ABSTRACT

XU, DAHAN. Minimum-Fuel Hidden-Layer Heuristic for Waypoint Navigation of Small-Class UAV. (Under the direction of Dr. Lawrence Silverberg).

This thesis develops a minimum-fuel algorithm for the improvement of waypoint navigation (WN) of small-class UAV, referring to it as the minimum-fuel hidden-layer (MFHL). The MFHL minimizes fuel over a horizon of three waypoints. Its purpose is to overcome the problem that we find in existing WN for small-class fixed-wing vehicles which results from a natural trade-off that exists between minimum turning radius and threshold radius. In the MFHL we introduce the concept of the turning circle and its reachability condition to adjust waypoints "in the background" of WN algorithms. The algorithm can serve as a background layer to an existing WN algorithm, hidden from the user. The note develops both minimum fuel and minimum distance heuristics and compares WN with and without the MFHL in the absence of disturbances, focusing on navigation errors, and with disturbances, focusing on the ability to correct for errors. The performance improvements suggest that MFHL be considered for adoption in WN in the small-class fixed-wing community.

Minimum-Fuel Hidden -Layer Heuristic for Waypoint Navigation of Small-Class UAV

by
Dahan Xu

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Mechanical Engineering

Raleigh, North Carolina
2018

APPROVED BY:

_____	_____
Lawrence Silverberg Committee Chair	Andre Mazzoleni
_____	
Scott Ferguson	

BIOGRAPHY

Dahan Xu takes his master's study in mechanical engineering at North Carolina State University from August 2017 to December 2018, mainly focusing on dynamics and relevant simulations. This thesis is based on his graduate research under the instructions of Dr. Lawrence Silverberg. The research is simulation-based and hopefully it will be of help in the UAV community.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Overall Background

The user community of unmanned systems has been growing significantly. The commercial market net worth reached a reported $8.3 billion in 2018 [1]. We find the largest growth in the commercial markets, in small-class (under 55 pounds) vehicles, in particular. 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 Missionplanner, 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, demanding higher performance. In RC flight, the racing community requires more precise path following and the growth in autonomous flight is expected to demand greater precision in path following resulting from its applications.

Ongoing research in improving flight performance of autonomous vehicles tends to address problems, more broadly. For example, in 2015, Bakolas and Tsiotras [2] developed a nice dynamic model of a UAV in pursuit of a target in the presence of disturbances. In 2016, Ratnoo [4] accounting for turning factors in an algorithm that pursues a moving target. In 2017, Maravall, Lope, and Fuentes [3] tried out a vision system. These development efforts are typical but would be difficult to transfer over to small-class fixed-wing vehicles because of the WN constraint. The authors did not find papers that specifically address WN path planning problems that hidden layers can remedy.

1.2 Waypoint Navigation Algorithm

The WN algorithm guides the vehicle in straight-line paths between waypoints. The feedback control has a limit on banking which creates, effectively, a minimum turning radius at a particular flight speed. When approaching a target waypoint, the threshold criterion updates the target waypoint when the vehicle is within a certain radius of it, called the threshold radius. The minimum turning radius and the threshold radius create a trade-off between overshooting the desired path and precisely reaching target points. Due to intermittent wind disturbances, the threshold criterion also results in a tradeoff between reaching the waypoint and missing the waypoint, the latter intermittently resulting in the vehicle turning around to try again and reach the waypoint. Indeed, the threshold criterion, because it is distinct from the smallest desirable turning radius and from wind disturbance considerations, can be difficult to tune, and can lead to undesirable performance. In addition to not reaching a waypoint precisely, the aforementioned problems result in a loss of fuel. Figure 1.1 gives three typical conditions where the performance of traditional WN algorithm can be improved.

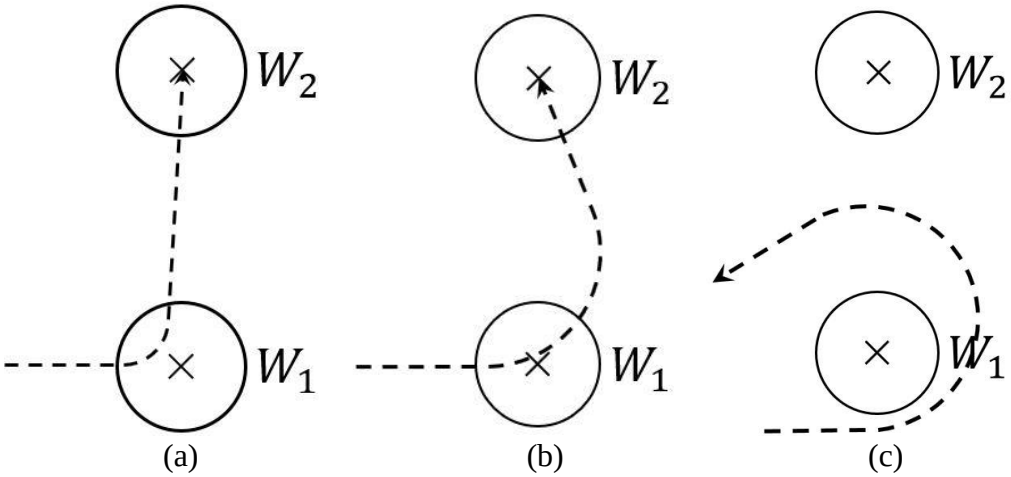


Figure 1.1 Typical conditions where the waypoints are not reached.

