OPTIMIZED FUZZY INFERENCE SYSTEM-BASED THREE-DIMENSIONAL RESOLUTION ALGORITHM FOR COLLISION AVOIDANCE OF FIXED-WING UNMANNED AERIAL VEHICLES

Shyam Rauniyar(1) **and Donghoon Kim**(2)
(1)(2)*University of Cincinnati, Cincinnati, OH 45221, USA, +1-513-556-5365,*
rauniysm@mail.uc.edu, Donghoon.Kim@uc.edu

Abstract: *Fixed-wing Unmanned Aerial Vehicles (UAVs) provide better endurance, high-speed motion, and higher payload operations in comparison to their rotary-wing counterparts, yet during conflict and potential collision scenarios aren't as much agile and maneuverable. There is thus a necessity of devising novel algorithms to ensure avoidance of multi-threat conflicts for fixed-wing UAVs maintaining normal flight. This paper proposes a decentralized Fuzzy Inference System (FIS)-based resolution algorithm for a fixed-wing UAV that computes changes to commands like heading, velocity, and altitude for Collision Avoidance (CA) during Three-Dimensional (3D) flight towards a target using PD control. It is assumed that the UAVs are equipped with high-end RADAR sensors to sense nearby intruders. Potential collision is determined using TCAS-based thresholds. The UAVs are in coordinated-turn flight in windless environment to reduce the complexity of the system. The FIS is optimized by determining its required parameters using a genetic algorithm such that maneuvers performed using obtained commands maintain a separation greater than the closest point of approach threshold. Three sets of results namely - without CA, with manually set 3D-FIS CA, and optimized 3D-FIS CA, are compared to verify the efficacy of the algorithm and subsequent optimization.*

Keywords: *Collision Avoidance, Resolution Algorithm, Fuzzy Inference, Genetic Algorithm, Fixed-wing UAVs,.*

1. Introduction

The increasing airspace traffic density would cause congested traffic scenarios, which require the developed safety procedures to resolve multi-threat conflicts. Fixed-wing Unmanned Aerial Vehicles (UAVs) provide better endurance, high-speed motion, and higher payload operations in comparison to their rotary-wing counterparts, yet during conflict and potential collision scenarios are not as much agile and maneuverable. There is thus a necessity of devising novel algorithms to ensure Collision Avoidance (CA) of multi-threat conflicts for fixed-wing UAVs maintaining normal flight. CA is a key enabler for the integration of manned and unmanned missions. Over the years there have been efforts to develop several reliable Collision Detection and Resolutions (CDRs) for both manned and unmanned aircraft. Modeling of such methods usually requires decisions on state dimensions, state propagation, detection thresholds, maneuvering dimensions, and aircraft numerals [1].

CA Systems (CASs) can be categorized into four main methods based on the design factors of the sense and avoid system, namely - preferred CA, protocol-based decentralized CA, optimized escape trajectory approaches and E-Field methods [2]. Today in civil and military aviation, the most used CDR method is proposed in Airborne CAS II (ACAS II). The most common and widely used implementation of ACAS II is Traffic alert and CAS II (TCAS II) version 7.1. It uses the time to go to the Closest Point Approach (CPA) as a basis for the decision of CA strategy. Two types of alerts are issued by TCAS II. They are Traffic Advisory (TA) for the visual acquisition of the conflicting aircraft and Resolution Advisory to advise pilots on the range of vertical speed to avoid threats [3, 4]. ACAS II can be categorized into geometric relationship-based CAS approaches and thus is deterministic. Thus, one usually has unnecessary RAs issued, and the process itself is costly with long update cycles. Moreover, TCAS only provides vertical maneuvers, thus having potential for improvement especially if to be applied to UAVs capable of horizontal maneuvers [5].

Considering limitations of current air traffic CA logics, increasing traffic, and increasing numbers of UAVs for which traditional ACAS II logics may not be ideal, transformations of air traffic management engaged through NextGen (US), and SESAR (Europe) led to the definition of a new ACAS based on new logic, namely ACAS X. There are two variations called ACAS Xa, for large aircraft, and ACAS Xu, for unmanned aircraft. ACAS II/Xa systems provide resolutions in the vertical plan (preferred) only since bearing measurements from sensors are inaccurate, the altitude provided by the transponder is precise and collision volume around an aircraft is only 100 ft vertically compared to 500 ft horizontally [6]. ACAS Xu ascertains the need of for both vertical and horizontal resolutions for UAVs since the speed ratio of UAV lesser than large aircraft making horizontal avoidance safer in some situations. Also, it could be useful for a non-cooperative intruder when there is inaccurate altitude estimation. It generally suggests that cooperative traffic resolution is ideal with vertical motions and non-cooperative traffic using horizontal motions. There is also a suggestion of using probabilistic models like two Markov decision process models, one each for horizontal and vertical maneuvers [7, 8].

There have been a vast number of research and algorithms proposed on CA pertaining to the different methods mentioned earlier. Using the geometric and kinematic models for fixed-wing aircraft, a simple pair-wise CA algorithm was proposed for short-range conflicts in the horizontal plane [9, 10]. However, there also have been proposals of algorithms providing 3D maneuvers for conflict resolutions using simple geometric models [11–15]. Some have even chosen to optimize the geometric-based CA algorithms [16–18]. Apart from the geometric methods, taking into consideration the notions of probabilistic models proposed in ACAS X logic, like fuzzy logic [19–22] among which some even use GAs for optimization of maneuvers [23].

The geometric methods discussed above especially that of TCAS and short-range conflict maneuvers for pair-wise conflict resolution does not consider continuous range of motion. It rather provides discrete parameters like climb rate and bank angles for resolution maneuvers. Thus, this research devices an optimized Three-Dimensional (3D) Fuzzy Inference System (FIS)-based Resolution Algorithm (RA) that can provide more efficient and continuous resolutions for fixed-wing UAV conflicts. The proposed decentralized 3D-FIS-RA for a fixed-wing UAV computes changes to commands like heading, velocity, and altitude for Collision Avoidance (CA) during a flight towards a target controlled using PD controller.

