



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

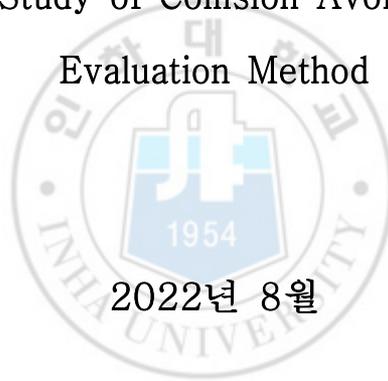
이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

공학석사학위 논문

머신러닝을 이용한 충돌 회피 알고리즘 개발 및 충돌 회피
알고리즘 평가 방법 연구

Constructing Collision Avoidance Algorithm Using Machine
Learning and Study of Collision Avoidance Algorithm
Evaluation Method



2022년 8월

인하대학교 대학원

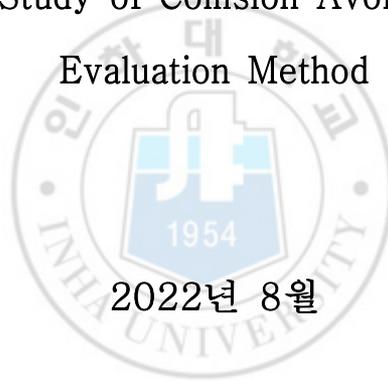
항공우주공학과(항공우주공학전공)

류재영

공학석사학위 논문

머신러닝을 이용한 충돌 회피 알고리즘 개발 및 충돌 회피
알고리즘 평가 방법 연구

Constructing Collision Avoidance Algorithm Using Machine
Learning and Study of Collision Avoidance Algorithm
Evaluation Method



2022년 8월

지도교수 이학태

이 논문을 석사학위 논문으로 제출함

이 논문을 류재영의 석사학위논문으로 인정함.

2022년 7 월 7 일



주심 김충한 (인)

부심 이학태 (인)

위원 유창경 (인)

초록

최근 무인 항공기 UAV (Unmanned aerial Vehicle)의 기술 발전과 더불어 활용이 증대되고, 비행 수요가 증가하고 있다. 이에 따라 유인 항공기 공역 내의 무인 항공기를 운용하고자 하는 수요가 증가하는 추세이며, 무인 항공기와 유인 항공기의 공중 충돌을 예방하기 위한 무인 항공기의 공중 충돌 회피 시스템에 대한 관심이 커지고 있다.

과거부터 무인 항공기의 공중 충돌 회피 시스템은 기존 영공에서 유인 항공기와 원격조종항공기시스템 (RPAS)의 통합 운영을 가능하게 하기 위해 연구되었다. 대표적으로 RTCA DO-365 Minimum Operational Performances Standards (MOPS)에 제시된 Detect and Avoid (DAA)가 있으며, 이를 National Aeronautics and Space Administration (NASA)에서 시스템으로 구현한 Detect and Avoid Logic for Unmanned Systems (DAIDALUS)가 존재한다.

본 논문에서는 DAIDALUS의 결과를 경험으로 하는 머신러닝을 이용하여 충돌 회피 알고리즘 시스템을 구현한다. 국내 공역에서 DAA Well-Clear (DWC) 위반이 일어난 ADS-B data와 MOPS에서 제시한 test vector를 input으로 하고, 각각 데이터의 DAIDALUS 결과를 output으로 하여 dataset을 구성하였으며 해당 dataset을 학습 시키는 방식으로 충돌 회피 알고리즘을 구현한다.

충돌 회피 알고리즘을 평가하기 위해, 항공기 운동 모델과, 항공기 의사 결정 모델을 이용하여 충돌 회피 시뮬레이션을 진행하였다. 항공기의 침입 시나리오는 test vector 중 고도와 속도를 변화시켜 생성하였으며 intruder가 두 개인 경우도 추가하였다. 충돌 회피 알고리즘은 연료 소모량과 DWC 위반이 일어난 시간으로 효율성과 위험도를 평가하였으며 연료소모량이 적을수록 효율적인 기동을 하였다고 판단하고 DWC 위반이 일어난 시간이 적을수록 충돌 위험도가 적었다고 판단하였다.

핵심어 : 충돌 회피 알고리즘, ADS-B, 머신러닝, 시뮬레이션, Detect and Avoid(DAA)

Abstract

Recently, with the technological development of unmanned aerial vehicle (UAV), the use of UAV and the demand for flight are increasing. Accordingly, the demand for operating an unmanned aerial vehicle within the airspace of a manned aircraft is increasing, and interest in an aerial collision avoidance system of UAV to prevent an aerial collision between an unmanned aerial vehicle and a manned aircraft has increased.

The aerial collision avoidance system of unmanned aerial vehicles has been studied to enable the integrated operation of the Remotely Piloted Aircraft System (RPAS) in the airspace with manned aircraft from the past. Representatively, there is a Detect and Avoid (DAA) presented in RTCA DO-365 Minimum Operational Performances Standards (MOPS), and Detect and Avoid Logic for Unmanned Systems (DAIDALUS) which implemented DAA as a system developed by National Aeronautics and Space Administration.

In this paper, implement a collision avoidance algorithm system using machine learning based on the results of DAIDALUS. The output of DAIDALUS was constructed using the ADS-B data in domestic airspace and the test vector presented in the MOPS. Collision avoidance algorithm was implemented by training the dataset in machine learning.

To evaluate the collision avoidance algorithm, a collision avoidance simulation based on the five Degrees-of-Freedoms aircraft dynamics model and pilot decision model was performed. Scenarios were constructed by changing the altitude and speed among the test vector, and the case of two intruders were also added. The collision avoidance algorithm was evaluated by the fuel consumption

and the time the DWC alerts sounded. The lower the fuel consumption, the more efficient the maneuver was. The shorter the time the DWC alerts sounded, the lower the risk of collision.

Key words : Collision Avoidance System, ADS-B, Machine Learning, Simulation, Detect and Avoid(DAA)



차례

제 1 장. 서론	1
1.1 연구 배경	1
1.2 관련 연구	2
1.2.1 지도 학습을 통한 충돌 회피 알고리즘	2
1.2.2 심층 강화 학습을 통한 충돌 회피 알고리즘	4
제 2 장. 머신러닝을 통한 충돌 회피 알고리즘	5
2.1 DAIDALUS	5
2.1.1 DWC(Detect and Avoid Well Clear)	5
2.1.2 DAIDALUS	9
2.2 머신러닝 기법	14
2.2.1 심층 신경망	14
2.3 Dataset 구성	18
2.3.1 ADS-B	19
2.3.2 Test Vector	21
제 3 장. 충돌 회피 시뮬레이션	26
3.1 항공기 운동 모델	26
3.1.1 주요 항공기 parameter 계산	26
3.1.2 BADA를 이용한 항공기 parameter 계산	29
3.1.3 충돌 회피 시뮬레이션 UI	31
3.2 항공기 의사 결정 모델	33

3.2.1 DAA 의사 결정 모델 (DAIDALUS 모드)	33
3.2.2 DAA AI 의사 결정 모델 (AI 모드)	36
제 4 장. 충돌 회피 알고리즘 결과	38
4.1 충돌 회피 알고리즘 평가 방법	38
4.2 시뮬레이션 결과	39
4.2.1. Converge 1 결과	39
4.2.2. Maneuver 6 결과	43
4.2.3. 2 Intruder 1 결과	47
4.2.4. 그 외의 시나리오 결과	51
제 5 장. 결론	52
제 6 장. 참고 문헌	53

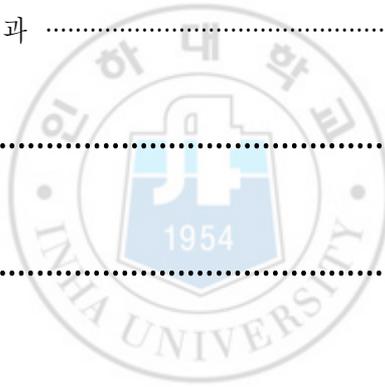


그림 차례

그림 1. ACAS-XU Table	3
그림 2. Neural Network result	3
그림 3. 유지 명령이 내려온 시뮬레이션 결과(PPO) [7]	4
그림 4. DWC 정의	6
그림 5. DAIDALUS Track conflict Bands [4]	10
그림 6. Maneuver Guidance Bands [4]	12
그림 7. DAIDALUS 구상도 [4]	12
그림 8. 심층 신경망	15
그림 9. 오차역전파법(Back propagation)	17
그림 10. FlighAware ADS-B data	19
그림 11. DWC Phase 1 위반이 일어난 국내 공역 [9]	21
그림 12. Test vector Converge 1	23
그림 13. Test vector Dynamic 1	23
그림 14. Test vector Head On 1	24
그림 15. Test vector High Speed 1	24
그림 16. Test vector Manuever 1	25
그림 17. Test vector Over Taking 1	25
그림 18. 충돌 회피 시뮬레이션 구상도	26
그림 19. 속도와 고도에 따른 PID 제어기 [10]	29
그림 20. 뱅크각에 따른 P 제어기 [10]	29
그림 21. 항공기 UI	32
그림 22. 항공기 DAA Viewer	32
그림 23. DAIDALUS 모드 구상도	35

그림 24. AI 모드 구상도	37
그림 25. DAIDALUS mode Alert level for Converge 1	40
그림 26. AI mode Alert level for Converge 1	40
그림 27. DAIDALUS mode 2-D trajectory for Converge 1	41
그림 28. AI mode 2-D trajectory for Converge 1	41
그림 29. DAIDALUS mode 3-D trajectory for Converge 1	42
그림 30. AI mode 3-D trajectory for Converge 1	42
그림 31. DAIDALUS mode Alert level for Maneuver 6	44
그림 32. AI mode Alert level for Maneuver 6	44
그림 33. DAIDALUS mode 2-D trajectory for Maneuver 6	45
그림 34. AI mode 2-D trajectory for Maneuver 6	45
그림 35. DAIDALUS mode 3-D trajectory for Maneuver 6	46
그림 36. AI mode 3-D trajectory for Maneuver 6	46
그림 37. DAIDALUS mode Alert level for 2 Intruder 1	48
그림 38. AI mode Alert level for 2 Intruder 1	48
그림 39. DAIDALUS mode 2-D trajectory for 2 Intruder 1	49
그림 40. AI mode 2-D trajectory for 2 Intruder 1	49
그림 41. DAIDALUS mode 3-D trajectory for 2 Intruder 1	50
그림 42. AI mode 3-D trajectory for 2 Intruder 1	50

표 차례

표 1. DWC thresholds 정의	8
표 2. DWC Phase 1 thresholds	8
표 3. DWC Phase 2 thresholds	9
표 4. Daidalus configurable parameter [4]	13
표 5. Layer unit	15
표 6. 심층신경망 Epoch와 Learning rate	18
표 7. Lambert Conic Conformal Projection 지표	20
표 8. 시나리오별 추가 연료소모량 및 DWC 위반이 일어난 시간	51



제 1 장. 서론

1.1 연구 배경

무인 항공기는 초기에 군사용 목적으로 연구되었지만 최근 비약적인 발달과, Traffic Collision Avoidance System, Automatic Dependent Surveillance-Broadcast (ADS-B) 등과 같은 항공 전자 장비에 발달로 인해 민간 분야까지 활용도를 확대하고 있다. 감시, 재난 구호 지원, 산불 감시, 통신 중계 등에 무인 항공기가 사용되고 있으며 향후 드론 택시나 전기 동력 수직 이착륙기 같은 미래 교통 수단으로 사용될 것으로 예상된다.

