Detect and Avoid AI System Model Using a Deep Neural Network.

Jae young Ryu
dept. Aerospace Engineering
Inha University
Incheon, Republic of Korea
dbwodud1213@gmail.com

Hyeonwoong Lee
dept. Aerospace Engineering
Inha University
Incheon, Republic of Korea
hyeonwoong.lee@inha.edu

Hak-Tae Lee
dept. Aerospace Engineering
Inha University
Incheon, Republic of Korea
haktae.lee@inha.ac.kr

***Abstract*—Numerous Detect and Avoid (DAA) systems have been researched in the past to enable integrated operation of Remotely Piloted Aircraft System (RPAS) in the existing airspace with manned aircraft. This paper describes a process of constructing a DAA system using a Deep Neural Network (DNN). Training data are generated using the Detect and Avoid Alerting Logic for Unmanned Systems (DAIDALUS). DAIDALUS calculates the alert levels defined by the RTCA DO-365 MOPS document and outputs the conflict bands represented by ranges of heading, altitude, ground speed, and vertical speed that are predicted to cause well-clear violations with one or more aircraft. As the training data are required to cover a wide range of encounter geometries, two different data sets are combined. The first set of data are generated based on the historic operations that can reflect the characteristics of the airspace. Recorded trajectory data in a highly congested airspace, Incheon FIR of Republic of Korea, is used. Second set of data are generated based on the test vectors given by the MOPS that contains numerous combinations of encounter angles or speeds among many others. The DNN based DAA model is tested through DAA simulations. For this purpose, a previously developed pilot decision model is used along with a aircraft dynamics model. Flight Scenarios are created by modifying some of the test vectors or adding additional intruders. The DNN based model kept the aircraft free of loss of well-clear situation for all the test cases. Especially, it handled the multiple intruder situations that were not part of the training set. However, when compared with the same simulations directly using DAIDALUS, DNN based DAA model resulted in significantly increased fuel consumption, which suggest that the avoidance solutions were less efficient.**

***Index Terms*—Neural Network, DWC, DAIDALUS, Pilot decision model, DAA simulation, Fuel consumption**

I. Introduction

Enabling the integrated operations of unmanned aircraft in the existing airspace with the manned aircraft has been one of the major research topics of the community. The International Civil Aviation Organization (ICAO) divides the unmanned aircraft system in to three categories, Remote Piloted Aircraft System (RPAS), model aircraft used only for recreational purposes, and autonomous aircraft that does not allow pilot intervention. ICAO leads the standardization work for the integrated operation of the RPAS and manned aircraft [1]. Radio Technical Commission for Aeronautics DO-365 Minimum Operational Performances Standards (MOPS) [2] established the standards for the Detect and Avoid (DAA) system, which is the key component that ensures proper separation of Remotely Piloted Aircraft with other surrounding aircraft. DO-365 has also defined DAA Well Clear (DWC) to provide quantified alerting levels to allow systems to be designed and measured against. With the establishment of the standards, numerous DAA algorithms has been researched. Detect and Avoid Alerting Logic for Unmanned Systems (DAIDALUS) [3] was developed at NASA as a representative avoidance algorithm that conforms to the DO-365 standards.

With recent success of machine learning techniques, many DAA algorithms have been studied through supervised learning using Deep Neural Networks (DNN) [4]. Reference [5] used ACAS-Xu as the training data set. In Reference [6], the initial weights of the Proximal Policy Optimization's policy network is obtained through supervised learning.

In this paper, DNN is used for constructing a DAA system that mimics the behavior of the DAIDALUS. Two different style of encounter geometries are used to generate the training inputs. One is based on the test vectors provided by DO-365 and the other is extracted from historic flight data to reflect the characteristics of the air traffic where the DAA system will operate.

To evaluate the DNN based DAA algorithm, a fast-time simulation environment based on simplified aircraft dynamics model is developed. As DAIDALUS only provides the bands of headings and altitudes to avoid, a pilot decision model is necessary. A modified version of previously developed pilot decision model [7] is used for this study.

In order to evaluate the performance of the DNN based DAA algorithm, scenarios that are different from the ones present in the training set are developed. The DAA system is evaluated by additional fuel consumption and time spent in DWC alerts and compared to the ones that directly use DAIDALUS. DNN based DAA model along with the pilot decision model were able to maneuver the aircraft to prevent Loss of Well Clear (LoWC) situation. However, compared with the DAIDALUS, the avoidance paths were longer, resulting in larger fuel consumption. Notably, the DNN based model safely handled two scenarios where the ownship has to avoid two intruders.

Ministry of Trade, Industry and Energy and Ministry of Land, Infrastructure and Transport, Republic of Korea.

Section II describes the DNN configuration. Detect and avoid simulation setup using aircraft dynamics model and pilot decision models are presented in Section III. Section IV describes the evaluation of DNN based DAA system including the comparison with the direct simulation result using DAIDALUS. Finally, Section V concludes the paper.