In Figure 1.1, the crossings labelled as W_1 and W_2 are two waypoints, the dashed lines are the trajectories of vehicles going from W_1 to W_2 with WN algorithm, and the circles centered at the waypoints represent the threshold. In Figure 1.1(a), the threshold radius is greater than the turning radius of the vehicle. Since the vehicle is navigated to make a turn right after it touches the threshold circle, the waypoint is missed. The error results from a large threshold radius can be decreased by simply decreasing the threshold radius but it will lead us to Figure 1.1(b), where the turning radius is smaller than the threshold radius and there is a significant overshoot. This overshoot of course causes a waste of fuel. Figure 1.1(c) demonstrates another possible condition when the turning radius is greater than the threshold radius. If the vehicle has already deviated from ideal path due to strong disturbance, it is likely that the vehicle will swirl around certain waypoint without ever touching it. The motivation to propose the MFHL algorithm in this thesis is to address these problems and improved the vehicle performance.

CHAPTER 2

Method

This chapter develops the minimum fuel hidden layer (MFHL). The MFHL preserves the waypoint navigation strategy in the foreground (The interface and waypoints specified by the user do not change) but changes the waypoints and the criterion for updating waypoints in the background. The MFHL replaces the traditional threshold criterion with a new criterion that, in effect, gives priority to a minimum turning radius, to reaching the waypoint, and to avoiding missing it – without a trade-off. Toward this end, the MFHL 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.

2.1 Flow Chart

The MFHL approaches navigation as an optimization problem within 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. We provide in the appendix closed-form geometric expressions needed for real-time implementation. See Figure 2.1 showing the MFHL flow chart.

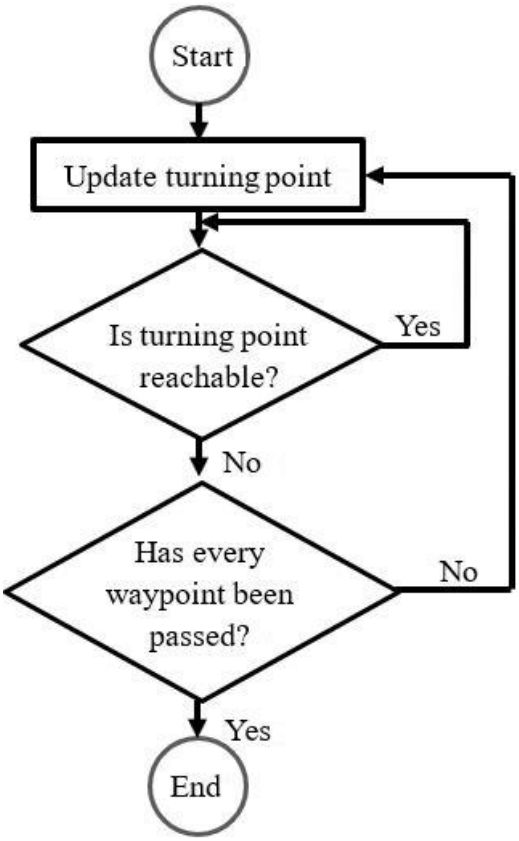


Figure 2.1 Flow chart of MFHL.

2.2 Illustration with an Example

To illustrate the MFHL, consider the 6-waypoint navigation problem shown in Figure 2.2. We will first treat the disturbance-free and error-free navigation problem, that is, the ideal navigation problem, followed by navigation in the presence of disturbances. The former focuses on the navigation part of the problem and the latter on performance in the presence of disturbances (feedback).