2. Proposed Algorithm: FIS-based 3D RA

Taking the inspiration from the CA logic of ACAS Xu, the solution proposed here uses two FISs for each of the horizontal and vertical maneuvers' parameters. The states considered are the relative heading, separation distance, and relative velocity with respect to the intruder. Later, a separate FIS is used to determine the weights of each maneuver based on the relative distance and relative velocity. Genetic Algorithm (GA) is used with different training scenarios of conflicting aircraft to design the appropriate FISs (i.e., their membership functions and rule bases), to achieve time or energy-efficient CA maneuvers with minimal likelihood of collisions.

2.1. UAV Model and Assumptions

Here, a basic point-mass model of a fixed-wing aircraft [24] is explained via following equations of motions:

$$\dot{x} = V\cos\gamma\cos\psi \tag{1}$$

$$\dot{y} = V\cos\gamma\sin\psi \tag{2}$$

$$\dot{z} = V\sin\gamma \tag{3}$$

$$\dot{V} = \frac{T-D}{m} + g\sin\gamma \tag{4}$$

$$\dot{\gamma} = -\frac{L\cos\phi}{mV} + \frac{g\cos\gamma}{V} \tag{5}$$

$$\dot{\psi} = -\frac{L\cos\phi}{mV} + \frac{g\cos\gamma}{V} \tag{6}$$

Where, $\begin{bmatrix} x & y & z \end{bmatrix}^\top$ is the position of the UAV and $\begin{bmatrix} \psi & \gamma & \phi \end{bmatrix}^\top$ are 3-2-1 Euler angles providing the orientation of the UAV with respect to the inertial frame. Also, T, D, and L are thrust, drag, and lift on UAV, and m is its mass.

For feedback of the UAV and environment map, it is assumed that the UAV is equipped with proprioceptive sensors like Inertial Measurement Units (IMUs) and exteroceptive sensors capable of providing 3D position mapping of environments like 3D RADAR providing us range, azimuth, and elevation data. The relative velocity of the intruder is determined based on the change of its position estimated by the exteroceptive sensors over time.

For our case, the kinematic part of the model is only used modified for coordinated turned flight with zero side-slip and windless conditions [25]. This also includes the PD controller to achieve the commanded states. The equations of motion thus obtained are:

$$\dot{p}_n = V_g\cos\psi\cos\gamma \tag{7}$$

$$\dot{p}_e = V_g\sin\psi\cos\gamma \tag{8}$$

$$\dot{p}_d = -V_g\sin\gamma \tag{9}$$

$$\dot{\psi} = \frac{g}{V_g}\tan\phi \tag{10}$$

$$V_a\sin\gamma^c = \min\left(\max\left(k_h\left(h^c - h\right), -V_g\right), V_g\right) \tag{11}$$

$$\dot{\gamma} = k_\gamma(\gamma^c - \gamma) \tag{12}$$
$$\dot{V}_a = k_{V_a}(V_a^c - V_a) \tag{13}$$
$$\frac{g}{V_g}\tan\phi^c = k_\psi(\psi^c - \psi) \tag{14}$$
$$\ddot{\phi} = k_{P_\phi}(\phi^c - \phi) + k_{D_\phi}(-\dot{\phi}) \tag{15}$$

It is to be noted that UAV is in the North-East-Down (NED) coordinate earth frame. The control command input is $\begin{bmatrix}\psi^c & h^c & {V_a}^c\end{bmatrix}^\top$, given to a PD controller. Also, V_a is the airspeed, and V_g is the ground speed. For this research, wind speed is assumed to be zero, and hence $V_g = V_a$. Also, ψ is the heading angle, γ is the flight path angle, ϕ is the roll angle, and g is the acceleration due to gravity. In addition, k_h, k_γ, k_ψ, k_{V_a}, $k_{P\phi}$, and $k_{D\phi}$ are gain values for PD controller.

2.2. System Architecture

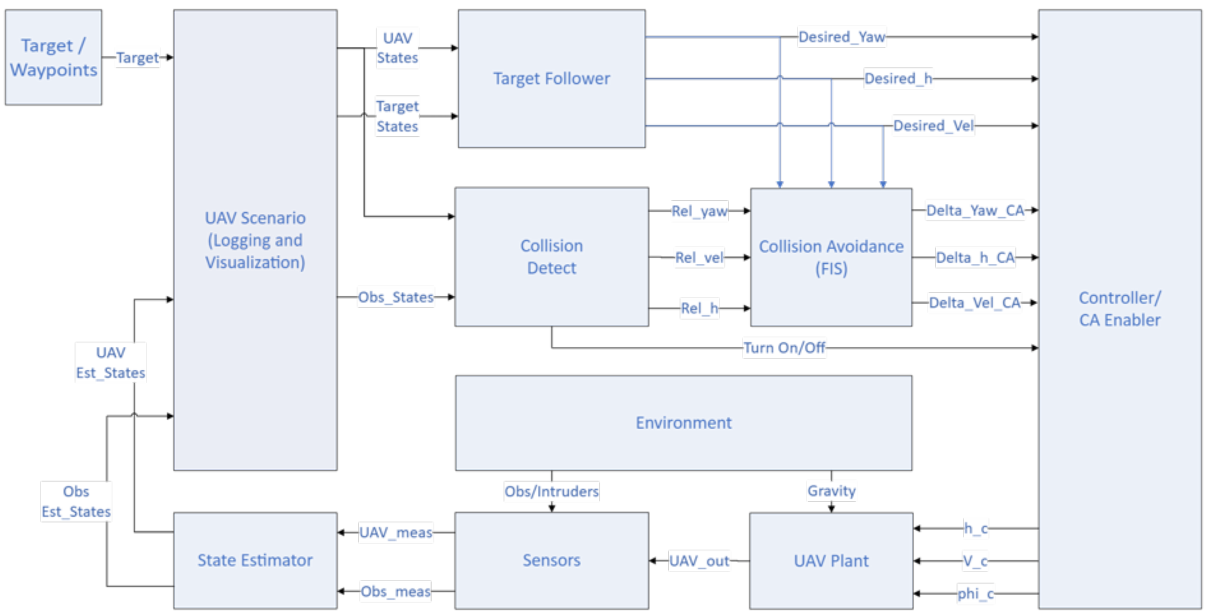

Figure 1. System Architecture and Control Flowchart

Figure 1 shows the architecture on which the proposed algorithm shall work. The UAV operates in an environment filled with intruders where the UAV is given a target position to reach. While doing so it can detect intruders and approaching intruders using its onboard sensors. Thereafter, the proposed CA algorithm shall provide necessary commands to the controller to calculate appropriate control inputs for the UAV. The sensor feedback is utilized for state estimation of the UAV itself and the environment.

