Scientific Research Publishing

Open Journal of Optimization, 2019, 8, 59-72
http://www.scirp.org/journal/ojop
ISSN Online: 2325-7091
ISSN Print: 2325-7105

Dubins Waypoint Navigation of Small-Class Unmanned Aerial Vehicles

Larry M. Silverberg, Dahan Xu

Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, USA
Email: lmsilver@ncsu.edu

How to cite this paper: Silverberg, L.M. and Xu, D.H. (2019) Dubins Waypoint Navigation of Small-Class Unmanned Aerial Vehicles. *Open Journal of Optimization*, **8**, 59-72.
https://doi.org/10.4236/ojop.2019.82006

Received: June 9, 2019
Accepted: June 27, 2019
Published: June 30, 2019

Abstract

This paper considers a variation on the Dubins path problem and proposes an improved waypoint navigation (WN) algorithm called Dubins waypoint navigation (DWN). Based on the Dubins path problem, an algorithm is developed that is updated in real-time with a horizon of three waypoints. The purpose of DWN is to overcome a problem that we find in existing WN for small-class fixed-wing unmanned aerial vehicles (UAV) of not accurately reaching waypoints. This problem results at times in high overshoot and, in the presence of wind disturbances, it can cause a vehicle to miss the waypoint and swirl around it. To prevent this, the DWN creates "new waypoints" that are in the background, called turning points. Examples illustrate the improvement of the performance of WN achieved using the DWN algorithm in terms of the targeting of waypoints while reducing fuel and time.

Keywords

Dubins Path, Waypoint Navigation, Unmanned Aerial Vehicles, Autonomy, Shortest Path, Fuel, Optimization

1. Introduction

The user community of unmanned aerial vehicle (UAV) systems has been growing significantly; the commercial market net worth reached a reported $8.3 billion in 2018 [1], the largest growth in the commercial markets being in small-class (under 55 pounds) vehicles. This paper develops an algorithm for a subset of this class of UAV, in particular, for fixed-wing vehicles. Users of small-class UAV are presently navigating autonomously by open-source algorithms such as Mission Planner, Cape, and Pix4D. These navigational systems employ waypoint navigation (WN), wherein the user enters waypoints, whether a priori (static) or not

(dynamic). The practice of WN in this community is a constraint to which development work adheres. The community is growing, and with that, is expected to demand higher accuracy. In RC flight, the racing community already requires reaching targets accurately and the growth in autonomous flight with its varied missions will create more demand to reach waypoints accurately.

The shortest path between two waypoints with a limited turning radius is called a Dubins path [2]. WN employing Dubins paths has been studied considerably. Ariff and Go conducted a thorough examination of the dynamics involved in WN with a focus on dynamic soaring algorithms [3]. Goerzen, Kong, and Mettler made rigorous comparisons between different path planning algorithms for unmanned vehicle navigation and identified advantages and disadvantages [4]. Manyam, *et al.* expanded the Dubins path problem from two waypoints to three waypoints where all three points and their headings are constrained [5]. Wolek and Woolsey implemented the Dubins path problem with estimates of unsteady velocity disturbances [6]. McGee and Hedrick considered the situations in which the disturbance is less significant and more predictable [7]. The vehicle was placed in a moving frame as a means of finding a piecewise continuous optimal path. Milutinovic *et al.* applied the characteristics of a Dubins vehicle to the circumnavigation algorithm, together with a feedback controller to improve the vehicle's performance [8]. Ketema and Zhao formulated the optimization problem of path planning under wind disturbances and identified situations in which the target points may be not reachable [9]. Meyer, Isaiah and Shima examined and proposed a solution for a vehicle to intercept a moving target with a Dubins path [10]. Criteria were given whether a target can or cannot be intercepted and an analytical solution was obtained by a path elongation algorithm. Wang *et al.* proposed a solution using a Dubins path to solve the formation rendezvous problem of multiple UAVs [11]. Medeiros and Urrutia optimized both path length and direction changes for a Dubins path problem, by dividing the path planning process into several stages, bringing in the algorithm of the travelling salesman's problem [12].

This paper considers a variation on the Dubins path problem and proposes an improved navigation algorithm called Dubins waypoint navigation (DWN). The proposed DWN algorithm considers a current vehicle state (position and velocity) along with the next two prescribed waypoints to plan the vehicle's path. Compared with traditional WN, one more waypoint is involved in the determination of the path, enabling consideration of the manner in which the vehicle should circle around its previous waypoint. The objective is similar to the Dubins path problem in that a shortest path and a minimum fuel path between waypoints consists of straight-line segments and arcs. The major difference is that in DWN the heading at the waypoints are not specified and we consider three points instead of two. Also, for simplicity, the paper only considers the case in which the vehicle's turning radius is constant.

