A HEURISTIC CASCADING FUZZY LOGIC APPROACH TO REACTIVE NAVIGATION FOR UAV

YEW-CHUNG CHAK[1] AND RENUGANTH VARATHARAJOO[2]

Department of Aerospace Engineering, Faculty of Engineering, Universiti Putra Malaysia,
43400 UPM Serdang, Selangor, Malaysia.

[1] *samchak11@gmail.com,* [2] *renu99@gmx.de*

ABSTRACT: The capability of navigating Unmanned Aerial Vehicles (UAVs) safely in unknown terrain offers huge potential for wider applications in non-segregated airspace. Flying in non-segregated airspace presents a risk of collision with static obstacles (e.g., towers, power lines) and moving obstacles (e.g., aircraft, balloons). In this work, we propose a heuristic cascading fuzzy logic control strategy to solve for the Conflict Detection and Resolution (CD&R) problem, in which the control strategy is comprised of two cascading modules. The first one is Obstacle Avoidance control and the latter is Path Tracking control. Simulation results show that the proposed architecture effectively resolves the conflicts and achieve rapid movement towards the target waypoint.

ABSTRAK: Keupayaan mengemudi Kenderaan Udara Tanpa Pemandu (UAV) dengan selamat di kawasan yang tidak diketahui menawarkan potensi yang besar untuk aplikasi yang lebih luas dalam ruang udara yang tidak terasing. Terbang di ruang udara yang tidak terasing menimbulkan risiko perlanggaran dengan halangan statik (contohnya, menara, talian kuasa) dan halangan bergerak (contohnya, pesawat udara, belon). Dalam kajian ini, kami mencadangkan satu strategi heuristik kawalan logik kabur yang melata untuk menyelesaikan masalah Pengesanan Konflik dan Penyelesaian (CD&R), di mana strategi kawalan yang terdiri daripada dua modul melata. Hasil simulasi menunjukkan bahawa seni bina yang dicadangkan berjaya menyelesaikan konflik dan mencapai penerbangan pesat ke arah titik laluan sasaran.

KEYWORDS: *fuzzy logic; motion planning; obstacle avoidance; path tracking; reactive navigation; UAV*

1. INTRODUCTION

UAV has attracted growing attention in both military and civilian applications that are considered dull, dirty or dangerous, such as intelligence, surveillance, reconnaissance, search and rescue, power line inspection, fire detection, border patrol, coastline monitoring, weather forecasts, and volcanic activity tracking [1]. In order to fulfill various mission objectives, it is important to integrate intelligent control that would cause UAVs to perceive their environment and quickly to react in a reasoned manner to an unplanned situation more like a human pilot. To make scenarios sensibly pragmatic for some sort of intelligent-like behaviors for UAVs, the motion planning autonomy must be able to handle varying numbers of critical objectives under different constraints such as minimizing the path length, keeping the path as straight as possible, flying over some areas of interest, avoiding obstacles or no fly zones, and approaching the target location from a commanded direction [2]. When the UAV enters non-segregated or controlled airspace, it must have the sense and avoid capability (also referred to as collision detection and resolution systems, CD&R) to maintain safety and fluency of the air traffic.

Various approaches have been applied to the CD&R problem. Krozel [3] and Kuchar [4] presented surveys of collision detection and resolution methods. Zeghal [5] conducted a survey of force field collision detection and resolution methods and finally Albaker [6] introduced the survey of CD&R methods for UAVs.

The human pilot approach to navigation is to make maps and assign flight waypoints, and at first glance it seems obvious that UAVs should operate the same way. However many tasks can be achieved without any map at all, using an approach referred to as reactive navigation, by reacting directly to its environment. Duong et al. [7] presented a technique in which a force field generated by an intruding airplane produces a conflict resolution action and a force from the flight plan generates an attracting action. However, the force field method is not without any shortcomings because of the aircraft's tendency to get caught in a local minima due to a spot with zero force where repelling and attracting forces cancel each other out.

The more familiar human-style map-based navigation is used by more sophisticated UAVs. This approach supports more complex tasks but is itself more complex. In the scenario where multiple UAVs are assigned to traverse known several target locations in the presence of dynamic threats, Beard, McLain, Goodrich, and Anderson [8] used a modified Voronoi diagram to generate possible paths to the targets. In Jung and Ariyur's work [9], the Dijkstra Method algorithm is used to calculate the shortest path from a starting point to the target destination. In 2003, Dogan [10] introduced a new probabilistic roadmap methods (PRM) approach to the problem of mission planning for UAVs flying through an area of multiple sources of threat. In a paper [11], Kothari, Postlethwaite and Gu proposed path planning algorithms using Rapidly-exploring Random Tree (RRT) to generate paths for multiple UAVs in the dense obstacle fields.

2. NAVIGATIONAL STRATEGY

The study focusses on non-cooperating UAVs and intruding obstacles. The navigational strategy consists of a mechanism and a procedure, which they are usually very strongly interdependent. The mechanism we used to navigate the UAV to avoid static and moving obstacles is a fuzzy logic controller. In this paper, we design a set of conditional statements derived from a pilot's knowledge to construct the control surface for the Sense and Avoid reactive navigation. The procedure specifies how the Sense capability enables the UAV to detect other aircraft in the airspace, and to determine whether it poses a potential conflict. The Avoid capability enables the UAV to take action to circumvent an impending collision in situations where a loss of separation has occurred. This endeavor will provide the fuzzy controller with data to intelligently determine any course to autonomously maneuver the UAV to eliminate the potential conflict.

2.1 Mechanism – Fuzzy Logic Controller

Many real dynamical systems and real life processes are too complicated to model mathematically. Instead, if we try to construct a perfect model for Sense and Avoid system, it may be too difficult for a successful application of a control theory. Therefore, an intelligent human operator is sometimes more efficient than an automatic controller. These heuristic control strategies, which provide a viable method to handle qualitative information, may be modeled by a fuzzy logic controller as depicted by a block diagram such as that shown in Fig. 1. This conventional architecture is by far the most common one because it generates low-level direct control signals. The knowledge-base module of a fuzzy logic controller contains theoretical knowledge and practical experience about all the input and output fuzzy partitions [12].

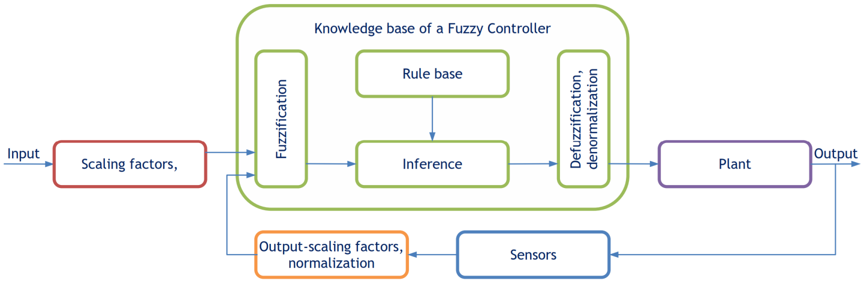

Knowledge base of a Fuzzy Controller
Input
Scaling factors,
Fuzzification
Rule base
Inference
Defuzzification, denormalization
Plant
Output
Output-scaling factors, normalization
Sensors

Fig. 9: A simple fuzzy logic control system block diagram.

