Design of Automatic Landing Fuzzy Controller and Path Manager for Fixed Wing Unmanned Aerial Vehicle

***Abstract*—A large number of accidents happened at the stage of landing, especially with presence of severe wind. Control of a flight vehicle for safe landing becomes one of the most critical and demanding tasks. The purpose of fuzzy logic controller (FLC) is to improve the performance, efficiency, and reliability of automatic landing system (ALS) under severe wind condition. The second part of the paper is to present a solution for switching from one waypoint segment to another. It combines straight-line path and orbits to provide a smooth transition path manager which is particularly useful for performing autonomous operation for Unmanned aerial vehicles (UAV). MATLAB and SIMULINK is implemented for simulation demonstration.**

I. Introduction to Fuzzy Logic Controller

ATMOSPHERIC disturbance has negative effect on the airplane regardless how big it is. Wind shear is one of the most dangerous wind disturbances. It has relatively significant difference in wind speed or direction over a short distance. It exists in both vertical and horizontal directions. It occurs at all altitudes and increases with increasing height. This is the reason the wind shear has caused a number of aircraft crashes over past decades[1]. According to a survey of the National Transportation Safety Board (NTSB), 22.6% of airplane crashes were weather related between 1989 and 1999 [1]. From this point, instead of using a traditional ALS, a fuzzy logic system can be applied to flight control to improve the performance of the aircraft in terms of adapting different environmental conditions. The objective of first part of the project is to design and investigate a robust controller, Fuzzy Logic Controller, for fixed wing UAV to achieve better automatic landing control with presence of steady wind and wind shear. Simulation results will be discussed.

A. Basic idea of Fuzzy Logic Controller

FLC computes based on ”degrees of truth” rather than typical Boolean logic, 1 or 0. Application of knowledge-based approach can result in efficiency, time and cost savings[3]. FLC is to imitate the control actions of a human operator. It is generally represented as a collection of if-then rules. For instance, If error1 is negative big and error2 is zero, then control action is negative big. The first part of the rule is called antecedent which specifies the conditions under which the rule holds. The second part of the rule is called consequent which prescribes the corresponding control action. It uses linguistic terms to make vague description that reflect the operator's knowledge[2].

B. Fuzzy Sets

A fuzzy set is an ordered set. Each value of a variable is related to its grade of membership in the set. The membership function represents the grades of membership. The fuzzy variables are converted from control input and output[2]. The position and shape of the membership function is highly dependent to specific application. In this project, the triangular and trapezoidal shaped membership function is used to represent fuzzy sets. Fuzzy set operations are used to link linguistic terms.

C. Fuzzy Set Operations

Fuzzy set operations contain logical connectives. The most commonly used operators are AND, OR, or NOT, which represents conjunction, disjunction and compliment respectively. Minimum operator is used for conjunction operators while maximum operator is used for the disjunction.

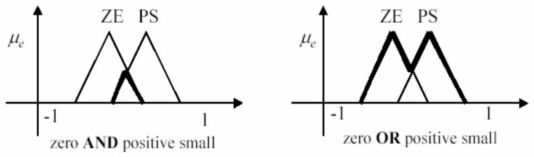

Fig. 1. Conjunction and disjunction of 2 fuzzy sets

D. Fuzzy Logic Control

1) Fuzzification: In this stage, it computes the membership degrees of the antecedent variables.

2) Degree of fulfillment: The degree of fulfillment for the antecedent of each rule is computed[2].

3) Implication: The degree of fulfillment is used to modify the consequent of the corresponding rule. This operation is the if-then implication which defines as a conjunction[2].

4) Aggregation: The scale consequents of all rules are combined into one single fuzzy set. The aggregation operator depends on the implication function[2].

5) Defuzzification: The resulting fuzzy set is defuzzified to yield a crisp value. There are mainly two types of diffuzification method, Takagi-Sugeno and Mamdani.

II. Fuzzy Logic Controller Design

At the aircraft landing phase, the aircraft is descending from cruising phase to a height which is about 100m above ground. And then the aircraft is entering the landing zone to

approach to runway as shown in figure 2. As the airplane descends along the path, the pitch, attitude, and speed must be controlled properly to achieve desired performance. In this study, the autonomous navigation starts from 100m and ends when UAV touches the ground.

