of wind for performance comparison.

A. Carrot chasing algorithm

A simplest approach to follow a desired path is to introduce a virtual target point (VTP) on the path and direct the UAV to chase VTP. As the engagement progresses, the UAV approaches towards the path and eventually follow the path. In this approach, the VTP is called the Carrot and the UAV chases the carrot, hence the name Carrot chasing algorithm.

1) Straight line following: Consider a straight line as shown in Figure 1(a). Following a straight line path involves two steps: (i) determining the cross-track error (d) and (ii) updating the location of the virtual target. Let the cross-track error of the path be d. The VTP (s) is located at a distance δ from q as shown in Figure 1(a). The UAV desired heading (ψ_d) is updated based on the location of s. The process of finding new ψ_d continues until the UAV reaches W_{i+1}. The procedure to update the location of VTP and ψ_d is given in Algorithm 1.

The control input u in Algorithm 1 uses a proportional controller with gain $\kappa > 0$. The path parameter δ can modify the performance of the carrot chasing algorithm as shown in Figure 2(a). For low $\delta = 0 - 10$, the UAV is forced to move towards the LOS and it takes longer time to settle on the path resulting in higher cross-track error. With increase in δ the UAV is able to settle on the path quickly, thus reducing the cross-track error. However, with large δ, the cross-track

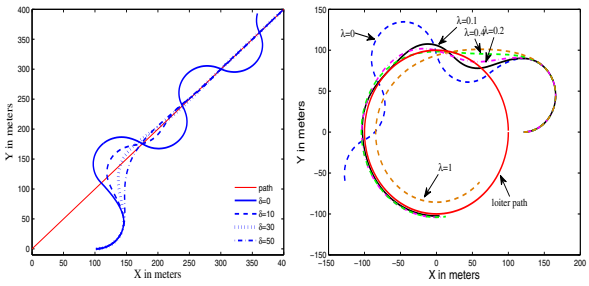
Algorithm 1 Algorithms for carrot chasing algorithm

Straight Line following

Initialize: $W_i = (x_i, y_i), W_{i+1} = (x_{i+1}, y_{i+1}), p = (x, y), \psi$
$R_u = ||W_i - p||, \theta = atan2(y_{i+1} - y_i, x_{i+1} - x_i)$
$\theta_u = atan2(y - y_i, x - x_i), \beta = \theta - \theta_u$
$R = \sqrt{R_u^2 - (R_u \sin(\beta))^2}$
$(x'_t, y'_t) \leftarrow ((R + \delta)\cos\theta, (R + \delta)\sin\theta)$ % $s = (x'_t, y'_t)$
$\psi_d = atan2(y'_t - y, x'_t - x)$
$u = \kappa(\psi_d - \psi)$

Loiter following

Initialize: $O = (x_l, y_l), r, p, \psi$
$d = ||O - p|| - r$
$\theta = atan2(y - y_l, x - x_l)$
$(x'_t, y'_t) \leftarrow (r\cos(\theta + \lambda), r\sin(\theta + \lambda))$ % $s = (x'_t, y'_t)$
$\psi_d = atan2(y'_t - y, x'_t - x)$
$u = \kappa(\psi_d - \psi)$



(a) (b)

Fig. 2. Performance of the carrot chasing algorithm with variation in δ and λ (a) UAV trajectories for different δ (b) The vehicle trajectories when λ is varied.

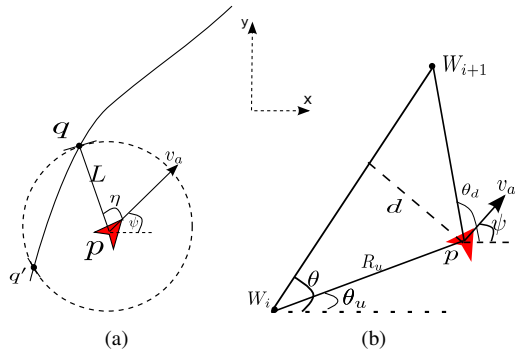
again increases but the time taken to reach W_1 decreases as the UAV moves ahead towards W_1.