2.2 Procedure – Sense and Avoid

Sense and Avoid is the key factor for the integration of unmanned aerial vehicles into the common airspace. In general it is a safety system for Unmanned Aerial Vehicles (UAV). Basically, Sense and Avoid can be separated into two tasks as follows [13]:

i) Sense is the ability to detect and track obstacles, and
ii) Avoid is the ability to steer UAV around obstacles.

We regard the mission planning problem of UAV trajectory tracking with only minimal amount of information about the airspace environment, which are the positions of the UAV and waypoints, are available from a communication system. We assume that the vehicle dynamics are considered as point masses in Cartesian coordinates on $\mathbb{R}^2$ for the ease of analysis. The deterministic position and velocity vectors of the intruding obstacles are presumed to be detectable by a UAV's sensors and estimator. To navigate reactively in the airspace, the UAV must be able to reliably detect the intruding obstacle within a certain radius with a field of regard of 270° as shown in Fig. 2. It is also assumed that the intruding obstacles cannot climb or descent, maneuver laterally, or change speed.

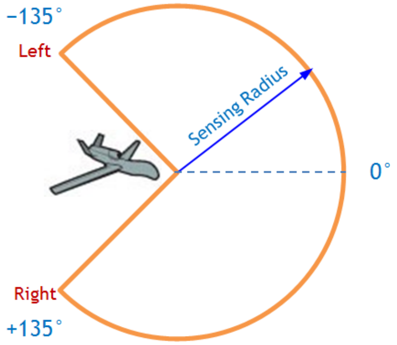


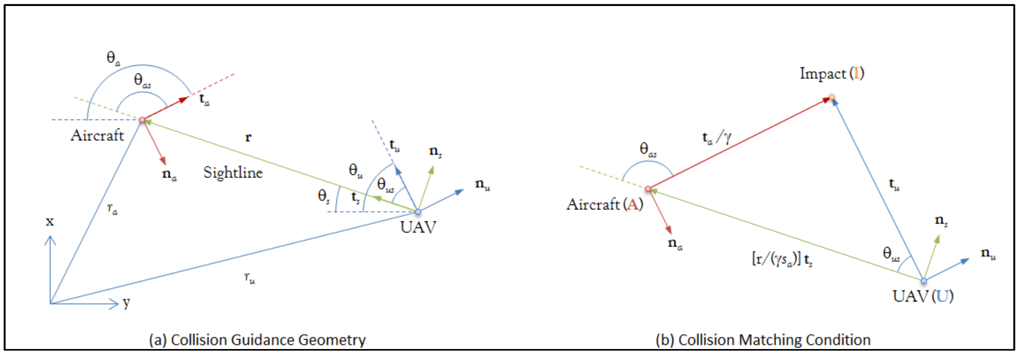
Fig. 10: The sensing range of the UAV's obstacle sensors.

2.3 Collision Kinematics

Figure 3(a) illustrates the guidance geometry of a UAV and an aircraft where the trajectory of the UAV might run into the aircraft's path with the likelihood of conflict.

The UAV detects the aircraft and initiates a sightline to track the aircraft. With a motion estimation algorithm, the aircraft velocity can be estimated and allow us to establish the motion geometry. Let's describe the tangent and normal axis sets as follows:

i) $(\mathbf{t}_a, \mathbf{n}_a)$ defines the aircraft,
ii) $(\mathbf{t}_s, \mathbf{n}_s)$ defines the sightline, and
iii) $(\mathbf{t}_u, \mathbf{n}_u)$ defines the UAV.



(a) Collision Guidance Geometry

(b) Collision Matching Condition

Fig. 11: Collision geometry (adapted from [14]).

As seen in Figure 3(a), the sightline range vector is given by

$$\mathbf{r} = \mathbf{r}_a - \mathbf{r}_u \tag{1}$$

If the velocities of both vehicles are constant, then Equation (1) can be expanded to characterize the components of the aircraft velocity relative to the UAV.

$$\dot{r}\mathbf{t}_s + r\dot{\theta}_s\mathbf{n}_s = v_a\mathbf{t}_a - v_u\mathbf{t}_u \tag{2}$$

Shin, Tsourdos and White [14] suggests that the collision matching conditions as illustrated in Fig. 3(b) must assessed before developing conflict detection and resolution algorithms. If the aircraft does not maneuver, the intercept triangle AIU gets smaller as both vehicles move along their respective straight path trajectories. Note that a favorable interpretation of the vectored condition can be seen in Fig. 4(a) when we consider the relative velocity of the UAV with respect to the aircraft, $\mathbf{v}_r$

$$\mathbf{v}_r = \mathbf{v}_u - \mathbf{v}_a. \tag{3}$$

Based on the intercept triangle in Figure 3(b), the matching condition is given by

$$\mathbf{t}_u = \frac{1}{\gamma}\left(\frac{r}{s_a}\mathbf{t}_s + \mathbf{t}_a\right) \tag{4}$$

where

$$\frac{s_u}{s_a} = \frac{v_u}{v_a} = \gamma. \tag{5}$$

In equation (4), the ratio r/s_a can be obtained by applying the law of cosine in the intercept triangle AIU. By the law of cosine, we have

$$\left(\frac{r}{s_a}\right)^2 + 2\cos\theta_{as}\left(\frac{r}{s_a}\right) - (\gamma^2 - 1) = 0. \tag{6}$$

When the collision geometry fulfills the matching condition in equation (5) and the kinematic condition in equation (6), the intercept triangle AIU becomes invariant and the relative velocity $\mathbf{v}_r$ in equation (3) establishes the approach collision direction.

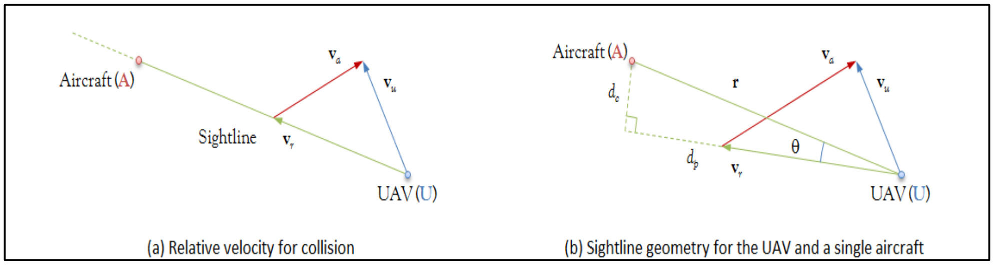


Fig. 12: Relative velocity approach (adapted from [15]).