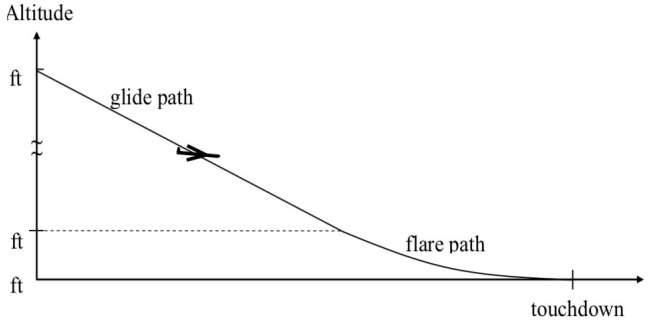

Fig. 2. Landing path [4]

FLC designed contains longitudinal and lateral component to cope with possible steady wind and wind shear. It is designed to give adequate performance and robustness in both east and down direction. The goal is to follow the landing path and minimize the east position error, such that the aircraft is able to land at the centre of runway. The labels used in membership functions are defined by $N \Rightarrow Negative, P \Rightarrow Positive, E \Rightarrow Extreme, VB \Rightarrow VeryBig, B \Rightarrow Big, M \Rightarrow Medium, S \Rightarrow Small, VS \Rightarrow VerySmall, ZE \Rightarrow Zero.$

A. Longitudinal Fuzzy Controller

The fuzzy logic based approach is implemented to achieve altitude hold, speed control and landing control. It plays a positive role in terms of autopilot heuristics to meet robustness requirements.

The longitudinal control loops include the throttle and the altitude controller. To control the altitude, it requires to translate the altitude error and speed error into a pitch angle command with integrated in to the inner pitch controller. The throttle control takes airspeed feedback and height error to regulate throttle. The altitude and velocity errors are generated by difference of reference commands and the corresponding measurements. A set of membership functions is created for pitch angle and throttle commands as shown in figure 3. The rule base of pitch angle and throttle control is shown in figure 4 and 5. The surface plot of the generated rules is illustrated in figure 6.

B. Lateral Fuzzy Controller

The successive loop closure is used for traditional autopilot controller to provide lateral track and turn performance. Lateral track error is fed back into the roll attitude control loop through FLC. The lateral fuzzy controller is designed to take roll angle error and lateral offset as inputs. The roll angle command is the consequent. Both inputs are using same

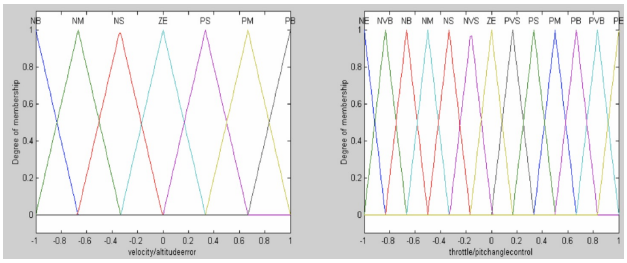

Fig. 3. Membership functions for pitch angle and throttle

h_e\v_e	NB	NM	NS	ZE	PS	PM	PB
NB	ZE	NVS	NS	NM	NB	NVB	NE
NM	PVS	ZE	NVS	NS	NM	NB	NVB
NS	PS	PSV	ZE	NVS	NS	NM	NB
ZE	PM	PS	PVS	ZE	NVS	NS	NM
PS	PB	PM	PS	PVS	ZE	NVS	NS
PM	PVB	PB	PM	PS	PVS	ZE	NVS
PB	PE	PVB	PB	PM	PS	PVS	ZE

Fig. 4. Rule base of pitch angle

triangular shaped lateral membership functions as shown in figure 7. The output, roll angle command, is generated by membership function showing in figure 8. The surface plot is illustrated in figure 10. A second type of lateral controller with different membership functions and rule base is also implemented for comparison. Its surface plot is demonstrated in figure 11.

III. Fuzzy Logic Controller Simulation

A wind shear model in SIMULINK is calculated by:

$$u_W = W_{20} \frac{ln(\frac{h}{z_0})}{ln(\frac{20}{z_0})}$$

where $z_0 = 0.15$ for category C flight phases. W_{20} is measured wind speed at 20ft. The wind direction applied is North East 33 degrees.