2) Loiter: In loiter maneuver, the UAV is tasked to circle around a point (O) with a standoff distance of r meters as shown in Figure 1(b). To loiter, the UAV must be located on the circle and its heading direction must be orthogonal to the line Oq. Assume the UAV is located at p and the cross-track error is $d = ||O - p|| - r$. Using the cross-track error information and the direction of loiter, we can generate a VTP (s in the Figure 1(b)). The procedure to update the UAV and the VTP is given in Algorithm 1.

The path parameter λ needs to be selected properly to ensure quicker path convergence. When $\lambda = 0$, the UAV is unable to track the path as the desired VTP always points orthogonal to the UAV position causing it to perform sinusoidal path along the loiter as shown in Figure 2(b). When λ is increased to $\lambda = 0.1, 0.2, 0.4$ radians, the UAV tracks the loiter circle quickly. However, with further increase in $\lambda = 1$ radian, the UAV is unable to track the path as the VTP is always further away.

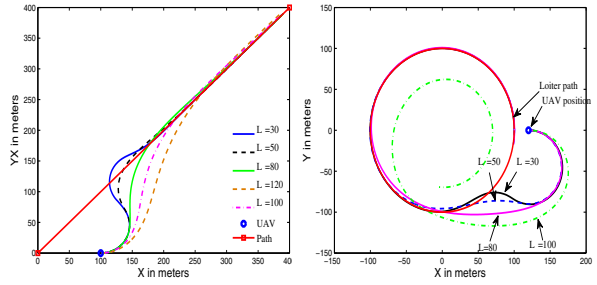
B. Non-linear guidance law

The non-linear guidance law (NLGL) was developed by Park et al. [1], which uses the VTP concept, but the mecha-



(a) (b)

Fig. 3. (a) Determining the intersection point for VTP q and q' (b) Geometry for pure pursuit and LOS based path following algorithm



(a) (b)

Fig. 4. The vehicle trajectories when the parameter L_1 is varied (a) for straight lines and (b) loiter paths

nism by which the VTP position is updated is different. For simplicity, assume the UAV is following a path as shown in Figure 3(a). From the current UAV position p, draw a circle of radius L. The circle will intercept the path at two points q and q'. However, the vehicle has to move northwards and hence the point q is selected as the VTP while q' is neglected. The advantage of this algorithm is that the same algorithm can be applied to any type of trajectories.

Algorithm 2 Algorithm for straight line following using NLGL algorithm

1: Initialize: $W_i = (x_i, y_i), W_{i+1} = (x_{i+1}, y_{i+1}), p, L$
2: Determine $q = (x_t, y_t)$
3: $\theta = atan2(y_t - y, x_t - x)$
4: $\eta = \theta - \psi$
5: $u = 2v_a^2 \sin(\eta)/L$

1) Straight line following: The VTP location $q = (x_t, y_t)$ is directly determined from the geometry of a straight line and a circle intersection with radius L (L is assumed to be constant). Due to this assumption if the distance is greater than L then there will not be any intersection resulting in no VTP. Figure 4(a) shows the effect of L. If L is low or high then the performance is marginal. Hence, L must be selected appropriately.

2) Loiter: The procedure for a loiter pattern is the same as that for straight line, except that, the circle of radius L will intersect with the loiter circle centered at O.

Even for loiter maneuver, the selection of L determines the performance of the vehicle. Figure 4(b) shows the performance of NLGL for loiter maneuvers with different L. Similar to the straight line case, with increase in L, the performance of the algorithm gets better. However, with further increase in $L > 100$, the VTP for the vehicle is far ahead and it tries to moves towards the target resulting in large cross track error.