2.4 Conflict Detection

From the sightline geometry shown in Fig. 4(b), White et al. [15] showed that the minimum separation, d_m, the closest distance of approach, d_c and the time to closest point of approach, τ can be used to detect the conflict between the UAV and the intruding obstacle. If the relative position vector is projected along the sightline, and the angle θ from the sightline to the relative velocity vector $\mathbf{v}_r$ is known, then d_c is given by

$$d_c = r \sin \theta \tag{7}$$

Similarly, the relative distance to the closest point of approach, d_p can also be determined

$$d_p = r \cos \theta \tag{8}$$

This gives the time to closest point of approach, τ

$$\tau = \frac{d_p}{v_r} \tag{9}$$

According to White et al. [15], the UAV and the intruding obstacle are said to be in conflict of interception, if d_c is strictly smaller than the minimum separation of d_m and the time to closest point of approach, τ is in the future but before the look-ahead time T, i.e.

$$d_c < d_m \quad \& \quad \tau \in [0, T). \tag{10}$$

2.5 Conflict Resolution

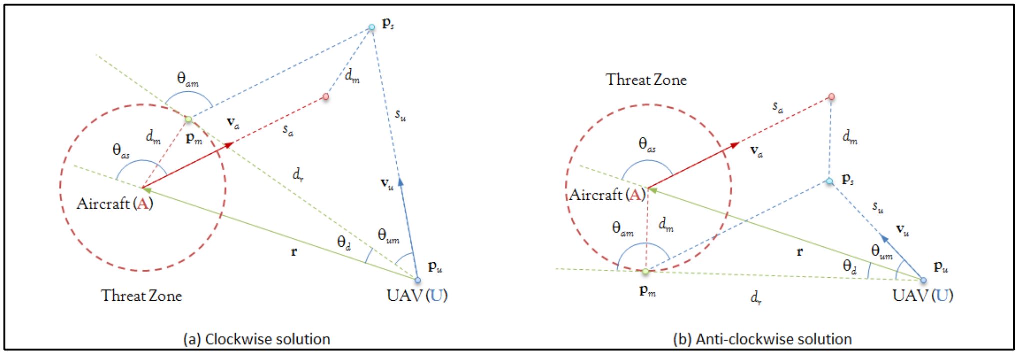
As investigated by Shin et al. [14], the collision geometry from Fig. 3(a) can be modified to become the resolution geometry as shown in Fig. 5. Using $\{\mathbf{p}_u, \mathbf{p}_m, \mathbf{p}_s\}$ to mark the resolution triangle, the figure shows for both clockwise and anti-clockwise trajectories. It should be noted that the resolution triangle and the intercept triangle are very much alike, though the resolution triangle is scaled and rotated about the point $\mathbf{p}_u$.

For the conflict resolution, Shin et al. [14] proposed that the direction of the relative velocity vector, θ_r, should become

$$\theta_r = \theta_m \equiv \begin{cases} \theta_s + \theta_d & \text{for the clockwise solution,} \\ \theta_s - \theta_d & \text{for the anti-clockwise solution.} \end{cases} \tag{11}$$

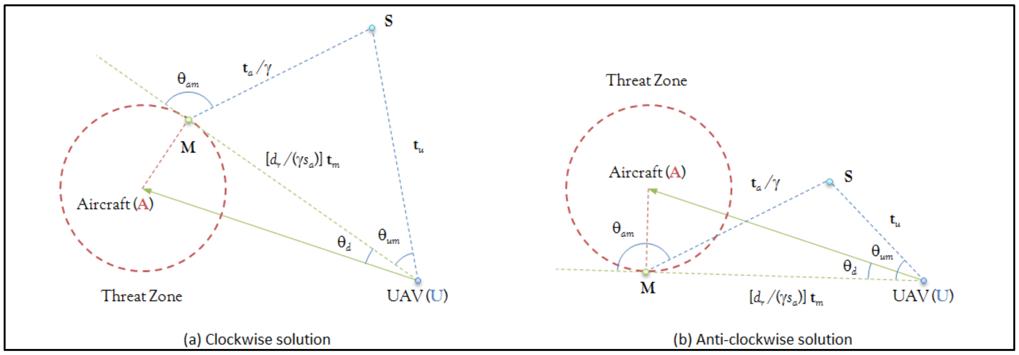
In order to resolve the conflict, the matching conditions for the clockwise and anti-clockwise solutions shown in Fig. 6 must be evaluated. As can be seen in the figure, the resolution matching condition is worked out from the vector addition for $\mathbf{t}_u$ calculated with respect to the resolution triangle MSU.

$$\mathbf{t}_u = \frac{1}{\gamma}\left(\frac{d_r}{s_a}\mathbf{t}_m + \mathbf{t}_a\right) \tag{12}$$



(a) Clockwise solution (b) Anti-clockwise solution

Fig. 13: Conflict resolution geometry for minimum separation (adapted from [14]).



(a) Clockwise solution (b) Anti-clockwise solution

Fig. 14: Resolution matching condition for minimum separation (adapted from [14]).

Application of the law of cosine to the resolution geometry yields the kinematic condition

$$\left(\frac{d_r}{s_a}\right)^2 + 2\cos\theta_{am}\left(\frac{d_r}{s_a}\right) - (\gamma^2 - 1) = 0 \tag{13}$$

For more comprehensive treatment of UAV conflict detection and resolution using differential geometry concepts, readers are encouraged to consult Shin [14].

2.6 UAV Autopilot Scheme

The heart of the block diagram in Fig. 7 shows the control scheme consists of two major parts. The first part is derived using the inverse dynamics method, which functions as normal navigation strategy for cruising flight. The second part is the cascading fuzzy controllers which is integrated to achieve obstacle avoidance and path tracking.

As reported in Sun et al. [16], the switching of the autopilot signal between the cruising mode and the steering mode can algorithmically computed by

$$U = \eta U_I + (1 - \eta)U_F \tag{14}$$

where η = [0, 1] is a switching variable, and U_I is the inverse dynamics controller and U_F is the fuzzy logic controller, which is only triggered on when the obstacle sensor detects any obstacle within its radius field, and at the same time the UAV enters the safety boundary of the obstacle.

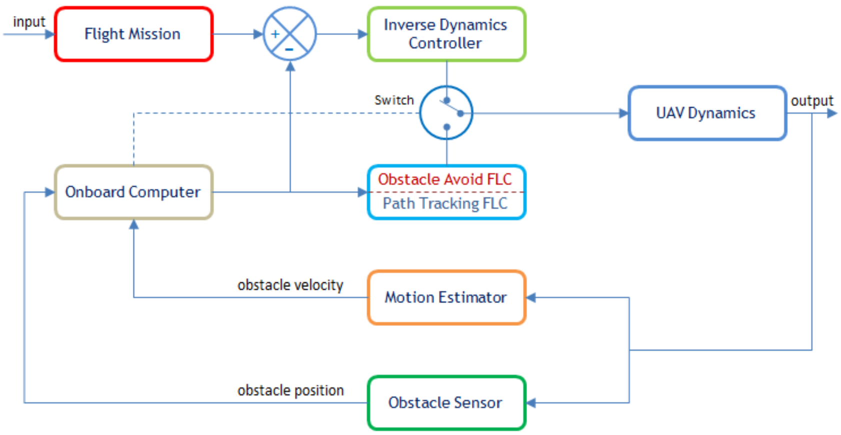


Fig. 15: Control block diagram of UAV autopilot system.