The interest here lies in an algorithm that is updated in real-time with a hori-

zon of three waypoints. The proposed DWN algorithm employs a current vehicle position and heading for the first waypoint, uses a prescribed vehicle position for the second waypoint, not specifying its heading, and uses a third waypoint to determine how the vehicle should circle around the second waypoint.

The purpose of DWN is to overcome a problem that we find in existing WN for small-class fixed-wing vehicles of not accurately reaching waypoints. This problem results from the present use of a waypoint radius as a threshold that determines when to move on to a next waypoint. Presently, a vehicle approaches a first waypoint and switches to navigating to a second waypoint when the vehicle is within a waypoint radius around the first waypoint, thereby not accurately reaching the first waypoint. This leads the operator to desire a smaller waypoint radius. However, the smaller waypoint radius causes several problems. First, it causes high overshoot. Secondly, in the presence of wind disturbances, it can cause a vehicle to miss the waypoint and swirl around it.

When correcting this problem, it became important to preserve the simplicity of prescribing waypoints without adding operator complexity. Therefore, the DWN does not have to change anything in the foreground, including the operator's specified waypoints. Instead, the DWN creates "new waypoints" that can be in the background, called turning points, as explained in the method section.

Section 2 describes the DWN algorithm. Section3 compares performances of WN and DWN with and without unknown wind disturbances, and the paper finishes in Section 4 with a summary and conclusions.

2. Method

2.1. Flow Chart

This section develops the DWN. The DWN eliminates waypoint circles and, instead, employs the new concept of turning points along with the new criterion for updating turning points based on whether or not a turning point is reachable. See **Figure 1** showing the DWN flow chart. As shown, the DWN approaches navigation as an optimization problem with in a local optimization space that consists of a current point and two future waypoints. At any instant, the vehicle seeks to reach a turning point that is determined by minimizing fuel along the path up to the second future waypoint constituting the horizon. The trajectory, and hence the optimization problem, is updated once the turning point is unreachable. The reachability condition is determined from the vehicle's position, heading, and turning radius. For the purposes of real-time implementation, the trajectory *may* be updated more frequently than when the turning point is updated (like under the conditions of high winds). The stated optimization problem yields desired paths that are determined in closed-form.

2.2. Finding Turning Points

Consider the illustrative WN problem shown in **Figure 2** and, in particular, the navigation to the second and third waypoints shown in **Figure 3**.

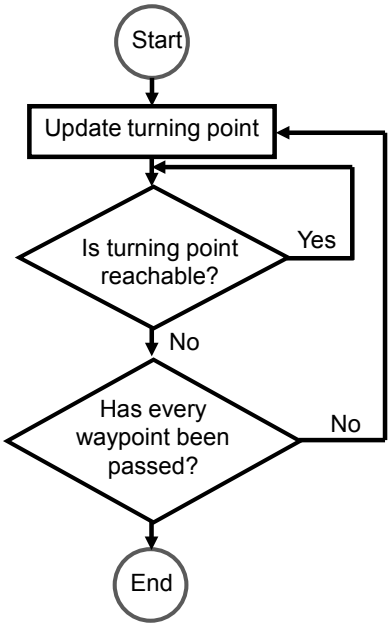

Figure 1. Flow chart of the DWN.

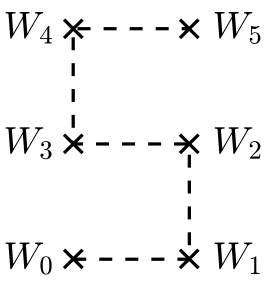

Figure 2. Illustrative WN problem.

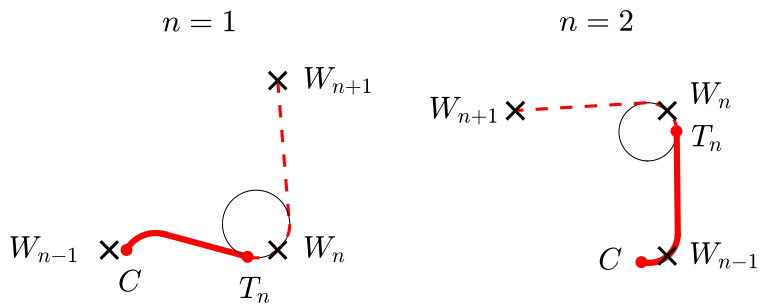

Figure 3. Navigating to the 2nd and 3rd waypoints.

The vehicle starts at W_0 and heads to W_1. At this point, the DWN considers waypoints W_0, W_1, and W_2. The other points are beyond the horizon.

Figure 3 shows the paths from W_0 to W_2 (left drawing) and from W_1 to W_3 (right drawing).