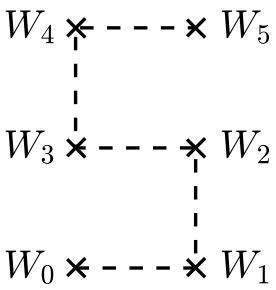
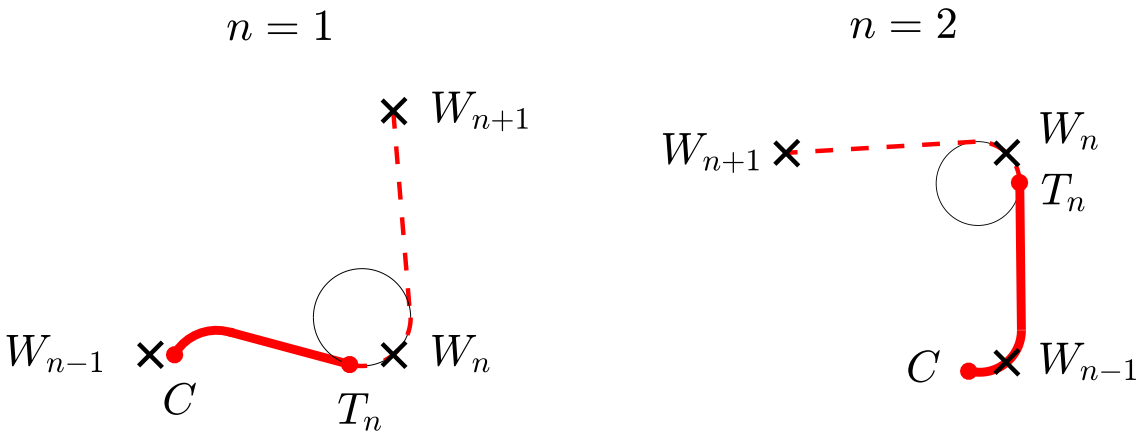


Figure 2.2 Set up of a 6-waypoint example.

Referring to Figures 2.2 and 2.3, the vehicle starts at W_0 and heads to W_1. At this point, the MFHL is considering waypoints W_0, W_1, and W_2. The other points are beyond the horizon. Figure 2.3 shows the paths from W_0 to W_2 (top drawing) and from W_1 to W_3 (bottom drawing).



(a)

(b)

Figure 2.3 Navigating to the 2nd and 3rd waypoints.

As shown, T_1 denotes the turning point associated with W_1. It lies on the line tangent to a turning circle that has a turning radius R. The turning radius is the minimum turning radius specified by the user. The 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. Typically, the fuel can be as much as 40% greater when turning than when following a straight line depending on turning radius, speed, and vehicle type. Even though the minimization is over the distance extending to W_2, the MFHL 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; the MFHL recalculates that section of the path in the next iteration. Under ideal conditions, MFHL updates the planned path when the vehicle crosses T_1, at which point T_1 is determined to be unreachable. Under real conditions, disturbances result in the vehicle *not* reaching point T_1 precisely. Point T_1 is determined to be unreachable at a point C that is different but close to 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 MFHL updates the iterations until the vehicle reaches its last waypoint.

2.3 Geometry

In order to derive a closed-form solution for the path planning, we have to construct a coordinate system involving the vehicle and the waypoints and identify the key parameters. For 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 the calculations after the geometric parameters are normalized with respect to R. The system is shown in Figure 2.4.

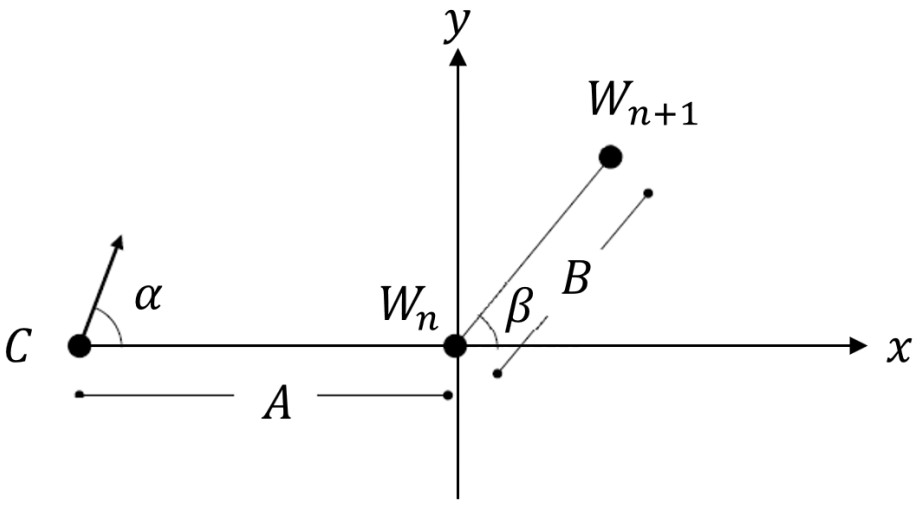


Figure 2.4 Set up of the coordinate system.

As shown, α is the heading angle (between -180^0 and 180^0) and β is the turn angle (between 0 and 90^0)[1]. At point C, the vehicle turns 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 minimum fuel solution is calculated for each case and the smallest of them yields the true minimum (See Figure 2.5).