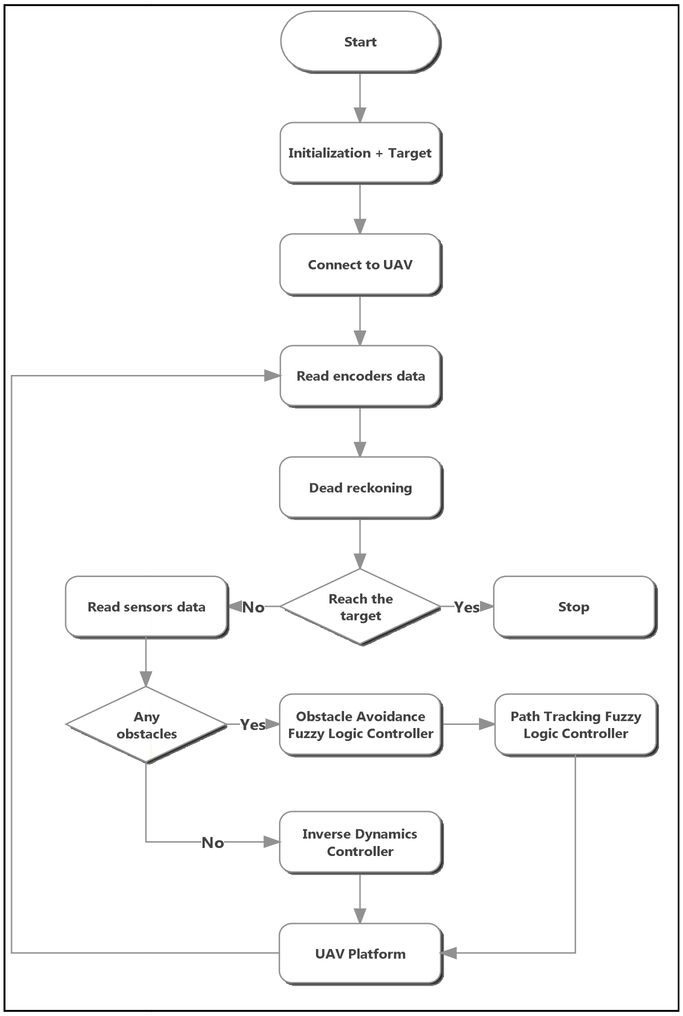


Fig. 16: Control flow and architecture of cascaded fuzzy modules of OAFLC and PTFLC.

3. FUZZY METHODOLOGY AND DESIGN

Figure 8 illustrates the control architecture of cascaded fuzzy modules of Obstacle Avoidance fuzzy logic control (OAFLC) and Path Tracking fuzzy logic control (PTFLC). Just like foot is part of the leg, and hand is part of the arm, the fuzzy control scheme is comprised of two cascading fuzzy modules that works by recursively breaking down the motion planning problem into two sub-problems, which are Obstacle Avoidance and Path Tracking.

3.1 Obstacle Avoidance Fuzzy Logic Control

In this maneuver, the algorithm starts with OAFLC when the sensors detect the obstacle within the compromising distance between of the UAV and the obstacle as shown in Fig. 9. The OAFLC is proposed to generate trajectory for the UAV in order to avoid obstacles in unknown dynamic environments. In this maneuver, the algorithm starts with OAFLC when the sensors detect the obstacle within the compromising distance between of the UAV and the obstacle. For this setup, two control inputs, "Obstacle Distance & Obstacle Angle" are used for fuzzification and two control outputs, "UAV Velocity & Heading Angle" are given after defuzzification.

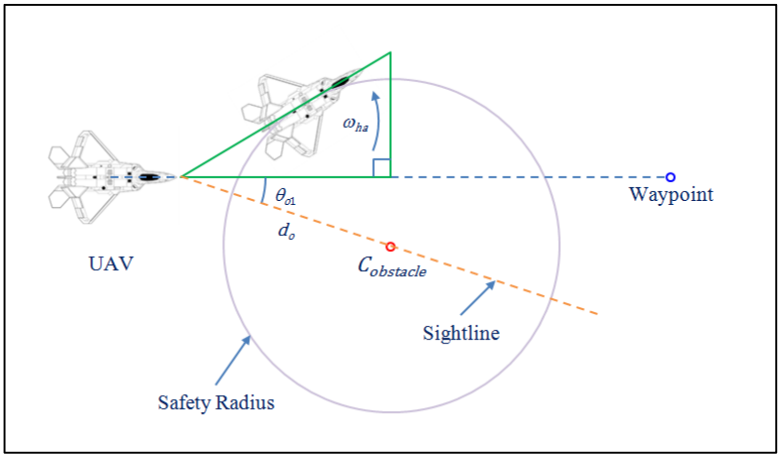


Fig. 17: The scheme for steering the UAV to avoid the obstacle.

The inputs of OAFLC are the distance from the UAV to the obstacle, d_o and the angle between the heading direction of the UAV and the sightline, θ_{o1}. Four linguistic terms of d_o are Very Close, Close, Medium, and Far respectively. The other input θ_{o1} is described by six membership functions: Negative Big, Negative Medium, Negative Small, Positive Small, Positive Medium, and Positive Big. The membership functions of d_o and θ_{o1} are shown in Fig. 10.

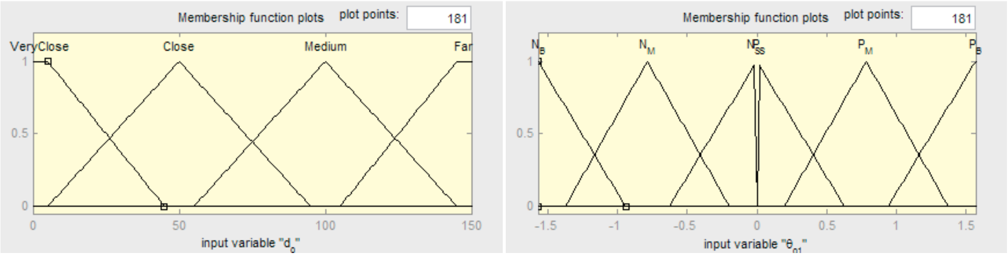


Fig. 18: OAFLC inputs – (Left) Obstacle-distance & (Right) Obstacle-angle.

The outputs of OAFLC are the linear velocity of the UAV, v and the angular velocity of the UAV, ω. Four linguistic terms of v are Very Slow, Low, Medium, and High respectively. The other output ω is described by seven membership functions: Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Big. The membership functions of v and ω are shown in Fig. 11.

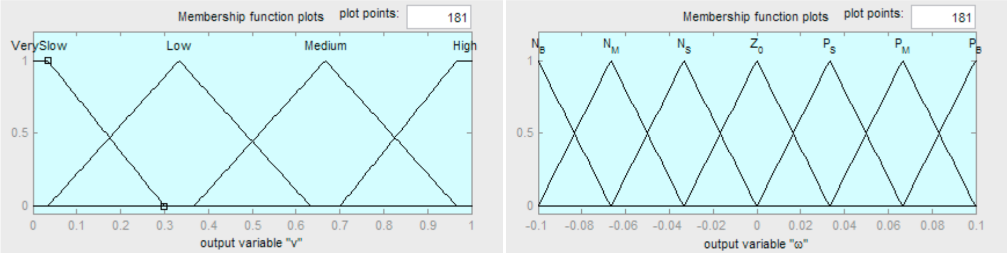


Fig. 19: OAFLC outputs – (Left) Linear velocity & (Right) Angular velocity.