As shown, T_1 denotes the turning point associated with W_1. It lies on the line tangent to a turning circle (not to be confused with the waypoint circle in classical WN) that has a turning radius R. The turning radius, specified by the user,

is the desirable radius of the turns around waypoints. An optimization problem minimizes the fuel from W_0 around and touching W_1 to W_2. The orientation of the turning circle is free to rotate about W_1 so the optimization problem is a minimization problem of fuel expressed in terms of the turning circle's orientation. The fuel consumed during a turn is greater than the fuel consumed while flying in a straight line over the same distance. Even though the minimization is over the distance extending to W_2, the DWN updates the vehicle's path once it reaches point T_1. Thus, the vehicle does *not* follow the section of the path from T_1 to W_2 (In Figure 3, the paths followed are solid lines and the paths not followed are dashed lines.) The DWN recalculates that not-followed section of the path in the next iteration. Under ideal conditions, DWN updates the planned path when the vehicle crosses T_1, at which point T_1 is determined to be unreachable. Under real conditions, disturbances can result in the vehicle *not* reaching point T_1 accurately. Point T_1 is determined to be unreachable at a point C that is different but near-point T_1. Referring to the right drawing, the new horizon is set to W_3 and the new turning point T_2 is determined from the vehicle's current point C and current direction of flight and the locations of points W_2 and W_3. As shown, the optimization problem now minimizes the fuel from C around and touching W_2 to W_3. The optimization problem is now a minimization problem of fuel from C to W_3 expressed in terms of the new turning circle's orientation. The DWN updates the iterations until the vehicle reaches its last waypoint.

2.3. Key Parameters

Figure 4 shows the n^{th} iteration. The vehicle is heading to W_n. A coordinate system was set up so its x-axis points from C to W_n and its y-axis is perpendicular to x in the direction of W_{n+1}. The data is scaled such that the x and y coordinates represent non-dimensional lengths of distance divided by turning radius R; equivalently $R = 1$. We perform all of the calculations after the geometric parameters are normalized with respect to R.

As shown, α is the heading angle (between –180˚ and 180˚) and β is the turn angle (β is between 0˚ and 90˚)[1]. We assume that there are no sharp turns between specified waypoints (the angle between any two lines connecting three consecutive waypoints is never greater than 90˚). At point C, the vehicle turns

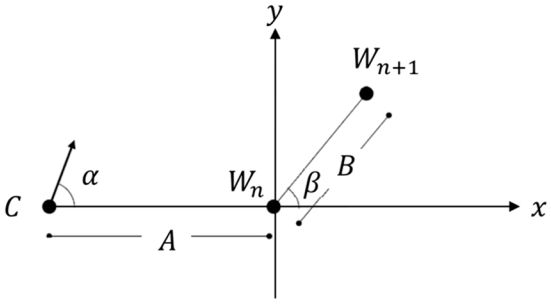

Figure 4. Set up of the coordinate system.

[1]The x axis, because it is along the line through C and W_n, can cause β to exceed 90˚.

counter-clockwise (CCW), or clockwise (CW). Likewise, it turns around W_n CCW or CW. In total, there are four cases: CCW-CCW, CCW-CW, CW-CCW, and CW-CW.

The interest lies in finding closed-form expressions for the paths as functions of A, B, α and β by curve fitting. Toward this end, note that the minimum fuel paths are continuous with respect to these parameters for each case, individually. Discontinuities occur when transitioning from one case to another. Therefore, the curve-fitting needs to be performed for each case and the case that consumes the smallest amount of fuel yields the true minimum (See Figure 5).

2.4. Optimization Problem

Figure 6 shows the geometry of the CW-CCW case. The fuel from C to W_{n+1} is a function of the orientation angle θ of the turning circle. (The other cases, not shown, are similar.) The fuel is

$$Fu = w_a L_a + w_s L_s \tag{1}$$

where L_a is the total length of the two arcs (around the first and second turning circles), L_s is the length of the two straight segments (after the first and second turning circles), and w_a and w_s are corresponding weights (in units of fuel per length). Letting $w_a = w_s = 1$ (in which the weights have units of 1), yields the shortest path problem:

$$L = L_a + L_s \tag{2}$$

In the results section, we will show that the shortest path and minimum fuel solutions are nearly indistinguishable under a broad range of conditions, allowing the shortest path solution to approximate the minimum fuel solution. This is important because the shortest path solution is independent of the vehicle, increasing ease of implementation and the versatility of DWN.

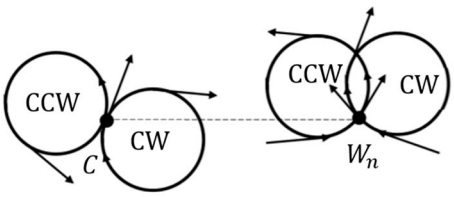

Figure 5. Different directions of turning.