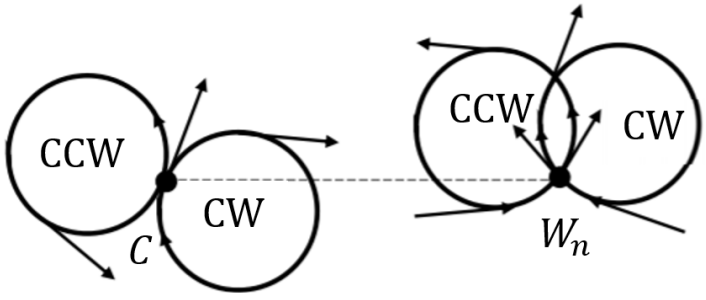


Figure 2.5 Different directions of turning.

[1] The x axis, because it is along the line through C and W_n, can cause β to exceed 90^0.

2.4 Construct the Optimization Problem

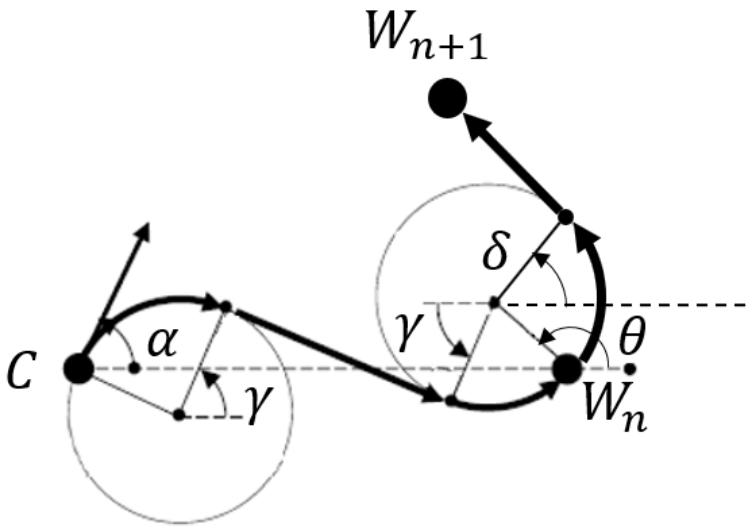


Figure 2.6 Geometry for the CW-CCW case.

Figure 2.6 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 $Fu = w_a L_a + w_s L_s$ where L_a is the length of the two arcs, L_s is the length of the two straight segments, and w_a and w_s are corresponding weights. Letting $w_a = w_s = 1$, yields the minimum distance problem: $L = L_a + L_s$. In the results section, we will show that the minimum distance and minimum fuel solutions are nearly indistinguishable under a broad range of conditions, allowing the minimum distance solution to approximate the minimum fuel solution. This is important because the minimum distance solution is independent of the vehicle's properties. This increases the versatility of the MFHL and ease of implementation.

Referring again to Fig. 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 2.1 and Table 2.2.

Table 2.1 Location of the end points of the first line segment.

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

Table 2.2 Location of the starting point of the second line segment

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

By appropriately manipulating the geometric relationships, the path length (the objective function) becomes a function of the orientation angle θ of the turning circle.

2.5 Updating Criterion

As described earlier, the MFHL guides a vehicle to a waypoint until it becomes unreachable, at which point it updates the waypoint. Fig. 2.7 shows the reachability condition. As shown, a reachability area consists of CCW and CW circles "enclosed" on the rear by a tangent line. At this time instance, the point T_{n-1} is already inside the shaded area so it is determined to be "unreachable". Consequently, point T_{n-1} is omitted and the vehicle is being navigated to T_n. At any given instant of time, T_n could be either inside or outside the reachability area. In this specific figure, T_n is outside of the reachability area so it is still reachable. As soon

as it is inside the reachability area, T_n becomes unreachable, and the MFHL updates the waypoint to T_{n+1}.

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

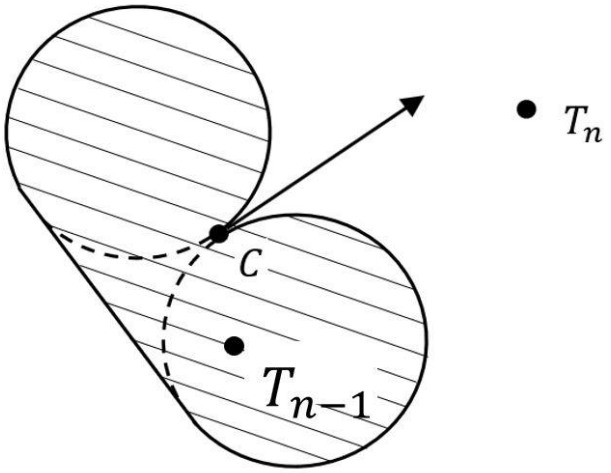


Figure 2.7 Examples of reachable and unreachable points.