II. Deep Neural Network (DNN)

A. DAIDALUS

In this paper, DNN is trained by the data set using the outputs of DAIDALUS. DAIDALUS is a software implementation intended to satisfy the operational and functional requirements detailed in NASA's DAA concept of integration for Unmanned Aircraft System (UAS) [8]. It aims to maintain well-clear for UAS operators and to restore well-clear when well-clear violation occurs or well-clear violation cannot be avoided. In the future, if a large number of training data set is available from either air traffic controllers' conflict detection and resolution results or human pilots' see and avoid efforts, these data can also be used.

Algorithms provided by DAIDALUS can be divided into detection, determine processing function, and alerting logic. Detection logic calculates the time interval of well-clear violation. It is predicted assuming the aircraft maintain the same speeds over a given look-ahead time. Determine processing function provides range of ownship maneuvers that leads to a well-clear violation in the form of conflict bands. If a well-clear violation occurs or it is unavoidable, it provides range of ownship maneuvers that recovers from a present or unavoidable well-clear violation in the form of recovery bands. Calculation proceeds assuming that ownship has constant turn rate, acceleration, and speed. As conflict and recovery bands, three types of bands are provided: track angle range (or heading, if wind information is provided), ground speed range (or airspeed, if wind information is provided), and vertical speed range. The alerting logic determines the corresponding alert types based on the given alerting schema.

Fig. 1 presents conflict bands for the track angles. α and β represent ownship's right and left limitations based on the performance limits. γ means range of maneuvers predicted to lead to well-clear violation. The shaded part is the region where well-clear violations occur.

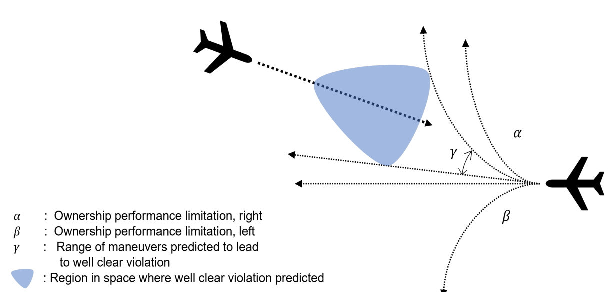

Fig. 1. Track conflict bands [3].

Fig. 2 shows the input and output of the DAIDALUS algorithm. Regarding separation alerts, the default values from the DO-365 MOPS, criteria for en-route phase I standards, are used. DAIDALUS also requires basic aircraft performance parameters. The values presented in Table I is used to represent the performances of commercial transport aircraft.

DADIALUS Java version 2.0.2 released in 2020 is used for the current study. In this version, conflict bands provide track, altitude, and speed commands at the same time. For the current study, only the horizontal maneuvers are considered. Consequently, the track bands represented by left limit, t_L, and right limit, t_R, are used along with the alert level, w. t_L and t_R range from 0 to 360. Discrete integer values of 1, 2, 3, and 4 are assigned to w that represent preventive alert, corrective alert, warning alert, and LoWC, respectively. When there is no risk, Not-a-Number (NaN) is provided as the conflict band. When there is no resolution value, infinity is provided as the conflict bands.

B. Constructing DNN

DNN [4] is composed of multiple layers of artificial neural networks and consists of nodes and links between input and output values. It has a structure as shown in Fig. 3. Input values to a node pass through a weight and a nonlinear function before being passed to the next artificial neural layer. This function is called an activation function. For this study, ReLU function that only allows positive inputs to pass through to the next layer is used as the activation function.

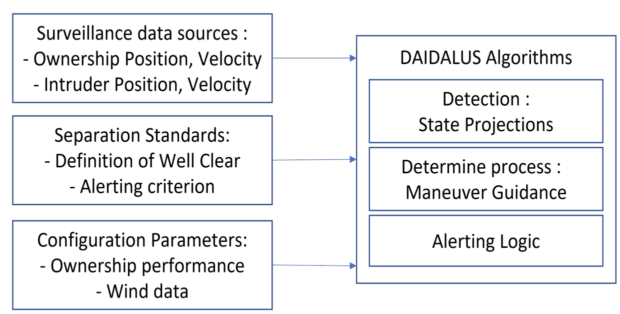

Fig. 2. Structure of DAIDALUS.

TABLE I
DAIDALUS Configurable parameters

Parameter	Default Value
Turn rate	3 °/s
Bank angle	30 °
Horizontal acceleration	2 m/s^2
Vertical acceleration	2 m/s^2
Min. ground speed	0 knots
Max. ground speed	700 knots
Min. vertical speed	-5000 ft/min
Max. vertical speed	5000 ft/min

The number of hidden layers is directly related to the number of weights. A large number of weights does not guarantee the performance of the trained network while being computationally costly. On the other hand, if the number of weights is too small, the DNN may not be able to sufficiently capture the characteristics of the training set. Initially, the model has four hidden layers, and more layers are added while monitoring the accuracy. When the model has seven hidden layers, there were minimum improvement of the accuracy compared with the case when the model has six hidden layers. Six layers and about 21,000 parameters are used as weight and biases. Table II summarizes DNN layers' units.