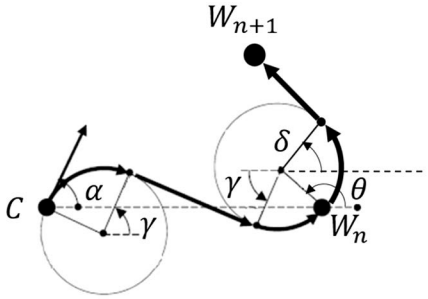

Figure 6. Geometry for the CW-CCW case.

Referring again to Figure 6, the first straight line segment starts at tangential point (x_1, y_1) and ends at turning point $T_n = (x_2, y_2)$. The second line segment starts at tangential point (x_3, x_4). Note that we define the first circle by the current position C and the vehicle's heading; it is either CCW or CW. The location of the second circle depends on the orientation angle θ. The second line segment is tangent to the second circle and passes through W_{n+1} and therefore is uniquely expressed in terms of the angle δ shown, which, in turn, depends on θ.

The coordinates x_1, y_1, x_2, y_2, x_3, and y_3 for each of the four cases are given in Table 1 and Table 2.

By appropriately manipulating the geometric relationships, the path length (the objective function) becomes a function of the orientation angle θ of the turning circle, and the solution of this 3-point optimization problem becomes a function of θ. This part of the iteration can be solved off-line and curve-fit to determine T_n. There are two ways to fit T_n. One way is to directly fit T_n to A, B, α, and β. The second way is to fit θ to A, B, α, and β followed by determining T_n from θ. The second way of curve fitting the data will inherently lead to a more accurate curve fit than the first way because the relationship between T_n and θ is rather complicated and its analytical relationship is available, too [13].

Table 1. Location of the end points of the first line segment.

	x_1	y_1	x_2	y_2
CW-CW	$-A+\sin\alpha+\cos\gamma$	$-\cos\alpha+\sin\gamma$	$\cos\theta+\cos\gamma$	$\sin\theta+\sin\gamma$
	$\gamma=\arctan\dfrac{\sin\theta+\cos\alpha}{\cos\theta+A-\sin\alpha}+\dfrac{\pi}{2}$			
CW-CCW	$-A+\sin\alpha+\cos\gamma$	$-\cos\alpha+\sin\gamma$	$\cos\theta+\cos\gamma$	$\sin\theta+\sin\gamma$
	$\gamma=\arctan\dfrac{\sin\theta+\cos\alpha}{\cos\theta+A-\sin\alpha}+\arccos\dfrac{2}{\sqrt{(\sin\theta+\cos\alpha)^2+(\cos\theta+A-\sin\alpha)^2}}$			
CCW-CW	$-A-\sin\alpha+\cos\gamma$	$\cos\alpha+\sin\gamma$	$\cos\theta-\cos\gamma$	$\sin\theta+\sin\gamma$
	$\gamma=\arctan\dfrac{\sin\theta-\cos\alpha}{\cos\theta+A+\sin\alpha}-\arccos\dfrac{2}{\sqrt{(\sin\theta-\cos\alpha)^2+(\cos\theta+A+\sin\alpha)^2}}$			
CW-CCW	$-A-\sin\alpha+\cos\gamma$	$\cos\alpha+\sin\gamma$	$\cos\theta-\cos\gamma$	$\sin\theta-\sin\gamma$
	$\gamma=\arctan\dfrac{\sin\theta-\cos\alpha}{\cos\theta+A+\sin\alpha}-\dfrac{\pi}{2}$			

Table 2. Location of second line segment.

	x_3	y_3
CW-CW & CCW-CW	$\cos\theta+\cos\delta$	$\sin\theta+\sin\delta$
	$\delta=\arctan\dfrac{B\cos\beta-\cos\theta}{B\sin\beta-\sin\theta}+\arccos\dfrac{1}{\sqrt{(B\cos\beta-\cos\theta)^2+(B\sin\beta-\sin\theta)^2}}$	
CW-CCW & CCW-CCW	$\cos\theta+\cos\delta$	$\sin\theta+\sin\delta$
	$\delta=\arctan\dfrac{B\cos\beta-\cos\theta}{B\sin\beta-\sin\theta}-\arccos\dfrac{1}{\sqrt{(B\cos\beta-\cos\theta)^2+(B\sin\beta-\sin\theta)^2}}$	

2.5. Updating the Turning Point

As described earlier, the DWN guides a vehicle to a waypoint until it becomes unreachable, at which point it updates the turning point. Figure 7 shows the reachability condition. As shown, a reachability zone consists of CCW and CW circles "enclosed" on the rear by a tangent line. The reachability zone is fixed to the vehicle. The vehicle is trying to reach waypoint T_n. At agiven instant of time, T_n could be either inside or outside the reachability zone. (In the figure, T_n is outside of the reachability zone.). As soon as it is inside the reachability area, T_n becomes unreachable, and the DWN updates the turning point to T_{n+1}.