2.6 Parameterization

Clearly, 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. For each of the four cases pertaining to turning CW and CCW, the orientation angle is a continuous function of the four parameters. Discontinuities, however, arise when transitioning from one case to another. Thus, we parameterized the

orientation angle by smoothly fitting the data to the four parameters separately for each case. 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 Fig. 2.8 for particular parameters ($A = B = 4$, $w_a = w_s = 1$).

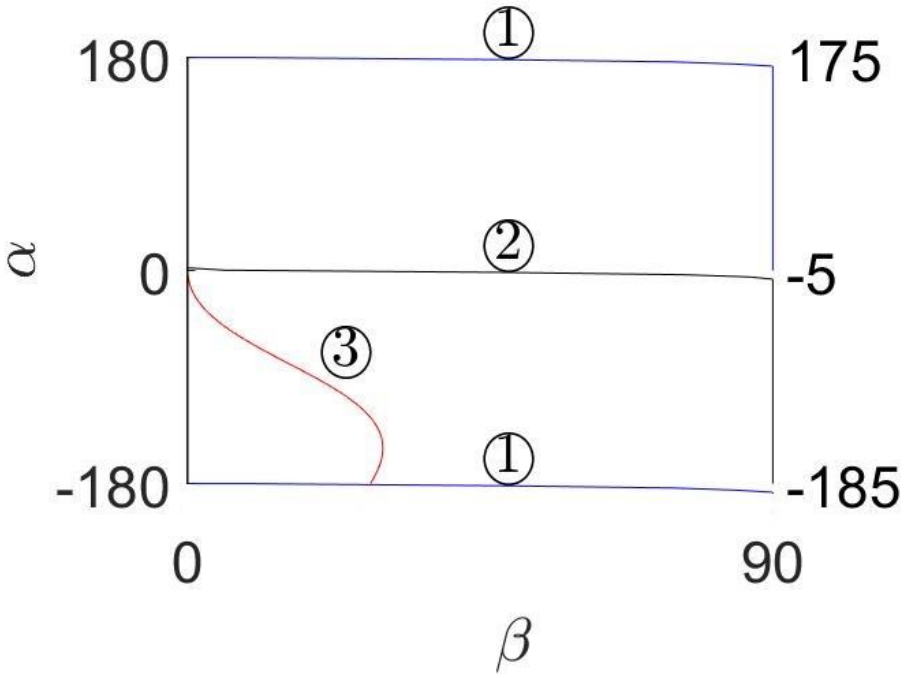


Figure 2.8 Distributions of different turning conditions.

As shown, there are three boundaries and three interior regions. (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 *B*:

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

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

We determined the coefficients b_{0i} through b_{6i} (i = 1, 2, 3) separately for each of the cases (See Appendix).

CHAPTER 3

Results

Let us continue with the 6-waypoint example and compare the current WN problem with and without the MFHL. In the WN problem, navigation performance depends on the minimum turning radius and the threshold radius. With the MFHL, navigation performance depends on minimum turning radius alone. Feedback control is the same whether or not the WN problem employs the MFHL. However, the resulting overshoots differ.

3.1 Results Comparisons for No Disturbance Cases

Figure 3.1 compares WN with and without the MFHL. The minimum turning radius R of the vehicle is the same in each of the cases. We see three cases that differ by the waypoint circles. The waypoint circles are 0.5R, R, and 2R, respectively.