Various wind speeds between 0 to 10 m/s and different directions are tested in the simulation. Due to space restriction, only the wind speed of 10m/s and 6m/s are presented here. The simulation results under wind speed of 10m/s at 20ft are illustrated in figure 12 and figure 13 for FLC and traditional controller. In final results, the throttle control and rudder deflection fuzzy control are turn off for both controllers. After implementing the throttle fuzzy controller mentioned above, it generated very unstable performance. The possible reasons are the throttle membership function provided in paper [3] has a range from -1 to 1 plus the rule base depends on the membership functions. This is very different our model where the throttle is between 0 and 1. In addition, the aircraft used in this paper is a twin jet fighter which is a very different aircraft comparing to a small UAV.

h_e\v_e	NB	NM	NS	ZE	PS	PM	PB
NB	ZE	NVS	NS	NM	NB	NVB	NE
NM	PVS	ZE	NVS	NS	NM	NB	NVB
NS	PS	PSV	ZE	NVS	NS	NM	NB
ZE	PM	PS	PVS	ZE	NVS	NS	NM
PS	PB	PM	PS	PVS	ZE	NVS	NS
PM	PVB	PB	PM	PS	PVS	ZE	NVS
PB	PE	PVB	PB	PM	PS	PVS	ZE

Fig. 5. Rule base of throttle

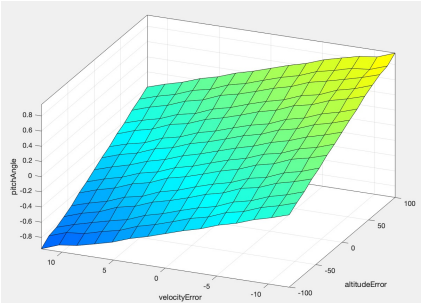

Fig. 6. Surface plot of pitch angle

As shown in figure 12, the aircraft overshot a little bit at first 5s for less than 20m due to severe wind speed of 16 m/s at 100m altitude. However, after 5s it maintained a smooth landing till the end with landing rate of 3 m/s. In contrast, the aircraft overshot for another 90m by using transitional controller which is more than 4 times of that of fuzzy controller. Not to mention the aircraft was fluctuating after the overshooting. In addition, the fuzzy controller significantly facilitated the elevator performance with value of less than -6 degrees. The angle of attack was stabilized at 4 degrees. The pitch angle was less than -6 degrees. Yet, as for transitional controller, elevator deflection, angle of attack, and pitch angle varied between -20 to 20 degrees. It has high possibility to lead to a dangerous landing for the aircraft under a severe wind condition.

The aircraft performed a very stabilized landing on east direction. The lateral fuzzy controller worked extremely well where the lateral offset is approximately 0. The small changes on Pe direction can be ignored for fuzzy controller whereas the traditional controller provided a much larger lateral offset with pe up to 2 meters.

As mentioned in lateral fuzzy controller design section, two different types of lateral fuzzy controller were simulated. Figure 14 is the performance of fuzzy type 2 controller with wind speed of 6m/s. It provides very similar results compared to type 1 FLC in terms of altitude, angle of attack, lateral offset, and pitch angle. Yet, it gives very small oscillations on roll angle, course angle, aileron deflection, and rudder. A possible reason is tuning of the gains is not perfect. The rudder and throttle fuzzy controller were excluded in the final simulation. Applied to a small UAV which is a very different aircraft than a jet fighter in paper [3] could also be one of the reasons.

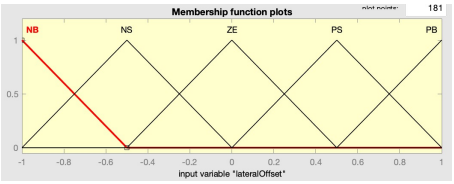

Fig. 7. Membership function of roll error and lateral offset

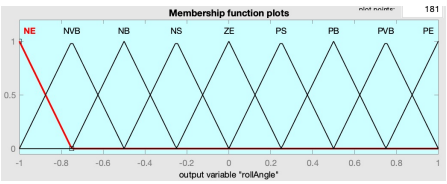

Fig. 8. Membership function of roll angle command

IV. Introduction to Path Manager

The small UAVs are used for a wide range of tasks such as rescue mission, terrain mapping, even delivery service. In this second part of the project, the objectives are to combine straight-line paths and orbits and control the aircraft to follow a desired path with smooth transition. It allows the aircraft to switch from one waypoint segment to another which is useful for autonomous operation of UAVs [5]. The straight-line and orbit guidance strategies are used to synthesize Dubins paths. Some assumptions made for this path manager are constant-altitude, constant-velocity vehicles with turning constraints[5].