Note that the purpose of "enclosing" the CCW and CW with the line segment was to prevent the possibility of point C passing T_n undetected, which would otherwise be possible because the detection is not truly performed continuously but only at discrete instances of time.

Comparing the reachability zone used in DWN with the waypoint circle used in WN, first notice that the reachability zone, in the absence of disturbances, allows the vehicle to accurately reach a first waypoint, as opposed to beginning its turn to a second waypoint before reaching the first waypoint. Secondly, in WN, the operator can easily be fooled into the natural desire to select a waypoint radius that is smaller than the vehicle's turning radius toward the goal of more closely reaching a waypoint. However, when doing so in the presence of a wind disturbance, this can result in missing the waypoint, finding it difficult to reach, and causing the vehicle to swirl around the waypoint. In the DWN, no waypoint radius is considered.

2.6. Parameterization

We formulated the 3-point horizon optimization problem to keep the computational effort minimal for real-time implementation. To further reduce computational effort and for robustness, we parameterized the solution, effectively reducing it to a look-up table. Toward this end, we determined the orientation angle θ of a turning circle as a function of the four parameters A, B, α, and β. We parameterized the orientation angle by smoothly fitting the data to the four parameters separately for each case (CCW-CCW, CCW-CW, CW-CCW, and

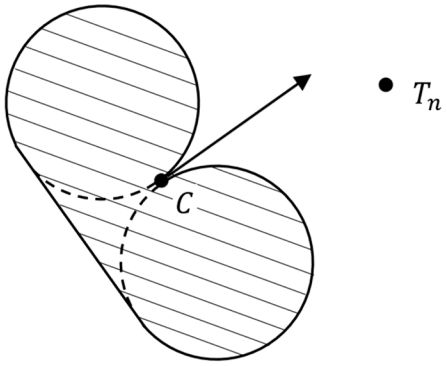

Figure 7. Examples of reachable and unreachable points.

CW-CW). Toward distinguishing between the different cases, we needed to parameterize the boundaries of each of the cases in terms of the four parameters. In particular, α and β transition from case to case due to changes in A and B. For example, consider the graph of α versus β shown in Figure 8 for particular parameters ($A = B = 4$, $w_a = w_s = 1$).

As shown, there are three boundaries and three interior regions. (It turns out that the CW-CW case never produces an optimal orientation angle.) We express the boundary of the turn angle β as a function of the heading angle α in which its coefficients are expressed as a function of the distances A and Bas follows:

$$\beta = \begin{cases} a_0 + a_1\delta + a_2\delta^2 & \alpha \geq -a_1/2a_2 \\ \dfrac{4a_0a_2 - a_1^2}{4a_2} & \alpha \leq -a_1/2a_{2\,2} \end{cases} \tag{3}$$

$$a_i = b_{0i} + b_{1i}A + b_{2i}A^2 + b_{3i}B + b_{4i}B^2 + b_{5i}AB \ (i = 1, 2, 3) \tag{4}$$

We determined the coefficients b_{0i} through b_{6i} (i = 1, 2, 3) separately for each of the cases. The real-time procedure of updating the turning points is as follows:

1) Calculate A, B, α, and β at an instant of time from a current state of the vehicle.

2) Calculate the coefficients associated with the boundaries between the interior regions (See [13]).

3) Determine the interior region in which the vehicle lies.

4) Calculate the orientation angle θ of the next turning circle (See [13]).

5) Calculate the next turning point T_n (expressed in terms of θ analytically).

3. Results

Let us continue with the 6-waypoint example and compare WN and DWN. In WN, navigation performance depends on the minimum turning radius and the waypoint radius. In DWN, navigation performance depends on minimum turning radius alone. The feedback control algorithms are the same for WN and DWN. However, the resulting overshoots differ. An illustrative feedback control algorithm described below accounts for turning toward an endpoint limited by the vehicle's minimum turning radius R and serves as a basis for the comparison between WN and DWN.

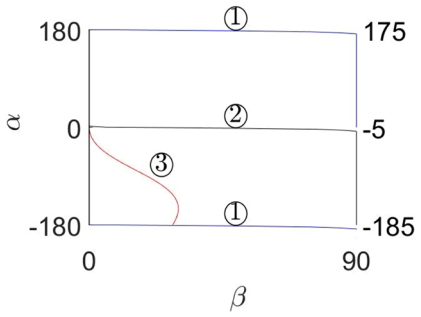

Figure 8. Distributions of different turning conditions.