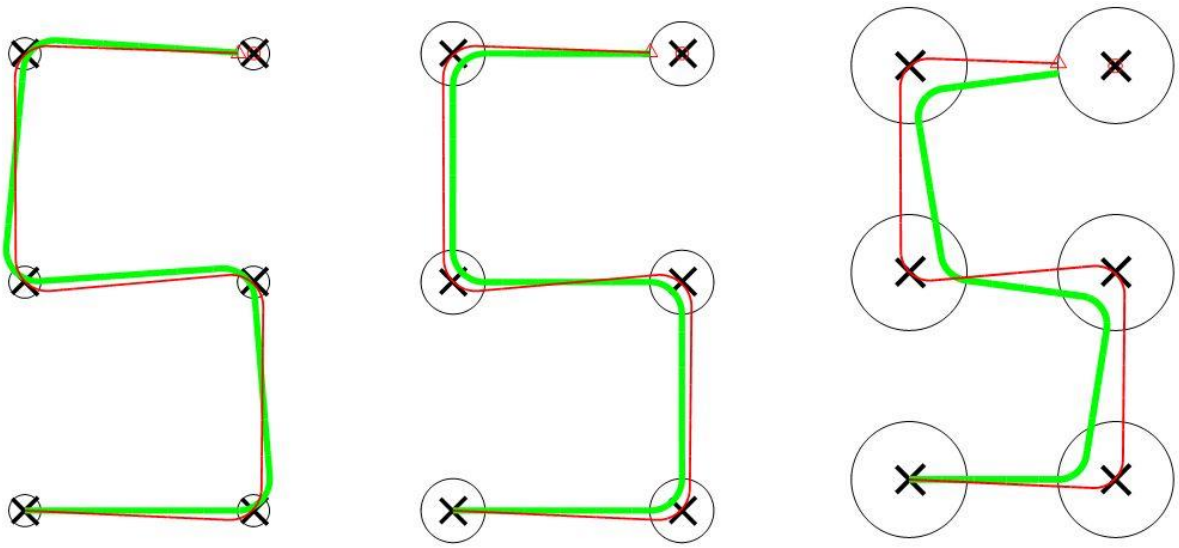


Figure 3.1 WN with and without the MFHL under no disturbances.

In each case, there are no disturbances. With the MFHL, the vehicle passes through the waypoints exactly and follows the minimum fuel paths by design. With the standard *WN*, none of the vehicles reach their waypoints. The vehicle overshoots the desired path when the waypoint radius is less than *R*. The vehicle follows close to the desired path when the waypoint radius is equal to *R*, and the vehicle undershoots the desired path when the waypoint radius is larger than *R*.

3.2 Weight of the Arc Sections During Calculations

The question arises as to the differences between the minimum fuel and minimum distance solutions, the latter determined when $w_a = w_s = 1$. When the distances *A* and *B* are very large compared to the turning radius *R*, the vehicle is flying in a straight line most of the time and the difference between the two solutions will be very small. The distances tend to be at least four times larger than the turning radius and the expected differences between the two solutions largest in these cases. Figure 3.2 shows a typical example. As shown, the orientation angle of the turning circle does not change more than 4^0 in this worsted case. In that the penalty of not following the minimum-fuel path is a marginal reduction in fuel, the minimum-distance path approximation would be acceptable in the small-class UAV problem.

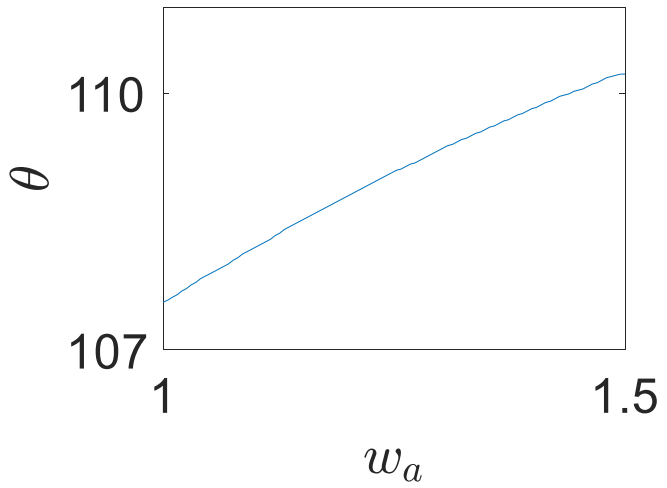


Figure 3.2 Example of a CCW-CCW condition. (A = B = 4, α = -45^0, β = 30^0)

3.3 Results Comparisons with Disturbances

To illustrate the effect that feedback corrections have on path following, we first constructed a simplified feedback controller and then performed the comparisons with the simplified controller. Figure 3.3 shows a vehicle at an instant in time heading in a given direction and a reference line with endpoints that are waypoints in the case of standard *WN* and that are target points when the MFHL is used. The vehicle's turning angle, $\phi = \phi_T + \phi_R$, consists of two parts: the target component ϕ_T toward the target and the reference part ϕ_R toward the reference line. Also shown, the distance between the vehicle and the reference line is denoted by *s*. The target component of the turning angle is determined from geometry and the reference part from the feedback law $\phi_R = |gs + h\dot{s}| \dfrac{\phi_T}{|\phi_T|}$

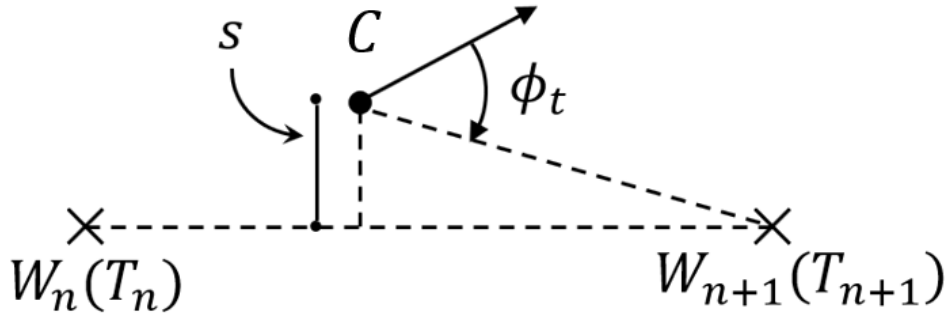


Figure 3.3 Parameters for the controller.

where g and h are control gains and the term $\phi_T / |\phi_T|$ dictates the direction of ϕ_R. Figure 3.4 shows WN with and without the MFHL, both employing the simplified controller and influenced by a randomly generated disturbance. 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.