2.3. Target Follower

The UAVs are initiated at different points in space and each of them have a specific target to reach. The state vector for the UAVs is, $\mathbf{y} = \begin{bmatrix}p_n & p_e & p_d & V_a & \psi & \gamma & \phi & \dot{\phi}\end{bmatrix}^\top$. And, the target states is, $\mathbf{y_t} = \begin{bmatrix}p_{nT} & p_{eT} & p_{dT}\end{bmatrix}^\top$. Using the current states of the UAVs and corresponding target states, the desired states for control are determined at each time interval (dt). The desired follower states for this case are $\begin{bmatrix}\psi^d & h^d & V_a{}^d\end{bmatrix}^\top$. Desired follower States calculated at each time interval as well. Equation 16 gives the desired heading (ψ^d).

$$\psi^d = \psi_T = \arctan\left(\frac{p_{eT} - p_{eA}}{p_{nT} - p_{nA}}\right) \tag{16}$$

Similarly, the desired height (h^d) is simply given by $h_d = -p_{d_T}$. However, height with respect to own-ship (Δh_T) calculated in Eq. 17 is also taken into consideration for the collision avoidance as will be later mentioned in 2.6..

$$\Delta h_T = -\left(p_{dT} - p_{dA}\right) \tag{17}$$

Lastly, the desired speed (V_{a_d} is calculated proportionally using the separation of from the target states from the current state (d_{slant}) and an optimal speed (V_{opt} chosen by the user. Firstly, a gain value $k_{V_{a_d}}$ is determined using optimal speed and initial separation (d_{slant_0}) as shown in equation below:

$$k_{V_{a_d}} = \frac{V_{opt}}{d_{slant_0}} \tag{18}$$

Then, Eq. 19 is used to determine the desired speed.

$$V_{a_d} = k_{V_{a_d}}\left(d_{slant} - d_{slant_0}\right) \tag{19}$$

2.4. Intruder Detection

The intruders are detected using the sensors installed on the UAVs. The research here is assuming the UAVs are installed with a high-end RADAR like Echodyne . The range of the sensor is assumed to be around 0.4 nm, with field of view of $\begin{bmatrix}-60 & 60\end{bmatrix}$ deg in azimuth and $\begin{bmatrix}-30 & 30\end{bmatrix}$ deg of elevation. The line of view of the sensor is set to be along the North direction in body reference frame of the UAVs. Only once the UAVs detect the intruders, the collision detection mode is activated.

2.5. Collision Detection

During the collision detection mode, the system of the UAVs calculated the necessary relative variables to determine if the closest point of approach is under the threshold of RA to initiate collision avoidance mode using 3D RA FIS. The collision detection scenario with different considered variables are visualized in Fig. 2.

Firstly, the relative velocity between the nearest intruder (identified using subscript i) and the own-ship (identified using subscript o) is calculated as:

$$\mathbf{V_{o/i}} = \mathbf{V_{own}} - \mathbf{V_{int}} \tag{20}$$

The velocity of the UAV can be then calculated in NED frame from following expression:

$$\mathbf{V} = \begin{bmatrix} V_a \cos\psi \sin\gamma \\ V_a \sin\psi \sin\gamma \\ -V_a \sin\gamma \end{bmatrix} \tag{21}$$

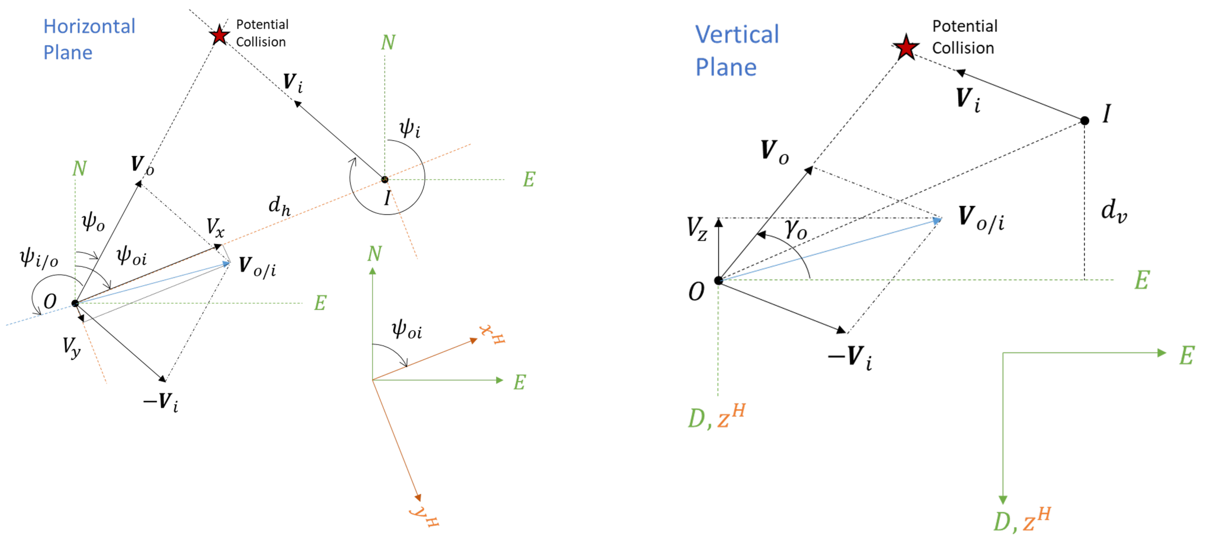

Figure 2. Collision Detection Scenario

Direction of position of intruder with respect to own-ship is given by:

$$\psi_{oi} = \tan^{-1}\left(\frac{p_{e_i} - p_{e_o}}{p_{n_i} - p_{n_o}}\right) \tag{22}$$