V. Path Manager Implementation

A. Smooth transition between waypoints

A waypoint path is defined as an ordered sequence of waypoints:

$$W = W_1, W_2, W_3, ..., W_N$$

where $w_i = (w_{ni}, w_{ei}, w_{di})^T$.
When switching from one straight line to another, instead of transiting with a very sharp angle, a fillet can be used to smooth between the straight-line segments [5]. The geometry associated with a smooth transition is shown in figure 15. Two half planes $H_1 and H_2$ are defined at tangential point of straight-line path and orbit path. It is the point where aircraft should switch from following straight line to orbit. A unit vector q_i is defined to be inertial direction of waypoint path. It is used to identify if the aircraft is at the tangential point where the state machine has to generate a new command to switch current state. To be more specific, when the state machine is in $state = 1$, the aircraft is commanded to follow the straight-line path. If the aircraft has entered H_1, the aircraft is switched to follow orbit. If the aircraft has transitioned into H_2, then the state machine is updated to state=2 [5]. The length of the waypoint path W is defined as:

$$|W| = \sum_{i=2}^{n} ||w_i - w_{i-1}||$$

Delta x/delta phi	NB	NS	ZE	PS	PB
NB	NE	NVB	NB	NS	ZE
NS	NVB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PVB
PB	ZE	PS	PB	PVB	PE

Fig. 9. Rule base for roll angle and rudder

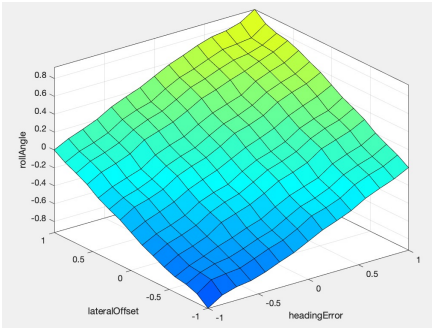

Fig. 10. surface plot of roll angle command type 1

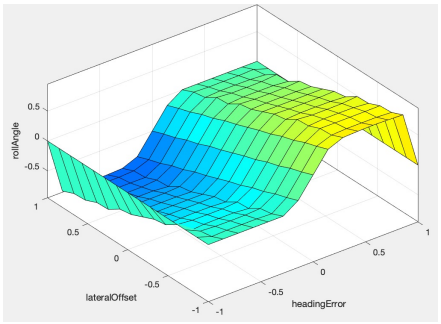

Fig. 11. surface plot of roll angle command type 2

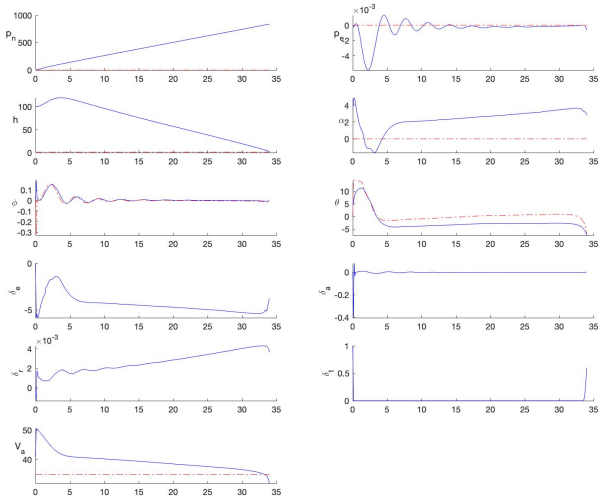

Fig. 12. Fuzzy controller type 1 performance under $w_{20} = 10m$

B. Dubins paths

In this section, we focus on Dubins paths. Instead of following a waypoint path, the goal is to transition from one configuration (position and course) to another[5]. Velocity is assigned to be constant. The time-optimal path between two different configurations contains a circular arc, followed by a straight line, and concluding with another circular arc to the final configuration [5]. These turn-straight-turn paths are one of the dubin paths. They are defined for optimal transitions between diffrent configurations. The two arcs in the dubin path is assumed to have same radius R. The start configuration of dubin path is (P_s, χ_s). The end configuration is denoted as (P_e, χ_e). Four different dubin path configurations are considered in this part of project (figure 16).