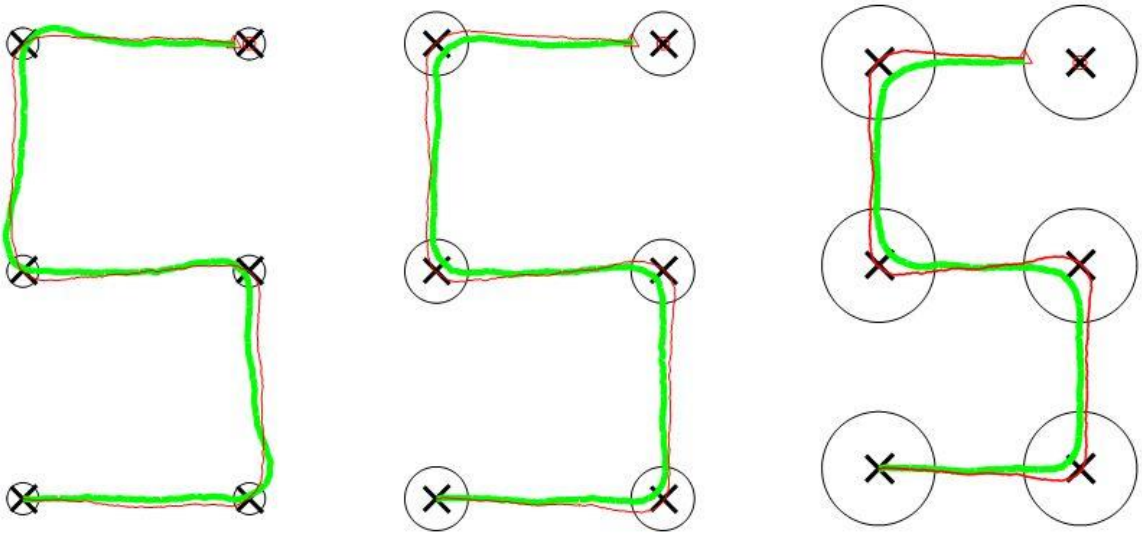


Figure 3.4 WN with and without the MFHL under disturbances (R_{WP} = 0.5R, R, and 2R).

From Figure 3.4 we can observe similar patterns as from Figure 3.1. The red lines which represent navigation with MFHL always hit the waypoints while the green lines, which represent navigation without MFHL tends to deviate from the waypoints and cost more fuels. It is

reasonable to conclude that MFHL proposed here is applicable to various conditions, even with disturbances.

CHAPTER 4

SUMMARY

In the growing class of small-class unmanned aerial vehicles, waypoint navigation suffers from an existing trade-off between minimum turning radius and threshold radius that prevents vehicles from reaching waypoints and closely following desired paths. This paper developed an algorithm, called the minimum-fuel hidden layer (MFHL) that remedies this problem in a way that is invisible to the user. The paper showed how, by establishing a horizon that includes two future waypoints, to improve the performance of WN in terms of fuel and time. For real-time implementation, we reduced the computational effort, essentially eliminated it, by parameterizing the solution to the associated minimum-fuel problem. We also compared current WN and WN with the MFHL, illustrating tracking performance (in the absence of disturbances) and regulation performance (in the presence of disturbances). The comparisons illustrated the improvements in path following (tracking) and overshoot (regulation) obtained by the MFHL.

References

[1] Drubin, Cliff. UAV Market Worth $8.3 B by 2018. Microwave Journal, International ed.; Dedham Vol. 56, Iss. 8, (Aug 2013): 37.

[2] Bakolas, Efstathios and Tsiotras, Panagiotis. Feedback Navigation in an Uncertain Flow Field and Connections with Pursuit Strategies. Journal of Guidance, Control, and Dynamics, Vol. 38, No. 4 (2015), pp. 631-642.

[3] Maravall , Darí; Lope, Javier de and Fuentes , Juan P. Navigation and Self-Semantic Location of Drones in Indoor Environments by Combining the Visual Bug Algorithm and Entropy-Based Vision. Frontiers in Neurorobotics. 2017;11 DOI 10.3389/fnbot.2017.00046.

[4] Ratnoo, Ashwini. Three-Point Guidance for Intercepting Weaving Targets. Journal of Guidance, Control, and Dynamics, Vol. 39, No. 8 (2016), pp. 1881-1886.