C. Pure pursuit and LOS based path following (PLOS)

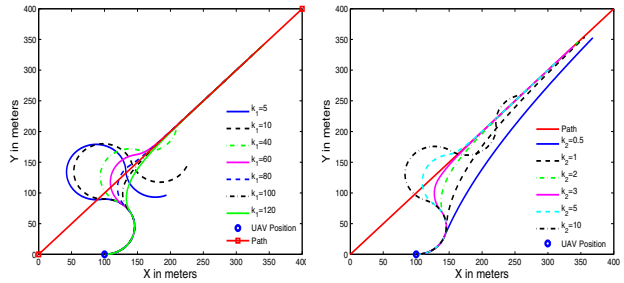
The pure pursuit (P) and line-of-sight (LOS) guidance are derived from missile guidance literature. The pursuit guidance law directs the UAV towards the waypoint W_{i+1}, while the LOS guidance law pushes the vehicle towards the LOS (as shown in Figure 3(b)). Thus the result of P and LOS will enable the vehicle to follow a path. The guidance law for pure pursuit is given as $\psi_p = k_1(\theta_d - \psi)$, where k_1 is the gain and θ_d is the desired angle. The LOS guidance law ensures that the angle between W_i and the UAV is the same as that of the angle between W_i and W_{i+1} (in Figure 3(b)). The LOS guidance law is given by $\psi_l = k_2 d \sin(\theta - \theta_u)$ where, d is the cross-track error, θ is the LOS angle between the two way points and θ_u is the angle between the UAV and W_i. The resultant motion of the UAV when pursuit and LOS guidance laws act together will allow the UAV to steer towards the path quickly. The combined guidance law is given as $\psi_d = k_1(\theta_d - \psi) + k_2 d \sin(\theta - \theta_u)$.

1) Straight line following: The cross-track error for PLOS guidance law is determined similar to the carrot following algorithm. Based on the cross track error and heading error, the desired heading is determined. The guidance law is stable when $k_1 > 0$ and $k_2 > 0$ [4]. When k_1 is small, the LOS guidance is dominating because of which the vehicle steers towards the path and crosses it resulting in a path that oscillates along the desired straight line path as shown in Figure 5(a) for $k_1 = 5$. With increase in k_1, the oscillation of the vehicle path decreases as the pursuit guidance weight increases which acts as a damping on the vehicle path oscillation. This effect can be seen in the figure for $k_1 = 10$ and $k_1 = 40$. With further increase in k_1, the vehicle settles onto the straight line path.

On the other hand, for a fixed k_1, with very low $k_2 = 0.5$, the pursuit guidance is dominant and the vehicle is directed towards W_{i+1} (as shown in Figure 5(b)). With increase in $k_2 = 1$, we can see that the vehicle tries to move towards the desired path, but takes a long time to converge onto the path. With increase in $k_2 = 2$ and $k_2 = 3$, the vehicle tracks the path quickly. However, with further increase in k_2, the LOS guidance dominates, resulting in oscillations on the path as shown in the figure for $k_2 = 5$ and $k_2 = 10$. Hence, the gains must be selected appropriately for best path performance.

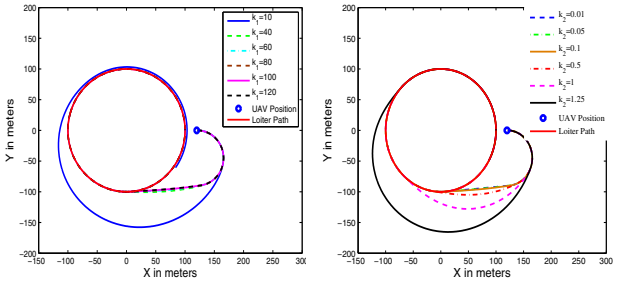
2) Loiter: Determining cross track error d for the loiter is similar to that of the carrot following algorithm. However, the reference heading angle θ_p is generated by the angle which is tangential to the line joining the aircraft and the loiter circle center.