The input value in DNN consists of ownship's altitude, ground speed, vertical rate, heading and intruder's altitude, ground speed, vertical rate, heading along with the relative horizontal position of the two aircraft, totalling ten parameters. This is almost identical to the input to the DAIDALUS except DAIDALUS takes position of the two aircraft separately.

The output value in DNN consists of the DAA alerting level, w, and left and right track resolutions (t_L and t_R). If DAIDALUS returns a NaN value, -1 is used as an output instead. If an Infinity is returned, it means it cannot be avoided in that direction, so the aircraft's heading +180° value is used as an output instead. To make the three values within a similar range, a scaling is applied to construct the output vector, $\mathbf{Y}$, as shown in Eq. (1).

$$\mathbf{Y} = \begin{bmatrix} 10w & \frac{t_L}{10} & \frac{t_R}{10} \end{bmatrix} \tag{1}$$

The loss function is used to calculate the error (or distance) between the original value $\mathbf{Y}$ and expected value $\hat{\mathbf{Y}}$. For the current study, mean square error shown in Eq. (2), which has the batch size to sixteen, is used. Learning proceeds while adjusting the weights to reduce this error. The gradient of the loss function with respect to weights is used to adjust weights through the back propagation operation.

$$e_{MS} = \sqrt{\sum_{i=1}^{n} \left| \mathbf{Y}_i - \hat{\mathbf{Y}}_i \right|^2} \tag{2}$$

During the training, the optimizer and learning rate play an import role. If the optimizer is not appropriate, it may fall into local minimums, and the learning will not proceed any further. In this study, Stochastic Gradient Descent optimizer with Momentum is used to widen the search area. Momentum has a value of 0.9. The learning rate is different for each epoch, and it is shown in Table III.

C. Constructing Data Set

The data set needs to be generated for a wide range of encounter geometries. There are two types of training data set for the DNN model. First set of data are generated based on the actual operations that can reflect the characteristics of the airspace. ADS-B data within Incheon Flight Information Region (FIR) during year 2019 are analyzed to find conflict pairs [9]. Fig. 4 shows all trajectories from 13:00 on May 31, 2019 to 07:30 on June 2, 2019. From around one million flights, a total of 5600 encounters are identified that caused DWC phase 1 alert level of preventive alert or higher. All the computed alerts and corresponding positions are shown in Fig. 5 [9]. The original ADS-B data, purchased from FlightAware, are not at a regular time interval. All the trajectories are time synchronized at a regular one second interval using linear interpolation. With the interpolated trajectories, the 5600 encounters amounts to 300,000 seconds of cumulative flight times.

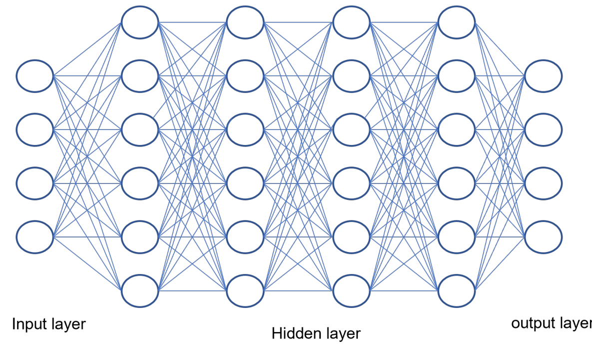

Fig. 3. Deep Neural Network structure.

TABLE II
LAYER UNITS

Input	Hidden 1	Hidden 2	Hidden 3
10	64	64	64
Hidden 4	Hidden 5	Hidden 6	Output
64	64	64	3

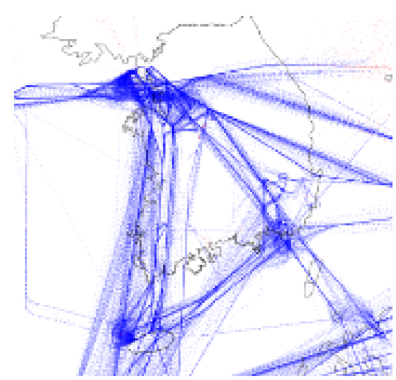

Fig. 4. ADS-B data for year 2019.