무인 항공기의 개발과 더불어 무인 항공기의 비행 공역 범위도 다양해지면서 유인 항공기와 무인 항공기의 통합 운용에 대한 관심도 커지고 있다. The International Civil Aviation Organization (ICAO)에서는 무인 항공기 시스템을 원격조종항공시스템 (RPAS)와 레크레이션을 목적으로 하는 model aircraft, 조종사의 개입을 허용하지 않는 autonomous aircraft로 나누었다. ICAO는 유인 항공기가 있는 공역에서 RPAS 통합 운용을 위한 표준화 작업을 선도하고 있으며 [1] 국내에서도 관련 연구가 진행되고 있다 [2]. National Aeronautics and Space Administration (NASA)에서는 유인 항공기와 무인 항공기의 통합 운용을 위해 RTCA DO-365 Minimum Operational Performances Standards (MOPS) [3]에서 제시한 Detect and Aoid (DAA)를 소프트웨어화하여 Detect and Aoid Alerting Logic for Unmanned System (DAIDALUS) [4]를 개발하였으며 European Union Aviation Safety Agency (EASA)에서는 MIDCAS 프로젝트를 진행하였다.

한편 무인 항공기와 유인 항공기가 같은 공역을 쓰게 될 경우, 국내 공역의 복잡도는 당연히 늘어나게 될 것이다. 이는 관제사의 과부하로 연결될 수 있는

며 심각할 경우, 관제사가 영똥한 충돌 회피 지시를 내릴 가능성이 존재하기 때문에 AI 관제사 모델과 시뮬레이션에 사용할 AI 조종사 모델이 필요하다.

본 연구에서는 머신러닝 중 심층 신경망 (DNN) [5]의 지도 학습을 통한 충돌 회피 알고리즘을 제안한다. 심층 신경망은 dataset의 구성에 따라 다양한 알고리즘 구성이 가능하며 본 논문에서는 DAIDALUS의 결과를 output으로 하여 충돌 회피 알고리즘을 개발하였다. 향후 조종사나 관제사의 지시에 따른 dataset 구성이 가능하다면 AI 관제사와 조종사 모델을 만들 수 있을 것으로 예상된다.

1.2 관련 연구

1.2.1 지도 학습을 통한 충돌 회피 알고리즘 [6]

지도 학습을 통해 충돌 회피 알고리즘을 개발한 연구는 과거에 진행된 바 있으며 해당 연구에서 dataset으로 ACAS-Xu table을 이용하였다. ACAS-Xu는 partially observable Markov Decision Process를 이용하여 충돌 회피 문제를 공식화한 ACAS-X의 하나의 종류이다. ACAS-X에는 유인 항공기 버전인 ACAS-Xa와 무인 항공기 버전인 ACAS-Xu로 나뉘며 ACAS-Xu는 충돌을 회피하면서 방해물 최소화하기 위한 수평 또는 수직 방향으로 해결책을 제시한다. 수평 방향으로는 Clear-of-Conflict (CoC)와 왼쪽, 오른쪽으로 회피할 때의 구체적인 heading rate를 제시하며 heading rate는 1.5 deg/sec (weak), 3.0 deg/sec (strong)로 구성된다. 해당 연구에서 input으로 항공기 사이의 거리, intruder와 ownship 서로 간의 상대 각도와 ownship의 속력, intruder의 속력, 수직 분리까지 남은 시간, 현재 명령값이 들어간다. output은 CoC, weak left, weak right, strong left, strong right의 score value로 구성된다. Loss function으로는 비대칭 MSE function을 사용하였으며, 6개의

hidden layer를 가지고 optimizer로 AdaMax를 사용하였다. 그림 1은 수직 분리 위반까지 남은 시간이 0일 때의 ACAS-Xu table의 값이며 그림 2는 Neural Network의 결과값이다.

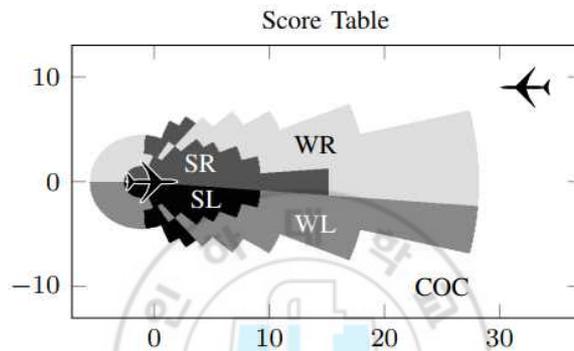


그림 1. ACAS-XU Table

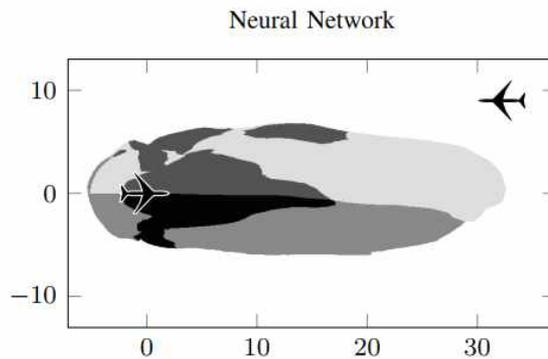


그림 2. Neural Network result

제 2 장. 머신러닝을 통한 충돌 회피 알고리즘

본 연구에서는 머신러닝 기법 중 심층 신경망 [5]의 지도 학습 기법을 이용하여 충돌 회피 알고리즘을 구성하였으며 dataset은 DAIDALUS 결과값을 이용하여 output을 구성하였다.

2.1 DAIDALUS [4]

DAIDALUS는 NASA에서 개발한 소프트웨어이다. 민간 영공에서의 UAS 통합 운영을 지원하기 위해 MOPS [3]를 기반으로 하는 DAA 알고리즘을 소프트웨어로 구현화 하였다.

2.1.1 DWC (DAA Well Clear)

유인 항공기에는 최종 충돌 회피를 조종사의 시각에 의존하여 방지하는 규정을 Title 14 of the Code of Federal Regulations part 91의 의거하여 만들었다. 이는 날씨 상태에 한하여 계기비행이나 시계비행에 관계 없이 항공기 조종사가 다른 항공기를 보고 피할 수 있도록 경계를 유지해야 하는 규정이다. 이에 상반되는 개념으로 MOPS [3]에는 무인 항공기가 주변 항공기들과의 적절한 분리를 유지할 수 있게 하는 Detect and Avoid (DAA) 시스템이 제시되어 있다. DAA 시스템은 원격조종항공기와 유인 항공기의 통합 운용을 가능하게 하는 것을 목적으로 만들어졌으며 UAS 운영자에게 명확한 well clear 상태에 대한 인식을 제공하기 위해 well clear의 수학적 정의와 well-clear boundary에 대한 알고리즘이 포함되어 있다. Well-clear boundary는 well-clear volume (WCV)의 경계로 정의하며, WCV에 항공기가 쌓으로 존

재할 경우 well-clear violation이 일어난 것으로 간주한다.

초기 well-clear violation을 방지하기 위한 Detect and Avoid Well Clear (DWC)의 정의는 항로 상의 비행 중인 항공기들을 대상으로 정의한 phase 1만이 존재하였으나 이후 2020년에 개정된 DO-365A MOPS [8]에는 phase 2를 재정의하여 접근 관제 구역에 완화된 기준을 제시했다.

그림 4는 DWC를 간략히 그림으로 나타낸 것이다. 그림 4에서 보여지는 수평면상에서 기준 분리 거리 (DMOD) 위반까지 남은 시간을 의미하는 modified tau(τ_{mod}^*), 최소 근접거리시의 수평 분리 거리를 의미하는 horizontal miss distance (HMD), 수직 분리 거리인 vertical separation (d_h), 최소 근접거리까지 남은 시간 (t_{cpa})은 식 2-1부터 2-6을 이용하여 계산할 수 있다.

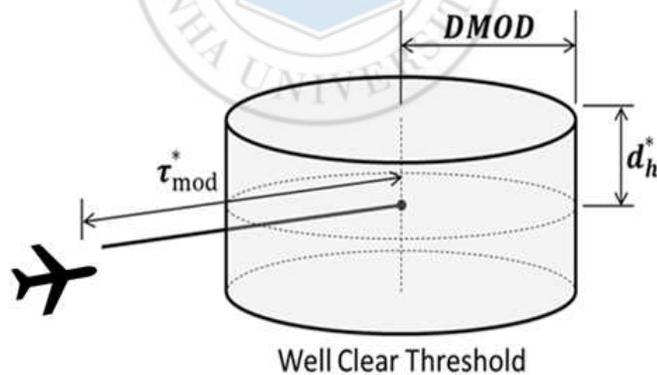


그림 4. DWC 정의

$$r = \sqrt{d_x^2 + d_y^2} \quad (2-1)$$

$$\dot{r} = \frac{d_x \cdot v_{rx} + d_y \cdot v_{ry}}{r} \quad (2-2)$$

$$\tau_{\text{mod}} = \frac{-(r^2 - DMOD^2)}{r\dot{r}} = \frac{DMOD - r^2}{d_x v_{rx} + d_y v_{ry}} \quad (2-3)$$

$$t_{cpa} = \frac{d_x \cdot v_{rx} + d_y \cdot v_{ry}}{v_{rx}^2 + v_{ry}^2} \quad (2-4)$$

$$HMD = \begin{cases} \sqrt{(d_x + v_{rx} t_{cpa})^2 + (d_y + v_{ry} t_{cpa})^2} & \text{for } t_{cpa} \geq 0 \\ -\infty & \text{for } t_{cpa} < 0 \end{cases} \quad (2-5)$$

$$d_h = h_2 - h_1 \quad (2-6)$$

위 식에서 $d_x = x_2 - x_1$, $d_y = y_2 - y_1$ 을 의미하며 $v_{rx} = \dot{x}_2 - \dot{x}_1$, $v_{ry} = \dot{y}_2 - \dot{y}_1$ 을 의미한다. 만일 표 2의 Time of Alert 시간 내의 계산된 τ_{mod} , HMD, d_h 가 표 2에서 제시하는 값을 기준으로 표 1의 조건을 만족하는 경우 phase 1에 해당 위험 단계를 가지게 된다. 표 3의 Time of Alert 시간 내의 계산된 파라미터가 표 3에서 제시하는 값을 기준으로 표 1의 조건을 만족하는 경우 phase 2에 해당 위험 단계를 가지게 된다.