The sensitivity of the algorithm to gains k_1 and k_2 is analyzed by keeping one gain constant and varying the other. Figure 6(a) shows the performance of PLOS algorithm with



(a) (b)

Fig. 5. (a) Effect of change in gain k_1 with constant $k_2 = 3$ (b) Effect of change in gain k_2 with constant $k_1 = 100$



(a) (b)

Fig. 6. (a) Effect of change in gain k_1 with constant $k_2 = 3$ (b) Effect of change in gain k_2 with constant $k_1 = 100$

constant $k_2 = 0.1$ and varying k_1. With low k_1, the vehicle takes longer time to reach the circle and then it follows the paths accurately. However, with increase in $k_1 > 10$, the vehicle tracks the circle accurately. This shows that for loiters, the effect of LOS is low and the effect of pursuit is high.

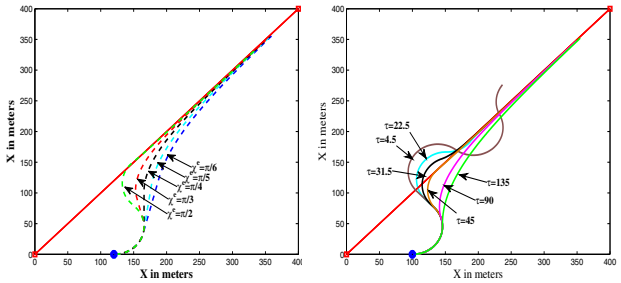
Figure 6(b) shows the results with constant $k_1 = 100$ and varying k_2. From the figure, we can see that vehicle is able to track for low values of $0 \leq k_2 < 1$, while with increase in k_2 the performance degrades.

D. Vector field based path following

The vector field based path following algorithm was developed by Nelson et al. [2]. The main idea is based on vector fields that determine the direction of flow which the vehicle follows to track the path.

1) Straight line following: The vector field based path following operates in two modes (i) when the vehicle is quite far away from the path and (ii) when the vehicle is close to the path defined by a transition boundary defined by τ. When the vehicle is far away, the natural strategy is to move towards the path. This is done by a parameter χ^e called as entry heading. When the vehicle enters into the transition region, χ^e is updated. The algorithm used to determine the desired heading angle ψ_d is the same as that given in [2] and hence we did not include in this section.

The performance of the vector following algorithm varies with different parameters values of χ^e, τ, k. When all the parameters are constant and χ^e is varied between $0 < \chi^e \leq \pi/2$, then the effect of path convergence can be seen in



(a) (b)

Fig. 7. (a) Trajectories of the vehicle for different initial heading angles (b) Effect of changing χ^e with constant $k = 1$ and $\tau = 45$ (c) Effect of change in τ with constant k and $\chi^e = \pi/2$

Figure 7(a). The other parameters are $\tau = 45, k = 1, \psi = 0$. With higher χ^e, the time to converge on the path decreases.

When τ is varied with $\chi^e = \pi/2, k = 1, \psi = 0$ constant, then we can see that with small values of τ the vehicle has to come closer to the path and then start to track thus increasing the convergence time. The parameter τ determines the width of the transition region near the path. However, with increase in τ, the vehicle converges quickly onto the path. With further increase in τ, it takes very long time to converge onto the path. This effect is similar to the VTP. The parameter k was kept constant at 1. The algorithm was very sensitive to $k > 1$, therefore we do not consider varying k.

2) Loiter: For loitering, the vector field is created such that it converges on the loiter circle. When the vehicle is outside or inside the loitering circles, the vector field pushes them onto the circle. The algorithm to determine the desired heading angle of the vehicle is the same as that described in [2]. The loiter following has only two parameters k, α. The variation in k is too sensitive and hence $k = 1$ is always kept constant. Through simulations, we found that varying the gain α did not affect the path convergence and hence we do not show its trajectories.