3.1. An Illustrative Feedback Control Algorithm

Consider Figure 9 showing a reference path with endpoints that are waypoints in the case of WN and that are turning points in the case of DWN. A vehicle at time t_i heads in a direction ϕ_{Hi} (not shown) a distance s_i from the reference path. The vehicle travels a distance $v\Delta t$ over a time step. At time t_{i+1}, the vehicle heads in a direction $\phi_{i+1} = \phi_i + \psi_i$ in which ψ_i is the turn angle over one iteration, given by

$$\psi_i = \begin{cases} \psi_{\max} & \phi_{Ti} + \phi_{Ri} > \psi_{\max} \\ -\psi_{\max} & \phi_{Ti} + \phi_{Ri} < -\psi_{\max} \\ \phi_{Ti} + \phi_{Ri} & \left|\phi_{Ti} + \phi_{Ri}\right| \le \psi_{\max} \end{cases} \quad \phi_{Ri} = \left|gs_i + h\dot{s}_i\right| \frac{\phi_{Ti}}{\left|\phi_{Ti}\right|} \quad \psi_{\max} = \frac{v\Delta t}{R} \tag{5}$$

Above, ϕ_{Ti} is a target angle and ϕ_{Ri} is a reference angle (all of the angles are positive counter-clockwise), g and h are control gains, and $\phi_T/\left|\phi_T\right|$ determines whether ϕ_R is counter-clockwise or clockwise.

The parameters used in the results, both with and without wind disturbances, are as follows: The minimum turning radius was $R = 0.7$, the time step was 0.02, the vehicle speed was $v = 0.8$, and the control gains were $g = 0.1$ and $h = 5.25$. With these parameters the turning radius was 0.7.

3.2. The Shortest Path Approximation

Let us compare the minimum fuel and the shortest path solutions, the latter determined when $w_a = w_s = 1$. In practice, the distance between waypoints is at least four times larger than the turning radius R. Therefore, we will only consider distances A and B that are more than or equal to 4 times greater than R. When the distances A and B are exactly 4 times greater than R, the vehicle is flying in a straight line the least amount of the time and, for this case, we expect the difference between the minimum fuel solution and the shortest path solution to be the greatest. Figure 10 shows minimum fuel solutions in which w_a varies from 1—the shortest path case—to $w_a = 1.5$, for which the fuel cost during turning flight is 50% greater than the fuel cost during straight flight (In small-class fixed-wing vehicles the fuel penalty of turning is less than 50%). As shown, the orientation angle of the turning circle does not change more than 4˚. The minimum-fuel path is very close to the shortest path and therefore the shortest path approximates the minimum fuel path. The use of the shortest path would therefore be

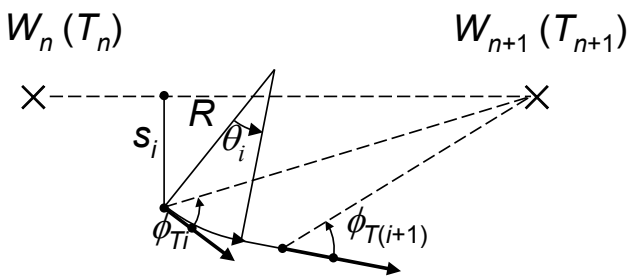

Figure 9. Parameters for the controller.

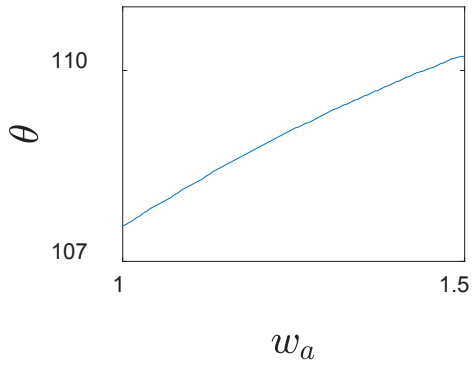

Figure 10. Example of a CCW-CCW condition ($A = B = 4$, $\alpha = -45^\circ$, $\beta = 30^\circ$, CCW-CCW).

acceptable in the vast majority of small-class UAV problems. This finding is significant because the shortest path solutions apply to any vehicle whereas the minimum fuel solutions need to consider vehicle parameters. In the results below, we employ the shortest path solutions.

3.3. Navigation in the Absence of an Unknown Wind Disturbance

Figure 11 compares WN and DWN in the absence of an unknown wind disturbance. Indeed, when the wind disturbance is completely accommodated for (by subtracting it out), DWN and WN exhibit the performance shown in **Figure 11**. The minimum turning radius R of the vehicle is the same in each case. The three cases consider WN waypoint radii of $0.1R$, $0.5R$, and R, respectively.