TABLE III
LEARNING RATES PER EPOCH

Epoch	500	1000	1500
Learning rate	10^{-7}	10^{-8}	10^{-9}

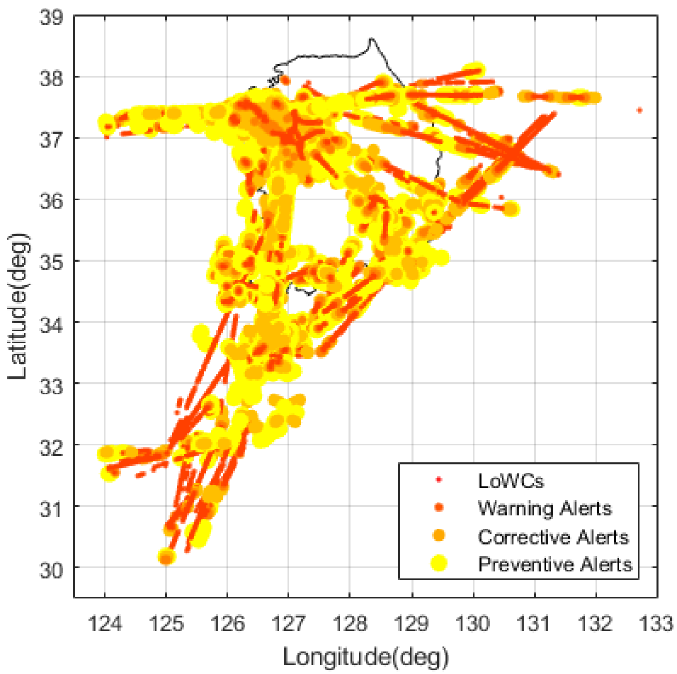

Fig. 5. DWC phase 1 alerts within Incheon FIR in 2019 [9].

The second set of data are generated based on the test vectors given by the MOPS [2] that has various situations with different encounter angles, speeds, or any other parameters. There are a total of 305 test vectors. Converge (C), Dynamic (D), Head on (H), High Speed (S), Maneuver (M), and Overtaking (O) encounter situations are used except for the Designer (D) situations that presented the encounter situations of more than two aircraft. A total of 50,000 seconds data set is constructed from the test vectors. Fig. 6 shows the Converge 11 situation.

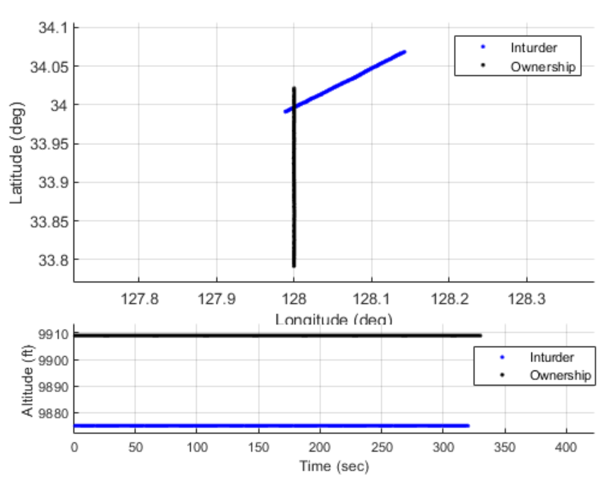

Fig. 6. Converge 11 encounter geometry.

III. DAA Simulation Setup

Fast-time simulations are performed to evaluate the trained DNN model. A five Degrees-of-Freedoms aircraft dynamics model [10] is used to generate the aircraft trajectories. Fig. 7 shows the architecture of the fast-time simulation. Initially aircraft are in waypoint navigation mode until maneuver commands are generated.

A. Aircraft model

Reference [10] calculates aircraft parameters based on a set of ten differential equations of motion, and uses Based of Aircraft Data (BADA) data to calculate engine thrust, drag, and fuel consumption. Since BADA is based on the mass model, the aerodynamic coefficient is not a function of the Angle of Attack (AOA).

$$C_L = C_{L_0} + C_{L_\alpha}\alpha \tag{3}$$

$$C_D = C_{D_0} + C_{D_2}{C_L}^2 \tag{4}$$

In Eq. (3), C_{L_0} is calculated using the aircraft cruising conditions suggested by BADA, and α is the amount of change in the AOA from the cruising state. In this study, performance parameters for Boeing 777-300 is used for the aircraft dynamics model. For the lift curve slope, $C_{L_\alpha} = 5.1$ is used. In Eq. (4), C_{D_0} and C_{D_2} are the drag coefficient according to the flap state suggested by BADA.

B. Pilot Decision model

The mode in which the aircraft maneuvers according to the DAIDALUS output with the pilot decision model of [7] is referred as to DAIDALUS mode, and the mode in which the aircraft maneuvers according to the DNN model with the simplified pilot model is called the AI mode.

In DAIDALUS mode, If the alert level of the ownship becomes higher than the corrective alert, the aircraft maneuvers using the track and altitude resolutions provided by DAIDALUS. The pilot model selects whether to use horizontal or vertical maneuver. It is assumed that the speed remains constant. Fig. 8 shows the details of this pilot decision model presented in [7].