E. LQR path following algorithm

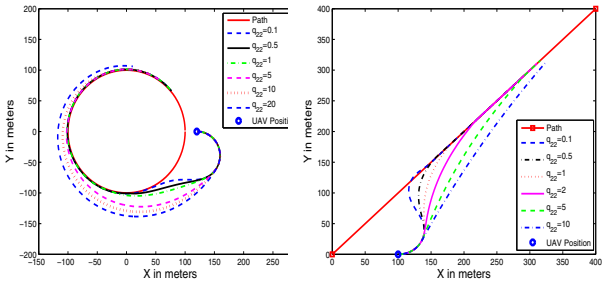
LQR based path following algorithm uses optimal control theory to compute u. The details of the LQR formulation for the path following problem are given in the reference [3]. The LQR control is given by

$$u = -\left[\sqrt{\left|\frac{d_b}{d_b - d}\right|}\, d + \sqrt{2\sqrt{\left|\frac{d_b}{d_b - d}\right|} + q_{22}}\, v_d\right] \qquad (2)$$

where, d_b is the bounded region along the path (similar to τ in vector field path following), q_{22} is the gain parameter of the $\mathbf{Q}$ matrix and $v_d = v\sin(\psi - \theta)$ is the cross track error velocity. The $\mathbf{Q}$ matrix is given as

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

where $q_{12} = q_{21} = 0$ and $q_{11} = |\frac{d_b}{d_b - d}|$. By selecting appropriate d_b and q_{22}, the performance of the path following algorithm can be tuned easily.



(a) (b)

Fig. 8. Performance of the adaptive LQR path following algorithm for (a) loiter path with different q_{22} values and (b) with different q_{22} values for a straight line path

The LQR path following algorithm uses a boundary around the path defined by d_b, similar to vector field technique. There are three parameters that control the trajectory of the vehicle - $\mathbf{R}$, q_{11} and q_{22}. For stable LQR controller, $\mathbf{R}$ and $\mathbf{Q}$ must be positive, therefore, typically, $\mathbf{R} = \mathbf{1}$, and $q_{11} > 0$ and $q_{22} > 0$ can be selected. The selection of q_{22} affects the performance of the algorithm.

1) Straight-line following: Consider a straight line segment as shown in Figure 1(a). Let R_u be the displacement of UAV with respect to point W_1 making an angle θ_u with the reference. The UAV position and velocity errors can be calculated as $d = R_u \sin(\theta - \theta_v)$ and $v_d = \dot{d} = v_a \sin(\psi - \theta)$ respectively. By selecting d_b and q_{22}, we can easily compute the required control effort.

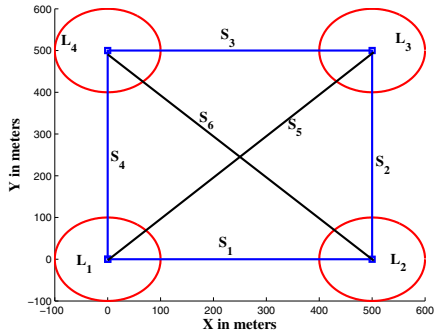
For a given heading angle of $\psi = 0$, the trajectories of the vehicle for different q_{22} is shown in Figure 8(b). With very low values of q_{22}, the vehicle tries to track the path very quickly as the penalty on the state is very low. For low values of q_{22} the vehicle needs to apply maximum acceleration and hence its control effort will be high. However, with increase in q_{22}, the time to converge on the path decreases and it takes a long time to converge on the path.

2) Loiter: Consider a circular path of radius r and let $R = d + r$, be the displacement of the UAV with respect to the center of the circular path. The UAV position and velocity errors can be calculated as $d = R - r$ and $v_d = v \sin(\psi - \theta)$, and substituted in 2.

For different values of q_{22}, the vehicle paths is shown in Figure 8(a). The vehicle converges onto the loiter path is proportional to the q_{22} value. Selecting q_{22} around $1 - 2$ results in a good performance with low control effort.