표 1. DWC thresholds 정의

Thresholds	Constraint
τ_{mod}^* (Modified Tau Threshold)	$0 \leq \tau_{mod} \leq \tau_{mod}^*$
HMD^* (Horizontal Miss Distance Threshold)	$HMD \leq HMD^*$
d_h^* (Vertical Separation Threshold)	$-d_h^* \leq d_h \leq d_h^*$

표 2. DWC Phase 1 thresholds

Threshold	Preventive	Corrective	Warning	Loss of Well Clear
τ_{mod}^*	35 sec	35 sec	35 sec	35 sec
HMD^*	4,000 ft	4,000 ft	4,000 ft	4,000 ft
d_h^*	700 ft	450 ft	450 ft	450 ft
Time of Alert	55 sec	55 sec	25 sec	0 sec

표 3. DWC Phase 2 thresholds

Threshold	Preventive	Corrective	Warning	Loss of Well Clear
τ_{mod}^*	-	-	0 sec	0 sec
HMD^*	-	-	1,500 ft	1,500 ft
d_h^*	-	-	450 ft	450 ft
Time of Alert	-	-	55 sec	0 sec

2.1.2 DAIDALUS [4]

DAIDALUS는 DAA 알고리즘을 소프트웨어로 구현한 것으로 UAS 운영자의 well clear 유지를 목표로 하며 well-clear violation이 일어나거나 well-clear violation을 피할 수 없는 경우, well clear를 회복시키는 것을 목표로 한다. DAIDALUS가 제공하는 알고리즘은 크게 detection, determine processing function, alerting logic으로 나눌 수 있다.

Detection은 well-clear violation이 일어날 때까지의 시간을 계산하며 계산 시, 항공기들이 같은 속도를 유지한다고 가정한 상태에서 선형 투영으로 계산한다. Determine processing function은 ownship이 well-clear violation을 유발할 수 있는 ownship의 기동 범위를 conflict bands 형식으로 제공하며, well-clear violation이 일어나거나, 피할 수 없는 경우 well clear로 회복시키는 ownship의 기동 범위를 recovery bands 형식으로 제시한다. 계산 시에는 ownship의 일정한 turn rate, acceleration, velocity를 투영하여 계산한다. conflict bands와 recovery bands는 track angle 범위, ground speed 범위, vertical speed 범위 등 세 가지 타입으로 제시된다. Alerting logic은 현재 상

응하는 alert type을 제시하며 이때 WCV와 alert의 기준값은 미리 설정해 놓을 수 있다. 기본적인 default로는 [3]에서 제시한 DWC를 사용한다.

그림 5는 track conflict bands 계산 시 ownship과 intruder의 상태를 도식화한 것이다. α 는 ownship의 오른쪽으로 회전할 수 있는 한계각을 의미하며 β 는 왼쪽으로 회전할 수 있는 한계각을 의미한다. γ 는 well-clear violation을 유발하는 각도를 의미한다. 음영으로 표현된 부분은 well-clear violation이 일어나는 구역을 의미한다. DAIDALUS는 해당 상황에서 γ 를 제공한다.

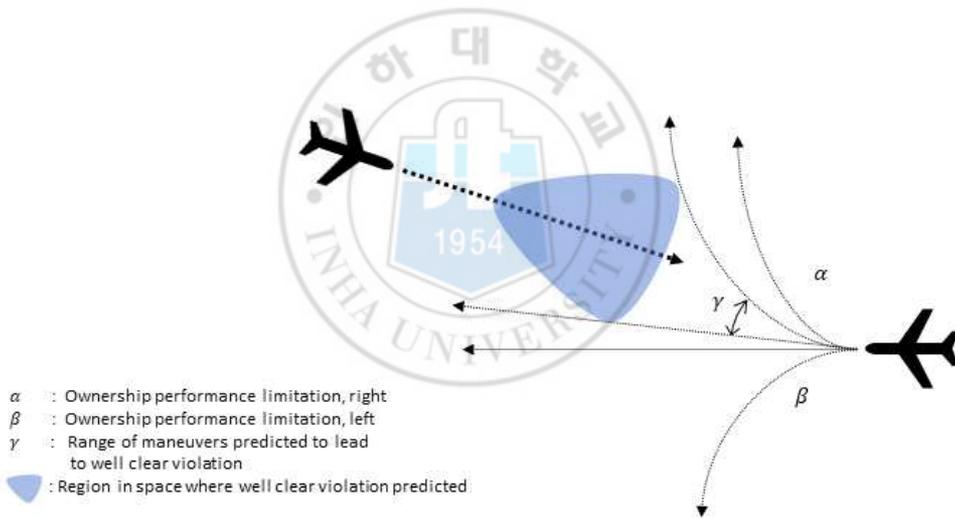


그림 5. DAIDALUS Track conflict Bands [4]

DAIDALUS는 conflict bands를 preventive 상황이거나 corrective 상황에 제시한다. 일정 시간 내에 well-clear violation이 일어나지 않지만 ownship의 기동이 well-clear violation을 일으킬 수 있을 경우 preventive한 bands를 제시하며, 현재 기동이 일정 시간 내에 well-clear violation를 유발할 경우 corrective한 bands를 제시한다. well-clear violation을 의미하는 Loss of well clear (LoWC)가 일어나거나 피할 수 없는 경우 DAIDALUS는 recovery bands를 제시하며, 이는 ownship의 기동 내에서 가장 빠른 시간내에 well clear로 회복 할 수 있는 기동 명령으로 구성된다.

그림 6은 상황에 따라 제공 받는 bands를 가시화한 것으로, 바깥쪽 원은 Self Separation Threshold (SST)를 나타내며 이는 determine processing function과 직접적인 연관이 있는 구역이다. 회색의 원은 ownship의 well clear violation을 일으키는 영역 (WCV)를 나타낸다.

DAIDALUS는 $t = t_0$ 인 경우, intruder가 SST 구역 밖에 있기 때문에 아무런 bands도 제시하지 않는다. $t = t_1$ 인 경우, intruder가 SST 구역 안으로 들어왔으나 현재 ownship의 진행 방향이 well-clear violation을 일으키지 않기 때문에 이를 유발할 수 있는 track 범위를 preventive한 conflict bands로 제시한다. $t = t_2$ 인 경우, 현재 ownship의 진행 방향이 well-clear violation을 유발하기 때문에 conflict bands는 preventive에서 corrective로 변화하며 이를 유발하는 track 범위를 제시한다. $t = t_3$ 일 경우 intruder가 ownship에 WCV 내로 들어온 것을 볼 수 있으며 UAS 운용자가 ownship의 기동 범위 내에서 well clear 상태로 회복할 수 있도록 recovery bands를 제시한다. 노랑색으로 표현된 부분은 conflict bands를 의미하며 초록색으로 표현된 부분은 recovery bands를 의미한다.

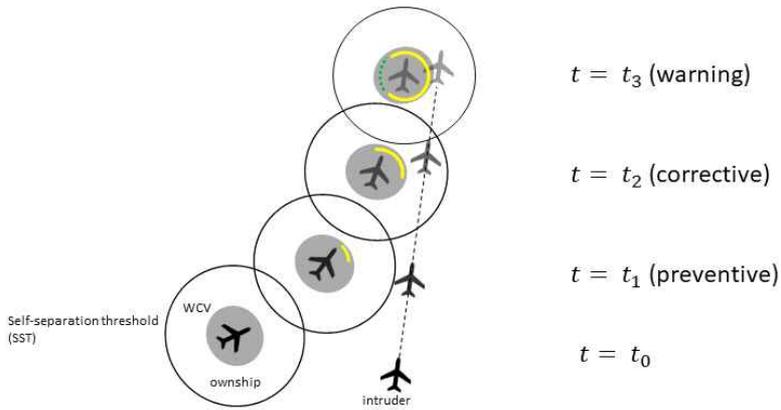


그림 6 Maneuver Guidance Bands [4]

그림 7는 DAIDALUS 알고리즘의 input과 output으로 들어가는 값을 표현하였다. 분리 경보 파라미터로는 MOPS [3]에서 제시한 Phase 1 값이 기본값으로 사용된다. DAIDALUS에서 사용하는 기본 항공기 파라미터는 표 4에 나타내었으며, 분리 경보 파라미터와 기본 항공기 파라미터는 변경이 가능하다.

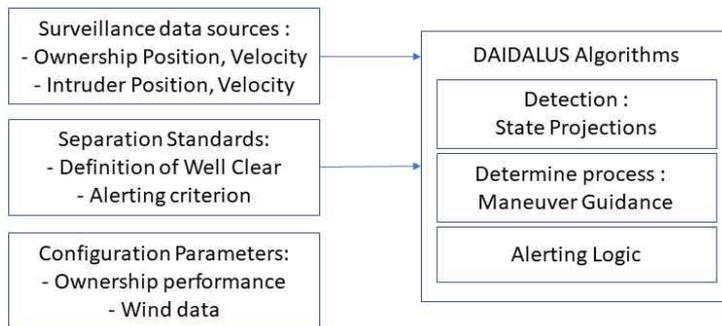


그림 7. DAIDALUS 구성도 [4]

표 4. Daidalus configurable parameter [4]

Parameter	Default value
Turn rate	3 deg/s
Bank angle	30 deg
Horizontal acceleration	2 m/s ²
Vertical acceleration	2 m/s ²
Min / Max ground speed	0 knots / 700 knots
Min / Max vertical speed	-5000 fpm / 5000 fpm
Track/ Ground/ Vertical speed step	1 deg / 1 knot / 10 fpm

DAIDALUS의 version으로는 JAVA 2.0.2 version을 사용하였으며 해당 version에서 conflict bands 같은 경우 track, altitude, speed 범위가 동시에 제시된다. 본 연구에서는 수평 기동만을 고려하였기 때문에, DAIDALUS에서 제시하는 conflict bands 중, track bands로 제시되는 왼쪽 범위 t_L , 오른쪽 범위 t_R 를 이용하였으며 더불어 alert level을 이용하여 output을 구성하였다. t_L 과 t_R 은 0-360도 까지의 각도로 제시된다. Alert level은 1, 2, 3, 4로 제시되며 이는 각각 preventive alert, corrective alert, warning alert, LoWC를 의미한다. 해당 version에서는 위험도가 없을 경우 Not a Number (NaN) 값이 제시되며, resolution 값이 존재하지 않을 경우 Infinity 값이 제시된다.

2.2 머신러닝 기법

지도 학습은 (x, y) , $x \in X$, $y \in Y$ 들의 집합이 주어졌을 때, $f: X \rightarrow Y$ 들의 목록 중에 가장 적합한 함수를 찾는 것을 목적으로 한다. 본 논문에서는 충돌 회피 알고리즘을 심층 신경망을 이용한 지도 학습으로 구성하였다.

2.2.1 심층 신경망 (DNN) [5]

심층 신경망 (DNN)은 입력값과 출력값 사이에 여러 층의 인공 신경망으로 이루어진 신경망이다. 인공 신경망은 기계학습과 생물학의 신경망에서 영감을 얻은 통계학적 알고리즘이다. 각각의 노드가 인간의 뉴런 역할을 하며 가중치가 시냅스의 역할을 하고 그림 8과 같은 구조를 가진다. 노드에 입력된 값이 가중치를 통과하고, 다음 층에 전달되기 전에 비선형 함수를 통과하는데 이 함수를 활성화 함수(Activation Function)이라고 한다. 선형함수만 사용할 경우 인공 신경망을 깊게 통과하는 의미가 줄어들기 때문에 활성화 함수를 사용하게 된다. 본 연구에서는 ReLU 함수를 활성화 함수로 사용하였으며 그 식은 2-7과 같다.

$$ReLU(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases} \quad (2-7)$$

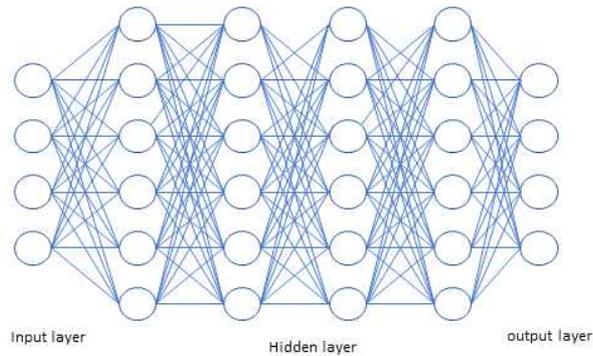


그림 8. 심층 신경망

DNN의 hidden layer의 수는 가중치의 수와 직접적으로 연결된다. 가중치가 많다고 학습이 적절하게 진행되지 않으며, 오히려 가중치 수가 많을수록 학습이 진행되는 시간이 오래 걸리게 된다. 반면에 가중치 수가 적으면 학습이 제대로 진행되지 않기 때문에 적절한 가중치 수를 찾는 것이 중요하다. 본 연구에서는 초기에 hidden layer의 수로 네 개로 설정하였으며 layer의 수를 늘려가며 정확도를 비교하였다. Layer의 수가 7개일 경우와 6개일 경우를 비교하였을 때, 정확도의 많은 차이가 없었기 때문에 학습 속도를 고려하여 6개의 hidden layer를 이용하였다. 총 21,000개의 파라미터가 사용되었으며 layer의 unit은 표 5와 같다.