To determine the Dubins path, it is necessary to compute the path length for the four configurations. The explicit formulas are derived for each case to calculate path length. The centers of the right and left turning circles are given by following formulas where the position p, the course χ, and the radius R are known.

$$c_r = p + R(cos(\chi + \frac{\pi}{2}), sin(\chi + \frac{\pi}{2}), 0)^T$$

$$c_l = p + R(cos(\chi - \frac{\pi}{2}), sin(\chi - \frac{\pi}{2}), 0)^T$$

Case I: R-S-R
$L_1 = ||c_{rs} - c_{re}|| + R < 2\pi + < \theta - \frac{\pi}{2} > - < \chi_s - \frac{\pi}{2} >>$
$+R < 2\pi + < \chi_e - \frac{\pi}{2} > - < \theta - \frac{\pi}{2} >>$

Case II: R-S-L
$L_2 = \sqrt{l^2 - 4R^2} + R < 2\pi + < \theta - \theta_2 > - < \chi_s - \frac{\pi}{2} >>$
$+R < 2\pi + < \theta_2 + \pi > - < \chi_e + \frac{\pi}{2} >>$
$\theta_2 = \theta - \frac{\pi}{2} + sin^{-1}(\frac{2R}{l})$

Case III: L-S-R
$L_3 = \sqrt{l^2 - 4R^2} + R < 2\pi - < \theta + \theta_2 > + < \chi_s + \frac{\pi}{2} >>$
$+R < 2\pi - < \theta + \theta_2 - \pi > + < \chi_e + \frac{\pi}{2} >>$

$\theta_2 = cos^{-1}(\frac{2R}{l})$

Case IV: L-S-L
$L_4 = ||c_{ls} - c_{le}|| + R < 2\pi - < \theta + \frac{\pi}{2} > + < \chi_s + \frac{\pi}{2} >>$
$+R < 2\pi - < \chi_e + \frac{\pi}{2} > + < \theta + \frac{\pi}{2} >>$

C. Algorithm for Tracking Dubins Paths

Take case III as an example, the guidance algorithm for tracking a Dubins path is illustrated graphically in figure 17. The algorithm is initialized in a left-handed orbit about c_{ls}. The aircraft continuously follows the orbit path till reach the half plane,H_1. After entering H1, the aircraft starts to follow a straight-line guidance until it enters H_2. After passing through H_2, it switches to a right-handed orbit around c_re, followed by entering H_3. It then switches back to straight-line path where it defines as the end of the dubin path. The dubin path parameters are denoted as following.

- c_s: Start circle location
- c_e: End circle location
- λ_s: Start circle direction
- λ_e: End circle direction
- z_1, q_1: The parameters of the half plane H_1

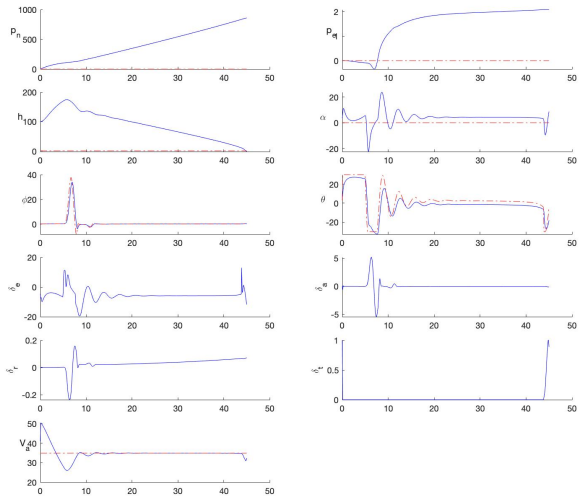

Fig. 13. Traditional controller performance under $w_{20} = 10m$

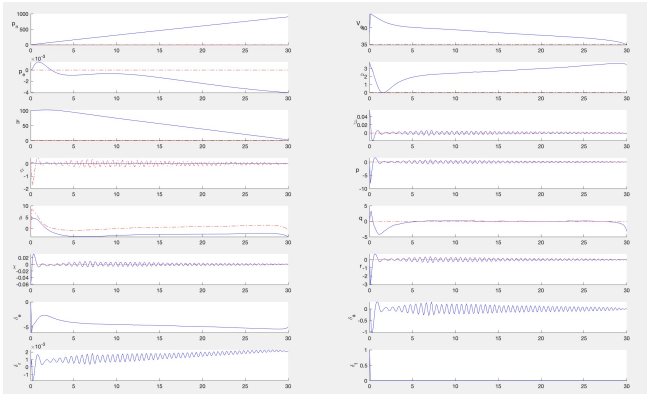

Fig. 14. Fuzzy controller type 2 performance under $w_{20} = 6m$

- z_2, q_2: The parameters of the half plane H_2, $q2 = q1$
- z_3, q_3: The parameters of the half plane H_3