Thereafter, Eq. 23 gives the relative heading of B with respect to own-ship.

$$\psi_{i/o} = \psi_i - \psi_o \tag{23}$$

The distance between the own-ship and the intruder in horizontal (NE) plane, d_h, is calculated by:

$$d_h = \sqrt{(p_{n_i} - p_{n_o})^2 + (p_{e_i} - p_{e_o})^2} \tag{24}$$

Likewise, the position of intruder in the down direction with respect to own-ship (d_v) is determined by Eq. 25 while the vertical separation is thus simply $\|d_v\|$.

$$d_v = p_{d_i} - p_{d_o} \tag{25}$$

Now, we calculate the horizontal and vertical closure rates that are vital for determining the tau values for closest point of approach to estimate if the UAVs are in line of potential collision. For that, a NED to H Frame transformation is carried out, which is a Down axis rotation by ψ_{oi}. The Direction Cosine Matrix (DCM) for the transformation is expressed as:

$$C^H_{NED} = \begin{bmatrix} \cos\psi_{oi} & \sin\psi_{oi} & 0 \\ -\sin\psi_{oi} & \cos\psi_{oi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{26}$$

Then, DCM from Eq. 26 is used to get $V_{o/i}$ in H frame as follows:

$$\mathbf{V^H_{o/i}} = \mathbf{C^H_{NED}}\mathbf{V_{o/i}} = \begin{bmatrix} V_x & V_y & V_z \end{bmatrix}^\top \tag{27}$$

Here, V_x is *horizontal closure rate* (usually expressed in knots), and V_y is another component of $\mathbf{V^H_{o/i}}$ in horizontal plane, and V_z is *vertical closure rate* (usually expressed in ft/min).

The range tau (τ_{range}) and vertical tau ($\tau_{vertical}$) use closure rates obtained from Eq. 27 in Eqs. 28 - 29.

$$\tau_{range} = \frac{d_h}{V_x} \tag{28}$$

$$\tau_{vertical} = \frac{d_v}{V_z} \tag{29}$$

These values are compared with the RA threshold specified in TCAS to detect potential collision. However, since the UAVs are assumed to be small operating at low altitudes ($<$ 1000 ft), the thresholds are reduced by a factor of $2/3$. In case of lower closure rates, the separation themselves are compared to the DMOD and ZTHR for the given sensitivity level for $<$ 1000 ft specified in TCAS reduced by a factor of $2/3$. For our case, DMOD is set to be 20 s, ZTHR to be 570 ft. Also, collisions are said to have occurred when distances are in the range of 30 % of DMOD and ZTHR.

2.6. 3D CA FIS

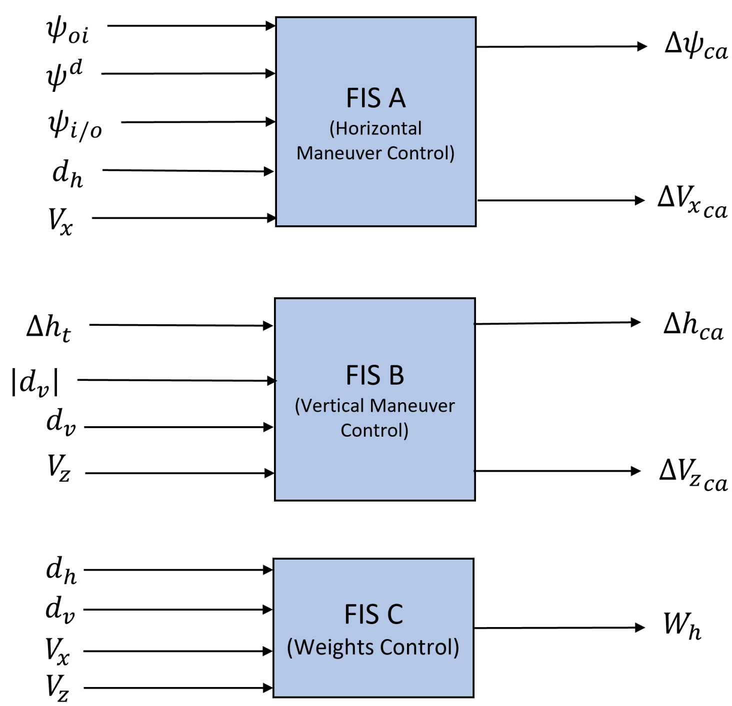

Figure 3. Proposed Fuzzy Inference System for Collision Avoidance

The FIS devised here is combination of FIS trees with three major blocks namely - FIS-A, FIS-B and FIS-C as shown in Fig. 3. FIS-A block escalates horizontal maneuver control by with outputs to change current states pertaining to horizontal motion, ψ and V_x, for avoiding collision. The input set for FIS-A is $\begin{bmatrix}\psi_{oi} & \psi_{i/o} & \psi^d & d_h & V_z\end{bmatrix}^\top$, and the output set is $\begin{bmatrix}\Delta\psi_{ca} & \Delta V_{xca}\end{bmatrix}^\top$

FIS-B block escalates vertical maneuver control by with outputs to change current states pertaining to vertical motion, h and V_z) for avoiding collision. The input set for FIS-B is $\begin{bmatrix}\Delta h_t & d_v & |d_v| & V_z\end{bmatrix}^\top$. This is where the desired control state Δh_t discussed in 2.3.is required. And, the output of the vertical maneuver FIS is $\begin{bmatrix}\Delta h_{ca} & \Delta V_{zca}\end{bmatrix}^\top$

Lastly, FIS-C block is a weight determination block that acts a bridge between the horizontal and vertical FISs' outputs so that each of them have their benefits based on the given collision situation. The inputs for FIS-C is $\begin{bmatrix}d_h & d_v & V_x & V_z\end{bmatrix}^\top$ and the output is weight for horizontal maneuver control (W_h). Weight for vertical maneuver control is simply, $W_v = 1 - W_h$