표 5. Layer unit

Input	Hidden 1	Hidden 1	Hidden 1
10	64	64	64
Hidden 1	Hidden 1	Hidden 1	Output
64	64	64	3

심층 신경망의 input value는 DAIDALUS와 마찬가지로 ownship의 고도, ground speed, vertical rate, heading과 intruder의 고도, ground speed, vertical rate, heading을 가지도록 구성하였다. 하지만 항공기 각각의 위도와 경도가 들어가는 DAIDALUS와는 다르게 ownship과 intruder의 상대 위도와 경도가 input 파라미터로 들어가게 구성하였는데 이는 같은 조우 상황을 가지더라도 다른 위치에서 일어나는 경우가 있기 때문이다. 합계 10개의 파라미터가 input value로 들어가게 된다.

심층 신경망의 output value는 DAIDALUS에서 제시하는 track bands의 왼쪽 범위 (t_L)과 오른쪽 범위 (t_R), alert level (w)로 구성된다. DAIDALUS가 NaN 값을 제시할 경우, -1값이 output으로 대신 들어가게 된다. DAIDALUS가 Infinity 값을 제시할 경우, 이는 해당 방향으로 회피할 수 없는 것을 의미하기 때문에 현재 항공기 heading $+180^\circ$ 의 값이 output으로 대신 들어가게 된다. 심층 신경망의 계산시, output value의 값이 유사할 필요가 있기 때문에 식 2-8과 같이 scaling이 진행되었다.

$$Y = [10w, \frac{t_L}{10}, \frac{t_R}{10}] \quad (2-8)$$

손실 함수는 입력값 X 를 f 의 함수에 넣었을 경우의 예측값인 \hat{y} 과 본래의 label 값인 y 를 이용하여 만드는 함수이다. 보통 평균 제곱 오차 (MSE)를 손실함수로 많이 사용하지만 크로스 엔트로피 (Cross-Entropy) 손실 함수나 소프트맥스 (Softmax Function) 손실 함수를 사용하기도 한다. 본 논문에서는

batch size가 16인 MSE를 사용하였고 식은 2-9와 같다. DNN은 여러 개의 인공 신경망 가중치로 이루어져 있으며 학습한다는 것은 가중치 값을 조정하는 것을 의미한다. 가중치를 조정하는 방법에는 많은 알고리즘이 존재하지만 본 연구에서 가중치는 손실 함수의 기울기를 이용하여 조정하였다. 손실 함수의 기울기는 연쇄 법칙 (Chain rule)을 이용한 오차역전파법(Back propagation)을 사용하여 계산하였으며 그림 9는 오차역전파법을 나타낸다.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2-9)$$

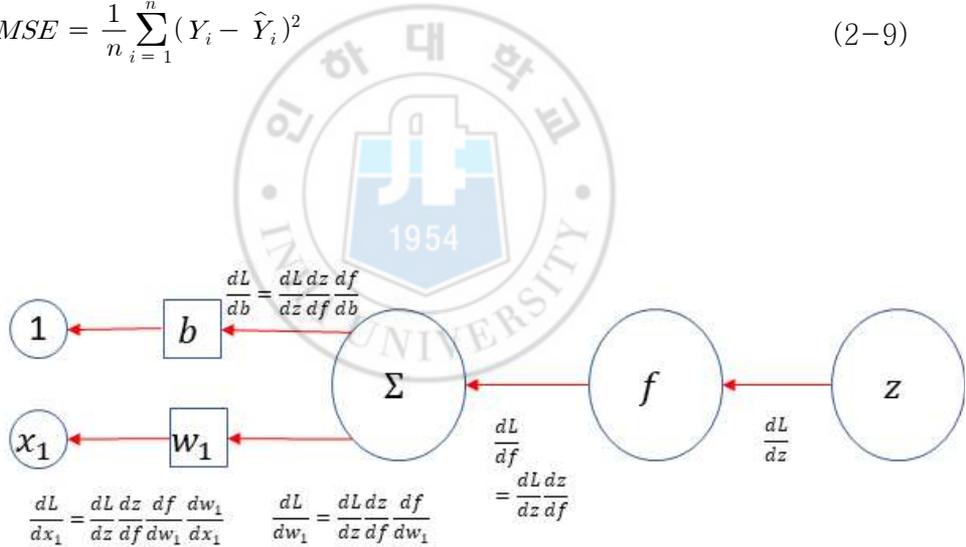


그림 9. 오차역전파법(Back propagation)

Optimizer와 learning rate는 학습 시, 중요한 역할을 한다. Optimizer는 손실 함수의 기울기를 이용하여 가중치를 조절하는데 이용된다. Optimizer에는 여러 종류가 있으며 본 연구에서는 local minimum의 빠지더라도 해결할 수 있

도록 모멘텀을 이용한 경사 하강법을 이용하였다. 모멘텀값은 0.9로 설정하였으며 해당 식은 2-10부터 2-11과 같다. Learning rate는 가중치 조절 시 사용하는 파라미터로, 너무 클 경우 인공 신경망의 가중치의 값이 커지는 이유로 학습이 진행되지 않을 수 있으며, 진행되더라도 오차가 큰 상태에서 더 이상 학습이 진행되지 않을 수 있다. 반면에 너무 작을 경우에는 local minimum에 빠져 학습이 진행되지 않을 수 있으며, 진행되더라도 학습 시 시간이 오래 걸릴 수 있기에 적절한 파라미터로 설정하는 것이 중요하다. 본 연구에서 사용한 Epoch별 Learning rate는 표 6과 같다.

$$v_{ij}(t+1) = \gamma * v_{ij}(t-1) + \eta \frac{dL}{dw_{ij}} \quad (2-10)$$

$$w_{ij}(t+1) = w_{ij}(t) - v_{ij}(t) \quad (2-11)$$

표 6. 심층신경망 Epoch와 Learning rate

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

2.3 Dataset 구성

본 연구에서는 광범위한 항공기들의 조우 상황을 이용하여 dataset을 구성하기 위해 두가지 타입의 dataset을 구성하였다. ADS-B data를 이용하여 국내 공역의 위험도를 판단한 결과 [9]와 MOPS에서 제시한 test vector를 이용하

여 dataset을 구성하였다.

2.3.1 ADS-B

ADS-B 데이터는 항공기의 위도, 경도, 고도 등의 위치 정보와 지상 속도, 상승률, 방위각 등을 담은 메시지를 송출하는 데이터이다. 2020년부터 미국과 유럽의 모든 민간 항공기는 ADS-B 탑재 장비 장착을 의무화 하였으며 점차 의무화 하는 국가가 늘어나고 있다. 그림 10은 FlightAware의 ADS-B data로 2019년 5월 31일 13:00 ~ 6월 2일 07:30 분까지 국내 공역 및 중국과 일본 공역을 통과한 항공기의 궤적이다.



그림 10. FligthAware ADS-B data

본 연구에서는 ADS-B data를 이용하여 인천 Flight Information Region (FIR) 내의 MOPS [3]에서 제시된 DAA 기반 공중 충돌 위험도를 계산한 다음, DWC 위반이 일어난 지점의 데이터를 이용하여 dataset을 구성하였다. ADS-B 데이터는 일정 단위 시간의 비행 데이터로 구성되지만 항공기 상황이 나 지상 수신기의 상황에 따라 데이터를 받지 못하는 경우가 존재한다. 항공기

들의 시간 별 상태에 따라 공중 충돌 위험도 계산이 진행되어야 하기 때문에 시간 동기화 작업이 필요하며, 1초 간격의 시간 동기화를 진행하였다. 하지만 ADS-B 데이터의 위치 정보는 위도와 경도로 구성되기 때문에 해당 데이터로 시간 동기화를 진행하기에는 한계가 존재한다. 이를 해결하기 위해 데이터의 위도와 경도를 상대 거리로 전환하여 시간 동기화를 진행하였으며, 상대 거리로 전환하기 위하여 Lambert Conic Conformal Projection을 이용하였다. 위도와 경도를 지정한 위치로부터의 상대거리 x, y 로 전환하였으며 전환한 x, y 를 선형보간법을 이용하여 1초 간격으로 동기화 하였고 이를 다시 위도와 경도로 전환하였다. 사용한 Lambert Conic Conformal Projection 지표는 표 7에 표현하였다. 해당 지표는 인천 FIR의 위도와 경도를 고려하여 선정하였으며, 지구 반지름은 6400km로 가정하였다.

표 7. Lambert Conic Conformal Projection 지표

Thresh hold	Standard Parrallel 1	Standard Parrallel 2	Origin Position
latitude	33°	38°	33°
longitude	125°	127°	126°

2019년 1월 1일 12:00 AM을 기준으로 데이터 보간을 진행하여 충돌 위험도를 계산하였으며 해당 결과는 [9]에서 확인할 수 있다. 그림 11은 [9]에서 제시한 국내 공역에서 DWC Phase 1 위반이 일어난 지역이다. [9] 결과 중, DWC 위반이 일어난 data의 속도 정보와 위치 정보를 토대로 하여 input을 구성하였으며 해당 data의 DAIDALUS 결과를 토대로 output을 구성하여 dataset을 생성하였다. 약 5600개의 조우상황을 이용하여 30만개(초)의

dataset을 생성하였다.

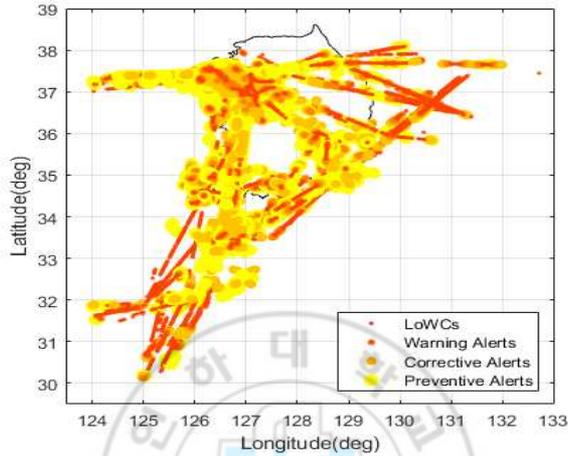


그림 11. DWC Phase 1 위반이 일어난 국내 공역 [9]

2.3.2 Test Vector

MOPS APPENDIX P에는 DAA 시스템의 작동이 원활하게 진행되는지를 확인하기 위해 test vector가 제시되어 있다. Test vector는 구체적인 항공기 궤적으로 ADS-B data 형식을 가지고 있다. 2000년에서 2010년 사이 발생한 공중 충돌 상황, Student Airborne Research Program에서 제시한 95가지 스트레스를 유발하는 Well clear boundaries 상황, NASA에서 수행한 비행 테스트를 토대로 만들어졌으며 총 305개의 test vector가 존재한다. 본 연구에서는 여러 intruder가 존재하는 Designer 상황을 제외한 Converge, Dyanamic, Head on, High Speed, Maneuver, Over taking 상황을 이용하여 dataset을 구성하였으며 그림 12-17는 각각의 상황 중 대표적인 궤적을 제시하였다.