In AI mode, if the alert level estimated by the DNN model is higher than the corrective alert, the aircraft maneuvers using the track resolution provided by DNN model. The pilot decision model selects whether to use the track resolution or to maintain current heading. Identical to the DAIDALUS mode,

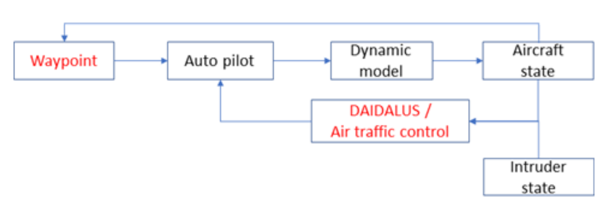

Fig. 7. Trajectory generation architecture.

the speed remains constant. New track resolution is provided every second. When executing the horizontal maneuver, the pilot decision model chooses the one requires smaller heading change from the current direction. The ownship maintains the last heading for five additional seconds after the alert level becomes preventive alert or lower and then starts maneuvering back to the original flight path. Fig. 9 shows decision model used for the AI mode. Compared with the decision model of [7], this model lacks several features that are altitude resolution, preferred heading change directions, and remaining time until the closest point of approach.

IV. DAA Simulation Results

A. Evaluation of the DAA system

In order to test the DNN based DAA model, two types of scenarios are created. One type of scenarios are created by changing the speed and the altitude of the test vector scenarios. Since their encounter types are identical, they are called the same name as the existing scenarios. Total of 8 scenarios, Converge 1 and 6, Head on 1 and 13, Maneuver 1 and 6, and Overtaking 12 and 15, are generated for this category. The other type of scenarios are created by adding another intruder to the existing test vectors. They are called 2 Intruder 1 and 2. These scenarios are simulated using the a Boeing 777-300 with a mass of 237 tons for both the ownship and the intruder.

The DAA system is evaluated by risk and efficiency. Risk assessment can be performed by measuring the time spent in one of the DWC alerts. For this study, time spent in warning alert is used for the risk metric because LoWC was avoided in all of the cases. Additional fuel consumption due to DAA maneuver is used for the efficiency metric.

B. Simulation results

Fast time simulations are conducted for DAIDALUS mode and AI mode for all the scenarios. The 2-D and 3-D trajectories and alert levels of the two DAA systems are illustrated.

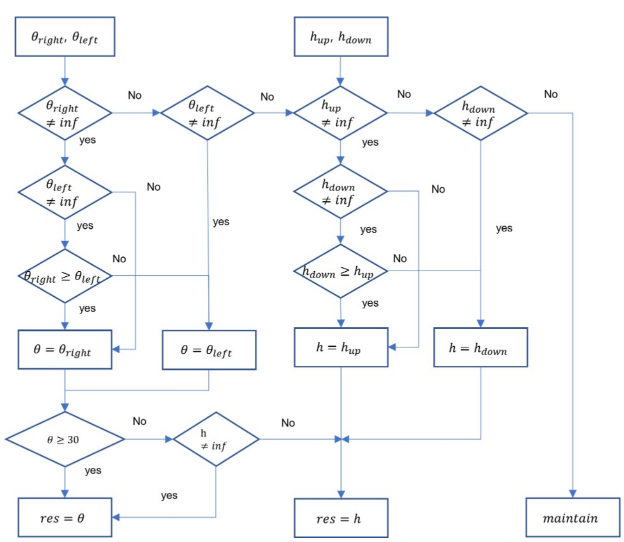

Fig. 8. Pilot decision model for DAIDALUS mode.

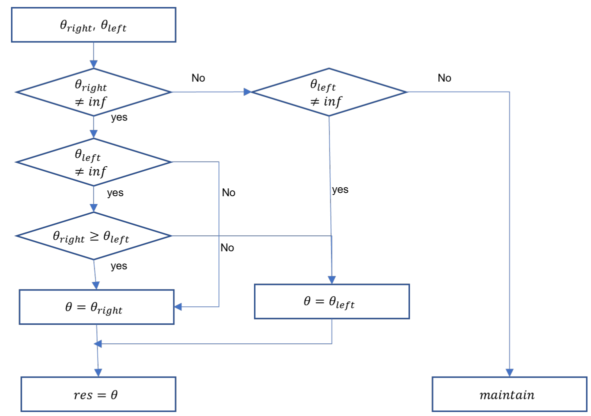

Fig. 9. Pilot decision model for AI mode.

In the figures that show maneuvering trajectories, the filled circle means the starting point, and the unfilled circle means the waypoint. The part indicated by the solid line is the maneuvering trajectory of the aircraft, and the part indicated by the dotted line is the trajectory of the original flight path. The black trajectory is the ownship's trajectory, the blue trajectory is the intruder's trajectory. On the ownship's route, points of preventive alert, corrective alert, warning alert, and LoWC according to DAIDALUS are expressed in green, orange, red, and purple, respectively.

Figs. 10 shows the alerts calculated by DAIDALUS and estimated by the DNN model for Converge 1 scenario while maneuvering according to DAIDALUS. As can be seen, the DNN model estimates the pair to be in the warning alert for much longer period. Figs. 11 and 12 show the 2-D and 3-D trajectories. As can be seen, the ownship was able to avoid LoWC by a moderate path stretch maneuver.