2.6.1. Membership Functions

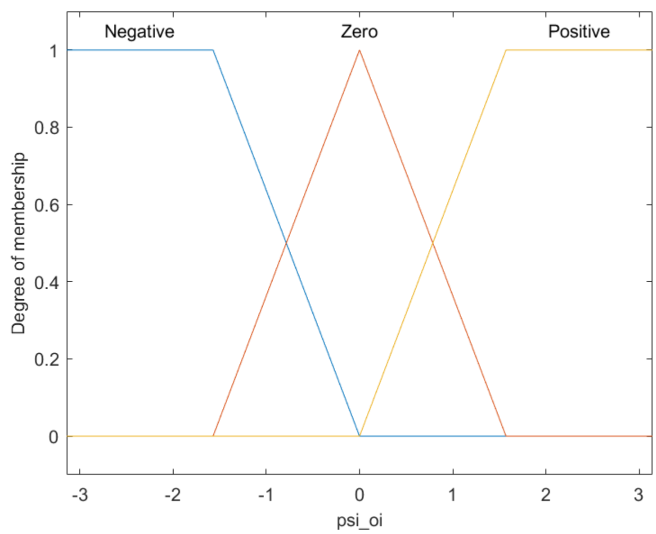

Figure 4. Membership Functions of FIS-A1 Non-optimized Input of ψ_{io}

All the inputs and outputs of the FIS structure includes 3 membership functions each. The extreme ends ones are trapezoidal and the middle one is triangular in nature. The parameters of the membership functions are symmetric at the beginning, later tuned using Genetic Algorithm for better performance of CA. For instance, Membership functions for ψ_{oi} of FIS-A1 is shown in Fig. 4.

2.6.2. Rule-base

Table 1. FIS A2 Rule-base

$\psi_{o/i}$	d_h		
	Low	Medium	High
Negative	High	Medium	Medium
Zero	Medium	Low	Low
Positive	Medium	Low	Low

The rule-base for the FIS is a set of rules that builds a relation between each of the inputs' membership function to the membership functions of outputs. Initially, the rule-base was designed intuitively which later was tuned using GA to improve the performance of CA. For instance, the non-tuned rule-base for FIS A1 is show in Tab. 1

2.6.3. GA Optimization

GA is run alongside the FIS structure outputs with a training scenario of UAVs in collision course, so that membership functions and rule-base are optimized for CA. The fitness function used for GA reduces the total distance traveled (dt) for each uav(i) and CPA penalties applied when any collision occurs. CPA penalties are reduced as distance at CPA (d_{CPA}) increases.

$$\min F = \sum_{i=1}^{n} d_t + \rho_{CPA} \tag{30}$$

$$\rho_{CPA} = \frac{\rho_x}{d_{CPA}} \tag{31}$$

Where, ρ_x is a large penalty for undesired collisions, set to be 30.

2.7. CA Controller

The controller block identifies whether CA is required and determines command control states ($\begin{bmatrix}\psi^c & h^c & V_a{}^c\end{bmatrix}^\top$) subjecting current states to changes using FIS outputs to avoid potential collision. For the case of no collision detected the command control states are basically equal to the desired control states. That is, command heading (ψ_c) is equal to ψ^d, command height (h^c) is equal to h^d, and commanded speed ($V_a{}^c$) is equal to V_a^d

However, when a collision is detected the command control states of heading and height are calculated as follows:

$$\psi^c = \psi + W_h \Delta\psi_{ca} \tag{32}$$

$$h^c = h + W_v \Delta h_{ca} \tag{33}$$

Command speed (V_{ac}) to avoid collision is obtained modifying horizontal closure rate and vertical closure rate as follows:

$$V_x{}^{ca} = V_x + W_h \Delta V_{xca} \tag{34}$$

$$V_z{}^{ca} = V_z + \ W_v \Delta V_{zca} \tag{35}$$

V_y is considered the same as previous to avoid extreme deviation from desired path. That is, $V_y{}^{ca} = V_y$. Then, the relative velocity with application of CA is given by Eq.

$$\mathbf{V^c_{o/i}} = \left(\mathbf{C^H_{NED}}\right)^{\mathbf{T}} \begin{bmatrix}V_x{}^{ca} & V_y{}^{ca} & V_z{}^{ca}\end{bmatrix}^\top \tag{36}$$

And, the desired CA velocity for the own-ship aircraft is:

$$\mathbf{V^c_o} = \mathbf{V^d_{o/i}} + \mathbf{V_i} \tag{37}$$

Finally, we have a command speed simply determined as:

$$V_a{}^c = \left|\left|\mathbf{V^d_o}\right|\right| \tag{38}$$

3. Results

There are three set of results to compare the efficacy of the algorithm used. The environment is so set that two UAVs are moving towards each other and are in a collision course. The simulation was run for 150 *s* for each set of results. The first of the three results is when collision avoidance is not applied. This is then compared to the results generated our of 3D-FIS-RA algorithm which is not optimized. Lastly, the results are then compared to the one obtained out of 3D-FIS-RA optimized using Genetic Algorithm.