III. Comparison of Algorithms

For comparison purpose, we define two metrics - total control effort (U) and total cross track error (D). The total control effort will determine the control demands of the algorithm while total cross track error will determine the offset of the vehicle from the desired path. Let $u(t)$ be the control effort and $d(t)$ be the cross track error of a given guidance law at time t, then the total control effort and total



(a)

Fig. 9. UAV mission with the sequence of visiting as $S_1, L_2, S_2, L_3, S_3, L_4, S_4, L_1, S_5, L_3, S_3, L_4, S_6$

cross track error metrics are defined as

$$U = \sum_{t=0}^{t=T} u(t)^2, D = \sum_{t=0}^{t=T} d(t). \quad (3)$$

In real-world, the UAVs have to be robust to wind disturbances as its magnitude and direction changes with time. In order to analyze the performance of the five path following algorithms in the presence of varying wind conditions, we consider a mission where the UAV has to follow straight line paths and perform loiter continuously. Figure 9(a) shows the sample mission plan and the sequence for the vehicle to navigate. The notation S_i denotes the i^{th} straight line path while L_j denotes the j^{th} loiter. The vehicle follows a straight line and when it reaches the loiter radius of the ending waypoint, it shifts to loiter. When the UAV is close to the next straight line path, it switches to straight line path.

A. Performance comparison

We carried out 1000 simulations with different wind characteristics while keeping the initial position $p = (-50\ -150)$ and heading angle $(\psi = 0)$ constant. For each simulation, the wind magnitude and direction changes randomly every 20 seconds. In order to compare the performance of a guidance law with respect to cross track error and control effort, we used the average percentage weight ζ metric, where ζ is evaluated as

$$\zeta = \Gamma \bar{U} + (1 - \Gamma)\bar{D}, \quad (4)$$

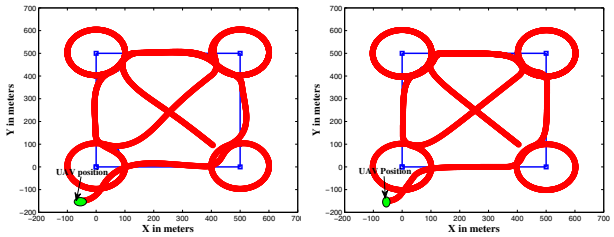
and the trade-off weight Γ is varied from 0 to 1, $\bar{U}$ is the normalized mean control effort, $\bar{D}$ is the normalized mean cross track. ζ is evaluated for each guidance law. The parameters used for different path following algorithms are given in Table I. The parameters given in the table correspond to the best performance of the path following algorithms during testing phase before performing the Monte-Carlo simulations.

Figure 11 shows the performance of different path following algorithms with change in trade-off weight ζ. If we consider only the cross track error as the metric ($\Gamma = 0$), then vector field path following algorithm performs the best. On the other hand, if we consider control effort as the

Algorithm	Parameters for the algorithm
Carrot	Straight line: $\delta = 100$ Loiter: $\lambda = 0.2\,\text{rad}$
NLGL	Straight line: $L_1 = 101$ loiter: $L_1 = 50$
PLOS	Straight line: $k_1 = 80, k_2 = 0.8$ Loiter: $k_1 = 100, k_2 = 0.1$
LQR	Straight line: $q_{22} = 5$ Loiter: $q_{22} = 10$
VF	Straight line: $\tau = 3v, \chi^e = \pi/3, \alpha = 5$ Loiter: $\alpha = 50$

TABLE I

PARAMETERS USED FOR DIFFERENT GUIDANCE LAWS



(a) (b)

Fig. 10. UAV trajectories using (a) NLGL and (b) VF path following.

performance metric ($\Gamma = 1$), then NLGL is the best. The cross track error and control effort are related. If U is more then D is less. Figures 10(a) and 10(b) show the trajectories of the UAV using NLGL and VF path following algorithm for one simulation respectively. From the figure, we can see that VF accurately tracks the path than the NLGL. Thus VF spends more control effort to track accurate paths.

When the weight is between $\Gamma = 0.1$ to $\Gamma = 0.4$, the performance of Carrot, NLGL and PLOS are close. The VF performance is poor as more weight is given towards control effort than cross track error. One reason that VF consumes more control effort is related to the chattering effect of VF algorithm which could be minimized using a sliding mode controller. The Carrot and PLOS guidance laws perform similar for different weights and they seem to be more robust to the effect of disturbances. The LQR also has low slope characteristics but with slightly more cross track error and requires higher control effort than PLOS and Carrot.