First consider DWN (solid lines). In each case, the vehicle passes through waypoints exactly and follows the minimum fuel paths—by design. Next, consider WN. In the first case, the vehicles passes close to the waypoints but with significant overshoot. In the other two cases, we see as the waypoint radius increases that the vehicle passes farther from the waypoints. In the third case, the vehicle undershoots the desired path. Note that in the WN and DWN cases given above, the feedback control had very little effect on the response because there was no unknown wind disturbance.

3.4. Navigation in the Presence of an Unknown Wind Disturbance

Figure 12 shows WN and DWN responses in the presence of a wind disturbance. The wind velocity is constant and unknown, set to 10% of the drone's speed directed from the bottom to the top of the figures (in the horizontal plane) and not accommodated for.

In the left figure, the waypoint radius was set to be small, intending to achieve higher accuracy. However, the wind exceeds the capability of the feedback controller and the vehicle just misses the second waypoint sending it into a swirl. DWN also misses the turning point paired with the second waypoint but moves on to complete its journey. It can do this because it recognizes the waypoint to be unreachable. In the middle and the right figures, the waypoint radii are larger and WN manages to navigate the vehicles. It is worth noticing that errors exist

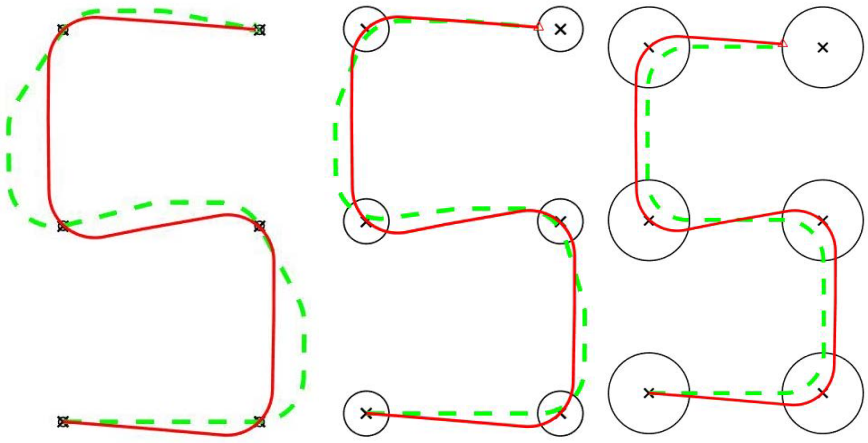

Figure 11. WN and DWN ($R_{WP} = 0.1R$, $0.5R$, and R).

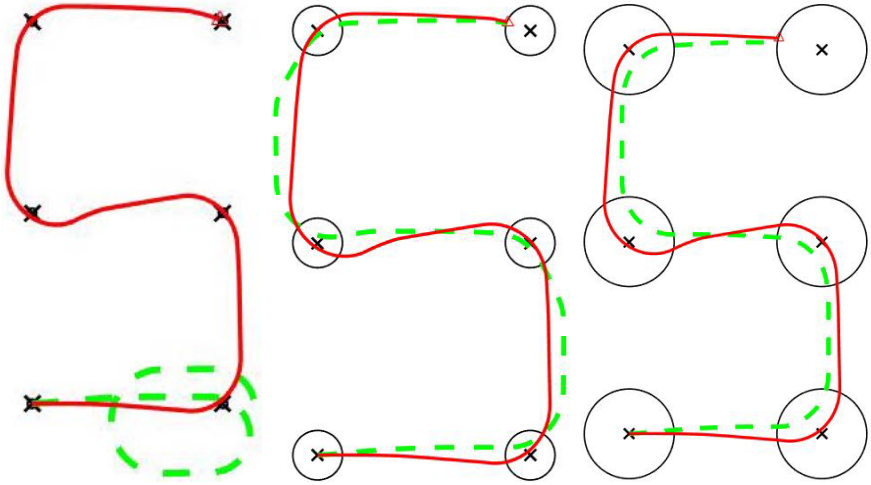

Figure 12. WN with and without the DWN in the presence of a disturbance ($R_{WP} = 0.1R$, $0.5R$, and R).

for both WN and DWN. The DWN errors result from using the turning points for navigation instead of waypoints. It is clear that the DWN errors are smaller than the WN errors due to its features already discussed, hence one still recognizes that DWN improves the performance of the WN navigation.

4. Conclusion

WN suffers from an existing trade-off between minimum turning radius and waypoint radius that prevents vehicles from reaching waypoints and closely following desired paths. This paper developed an algorithm, called Dubins waypoint navigation (DWN) that remedies this problem by *not* using waypoint circles. Instead, we introduced a reachability zone and turning points. The DWN significantly reduces undershoot, overshoot, and swirl. Furthermore, the algorithm remedies the problem in a way that can be hidden from the operator to avoid confusion. The paper showed, by establishing a horizon that includes two future waypoints, how to improve the performance of WN in terms of the tar-

geting of waypoints while reducing fuel and time. Pertaining to real-time implementation, we also reduced the computational effort, essentially eliminating it, by parameterizing the shortest path solution. We also compared WN and DWN tracking performance (in the absence of an unknown disturbance) and regulation performance (in the presence of an unknown disturbance). The comparisons are illustrative of the improvements in path following (tracking and regulation) obtained by DWN.