3.1. Without Collision Avoidance

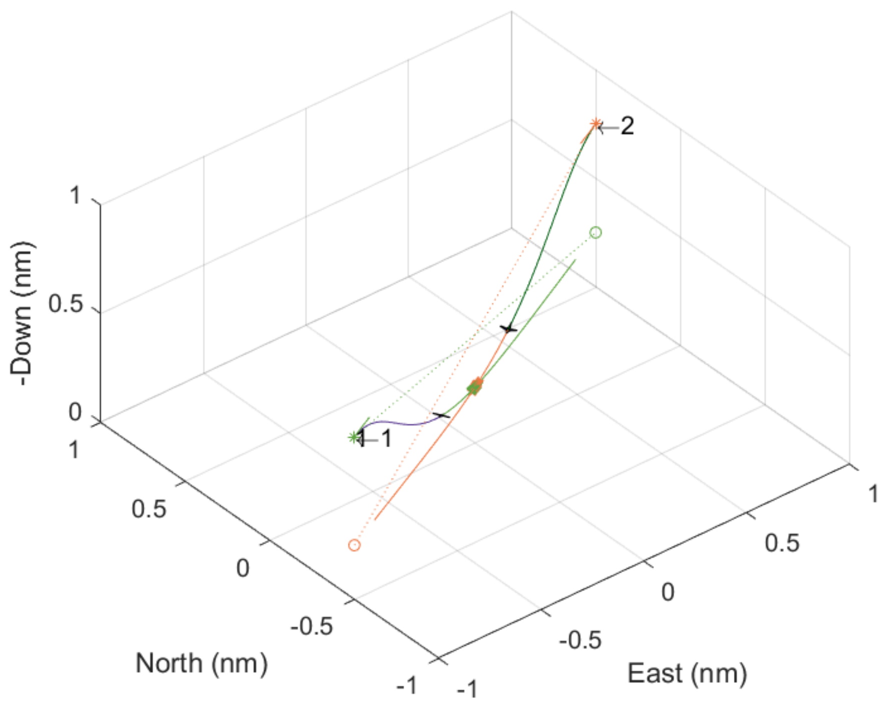

Figure 5. UAVs Colliding without Collision Avoidance

Figure 5 shows two UAVs that are moving towards each other when travelling towards their respective target positions. However, they are on a collision course as shown by the dark stars in the middle of the course. This scenario is later also used for optimization of parameters of 3D-FIS-RA. Since, collision avoidance is turned off for this case, the collision is shown to be happening.

Figure 6 shows smoother state transitions due to use of PD control to reach the targets. However, at the time of collision some maneuvers must be performed to avoid it. Hence, any deviation from this flow of states will suggest some maneuvers being performed to avoid collision.

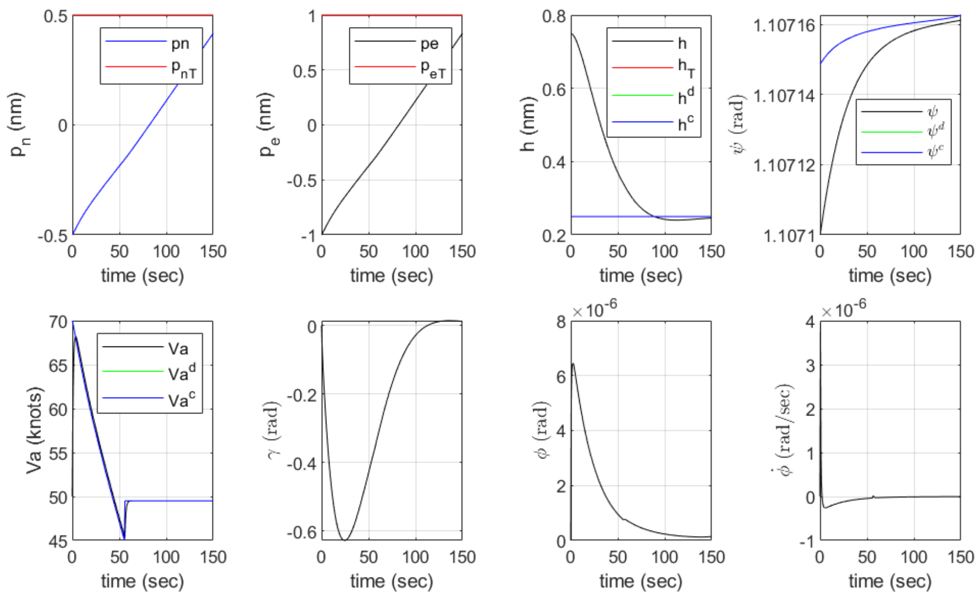

Figure 6. UAV 2 States in absence of Collision Avoidance

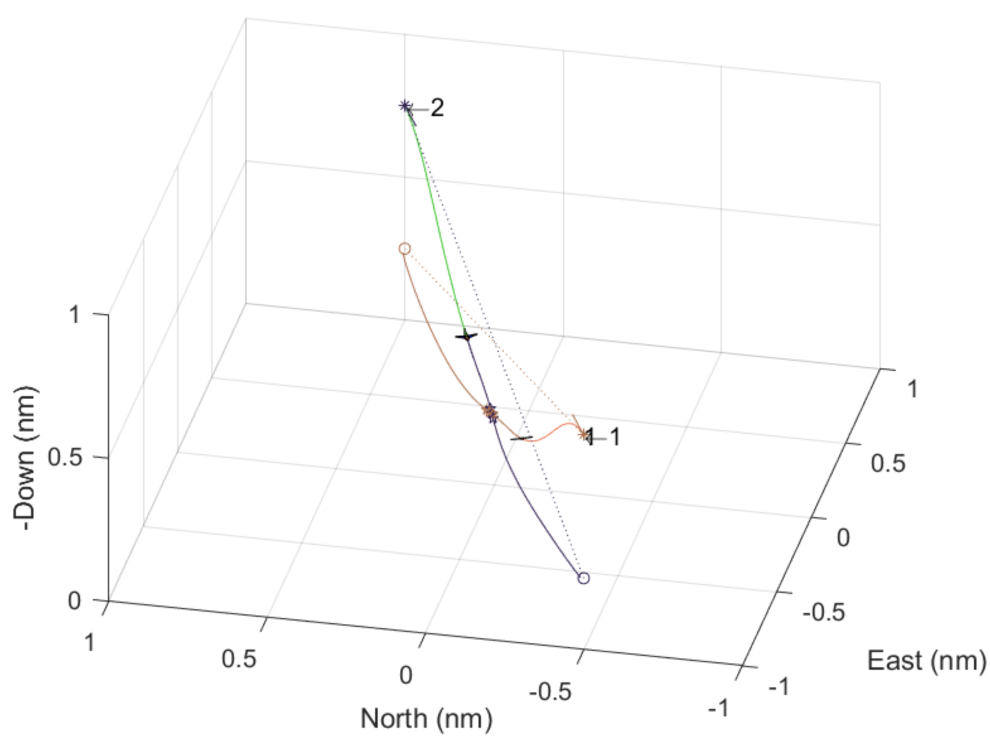

Figure 7. UAVs Maneuvering under manually set 3D-FIS-RA Collision Avoidance

3.2. CA With Non-Optimized 3D-FIS-RA

Here, the collision avoidance is set with a manually prepared 3D-FIS-RA, with rule-based defined intuitively. It can be seen Fig. 7 that their is slight change in paths of both the UAVs when approaching the CPA. However, it still undergoes collision inferred from the starts in the middle of the course. However, collision intensity is lower than that compared to the results of collision avoidance turned off. Since the collision was still not avoided using the manually set 3D-FIS-RA, optimization can be help improve it to avoid collisions.