Fig. 13 shows the alerts estimated by the DNN model for the same Convergence 1 scenario while maneuvering according to the DNN model. Figs. 14 and 15 show the 2-D and 3-D trajectories. The initiation of the maneuver is slower, which caused more pronounced maneuver. Also while returning to the original flight plan, another avoidance maneuver occurs resulting in a S-shaped pattern.

Figs. 16 through Fig. 18 show the alerts, 2-D, and 3-D trajectories of the DAIDALUS mode for 2 Intruder 1 scenarios, respectively. Figs. 19 through Fig. 21 show the alerts, 2-D, and 3-D trajectories of the AI mode, respectively. Although the AI mode maneuver consumed more fuel it can be noted that it was able to avoid LoWC with two intruders, which was not present in the training set.

Table IV shows the time spent in warning alert and additional fuel consumption for all scenarios. It can be observed that the AI mode maneuvers were significantly less efficient. However they were able to prevent the LoWC from happening even with the two multiple intruder scenarios that were not present in the training set.

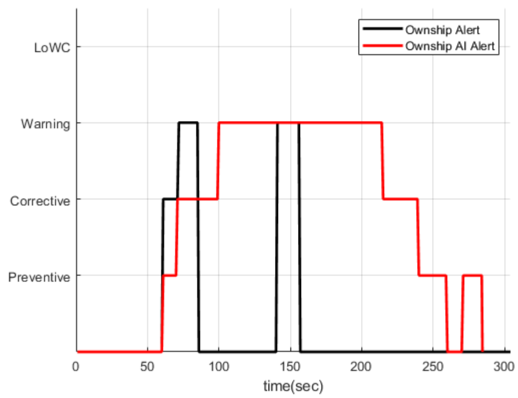

Fig. 10. DAIDALUS mode Alerts for Converge 1 scenario.

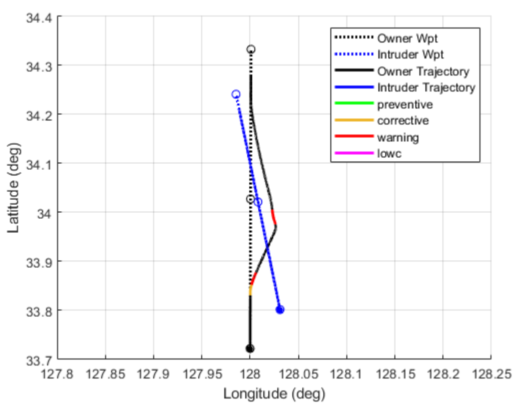

Fig. 11. DAIDALUS mode 2-D trajectory for Converge 1 scenario.

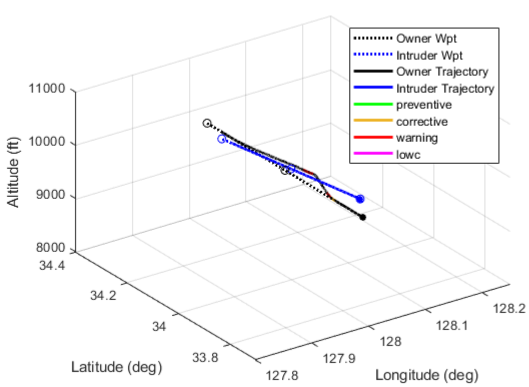

Fig. 12. DAIDALUS mode 3-D trajectory for Converge 1 scenario.

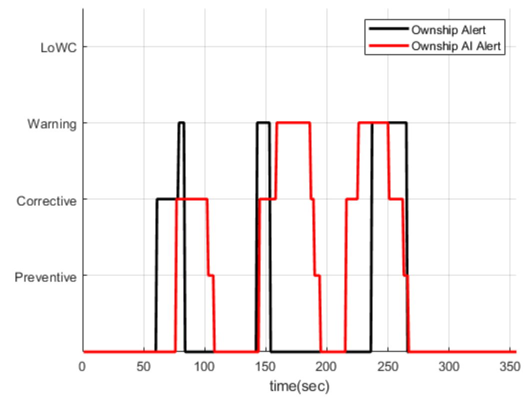

Fig. 13. AI mode Alerts Converge 1 scenario.

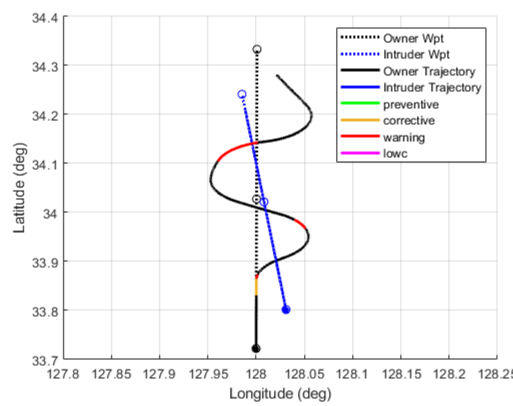

Fig. 14. AI mode 2-D trajectory for Converge 1 scenario.