The control module for determining v and ω commands contains a set of rules which are complicated nonlinear mappings of input signals into output signals. Rules relating the inputs and outputs for the fuzzy logic controller are set up in the form of if-then statements and are based on heuristics and human experience with navigating through an environment (similar to steering a car when entering and going around the roundabout). The rules for the fuzzy inference system can be summed up in some simple decision-making logic. There are a total of 24 rules for this setup, and the rules can be broken up into two output commands as shown in Table 1, Fuzzy Associative Matrix, where the Mamdani (max–min) model is used here.

Table 1: Fuzzy Associative Matrix – Fuzzy (max–min) Rules for OAFLC.

θ_{o1} / d_o	N_{Big}	N_{Medium}	N_{Small}	P_{Small}	P_{Medium}	P_{Big}	Output u
Far	High $Z_{0°}$	High $Z_{0°}$	Medium P_{Small}	Medium N_{Small}	High $Z_{0°}$	High $Z_{0°}$	v ω
Medium	Medium $Z_{0°}$	Medium P_{Small}	Low P_{Medium}	Low N_{Medium}	Medium N_{Small}	Medium $Z_{0°}$	v ω
Close	Low P_{Small}	Very Slow P_{Medium}	Very Slow P_{Big}	Very Slow N_{Big}	Very Slow N_{Medium}	Low N_{Small}	v ω
Very Close	Very Slow P_{Medium}	Very Slow P_{Big}	Very Slow P_{Big}	Very Slow N_{Big}	Very Slow N_{Big}	Very Slow N_{Medium}	v ω

By using this Fuzzy Associative Matrix. we can plot the control outputs corresponding to each value of the control inputs, and thus producing the control surfaces corresponding to the fuzzy control rules, as shown in Fig. 12.

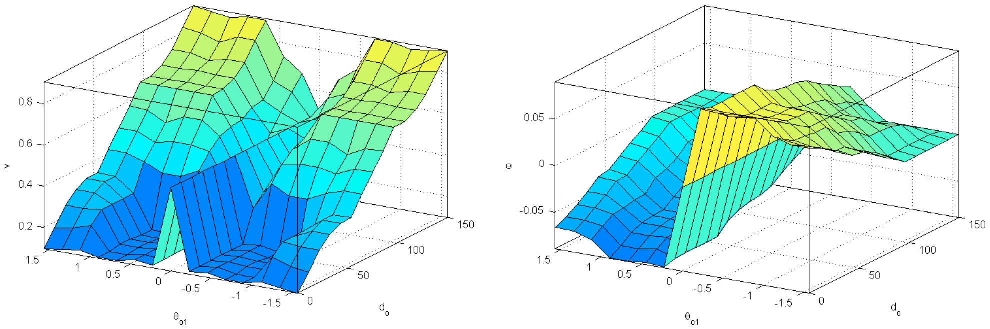


Fig. 20: OAFLC control surfaces – (Left) Linear velocity & (Right) Angular velocity.

3.2 Path Tracking Fuzzy Logic Control

With OAFLC, it will maneuver the UAV around the obstacle to avoid collision, causing the angle between the UAV heading direction and the sightline increases. If the collision is prevented and the obstacle angle is detected beyond 90°, that is when the UAV is moving at the tangent of the safety boundary as illustrated in Fig. 13, the control algorithm switches over to the PTFLC to steer the UAV to its waypoint smoothly.

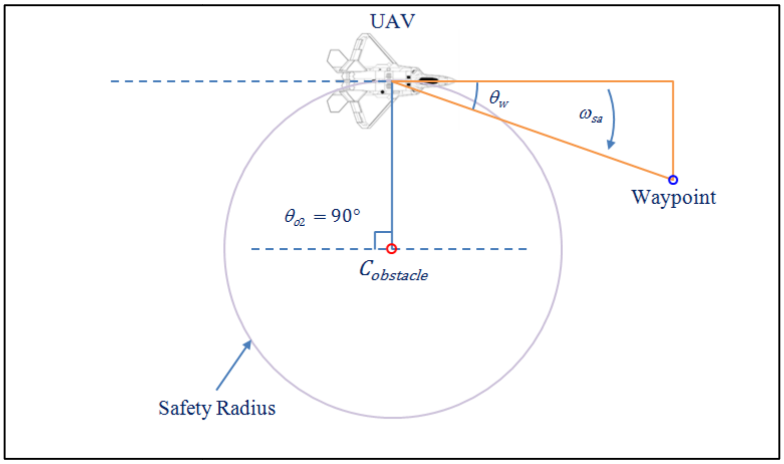


Fig. 21: The scheme for steering the UAV to its waypoint smoothly.

In the PTFLC setup, two inputs "Waypoint Angle θ_w & Obstacle Angle θ_{o2}" as shown in Fig. 14, are used for the fuzzification interface, and two outputs "Linear Velocity & Angular Velocity" as shown in Fig. 15, are given after defuzzification.

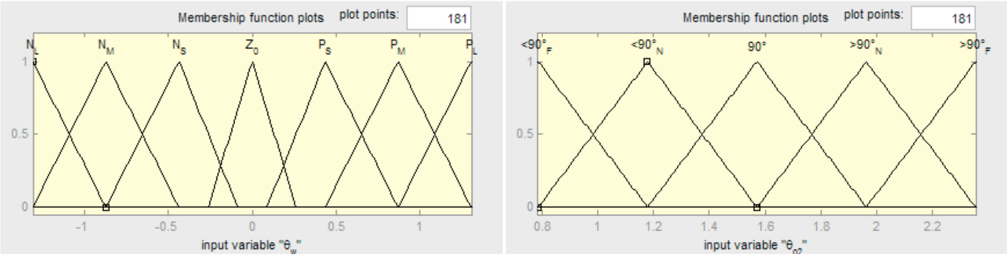


Fig. 22: PTFLC inputs– (Left) Waypoint-angle & (Right) Obstacle-angle.

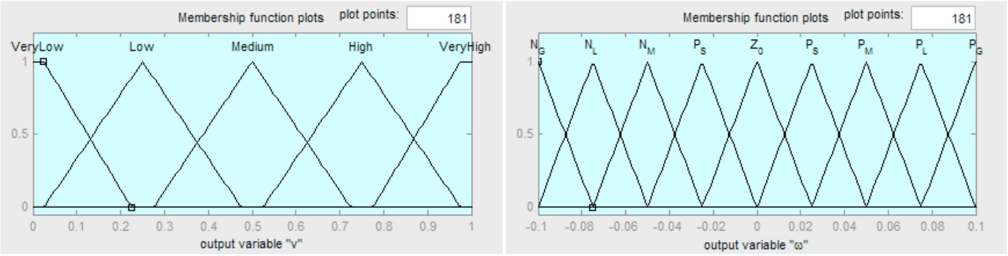


Fig. 15: PTFLC outputs – (Left) UAV-velocity & (Right) Steering-angle.

Likewise, the If-Then rules are based on human experience like steering a car when going around and exiting the roundabout, as shown in Table 2. Consequently, the control surfaces corresponding to the fuzzy rules are produced as shown in Fig. 16.

Table 2: Fuzzy Associative Matrix – Fuzzy (max–min) Rules for PTFLC.