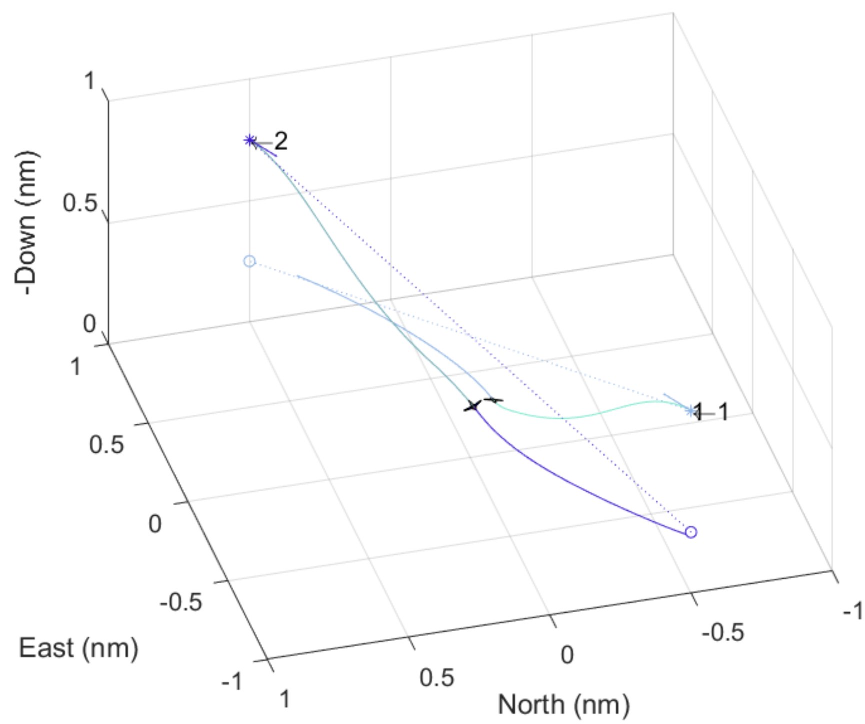

Figure 8. UAVs Maneuvering under optimized 3D-FIS-RA Collision Avoidance

3.3. CA With Optimized 3D-FIS-RA

The results obtained for this case are from simulation with collision avoidance using 3D-FIS-RA optimized using GA. It is clear from Fig. 8 that the UAVs completely avoid the collision setting up a 3D maneuver totally opposite to one another such that the closest point of approach is increase while maintaining grounds towards the target. When compared to the states shown in Fig. 6 the states obtained over time for this case shown in Fig. 9 have some variations, with change in command states due to collision detection at around 70 s showing the collision avoidance happens and is much smoother than the previous case of manual 3D-FIS-RA.

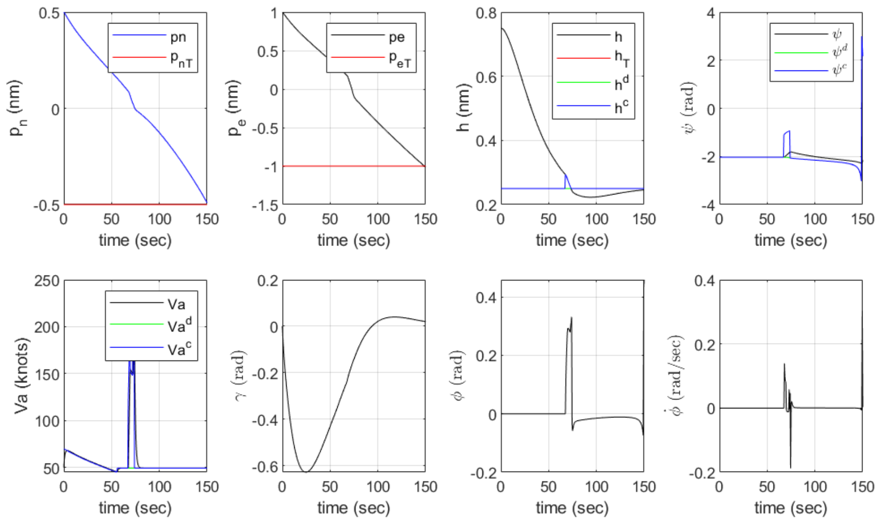

Figure 9. UAV 2 States when optimized 3D-FIS-RA used

4. Conclusion

The study proposed an optimized Three-Dimentional Fuzzy Inference System-based resolution Algorithm (3D-FIS-RA) that combines horizontal and vertical maneuvers to perform Collision Avoidance (CA) among UAVs in conflict. The proposed algorithm uses PD control for the UAVs to reach a target from an initial state. However, the 3D-FIS-RA is used to generated deviation commands that can be applied to the current states of the UAVs when potential collision is detected to avoid it. The FIS is optimized using Genetic Algorithm to improve the 3D maneuvers during CA while maintaining path to towards the target. The study compared different set of results namely - without CA, with CA using manually set 3D-FIS-RA, and optimized 3D-FIS-RA collision avoidance. It was clearly visible that the optimized algorithm proposed performed well enough to avoid the collision. However, the results could still be further optimized considering velocity limits and better cost function.

5. References

[1] Kuchar, J. K. and Yang, L. C. “A review of conflict detection and resolution modeling methods.” IEEE transactions on intelligent transportation systems, Vol. 1, No. 4, pp. 179–189, Dec 2000. doi:10.1109/6979.898217.