ADS-B data 형식을 가지고 있기 때문에 data의 속도 정보와 위치 정보를 토대로 하여 input을 구성하였으며 해당 data의 DAIDALUS 결과를 토대로 output를 구성하여 dataset을 생성하였다. Ownship과 intruder를 번갈아 가며 580개의 조우 상황, 총 5만개의 dataset을 구성하였다.



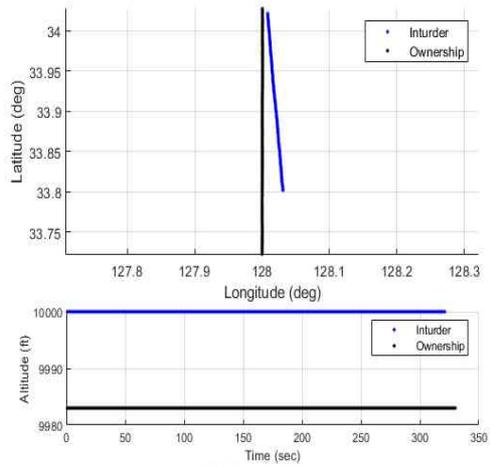


그림 12. Test vector Converge 1

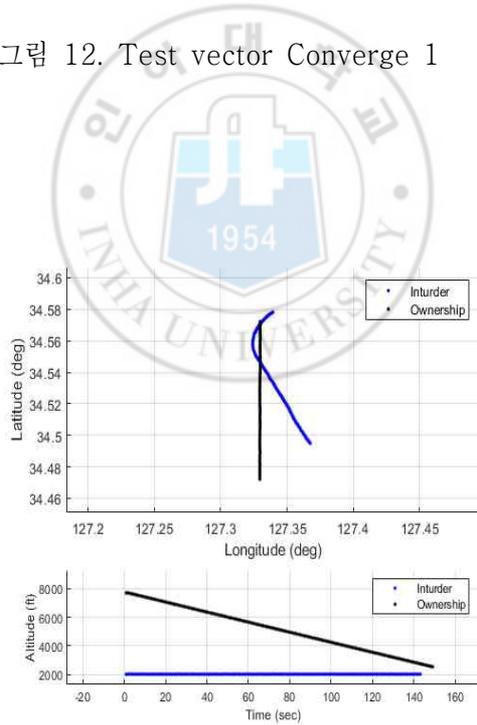


그림 13. Test vector Dynamic 1

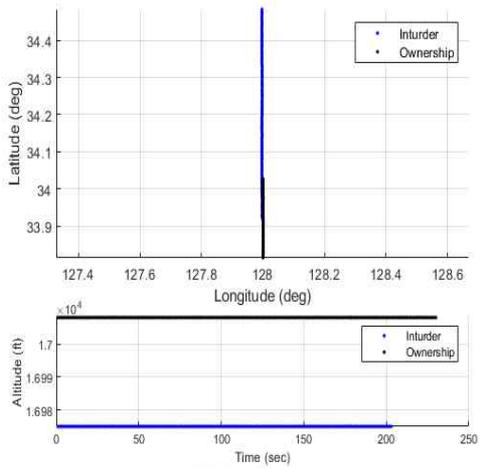


그림 14. Test vector Head On 1

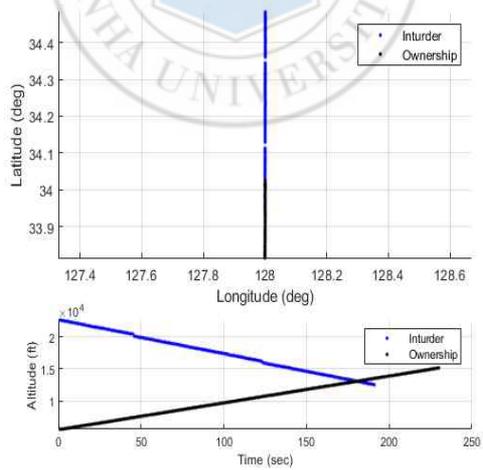


그림 15. Test vector High Speed 1

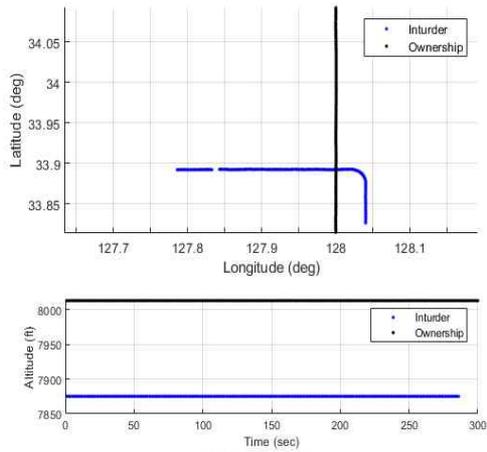


그림 16. Test vector Manuever 1

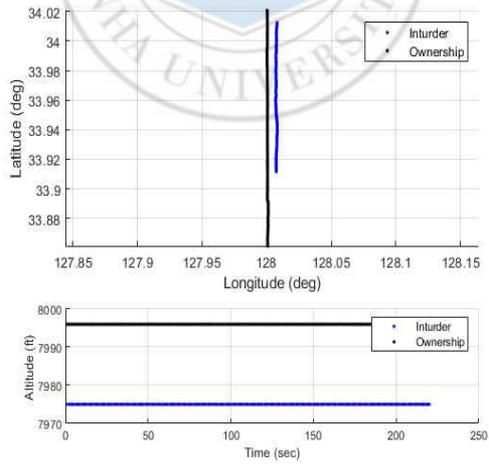


그림 17. Test vector Over Taking 1

제 3 장. 충돌 회피 시뮬레이션

본 연구에서는 충돌 회피 알고리즘을 평가하기 위해 항공기 운동 모델 [10]과 항공기 의사 결정 모델 [11]을 바탕으로 하는 충돌 회피 시뮬레이션을 진행하였다. 그림 18은 충돌 회피 시뮬레이션의 구상도이며 초기 항공기는 Waypoint (WPT)로 가는 것을 목적으로 하며, 기동 명령이 들어올 경우 해당 명령을 따르게 된다.

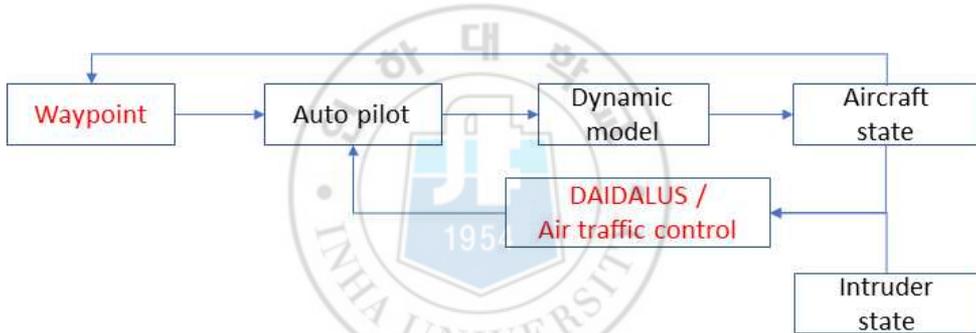


그림 18. 충돌 회피 시뮬레이션 구상도

3.1 항공기 운동 모델 [10]

3.1.1 주요 항공기 parameter 계산

항공기 운동 모델은 5 자유도 운동 방정식과 PID 제어를 통해 구현되며 아래와 같은 parameter를 가진다.

$$X = [\alpha, \phi, T, V, \gamma, \psi, \Phi, \Lambda, h, m]$$

α 는 받음각, ϕ 는 뱅크각, T 는 추력, V 는 항공기 ground speed, γ 는 항공기 경로각, ψ 는 방위각, Φ 는 항공기 위도, Λ 는 항공기 경도, h 는 항공기 고도, m 은 항공기 질량을 의미한다. 해당 parameter는 식 3-1부터 3-10을 통하여 계산된다.

$$\ddot{\alpha} = -2\zeta_{\alpha}\omega_{n_{\alpha}}\dot{\alpha} - \omega_{n_{\alpha}}^2 + \omega_{n_{\alpha}}^2\alpha_{cmd} \quad (3-1)$$

$$\dot{\phi} = \frac{\phi_{cmd} - \phi}{\tau_{\phi}} \quad (3-2)$$

$$\dot{T} = \frac{T_{cmd} - T}{\gamma_T} \quad (3-3)$$

$$\dot{V} = \frac{T\cos\alpha - D}{m} - g\sin\gamma \quad (3-4)$$

$$\dot{\gamma} = \frac{(L + T\sin\alpha)\cos\phi}{mV} - \frac{g}{V}\cos\gamma \quad (3-5)$$

$$\dot{\psi} = \frac{(L + T\sin\alpha)\sin\phi}{mV\cos\gamma} \quad (3-6)$$

$$\dot{\Phi} = \frac{V_N}{R_{\Phi} + h} \quad (3-7)$$

$$\dot{\lambda} = \frac{V_E}{(R_A + h) \cos \Phi} \quad (3-8)$$

$$\dot{h} = V \sin \gamma \quad (3-9)$$

$$\dot{m} = -fuel\ flow \quad (3-10)$$

식 3-1의 ζ_a 와 ω_{n_a} 은 항공기 형상과 동특성 분석을 통해 계산된 값을 사용하였으며 식 3-2의 τ_ϕ 는 항공기의 특성 정보를 통해 얻은 값을 사용하였다. 식 3-3의 τ_T 는 추력 반응 상수로써 제트 엔진의 경우는 감항 기준을 통해 예측된 값을 사용하였다. 항공기 모델은 정해진 궤적을 추종하기 위해 3가지 제어 입력(α_{cmd} , ϕ_{cmd} , T_{cmd})을 사용하였다. 제어기는 항공기 모델을 구성할 때 포함되며 항공기가 주어진 고도, 속도, 헤딩 명령을 추종할 때 사용된다. 비행 구간을 상승, 순항, 하강으로 나누고 각 구간에 맞는 제어기를 따로 구성하였으며, 속도와 고도 명령은 BADA의 비행 구간별 최대 추력 제한에 인해 서로 영향을 주기 때문에 각각의 PID 제어기와 공통의 PI 제어기를 사용하도록 구성하였다. 그림 19는 고도와 속도 제어기를 나타낸다. 헤딩 명령은 P제어기만을 이용하여 구현되며 그림 20와 같은 구조를 가진다.