IV. CONCLUSIONS

We carried out detailed analysis of five path following algorithms. All the algorithms allow the vehicle to follow the paths under different winds conditions. From the Monte-Carlo simulations, we found that vector field path following technique accurately follows the path than other techniques. However, it consumes large control effort. On the other hand, non-linear guidance law consumes the least control effort but takes longer time to track the paths. A trade-off between control effort and path accuracy is required depending on the application. From the analysis, one can easily choose the best path following algorithm and tune it appropriately. For mapping and urban terrain navigation where accurate path need to be tracked vector field path following would be better while for search and surveillance NLGL would be a likely choice. The Carrot, PLOS and LQR can also be used instead of NLGL as they perform close.

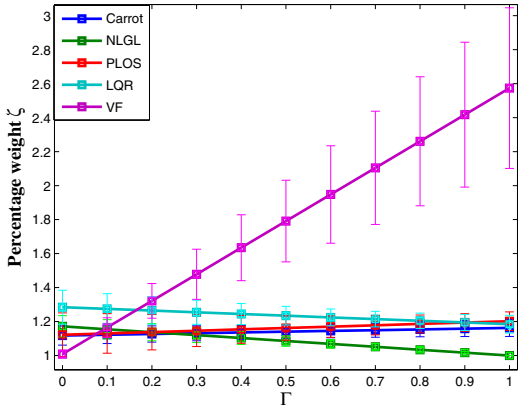


Fig. 11. Average performance of the mission with different guidance laws for varying weight on cross track error and control effort.

REFERENCES

[1] S. Park, J. Deystt, and J. How, "Performance and lyapunov stability of a nonlinear path-following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.

[2] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, pp. 519–529, June 2007.

[3] A. Ratnoo, P. Sujit, and M. Kothari, "Optimal path following for high wind flights," *Proc. of the IFAC World Congress*, Aug 2011.

[4] M. Kothari, I. Postlethwaite, and D. Gu, "A suboptimal path planning algorithm using rapidly-exploring random trees," *International Journal of Aerospace Innovations*, vol. 2, no. 1, pp. 93–104, 2010.

[5] J. Osborne and R. Rysdyk, "Waypoint guidance for small uavs in wind," *Proceedings of the AIAA Infotech@Aerospace*, 2005.

[6] M. Breivik and T. Fossen, "Principles of guidance-based path following in 2d and 3d," *Proceedings of the IEEE Conference on Decision and Control*, pp. 627–634, December 2005.

[7] S. Shehab and L. Rodrigues, "Preliminary results on uav path following using piecewise-affine control," *Proceedings of the IEEE Conference on Control Applications*, pp. 358–363, Aug 2005.

[8] I. Kaminer, O. Yakimenko, A. Pascoal, and R. Ghabcheloo, "Path generation, path following and coordinated control for time critical missions of multiple uavs," *Proceedings of the American Control Conference*, pp. 4906–4913, June 2006.

[9] C. Cao, N. Hovakimyan, I. Kaminer, V. Patel, and V. Dobrokhodov, "Stabilization of cascaded systems via l1 adaptive controller with application to a uav path following problem and flight test results," *Proceedings of the American Control Conference*, pp. 1787–1792, July 2007.

[10] A. Healey and D. Lienard, "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles," *Oceanic Engineering, IEEE Journal of*, vol. 18, no. 3, pp. 327 –339, jul 1993.

[11] P. Encarnacao and A. Pascoal, "Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 1, 2001, pp. 964 –969 vol.1.

[12] J. da Silva and J. de Sousa, "A dynamic programming approach for the motion control of autonomous vehicles," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, dec. 2010, pp. 6660 –6665.

[13] http://www.procerus.com.

[14] http://www.cloudcaptech.com.

[15] http://diydrones.com.

[16] http://paparazzi.enac.fr.

[17] http://openpilot.org.

[18] L. E. Dubins, "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, p. 497516, 1957.