The minimum turning radius is calculated by

$$R_{min} = \frac{Va_0^2}{gravity \times tan(\phi_{max})}$$

R used in the simulations has to be bigger than R_{min}.

The algorithm 7 shown in figure 18 is implemented to find dubin path parameter. Lastly, a guidance algorithm that is designed to switch between five different states to follow Dubins paths between the configurations is given in 19.

VI. Path Manager Simulation

A entire simulation is tested on MATLAB and SIMULINK as shown in figure 20. The aircraft was be able to follow the dubin path configurations successfully with different radius. The radius in figure 20 is 72m. 200m and 300m radius dubin path are tested with decent performance.

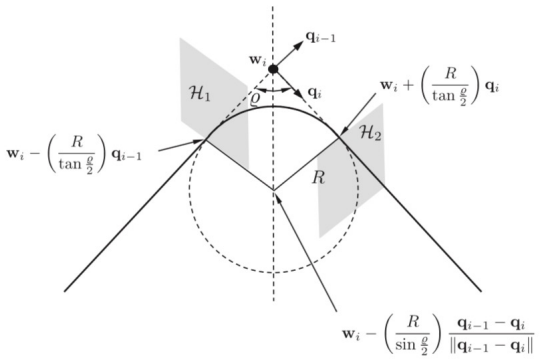

Fig. 15. The geometry of inserting fillet between waypoint segments [5]

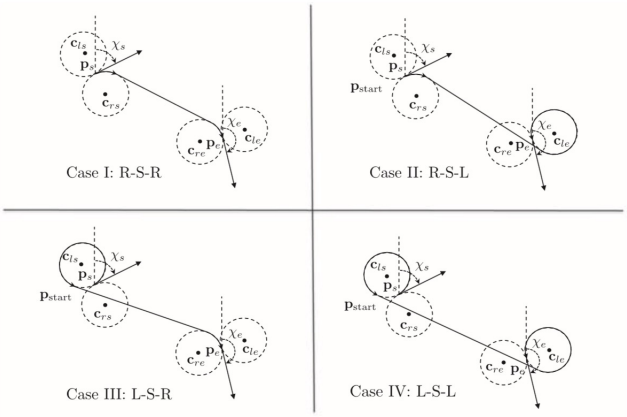

Fig. 16. Dubin path configurations [5]

VII. Conclusion

As for FLC, the aircraft was able conduct a safe and steady landing with presence of 10m/s wind shear in direction of North East 33 degress. FLC provides very promising results in both longitudinal and lateral directions compared to transitional autopilot controller we designed in class. Almost 0 lateral offset was obtained by applying lateral FLC under severe wind. Both types of lateral FLC generate similar results, except the type 2 FLC produce small oscillations on roll angle, course angle, aileron deflection, and rudder. The landing in longitudinal direction was very smooth and steady. However, there are some issues with the rudder and throttle fuzzy controller. The airspeed wasn't tracking the command properly. There is about 5m/s error in airspeed.

For the path manager implementation, the aircraft was tracking the dubin path successfully. Using minimum radius calculated gives relatively large roll angle. Considering the overall aircraft aspects, a radius of 200m or more is a more reasonable choice to obtain better performance. Another issue is since the autopilot controller is not tuned perfectly, the aircraft performed a unstable and fluctuated maneuver while tracking the dubin path.