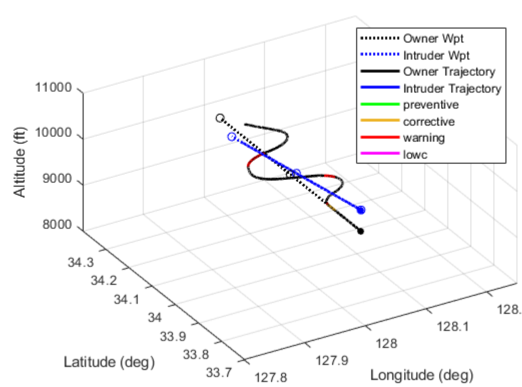

Fig. 15. AI mode 3-D trajectory for Converge 1 scenario.

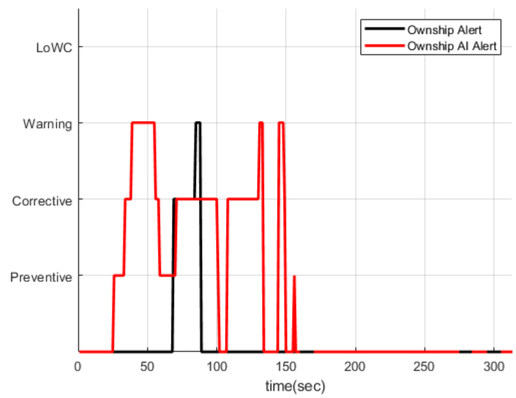

Fig. 16. DAIDALUS mode Alerts for 2 Intruder 1 scenario.

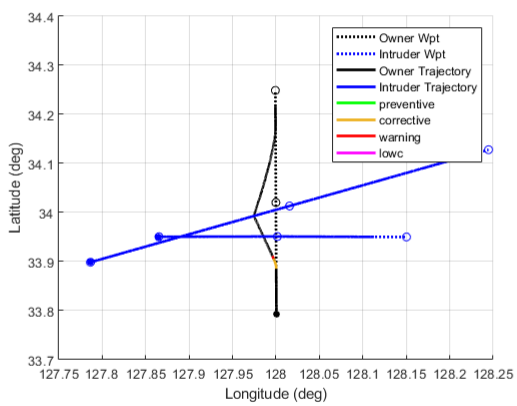

Fig. 17. DAIDALUS mode 2-D trajectory for 2 Inruder 1 scenario.

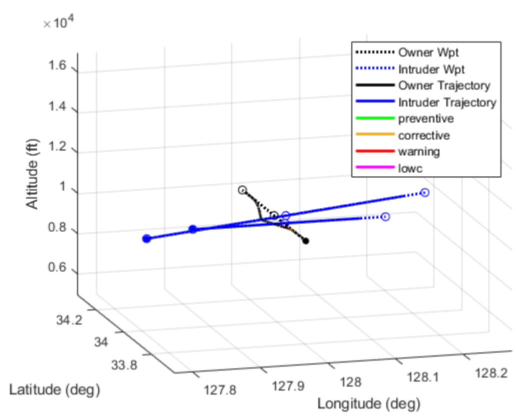

Fig. 18. DAIDALUS mode 3-D trajectory for 2 Inruder 1 scenario.

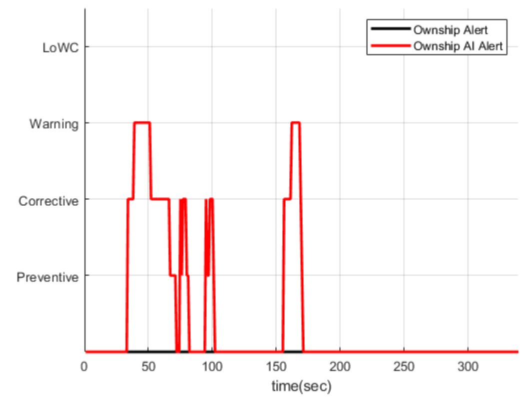

Fig. 19. AI mode Alerts for 2 Inruder 1 scenario.

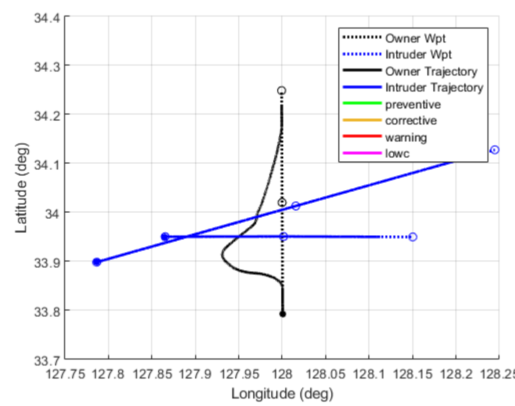

Fig. 20. AI mode 2-D trajectory for 2 Inruder 1 scenario.