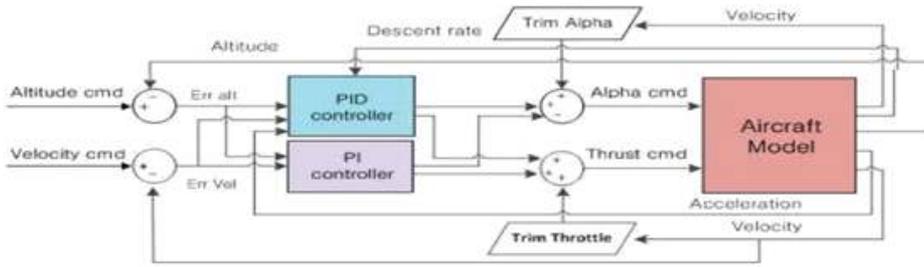


그림 19. 속도와 고도에 따른 PID 제어기 [10]

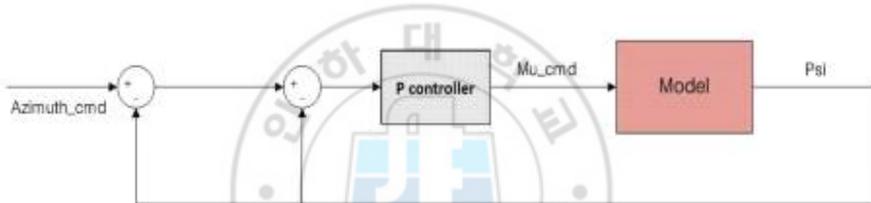


그림 20. 뱅크각에 따른 P 제어기 [10]

3.1.2 BADA를 이용한 항공기 parameter 계산

BADA는 항공기 운용 제한 수치, 항공기 성능 정보, 항공기 제원 등의 값을 제공한다. BADA의 항공기 성능 정보는 질점 모델 기반의 데이터로 구성되어 있으며 비행 구간, 고도에 따른 최대 엔진 추력과 연료 소모율을 계산하는데 사용된다. 최대 엔진 추력과 연료 소모율은 식 3-11부터 3-16을 따르며, 수식을 구성하는 계수들은 BADA의 .OPF 파일에서 얻을 수 있다.

$$T_{\max, climb} = C_{TC,1} \times \left(1 - \frac{h}{C_{TC,2}} + C_{TC,3} \times h^2\right) \quad (3-11)$$

$$T_{\max, cruise} = C_{TCR} \times T_{\max, climb} \quad (3-12)$$

$$T_{\max, descent} = C_{des} \times T_{\max, climb} \quad (3-13)$$

$$\eta_{nominal} = C_{f1} \times \left(1 + \frac{V_{TAS}}{C_{f2}}\right) \times T \quad (3-14)$$

$$\eta_{cruise} = \eta_{nominal} \times T \times C_{fcr} \quad (3-15)$$

$$\eta_{min} = C_{f3} \times \left(1 - \frac{h}{C_{f4}}\right) \times T \quad (3-16)$$

BADA는 질점 모델을 기반으로 데이터를 제시하기 때문에 공력계수가 받음각 함수로 구성되지 않는다. 5자유도 모델을 계산하기 위해 공력계수를 받음각 함수로 표현하였으며 식 3-17, 식 3-18과 같이 계산하였다.

$$C_L = C_{L_0} + C_{L_\alpha} \alpha \quad (3-17)$$

$$C_D = C_{D_0} + C_{D_2} C_L^2 \quad (3-18)$$

식 3-17의 C_{L_0} 는 BADA에서 제시한 순항 조건으로부터 받는 양력 계수를 의미하며 α 는 받음각을 의미한다. 본 연구에서 사용한 항공기 기종은 Boeing 777-300으로 $C_{L_a} = 5.1$ 을 가진다. BADA는 항공기 기종 별 날개 넓이, 질량과 flap 상태에 따른 stall speed, 양력 계수를 제시한다. Flap의 $stall\ speed \times 1.3$ 값을 flap 상태에 따른 최소 속도로 가정하였으며 flap 상태에 따른 양력 계수를 사용하여 양력을 계산하였다. 식 3-18의 C_{D_0}, C_{D_2} 는 BADA에서 제시한 flap 상태에 따른 양력 계수를 의미한다.

3.1.3 충돌 회피 시뮬레이션 UI

그림 21은 충돌 회피 시뮬레이션의 UI 이다. 시뮬레이션의 UI에는 Primary Flight Display가 존재하며 항공기의 현재 속도와 고도, 뱅크각을 가시화하였다. Navigation Display(ND)에서는 항공기의 현재 위치, 기수 방향, 비행 방향, 진로 및 intruder의 위치를 확인할 수 있으며 WPT까지 남은 시간과 거리를 확인할 수 있게 구성하였다.

DAA Viewer는 ND와 마찬가지로 ownship 주변에 intruder의 위치를 확인할 수 있게 구성하였으며, DAIDALUS에서 제시 받는 track resolution의 범위를 alert level에 따라 가시화 하였다. Altitude resolution으로 제시 받는 범위 또한 alert level에 따라 가시화 하였으며 그림 22는 warning alert 상황일 때의 DAA Viewer이다.

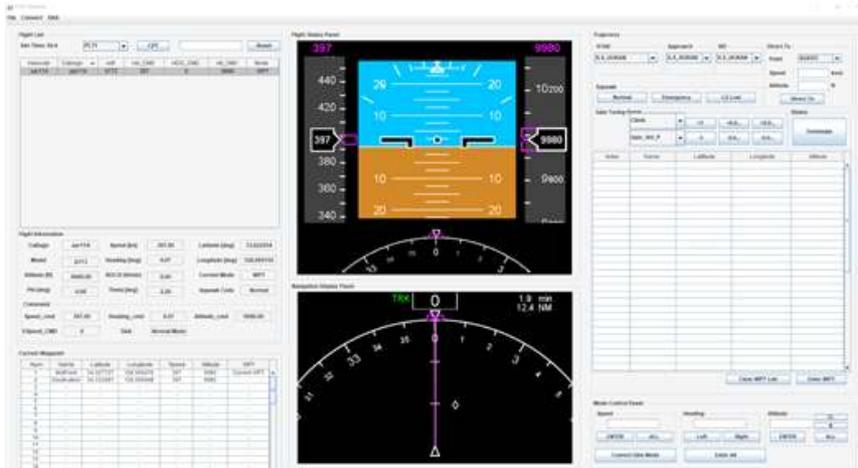


그림 21. 항공기 UI

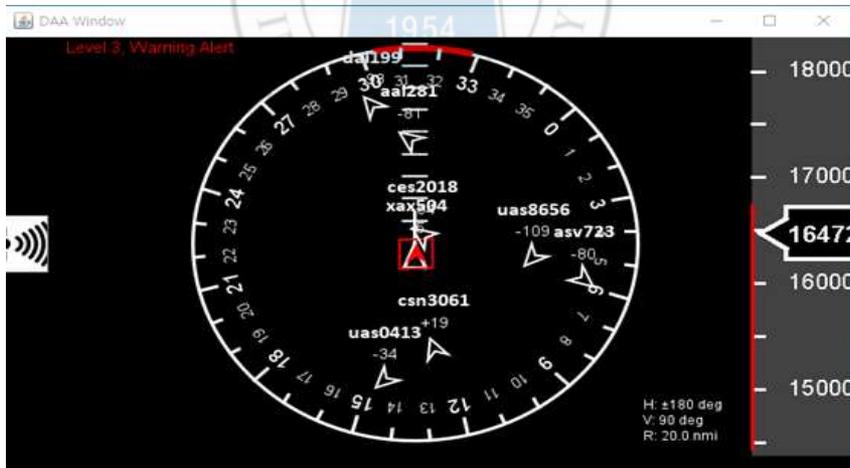


그림 22. 항공기 DAA Viewer

3.2 항공기 의사 결정 모델

DAIDALUS 결과와 달리 머신러닝으로 구현한 DNN 모델은 위험도 레벨과 track resolution만을 결과로 받기에 각각 다른 항공기 의사 결정 모델을 사용할 필요가 있다. DAIDALUS를 이용한 항공기 의사 결정 모델 [11]을 바탕으로 새로운 의사 결정 모델을 구성하였으며 DAIDALUS를 이용한 의사 결정 모델을 DAIDALUS 모드, 새로운 의사 결정 모델을 AI 모드로 명명하도록 한다. 본 연구에서는 DAIDALUS와 DNN 모델의 충돌 회피 알고리즘을 평가하기 위해 DAIDALUS 모드와 AI 모드의 경우에서 모두 시뮬레이션을 진행하였다.

3.2.1 DAA 의사 결정 모델 (DAIDALUS 모드) [11]

초기 ownship은 WPT를 향하여 기동한다. 기동 중 intruder와의 alert level이 corrective alert 이상이 될 경우 항공기는 WPT로 진행하던 기동에서 DAIDALUS가 제시하는 track resolution과 altitude resolution에 따라 기동한다. 기동은 수평 기동 또는 수직 기동 중 한 가지를 선택하며, 속도는 현재 속도를 유지한다.

Track resolution과 altitude resolution이 모두 존재할 경우, 현재 ownship의 heading과 track resolution의 차이가 30도보다 크면 ownship은 track resolution에 따라 수평 기동을 진행한다. 30도 보다 작을 경우 altitude resolution에 따라 수직 기동을 진행한다. Track resolution이 존재하고, altitude resolution이 존재하지 않는 경우, ownship은 track resolution에 따라 수평 기동을 진행한다. DAIDALUS는 오른쪽으로 회피할지, 왼쪽으로 회피할지에 대한 값을 제시하기 때문에 제시된 값으로 회피가 이루어지며 DAIDALUS가 제시하는 최소 근접 거리까지 남은 시간(t_{cpa})까지 ownship은

수평 기동을 유지한다. Altitude resolution이 존재하고, track resolution이 존재하지 않는 경우, ownship은 altitude resolution에 따라 수직 기동을 진행한다. DAIDALUS 같은 경우 위쪽으로 회피할지, 아래쪽으로 회피할지에 대한 값을 제시하기 때문에 제시된 값으로 회피가 이루어지며 항공기가 altitude resolution 조건에 도달할 때까지 수직 기동을 유지한다. 위험도가 corrective 이상이지만 ownship의 track resolution과 altitude resolution이 모두 존재하지 않을 경우, ownship은 현재 상태를 유지한다. DAIDALUS 모드의 전체적인 구상도는 그림 23과 같다.