θ_w / θ_{o2}	N_{Large}	N_{Medium}	N_{Small}	$Z_{0°}$	P_{Small}	P_{Medium}	P_{Large}	Output u
< 90° Far	High N_{Small}	High N_{Small}	Very High $Z_{0°}$	Very High $Z_{0°}$	Very High $Z_{0°}$	High P_{Small}	High P_{Small}	v ω
< 90° Near	Medium N_{Medium}	High N_{Small}	High N_{Small}	Very High $Z_{0°}$	High P_{Small}	High P_{Small}	Medium P_{Medium}	v ω
90°	Low N_{Large}	Medium N_{Medium}	High N_{Small}	Very High $Z_{0°}$	High P_{Small}	Medium P_{Medium}	Low P_{Large}	v ω
> 90° Near	Low N_{Large}	Low N_{Large}	Medium N_{Medium}	Very High $Z_{0°}$	Medium P_{Medium}	Low P_{Large}	Low P_{Large}	v ω
> 90° Far	Very Low $N_{Greatest}$	Low N_{Large}	Low N_{Large}	Very High $Z_{0°}$	Low P_{Large}	Low P_{Large}	Very Low $P_{Greatest}$	v ω

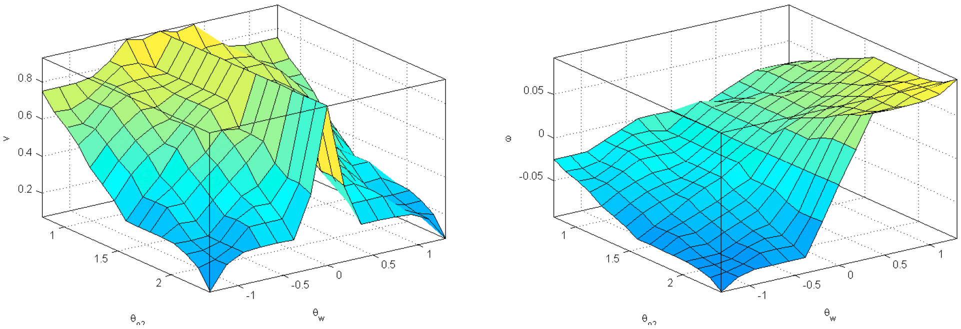


Fig. 16: PTFLC control surfaces – (Left) Linear velocity & (Right) Angular velocity.

4. SIMULATION STUDIES AND RESULTS

The performance of the developed fuzzy control strategy is investigated by conducting numerical simulations on path tracking in the presence of obstacles with three basic polygons (i.e., square, triangle and circle). For simplicity, the simulation is run in Cartesian coordinates on $\mathbb{R}^2$. This setting is not a restriction of the approach. Table 3 gives the initial conditions and parameter values used in the simulation. The following non-restrictive assumptions are used in this study [17]:

i) The vehicles are flying at a constant altitude and without wind turbulence.

ii) The mass and moment of the inertia of the vehicles are constant.
iii) The actuator dynamics of the UAV is approximated as a first order time delay system with a time constant of 0.01 s.
iv) The control surface deflection limit of the rudder is $\delta_R \in$ [-30°, 30°].

Table 3: Simulation settings for UAV maneuverability.

Parameters	**Values**
Sensing range of the UAV, r_s	10 m
Minimum separation distance, r_a	1.7 m
Max. ground speed, $v_{\max}$	2 m/s
Min. ground speed, $v_{\min}$	1 m/s
Max. acceleration, $a_{\max}$	0.6 m/s^2
Max. turn rate, $\omega_{\max}$	5°/s
Time delay for controlling the velocity, τ_v	3 s
Time delay for controlling the heading angle, τ_θ	0.3 s
Sampling rate, T	0.05 s

The simulation part of this work is divided into six distinctive scenarios:

i) In Scenario 1, the UAV navigates through multiple waypoints in a fairly straight manner with three static obstacles.
ii) In Scenario 2, the UAV navigates through multiple waypoints in a zigzag fashion with three static obstacles are placed in between the waypoints.
iii) In Scenario 3, In Scenario 5, the UAV navigates in an environment with three obstacles are aligned to overlap each other threat zones, and one waypoint is positioned in a close proximity with the obstacles.
iv) In Scenario 4, the UAV is set up to navigate in an environment that causes it to get caught in a “local minima” in the two-dimensional field.
v) In Scenario 5, a low-speed UAV navigates in an environment where an intrudering obstacle is crossing the path with the possibility of interception.
vi) In Scenario 6, a high-speed UAV navigates in an environment where an intrudering obstacle is crossing the path with the possibility of interception.

All simulations illustrated and presented here were implemented in C programming language using .NET Framework 4 and on a Windows 7 (64-bit) machine with an Intel CPU U7300 1.30GHz processor and 2.00GB of RAM.

Figure 17(a) illustrates the path performed by the UAV to reach the target. The generated path to the target is relatively smooth. In Fig. 17, the UAV trajectory is indicated by a solid blue line. The detection of the obstacle within the compromising distance between of the UAV and the obstacle incited the collision avoidance maneuver. In short, the generated path in Scenario #1 is reasonable because the UAV completely avoided the obstacles by maneuvering along the safe boundaries of the obstacles, and subsequently returning to the reference path.

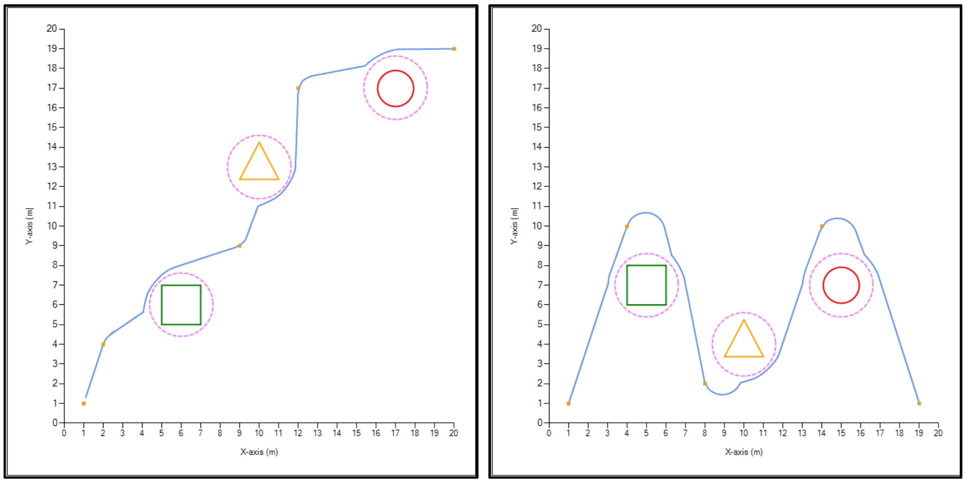


Fig. 17: (a) (Left) Scenario #1 and (b) (Right) Scenario #2.

The result in Fig. 17(b) indicates that the geometry of the path achieving complete avoidance is influenced by the positioning of the obstacles. In a nutshell, the generated path in Scenario #2 is reasonable because the UAV completely avoided the obstacles by maneuvering along the safe boundaries of the obstacles, even through the same obstacle is blocking between three waypoints.

As can be seen in Fig. 18(a), the control algorithm is designed so that the UAV is also capable of navigating around a group of aligned obstacles with overlapping boundaries and at the same time aborting certain flight submissions if the waypoint lies at a hazardous. In essence, the generated path in Scenario #3 is reasonable because the UAV completely avoided not only all obstacles, but also a group of aligned obstacles.