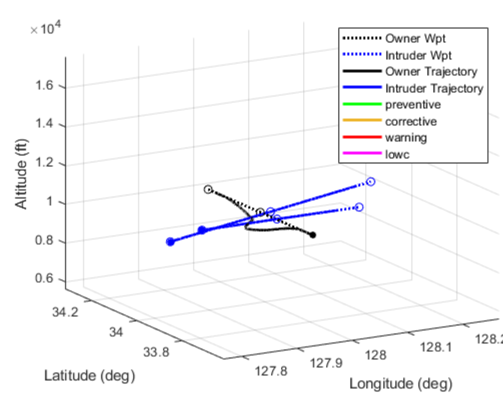

Fig. 21. AI mode 3-D trajectory for 2 Inruder 1 scenario.

TABLE IV
ADDITIONAL FUEL CONSUMPTION AND WARNING ALERT TIMES OF EACH SCENARIO

	Fuel consumption (kg)		Warning alert time (sec)	
Scenario	DAIDALUS	AI Mode	DAIDALUS	AI Mode
Converge 1	5.95 kg	156.90 kg	30 sec	45 sec
Converge 6	7.15 kg	39.38 kg	4 sec	8 sec
Head on 1	21.89 kg	286.46 kg	0 sec	0 sec
Head on 13	31.1 kg	400.34 kg	12 sec	7 sec
Maneuver 1	4.79 kg	3.88 kg	4 sec	17 sec
Maneuver 6	29.14 kg	281.01 kg	23 sec	16 sec
Overtaking 12	0.67 kg	175.01 kg	12 sec	0 sec
Overtaking 15	19.75 kg	189.76 kg	10 sec	4 sec
2 Intruders 1	7.15 kg	73.60 kg	4 sec	0 sec
2 Intruders 2	5.96 kg	319.77 kg	15 sec	0 sec

V. CONCLUSIONS

In this paper, a DAA system is constructed through supervised learning using DNN. The training data are generated using the historic aircraft trajectories in the Incheon FIR and test vectors provided in DO-365 MOPS. The DNN was trained using the output of DAIDALUS developed at NASA. Fast time simulations with aircraft dynamic model and two pilot decision models suggest that the DNN model generally estimates risk slightly later at a higher level resulting in inefficient horizontal maneuvers. However, the DNN based model was still able to prevent LoWC. Especially successfully handled the multiple intruder cases that were not a part of the training scenarios.

A follow study that includes other parameters such as vertical resolution and remaining time to the closest point of approach is expected to improve the DNN based model's performance. In addition, using air traffic controllers' instructions and pilots' see and avoid data are part of the future plan.

ACKNOWLEDGMENT

This work was supported by Development of eVTOL Flight Safety, Operability Evaluation Test and Collision Avoidance Managing Technology (No. 20016489) Project funded by the Ministry of Trade, Industry and Energy and Aviation Safety Management with Big Data Platform Implementation (No. 22BDAS-C158275-03) Project funded by the Ministry of Land, Infrastructure and Transport of the Republic of Korea.

REFERENCES

[1] ICAO, Manual on Remotely Piloted Aircraft Systems (RPAS), Doc 10019, 2015.

[2] DO-365: Minimum Operational Performance Standards for Unmanned Aerial Systems, RTCA Special Committee 228, 2017.

[3] C. Muñoz, A. Narkawicz, G. Hagen, J. Upchurch, A. Dutle, M. Consiglio, J. Chamberlain, "DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems," *in 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC),* IEEE, 2015, pp. 5A1-1-5A1-12

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[5] Julian, K. D., Kochenderfer, M. J., Owen, M. P. "Deep neural network compression for aircraft collision avoidance systems" *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 3, pp. 598–608, 2019.

[6] K. Park and J. Kim, "Aircraft Collision Avoidance Modeling and Optimization Using Deep Reinforcement Learning" *Institute of Control, Robotics and Systems*, vol. 27, no. 9, pp. 652-659, 2021.

[7] S.-H. Park, H. Lee, and H.-T. Lee, "Research on pilot decision model for the fast-time simulation of uas operation," *The Journal of Korea Navigation Institute*, vol. 25, no. 1, pp. 1–7, 2021.

[8] M. Consiglio, J. Chamberlain, C. Muñoz, K. Hoffler, "Concept of integration for UAS operations in the NAS," in *Proceedings of 28th International Congress of the Aeronautical Sciences, ICAS 2012*, Brisbane, Australia, 2012.

[9] J. Ryu, H. Lee, B.-S. Park, and H.-T. Lee "2019 Incheon FIR Aerial Collision Risk Analysis" *The Journal of Korea Navigation Institute*, vol. 25, no. 6, pp. 476-483, 2021.

[10] J. S. Kang, H. J. Oh, K. Y. Choi, and H. T. Lee, "Development and validation of an improved 5-DOF aircraft dynamic model for air traffic control simulation," *The Journal of Korea Navigation Institute*, vol. 20, no. 5, pp. 387–393, Oct. 2016.