Acknowledgements

The authors gratefully recognize the Namibia Wildlife Aerial Observatory (WAO) for its support of this work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Drubin, C. (2013) UAV Market Worth $8.3 B by 2018. *Microwave Journal*, **56**, 37.

[2] Tsourdos, A., White, B. and Shannugavel, M. (2011) Path Planning in Two Dimensions. In: Tsourdos, A., White, B. and Shanmugavel, M., Eds., *Cooperative Planning of Unmanned Aerial Vehicles*, Wiley, Chichester, 30. https://doi.org/10.1002/9780470974636

[3] Ariff, O. and Go, T. (2011) Waypoint Navigation of Small-Scale UAV Incorporating Dynamic Soaring. *The Journal of Navigation*, **64**, 29-44. https://doi.org/10.1017/S0373463310000378

[4] Goerzen, C., Kong, Z. and Mettler, B. (2010) A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *Journal of Intelligent and Robotic Systems*, **57**, 65-100. https://doi.org/10.1007/s10846-009-9383-1

[5] Manyam, S., Rathinam, S., Casbeer, D. and Garcia, E. (2017) Tightly Bounding the Shortest Dubins Paths through a Sequence of Points. *Journal of Intelligent & Robotic Systems*, **88**, 495-511. https://doi.org/10.1007/s10846-016-0459-4

[6] Wolek, A. and Woolsey, C. (2015) Feasible Dubins Paths in Presence of Unknown, Unsteady 5 Velocity Disturbances. *Journal of Guidance Control and Dynamics*, **38**, 782-786. https://doi.org/10.2514/1.G000629

[7] McGee, T. and Hedrick, J. (2007) Optimal Path Planning with a Kinematic Airplane Model. *Journal of Guidance, Control, and Dynamics*, **30**, 629-633. https://doi.org/10.2514/1.25042

[8] Milutinovic, D., Casbeerm, D., Cao, Y. and Kingston, D. (2017) Coordinate Frame Free Dubins Vehicle Circumnavigation Using Only Range-Based Measurements. *International Journal of Robust and Nonlinear Control*, **27**, 2937-2960. https://doi.org/10.1002/rnc.3718

[9] Ketema, Y. and Zhao, Y. (2010) Micro Air Vehicle Trajectory Planning in Winds. *Journal of Aircraft*, **47**, 1460-1463. https://doi.org/10.2514/1.C000247

[10] Meyer, Y., Isaiah, P. and Shima, T. (2015) On Dubins Paths to Intercept a Moving Target. *Automatica*, **53**, 256-263. https://doi.org/10.1016/j.automatica.2014.12.039

[11] Wang, Z., Lium L., Long, T. and Xu, G. (2018) Efficient Unmanned Aerial Vehicle

Formation Rendezvous Trajectory Planningf Using Dubins Path and Sequential Convex Programming. *Engineering Optimization*, **51**, 1412-1429. https://doi.org/10.1080/0305215X.2018.1524461

[12] Medeiros, A. and Urrutia, S. (2010) Discrete Optimization Methods to Determine Trajectories for Dubins' Vehicles. *Electronic Notes in Discrete Mathematics*, **36**, 17-24. https://doi.org/10.1016/j.endm.2010.05.003

[13] Silverberg, L. and Xu, D. (2018) Minimum-Fuel Hidden Layer Heuristic for Small-Class UAV. Master of Science Thesis, Mechanical Engineering, North Carolina State University, Raleigh.

Nomenclature

A = distance between current position and the following waypoint
a = coefficient for curve fitting β with respect to δ
B = distance between the following two successive waypoints
b = coefficient for curve fitting a with respect to A and B
C = current position
F_u = total fuel cost
g = proportional term for PID controller
h = derivative term for PID controller
L_a = total length of the arcs
L_s = total length of the straight segments
R = turning radius of the vehicle
T = target point
v = speed of the vehicle
W = waypoint
w_a = fuel cost per unit length travelled on an arc
w_s = fuel cost per unit length travelled on a straight line
α = angle of heading in local system
β = angle between the following two successive waypoints in local system
γ = angular position of the first tangential point in local system
δ = angular position of the last tangential point in local system
θ = angular position of the center of the turning circle
ϕi = angle that denotes the heading in global frame at the ith time instance
ϕ_R = angle of turning resulted from the feedback controller
ϕ_T = angle between heading and the target
ψ = angle of turn in one time step