Therefore, future work is to find a way to implement fuzzy throttle and rudder controller for auto landing system. Second

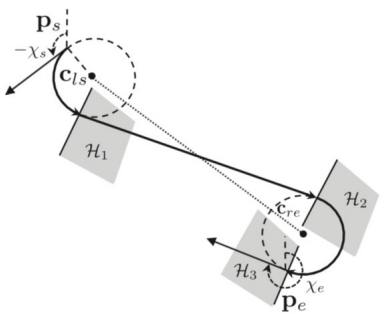

Fig. 17. Switching half planes for Dubin path [5]

$\mathbf{c}_{rs} \leftarrow \mathbf{p}_s + R\mathcal{R}_z\left(\frac{\pi}{2}\right)(\cos\chi_s, \sin\chi_s, 0)^\top$
$\mathbf{c}_{ls} \leftarrow \mathbf{p}_s + R\mathcal{R}_z\left(\frac{-\pi}{2}\right)(\cos\chi_s, \sin\chi_s, 0)^\top$
$\mathbf{c}_{re} \leftarrow \mathbf{p}_e + R\mathcal{R}_z\left(\frac{\pi}{2}\right)(\cos\chi_e, \sin\chi_e, 0)^\top$
$\mathbf{c}_{le} \leftarrow \mathbf{p}_e + R\mathcal{R}_z\left(-\frac{\pi}{2}\right)(\cos\chi_e, \sin\chi_e, 0)^\top$
Compute L_1, L_2, L_3, and L_4 using equations (11.9) through (11.12).
$L \leftarrow \min\{L_1, L_2, L_3, L_4\}$
if $\arg\min\{L_1, L_2, L_3, L_4\} = 1$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{rs}, \quad \lambda_s \leftarrow +1, \quad \mathbf{c}_e \leftarrow \mathbf{c}_{re}, \quad \lambda_e \leftarrow +1$
  $\mathbf{q}_1 \leftarrow \frac{\mathbf{c}_e - \mathbf{c}_s}{\|\mathbf{c}_e - \mathbf{c}_s\|}$
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z\left(-\frac{\pi}{2}\right)\mathbf{q}_1$
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z\left(-\frac{\pi}{2}\right)\mathbf{q}_1$
else if $\arg\min\{L_1, L_2, L_3, L_4\} = 2$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{rs}, \quad \lambda_s \leftarrow +1, \quad \mathbf{c}_e \leftarrow \mathbf{c}_{le}, \quad \lambda_e \leftarrow -1$
  $\ell \leftarrow \|\mathbf{c}_e - \mathbf{c}_s\|$
  $\vartheta \leftarrow \text{angle}(\mathbf{c}_e - \mathbf{c}_s)$
  $\vartheta_2 \leftarrow \vartheta - \frac{\pi}{2} + \sin^{-1}\frac{2R}{\ell}$
  $\mathbf{q}_1 \leftarrow \mathcal{R}_z\left(\vartheta_2 + \frac{\pi}{2}\right)\mathbf{e}_1$
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta_2)\mathbf{e}_1$
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(\vartheta_2 + \pi)\mathbf{e}_1$
else if $\arg\min\{L_1, L_2, L_3, L_4\} = 3$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{ls}, \quad \lambda_s \leftarrow -1, \quad \mathbf{c}_e \leftarrow \mathbf{c}_{re}, \quad \lambda_e \leftarrow +1$
  $\ell \leftarrow \|\mathbf{c}_e - \mathbf{c}_s\|,$
  $\vartheta \leftarrow \text{angle}(\mathbf{c}_e - \mathbf{c}_s),$
  $\vartheta_2 \leftarrow \cos^{-1}\frac{2R}{\ell}$
  $\mathbf{q}_1 \leftarrow \mathcal{R}_z\left(\vartheta + \vartheta_2 - \frac{\pi}{2}\right)\mathbf{e}_1,$
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta + \vartheta_2)\mathbf{e}_1,$
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(\vartheta + \vartheta_2 - \pi)\mathbf{e}_1$
else if $\arg\min\{L_1, L_2, L_3, L_4\} = 4$ **then**
  $\mathbf{c}_s \leftarrow \mathbf{c}_{ls}, \quad \lambda_s \leftarrow -1, \quad \mathbf{c}_e \leftarrow \mathbf{c}_{le}, \quad \lambda_e \leftarrow -1$
  $\mathbf{q}_1 \leftarrow \frac{\mathbf{c}_e - \mathbf{c}_s}{\|\mathbf{c}_e - \mathbf{c}_s\|},$
  $\mathbf{z}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z\left(\frac{\pi}{2}\right)\mathbf{q}_1,$
  $\mathbf{z}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z\left(\frac{\pi}{2}\right)\mathbf{q}_2$
end if
$\mathbf{z}_3 \leftarrow \mathbf{p}_e$
$\mathbf{q}_3 \leftarrow \mathcal{R}_z(\chi_e)\mathbf{e}_1$

Fig. 18. Algorithm 7 find dubin path parameter [5]

is to extend the FLC to a full-scaled autopilot control. Third is to integrate full-scaled FLC with path manager to investigate the path fallowing with presence of wind at different altitudes.