APPENDIX

Curve Fitting for the Boundaries in $\alpha - \beta$ Plane

Boundary #1:

Table A.1 Coefficients for Boundary #1.

	a_0	a_1	a_2
b_1	-360853	4010.94	-11.15
b	126331.6	-1398.93	3.87
b_3	-14947.5	166.36	-0.46
b_4	3826.06	-42.05	0.12
b_5	587.91	-6.57	0.02
b_6	-2448.85	27.29	-0.08
Normalized % error*	7.16	7.12	7.08

Boundary #2:

Table A.2 Coefficients for Boundary #2.

	a_0	a_1	a_2
b_1	23.9	20.98	8.4
b_2	-1.63	-11.9	-2.59
b_3	0.16	0.51	0.1
b_4	-0.39	-1.12	-0.34
b_5	0.01	0.05	0.02
b_6	0.02	-0.002	0.002
Normalized % error*	2.67	3.05	9.24

Boundary #3:

This boundary is calculated explicitly:

$$\gamma = \arctan\frac{\cos\delta}{-A-\sin\delta} + \arcsin\frac{1}{\sqrt{\cos^2\delta + (A+\sin\delta)^2}}$$

Curve Fitting for $\boldsymbol{\theta}$

$$\theta = f(A, B, \alpha, \beta)$$

Table A.3 Coefficients for calculating θ.

Term	CW-CCW	CCW-CW	CCW-CCW
1	1.3918	-1.4939	1.6151
A	0.0221	-0.0089	-0.0088
B	-0.0039	0.0025	0.0082
A^2	-0.0007	0.0003	0.0003
B^2	0.0002	-0.0001	-0.0003
$1/A$	0.7966	-0.4697	-0.5436
$1/B$	-0.1017	0.0415	0.1253
β	0.6679	-0.4009	0.4802
$A\beta$	-0.0272	0.0819	0.0096
$B\beta$	0.0088	0.052	-0.0098
$A^2\beta$	0.0009	-0.0032	-0.0003

Table A.3 (Continued).

$B^2\beta$	-0.0004	-0.002	0.0004
β/A	-0.6481	1.3486	0.3613
β/B	0.1841	0.9679	-0.2026
α	0.288	0.1453	0.0964
$A\alpha$	-0.031	-0.0162	-0.0122
$B\alpha$	0.0039	0.0053	0.0081
$A^2\alpha$	0.001	0.0005	0.0004
$B^2\alpha$	-0.0002	-0.0002	-0.0003
α/A	-1.5572	-1.1682	-1.116
α/B	0.1043	0.1066	0.1111
β^2	-0.0897	-0.4233	-0.0141
$A\beta^2$	0.013	0.1502	-0.0037
$B\beta^2$	-0.0025	-0.0495	0.0063
$A^2\ \beta^2$	-0.0004	-0.0077	0.0001
$B^2\beta^2$	0.0001	0.0014	-0.0002
$AB\beta^2$	0	0.0026	0
β^2/A	0.2039	1.4862	-0.2467
β^2/B	0.0165	-0.5278	0.2181
α^2	-0.0834	0.0306	0.0225
$A\alpha^2$	0.0087	-0.004	-0.0032
$B\alpha^2$	-0.0007	0.0021	0.0026

Table A.3 (Continued).

$A^2\alpha^2$	-0.0003	0.0001	0.0001
α^2/A	0.3168	-0.201	-0.2037
α^2/B	-0.0215	0.045	0.0416
$\beta\delta$	0.0236	-0.3978	-0.0069
$A\alpha\beta$	0.0004	0.0337	0.0002
$B\alpha\beta$	-0.0035	0.0262	0.0004
$A^2\alpha\beta$	0	-0.0015	0
$B^2\alpha\beta$	0.0001	-0.001	0
$\alpha\beta/A$	-0.0083	0.5157	-0.0337
$\alpha\beta/B$	-0.0537	0.567	0.0419
$1/\beta$	-0.0001	0	-0.0007
$1/(A\beta)$	0	0	0.0059
$1/(B\beta)$	0.0003	0	-0.0016
$1/\alpha$	-0.0001	0.0003	0.0014
A/α	0	-0.0005	-0.0002
B/α	0	0.0006	0
$1/(A\alpha)$	-0.0015	-0.0136	-0.006
$1/(B\alpha)$	0.0015	0.0096	0.0013
Normalized % error	0.4021	0.5329	0.3983

*The normalized percent error is $100\,(A\mathbf{x}-\mathbf{b})^T(A\mathbf{x}-\mathbf{b})/\mathbf{b}^T\mathbf{b}$ in which the numerator is the squared error in the curve fit and **b** is the vector of coefficients.