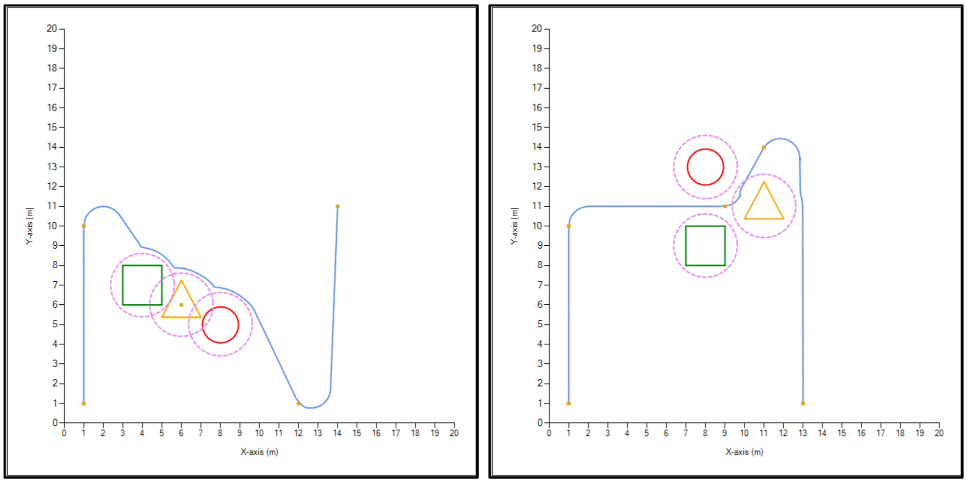


Fig. 18: (a) (Left) Scenario #3 and (b) (Right) Scenario #4.

In Fig. 18(b), the generated path to the target is relatively smooth. Even if the UAV is trapped in a "local minima", the control algorithm forcefully incites a steering maneuver to track the next waypoint and swiftly triggers the collision avoidance maneuver to cruise along the safe boundary until the conflict is cleared. To sum up, the resolution maneuver in Scenario #4 does not satisfy the minimum separation condition, although the UAV successfully broke through the cluster of obstacles without any collision. In reality, if a

UAV gets caught in a "local minima", it can climb up until the conflict is resolved, or to execute the aerobatic Immelman turn to escape.

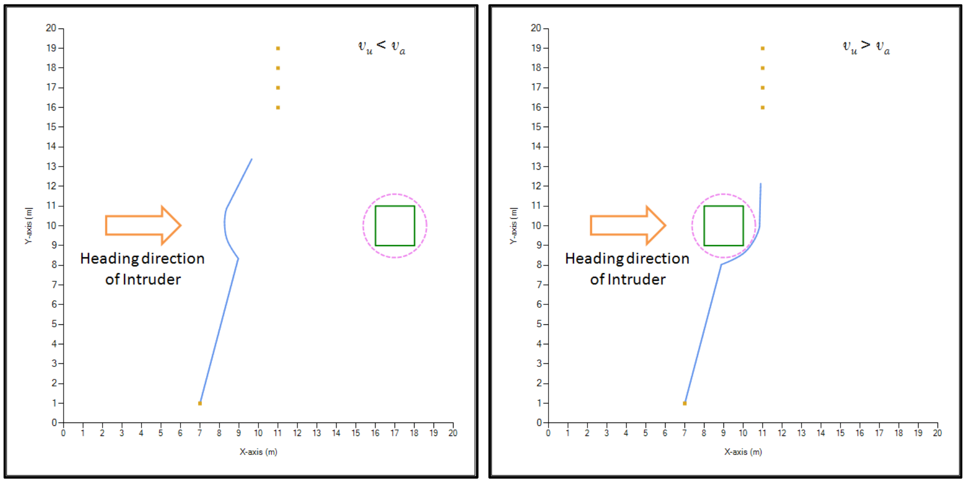


Fig. 19: (a) (Left) Scenario #5 and (b) (Right) Scenario #6.

Both scenarios in Fig. 19 show that the UAV is in conflict with the moving obstacle that is heading from the west to the east. In Scenario #5, the ground speed v_u of the UAV is less than the intruder speed, v_a. The simulation result shows that anticlockwise resolution approach successfully resolves the conflict. In Scenario #6, the UAV speed v_u is increased so that $v_u > v_a$. This triggers the UAV to take the clockwise resolution approach and it also successfully avoids the intruding obstacle.

All in all, it is important to note that the minimum distances between the UAV and obstacles are always greater than the minimum safe separation, unless it gets caught in a "local minima". Under normal circumstances, the proposed cascading fuzzy controllers effectively detect and resolve the conflicts.

4.1 Limitations

When carrying out the implementation of the proposed fuzzy controller at an early stage, the major disadvantage of an intelligent control for motion planning has been found to concur with the investigation of Sabo et al. [18] that the UAV's tendency to get caught in local minima. In fuzzy control, it is similar to suffering from left-right confusion, especially at very huge or symmetrical, curved-in obstacles. This drawback is an inherent characteristic in human intelligence like seen in badminton games. In doubles, a highly skilled player will play a tricky drop shot to the middle, aiming to place it exactly between the opponents to cause maximum confusion. Despite that, the benefit of using fuzzy logic is that the designer can slot in additional information and rules, thus generating multiple desired outputs with minimal control effort to circumvent these shortcomings.

From the simulation results, we ascertain that fuzzy logic is not the Holy Grail to the motion planning problem, though the optimum path can be achieved by tuning nonlinearity of the fuzzy membership functions. For that reason, neuro-fuzzy control is usually considered the complimentary form of the hybrid intelligence [19]. These sensors are able to determine the shape of obstacles that are about to be avoided. This ultimately translates into a smarter UAV that can not only avoid the obstacles, but also save on resources by determining the safe boundaries of the obstacles it should fly along.

4.2 Stability Analysis

Although fuzzy control employs the expert's linguistic description, in the form of rules, to control a process, it is generally a nonlinear controller. In fact, it does not have any differential equations like the linear control, and it is more difficult to analyze mathematically. In stability analysis, the availability of the mathematical model for the system is important, but, unfortunately, there are still difficulties with analyzing the stability of a fuzzy control system.

From a practical viewpoint, simulations are usually carried out for analyzing a dynamic system for gaining an insight to its dynamic behavior and to test stability of the fuzzy control system. However, it is useful to have a theoretical approach of analyzing the stability rather than just arbitrary simulations. Despite having difficulties with the stability analysis of fuzzy control, more or less advanced methods for proving the stability of fuzzy systems have been found. And one of the most general approaches to the stability analysis of nonlinear fuzzy control systems is using Lyapunov's direct method [20].