REFERENCES

[1] C. Lee and J. Juang, "Aircraft landing control in wind shear condition," *2011 International Conference on Machine Learning and Cybernetics*, Guilin, 2011, pp. 1180-1185. doi: 10.1109/ICMLC.2011.6016885

[2] R. K. David Solomon and T. Goutham, "Design of Fuzzy Logic Controller for Auto Landing Applications", *International Journal of Scientific and Research Publications*, vol. 3, no. 5, p. 9, 2013. [Accessed 13 April 2019].

[3] H. Farah, "The Fuzzy Logic Control of Aircraft", National Library of Canada, Ottawa, 1999.

[4] C. Lee and J. Juang, "Aircraft landing control in wind shear condition," *2011 International Conference on Machine Learning and Cybernetics,* Guilin, 2011, pp. 1180-1185. doi: 10.1109/ICMLC.2011.6016885

if state $= 1$ **then**
  flag $\leftarrow 2, \mathbf{c} \leftarrow \mathbf{c}_s, \rho \leftarrow R, \lambda \leftarrow \lambda_s$
  if $\mathbf{p} \in \mathcal{H}(\mathbf{z}_1, -\mathbf{q}_1)$ **then**
    state $\leftarrow 2$
  end if
else if state $= 2$ **then**
  if $\mathbf{p} \in \mathcal{H}(\mathbf{z}_1, \mathbf{q}_1)$ **then**
    state $\leftarrow 3$
  end if
else if state $= 3$ **then**
  flag $\leftarrow 1, \mathbf{r} \leftarrow \mathbf{z}_1, \mathbf{q} \leftarrow \mathbf{q}_1$
  if $\mathbf{p} \in \mathcal{H}(\mathbf{z}_2, \mathbf{q}_1)$ **then**
    state $\leftarrow 4$
  end if
else if state $= 4$ **then**
  flag $\leftarrow 2, \mathbf{c} \leftarrow \mathbf{c}_e, \rho \leftarrow R, \lambda \leftarrow \lambda_e$
  if $\mathbf{p} \in \mathcal{H}(\mathbf{z}_3, -\mathbf{q}_3)$ **then**
    state $\leftarrow 5$
  end if
else if state $= 5$ **then**
  if $\mathbf{p} \in \mathcal{H}(\mathbf{z}_3, \mathbf{q}_3)$ **then**
    state $\leftarrow 1$
    $i \leftarrow (i+1)$ until $i = N$.
    $(L, \mathbf{c}_s, \lambda_s, \mathbf{c}_e, \lambda_e, \mathbf{z}_1, \mathbf{q}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{q}_3) \leftarrow$ findDubinsParameters$(\mathbf{w}_{i-1}, \chi_{i-1}, \mathbf{w}_i, \chi_i, R)$
  end if
end if
return flag, $\mathbf{r}, \mathbf{q}, \mathbf{c}, \rho, \lambda$.

Fig. 19. Algorithm 8 follow waypoints with dubins [5]

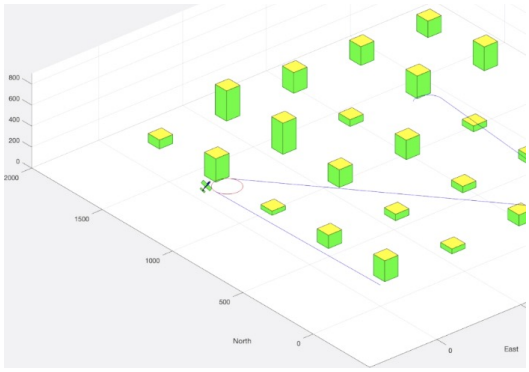

Fig. 20. Matlab simulation for path manager

[5] Beard, W. McLain, W.. *Small Unmanned Aircraft: Theory and Practice*. Princeton: Princeton University Press, 2012. Project MUSE