[2] Albaker, B. M. and Rahim, N. A. “A survey of collision avoidance approaches for unmanned aerial vehicles.” pp. 1–7. IEEE, Dec 2009. doi:10.1109/TECHPOS.2009.5412074.

[3] US Department of Transportation, F. “2011 TCAS II V7.1 Intro booklet.”

[4] Munoz, C., Narkawicz, A., and Chamberlain, J. A TCAS-II Resolution Advisory Detection Algorithm. AIAA Guidance, Navigation, and Control (GNC) Conference. doi:10.2514/6.2013-4622.

[5] Tang, J. “Review: Analysis and Improvement of Traffic Alert and Collision Avoidance System.” IEEE access, Vol. 5, pp. 21419–21429, 2017. doi:10.1109/ACCESS.2017.2757598.

[6] Manfredi, G. and Jestin, Y. “An introduction to ACAS Xu and the challenges ahead.” pp. 1–9. IEEE, Sep 2016. doi:10.1109/DASC.2016.7778055.

[7] Owen, M. P., Panken, A., Moss, R., Alvarez, L., and Leeper, C. “ACAS Xu: Integrated Collision Avoidance and Detect and Avoid Capability for UAS.” pp. 1–10. IEEE, Sep 2019. doi:10.1109/DASC43569.2019.9081758.

[8] Alvarez, L. E., Jessen, I., Owen, M. P., Silbermann, J., and Wood, P. “ACAS sXu: Robust Decentralized Detect and Avoid for Small Unmanned Aircraft Systems.” pp. 1–9. IEEE, Sep 2019. doi:10.1109/DASC43569.2019.9081631.

[9] Trapani, A., Erzberger, H., and Dunbar, W. Performance Analysis of a Horizontal Separation Assurance Algorithm for Short-Range Conflicts. AIAA Guidance, Navigation, and Control Conference. American Institute of Aeronautics and Astronautics, -06-14 2009. doi:10.2514/6.2009-5746.

[10] Erzberger, H. and Heere, K. “Algorithm and operational concept for resolving short-range conflicts.” Proceedings of the Institution of Mechanical Engineers. Part G, Journal of aerospace engineering, Vol. 224, No. 2, pp. 225–243, Feb 01, 2010. doi:10.1243/09544100JAERO546.

[11] Zhu, L., Cheng, X., and Yuan, F.-G. “A 3D collision avoidance strategy for UAV with physical constraints.” Measurement : journal of the International Measurement Confederation, Vol. 77, pp. 40–49, Jan 2016. doi:10.1016/j.measurement.2015.09.006.

[12] Lin, Z., Castano, L., and Xu, H. “A Fast Obstacle Collision Avoidance Algorithm for Fixed Wing UAS.” pp. 559–568. IEEE, Jun 2018. doi:10.1109/ICUAS.2018.8453307.

[13] Ferrera, E., Alcántara, A., Capitán, J., Castaño, A. R., Marrón, P. J., and Ollero, A. “Decentralized 3D Collision Avoidance for Multiple UAVs in Outdoor Environments.” Sensors, Vol. 18, No. 12, p. 4101, Nov 23, 2018. doi:10.3390/s18124101.

[14] Kim, Y., Lee, S., Lee, K., and Kang, J.-Y. "A Development of 3-D Resolution Algorithm for Aircraft Collision Avoidance." International journal of aeronautical and space sciences, Vol. 14, No. 3, pp. 272–281, 2013. doi:10.5139/IJASS.2013.14.3.272.

[15] gyun Woo, S., Park, C., Lee, D., and yeoun Lee, S. "Collision Avoidance for Unmanned Aerial Vehicles based on Safety Radius of the Formation Geometry." pp. 179–183. Institute of Control, Robotics, and Systems - ICROS, Oct 13, 2020. doi:10.23919/ICCAS50221.2020.9268210.

[16] Bilimoria, K. A geometric optimization approach to aircraft conflict resolution. 18th Applied Aerodynamics Conference. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2000-4265.

[17] Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T. "Dynamic Optimization Strategies for Three-Dimensional Conflict Resolution of Multiple Aircraft." Journal of guidance, control, and dynamics, Vol. 27, No. 4, pp. 586–594, Jul 2004. doi:10.2514/1.11168.

[18] Hu, J., Prandini, M., and Sastry, S. "Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution." Journal of guidance, control, and dynamics, Vol. 25, No. 5, pp. 888–900, Sep 2002. doi:10.2514/2.4982.

[19] Dong, T., Liao, X. H., Zhang, R., Sun, Z., and Song, Y. D. "Path Tracking and Obstacle Avoidance of UAVs -Fuzzy Logic Approach."

[20] Wang, K., Rong, H., Shi, M., Meng, Z., and Zhou, Y. "Reactive Obstacle Avoidance for Unmanned Helicopter Using Fuzzy Control." pp. 10026–10031. Technical Committee on Control Theory, Chinese Association of Automation, Jul 2018. doi:10.23919/ChiCC.2018.8483476.

[21] Kamthan, S. S., Singh, H., and Meitzler, T. "UAVs: on development of fuzzy model for categorization of countermeasures during threat assessment.", -05 2017. doi:10.1117/12.2275551.

[22] Du, B. and Liu, S. "A common obstacle avoidance module based on fuzzy algorithm for Unmanned Aerial Vehicle." pp. 245–248. IEEE, Jun 2016. doi:10.1109/CYBER.2016.7574830.

[23] Choi, D., Chhabra, A., and Kim, D. Collision Avoidance of Unmanned Aerial Vehicles Using Fuzzy Inference System-Aided Enhanced Potential Field. AIAA SCITECH 2022 Forum. American Institute of Aeronautics and Astronautics, -01-03 2022. doi:10.2514/6.2022-0272.

[24] Weitz, L. A. "Derivation of a point-ass aircraft model used for fast-time simulation." Tech. rep., April 2015.

[25] Beard, R. and McLain, T. Small Unmanned Aircraft Theory and Practice. Princeton University Press, 2 edn., 2012.