For ease of analysis, we assume the linear velocity of the UAV remain constant over the whole engagement to illustrate the use of Lyapunov's indirect method for stability analysis of Obstacle Avoidance fuzzy control. When the UAV to perform the obstacle avoidance trajectories is governed by the OAFLC, the error angle is given by

$$\theta_e = \theta_d - \theta \tag{15}$$

where θ_d is the desired UAV orientation. We control the UAV to fly to the desired orientation by using the following control law:

$$\dot{\theta}_e = -K_p\theta_e + \dot{\theta}_d \tag{16}$$

with the proportional constant $K_p > 0$, and $\dot{\theta}_d$ is the commanded angular velocity ω_c from the OAFLC output. Thus, we have rewrite equation (16) as

$$\dot{\theta}_e = -K_p\theta_e + \omega_c \tag{17}$$

Let's consider the following positive definite Lyapunov-function candidate

$$V = \frac{1}{2}\theta_e^2 \tag{18}$$

Taking the time-derivative of equation (18), we obtain

$$\dot{V} = \theta_e\dot{\theta}_e = -K_p\theta_e^2 + \theta_e\omega_c \tag{19}$$

The first term is clearly $-K_p\theta_e^2 < 0$ if $\theta_e \neq 0$. For $-90° \leq \theta_e \leq 90°$, we know that the relationship between θ_e and θ_o (obstacle angle) is given by

$$\theta_e = (90° - |\theta_o|)\,\mathrm{sign}(\theta_o) \tag{20}$$

Equation (20) tells that the sign of θ_e follows the sign of θ_o. From the OAFLC rule base in Table 1, it is observed that some values of θ_o yield zero ω_c, otherwise θ_o and ω_c are of opposite sign that allow us to interpret $\theta_e\omega_c < 0$ if $\omega_c \neq 0$. Now, taking the entire equation (19) into account, $\dot{V} \leq 0$, which is negative definite for any set of fuzzy rule. Hence, this guarantees the asymptotical stability of the proposed Obstacle Avoidance fuzzy controller. A more rigorous treatment to guarantee the stability of the fuzzy controller for Obstacle Avoidance and Path Following using the Lyapunov's stability theory can be found in Giap et al. [21].

5. CONCLUSION AND OUTLOOK

In this paper, we study the problem of motion planning of a fixed wing UAV, where the aircraft is required to travel to a known target location, passing through multiple waypoints while avoiding collision with various static and moving obstacles. All in all, the simulation results have validated the effectiveness of the proposed fuzzy control strategy and practical smooth paths have been generated by using the proposed cascading fuzzy logic modules in environments with reasonable static obstacles. We have also discussed the limitations of the fuzzy methodology and the stability analysis. Nevertheless, the high success rate from the simulation results is a distinguishing attribute of fuzzy logic control because it allows the user and system to capture much more information about the environment in a more probabilistic and efficient manner. In addition, the available tools such as MATLAB, fuzzyTECH, and DotFuzzy allow the user to easily develop and manipulate fuzzy inference systems. This makes fuzzy logic control a very powerful computational intelligence utility that allows the user to see the effects of reasoning with partial information and incorporating additional information with a minimal effort.

As demonstrated, the fundamental intelligent behavior of an aerial robot can be generally categorized into motion planning and reasoning on localization data and sensory feedback from its perceived environment. Some possible obvious future works that were not explored in this study are the motion planning in three-dimensional environments, avoidance of maneuvering aircraft, or shape-shifting obstacles, cooperative path planning of multiple UAVs using swarm intelligence and stability analytical testing of the fuzzy control system using other advanced methods such as the Describing Function technique, Popov criterion, Circle criterion and the Hyperstability criterion.

REFERENCES

[1] Austin R. (2010) Introduction to Unmanned Aircraft Systems (UAS); Unmanned Aircraft Systems: UAVS Design, Development and Deployment (Aerospace Series), John Wiley & Sons Inc, United Kingdom, p1-15.

[2] Samar R, Kamal WA. (2012) Optimal Path Computation for Autonomous Aerial Vehicles, edited by Amir Hussain and I. Aleksander, Cognitive Computation 4(4):515-525.

[3] Krozel J, Peters M, Hunter G. (1997) Conflict detection and resolution for future air transportation management. Technical Report NASA CR-97-205944.

[4] Kuchar J, Yang L. (2000) A review of conflict detection and resolution modeling methods. IEEE Transactions on Intelligent Transportation Systems 1:179-189.

[5] Zeghal K. (1998) A review of different approaches based on force fields for airborne conflict resolution. Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, p818-827.

[6] Albaker B, Rahim N. (2009) A survey of collision avoidance approaches for unmanned aerial vehicles,' 2009 International Conference for Technical Postgraduates (TECHPOS).

[7] Duong V, Zeghal K. (1997) Conflict resolution advisory for autonomous airborne separation in lowdensity airspace. Proceedings of the 36th IEEE Conference on Decision and Control, 3:2429-2434.

[8] Beard RW, McLain TW, Goodrich MA, Anderson EP. (2002) Coordinated target assignment and intercept for unmanned air vehicles. IEEE Transactions on Robotics and Automation 18(6):911-922.

[9] Jung S, Ariyur KB. (2011) Robustness for large scale UAV autonomous operations. IEEE International Systems Conference (SysCon), IEEE Press, p309-314.

[10] Dogan A. (2003) Probabilistic approach in path planning for UAVs. IEEE International Symposium on Intelligent Control, IEEE Press, p608-613.

[11] Kothari M, Postlethwaite I, Gu DW. (2009) Multi-UAV path planning in obstacle rich environments using Rapidly-exploring Random Trees, Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai, P.R. China, IEEE Press, p3069-3074.

[12] Ross T. (2010) Fuzzy Control Systems, Fuzzy Logic with Engineering Applications, 3rd edition, John Wiley & Sons, West Sussex, UK, p437-500.

[13] ASTM F2411 - 07: Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System, ASTM International, USA (2007).

[14] Shin, HS, Tsourdos A, White BA. (2012) UAS Conflict Detection and Resolution using Differential Geometry Concepts; Sense and Avoid in UAS (Aerospace Series), Plamen Angelov (Ed), John Wiley & Sons Inc, United Kingdom, p175-204.

[15] White BA, Shin HS, Tsourdos A. (2011) UAV obstacle avoidance using differential geometry concepts', IFAC World Congress 2011, Milan, Italy.

[16] Sun Z, Dong T, Liao XH, Zhang R, Song DY. (2006) Fuzzy Logic for Flight Control II: Fuzzy Logic Approach to Path Tracking and Obstacles Avoidance of UAVs, edited by Bai Y, Zhuang H, and Wang D, Advanced Fuzzy Logic Technologies in Industrial Applications, Advances in Industrial Control, Springer-Verlag, London, p223-235.

[17] Yamasaki T, Balakrishnan SN, Takano H. (2013) Separate-Channel Integrated Guidance and Autopilot for Automatic Path-Following. J. of Guidance, Control, and Dynamics 36(1):25-34.

[18] Sabo C, Cohen K. (2012) Fuzzy Logic Unmanned Air Vehicle Motion Planning. Advances in Fuzzy Systems, vol. 2012, Article ID 989051, 14 pages, doi:10.1155/2012/989051.

[19] Karray FO, De Silva CW. (2004) Neuro-fuzzy systems; Soft Computing and Intelligent Systems Design: Theory, Tools and Applications, Addison-Wesley, United Kingdom, p337-362.

[20] Michels K, Klawonn F, Kruse R, Nürnberger A. (2006) Stability Analysis of Fuzzy Controllers; Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers, Springer-Verlag, Berlin Heidelberg, p257-306.

[21] Giap NH, Shin JH, Kim WH. (2008) Adaptive robust fuzzy control for path tracking of a wheeled mobile robot. Artificial Life and Robotics 13(1):134-138.