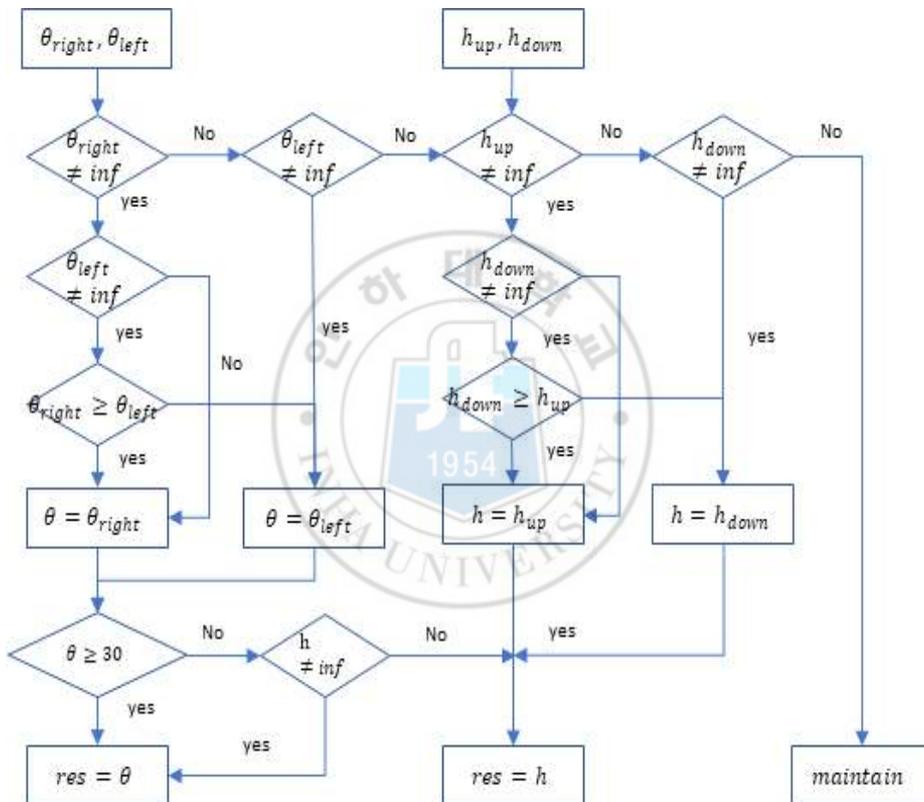


그림 23. DAIDALUS 모드 구상도

3.2.2 DAA AI 의사 결정 모델 (AI 모드)

DAIDALUS 모드와 마찬가지로 초기 ownship은 WPT를 향하여 기동한다. 기동 중 DNN 모델이 제시하는 intruder와의 alert level이 corretive alert 이상이 될 경우 항공기는 WPT로 진행하던 기동에서 DNN 모델이 제시하는 track resolution에 따라 기동한다. 기동은 수평 기동을 하며 [11]의 경우와 유사하게 하기 위해 속도는 현재 속도를 유지한다.

Track resolution은 left값과 right값이 존재할 수 있으며 매 초마다 제시된다. Left값과 right값이 ownship 항공기 방향의 왼쪽과 오른쪽에 존재하지 않는 경우도 존재하는데 해당 사항의 경우 Not a Number (NaN)값을 가진 것으로 판단한다.

Track resolution의 left값과 right값이 동시에 존재할 경우, DAIDALUS와 달리 DNN 모델은 왼쪽과 오른쪽의 우선 순위를 지정해 주지 않기 때문에 현재 ownship의 heading과의 차이가 더 작은 쪽을 우선 순위로 하여 회피를 진행한다. Left값만 존재할 경우 왼쪽으로 회피하며, right값만 존재할 경우는 오른쪽으로 회피하게 된다. DNN 모델은 DAIDALUS와는 다르게 최소 근접 거리까지 남은 시간(t_{cpa})을 제시하지 못하기 때문에 alert level이 preventive alert 이하가 되고 5초 동안 수평 기동을 유지한다. 위험도가 corrective 이상이지만 track resolution이 모두 존재하지 않을 경우, 현재 상태를 유지하게 되며 AI 모드의 전체적인 구상도는 그림 24과 같다.

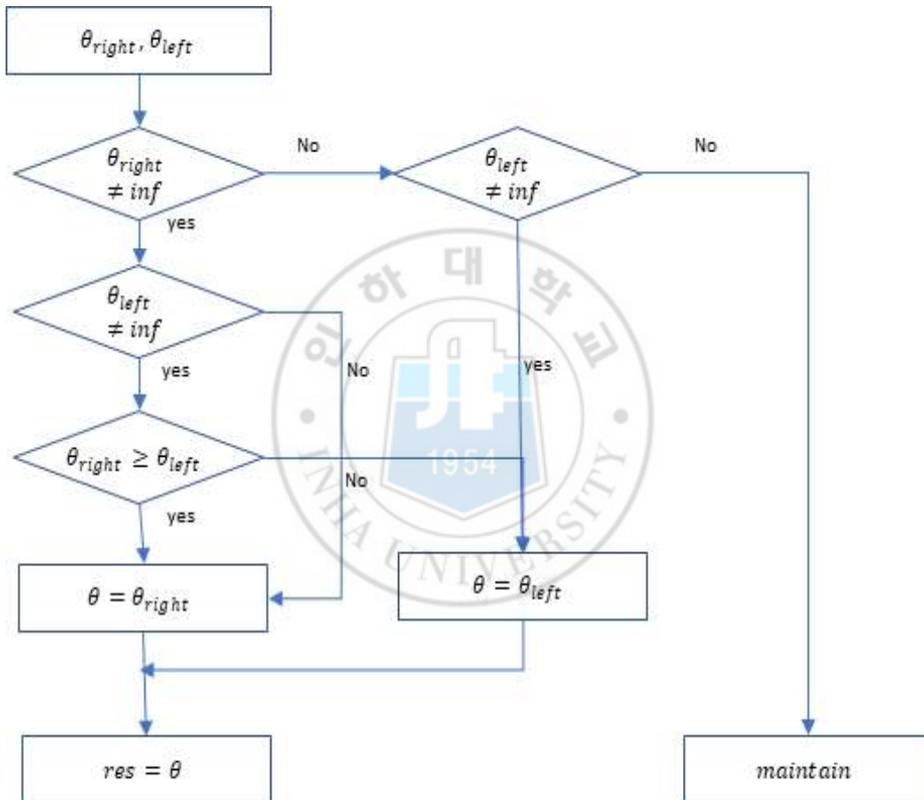


그림 24. AI 모드 구상도

제 4 장. 충돌 회피 알고리즘 결과

4.1 충돌 회피 평가 방법

DAIDALUS 모드와 AI 모드를 비교하기 위해 두 가지 타입의 시나리오를 생성하여 충돌 회피 시나리오를 진행하였다. 첫 번째 타입의 시나리오는 test vector에서 제시한 시나리오의 속도와 고도를 변형하여 생성하였으며 변형된 시나리오는 기존 test vector 시나리오와 조우 상황이 동일하기 때문에 같은 이름으로 명명하였다. Converge 1과 6, Head on 1과 13, Maneuver 1과 6, Over taking 12와 15 상황을 변형하여 시나리오를 생성하였다. 두 번째 타입의 시나리오는 기존의 test vector에 다른 intruder를 추가하는 방식으로 생성하였으며 총 두 가지 시나리오를 생성하였다. 각각 2 Intruder 1, 2 Intruder 2으로 명명하였다. 두 가지 타입의 시나리오 모두 초기 질량이 237톤인 Boeing 777-300을 항공기 모델로 하여 시뮬레이션을 진행하였다.

충돌 회피 알고리즘은 위험도 평가와 효율성 평가를 진행하였다. DWC 위반이 일어난 시간에 따라 위험도 평가를 진행하였으며 본 연구에서는 LoWC 상황이 일어나지 않았기 때문에 warning alert 이상의 DWC 위반이 일어난 시간이 많을수록, 충돌 위험성이 크다고 판단하였다. 효율성 평가를 위해 항공기의 연료소모량을 계산하였으며 충돌 회피를 하지 않고 지정된 경로로 기동한 항공기의 연료소모량을 계산한 다음, 충돌 회피한 항공기의 연료소모량과 비교 하는 방식으로 연구를 진행하였다. 차이가 적을수록 효율적으로 기동한다고 판단하였다.

4.2 시뮬레이션 결과

시뮬레이션은 각각의 항공기 의사 결정 모델 (DAIDALUS 모드, AI 모드)에 따라 진행하였으며 시나리오마다 항공기 의사 결정 모델의 2D 기동 궤적과 3D 기동궤적, alert level을 그림으로 나타냈다. 기동 궤적에서 안쪽을 채운 원은 start point를 의미하며 채워지지 않은 원은 WPT를 의미한다. 실선으로 표시된 부분은 항공기의 기동 궤적이며 점선으로 표시된 부분은 본래 항공기의 비행 계획 경로이다. 검정색 궤적은 ownship, 푸른색 궤적은 intruder의 궤적이며 ownship의 기동 궤적에서 preventive alert가 일어난 경우 초록색, corrective alert가 일어난 경우 주황색, warning alert가 일어난 경우 빨간색, LoWC가 일어난 경우 보라색으로 표현하였다. 표 8에서 시나리오별 연료소모율과 DWC 위반이 일어난 시간을 확인할 수 있다.

4.2.1 Converge 1 결과

그림 25는 DAIDALUS 모드일 경우 시간에 따른 DAIDALUS와 DNN 모델의 alert level이며, 그림 27과 29는 2-D와 3-D로 표현한 항공기 기동 궤적이다. 해당 궤적에서 충돌 상황을 회피하기 위해 수직 기동을 진행하였지만, 위험도가 떨어지지 않아 수평 기동을 다시 한 번 수행한 것을 볼 수 있다.

그림 26은 AI 모드일 경우 시간에 따른 DAIDALUS와 DNN 모델의 alert level이며, 그림 28과 30은 2-D와 3-D로 표현한 항공기 기동 궤적이다. 해당 궤적에서 DNN 모델의 경우 corrective alert로 인식하는 시간이 늦어져 수평 기동이 DAIDALUS 모드일 경우에 비해 과하게 진행한 것을 볼 수 있으며, 본래 기동 경로로 돌아올 때 다시 한 번 충돌 위험이 나타나 S자 모양의 기동을 한 것을 볼 수 있다. AI 모드일 경우, DAIDALUS 모드일 때 보다 약 100 kg의 연료를 더 소모하여 비효율적으로 기동한 것을 확인하였다.

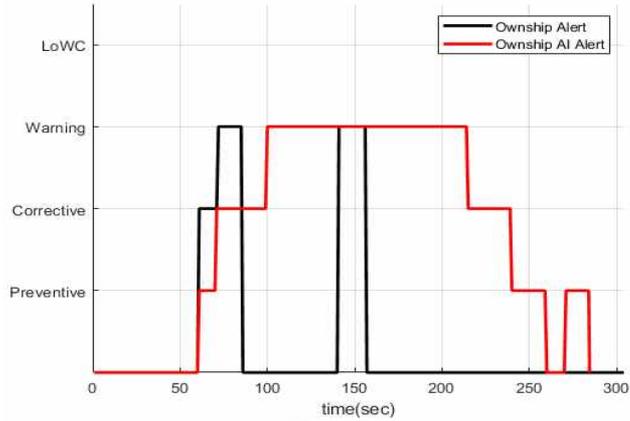


그림 25. DAIDALUS mode Alert level for Convergence 1

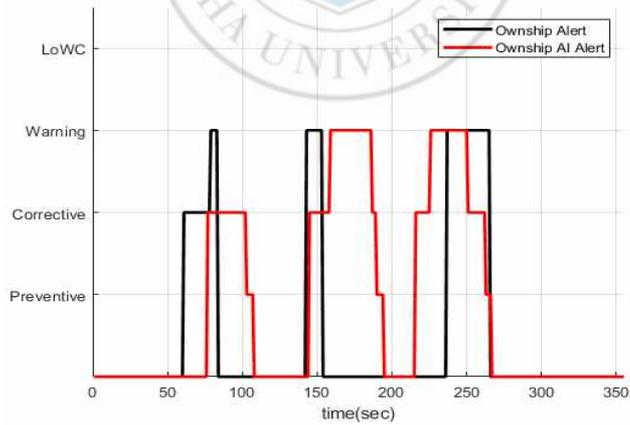


그림 26. AI mode Alert level for Convergence 1

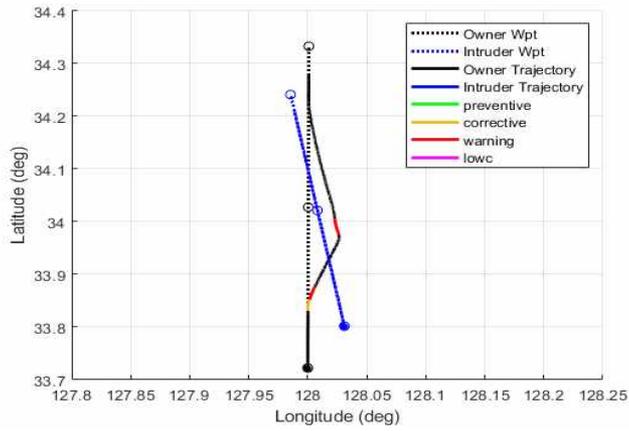


그림 27. DAIDALUS mode 2-D trajectory for Converge 1

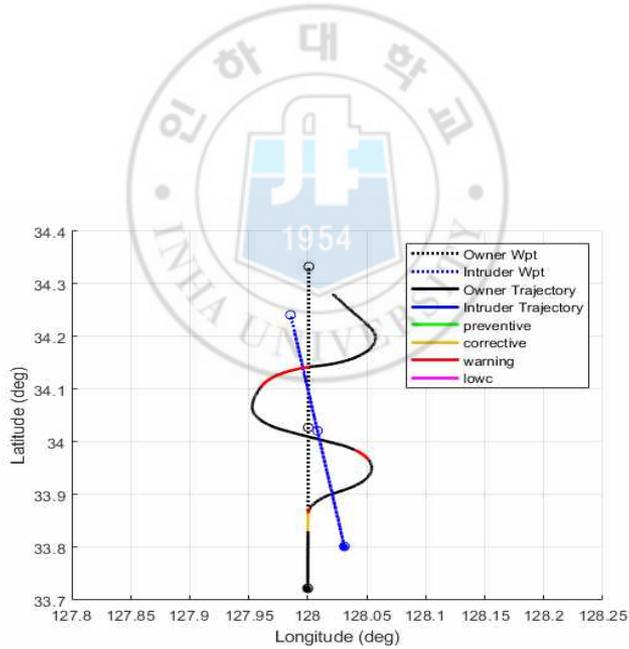


그림 28. AI mode 2-D trajectory for Converge 1

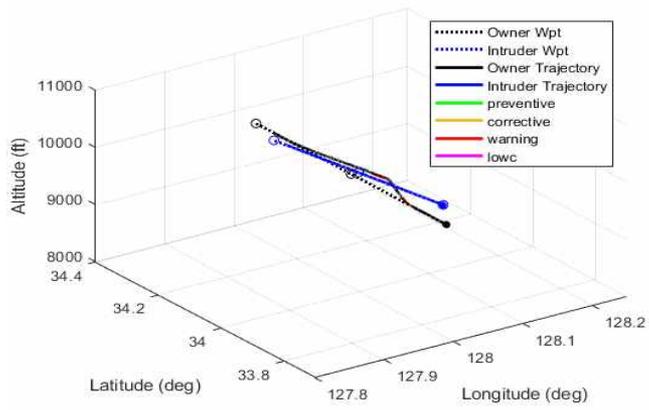


그림 29. DAIDALUS mode 3-D trajectory for Converge 1

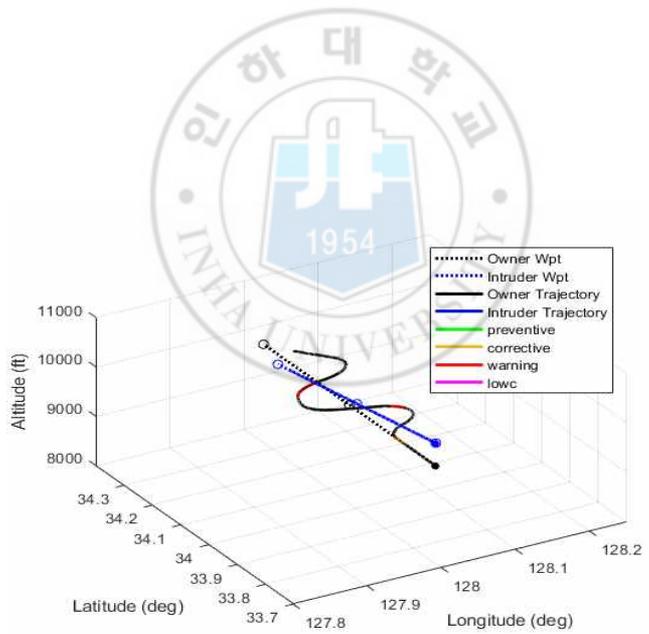


그림 30. AI mode 3-D trajectory for Converge 1

4.2.2 Maneuver 6 결과

그림 31는 DAIDALUS 모드일 경우 시간에 따른 DAIDALUS와 DNN 모델의 alert level이며, 그림 33과 35은 2-D와 3-D로 표현한 항공기 기동 궤적이다. 해당 궤적에서 Converge 1 상황과 비슷하게 충돌 상황을 회피하기 위해 수직 기동을 하였지만, 위험도가 떨어지지 않아 수평 기동이 다시 이루어진 것을 확인할 수 있다.

그림 32는 AI 모드일 경우 시간에 따른 DAIDALUS와 DNN 모델의 alert level이며, 그림 34과 36은 2-D와 3-D로 표현한 항공기 기동 궤적이다. 해당 궤적에서도 Converge 1과 마찬가지로 DNN 모델의 경우 corrective alert을 인식하는 시간이 늦어졌지만, warning alert이 일어난 시간이 DAIDALUS보다 적은 것을 보아 더 위험도가 낮게 기동한 것을 확인할 수 있었다. AI 모드일 경우, DAIDALUS 모드일 때 보다 약 250 kg의 연료를 더 소모하여 Converge 1과 마찬가지로 더 비효율적으로 기동한 것을 확인하였다.

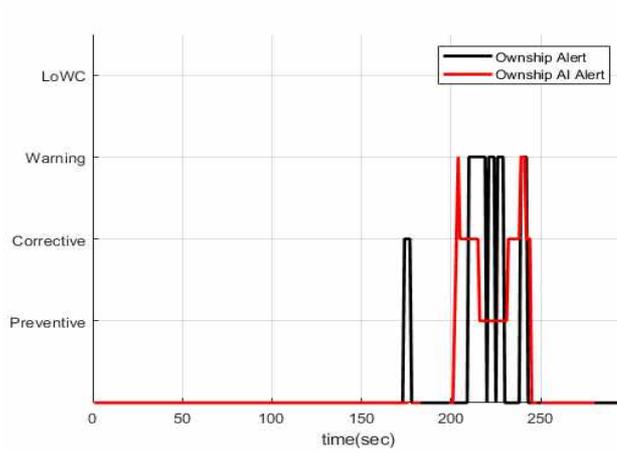


그림 31. DAIDALUS mode Alert level for Maneuver 6

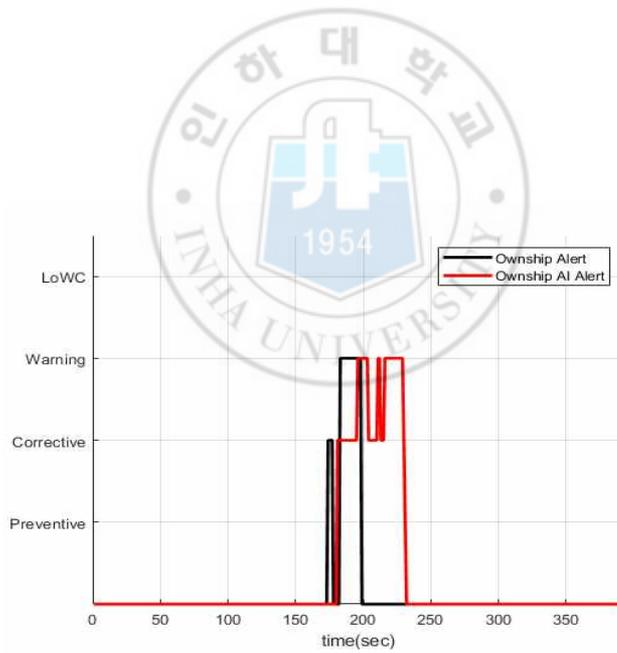


그림 32. AI mode Alert level for Maneuver 6

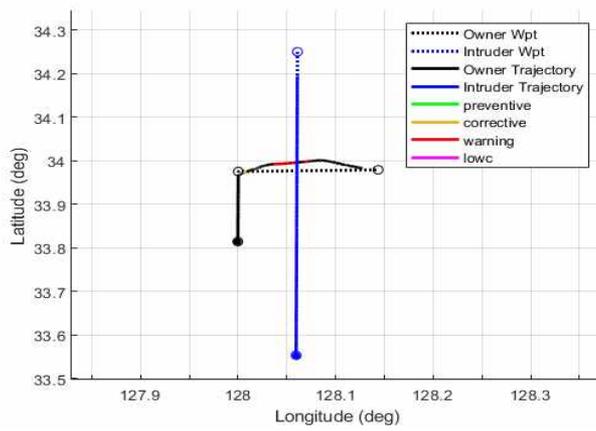


그림 33. DAIDALUS mode 2-D trajectory for Maneuver 6

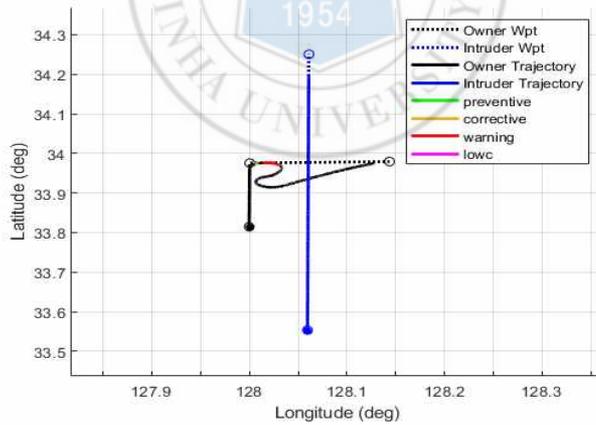


그림 34. AI mode 2-D trajectory for Maneuver 6

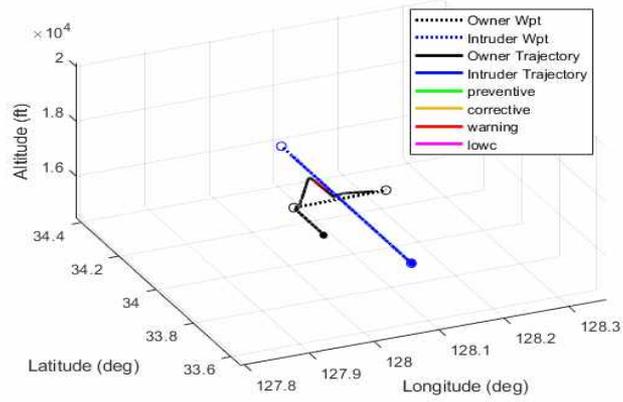


그림 35. DAIDALUS mode 3-D trajectory for Maneuver 6

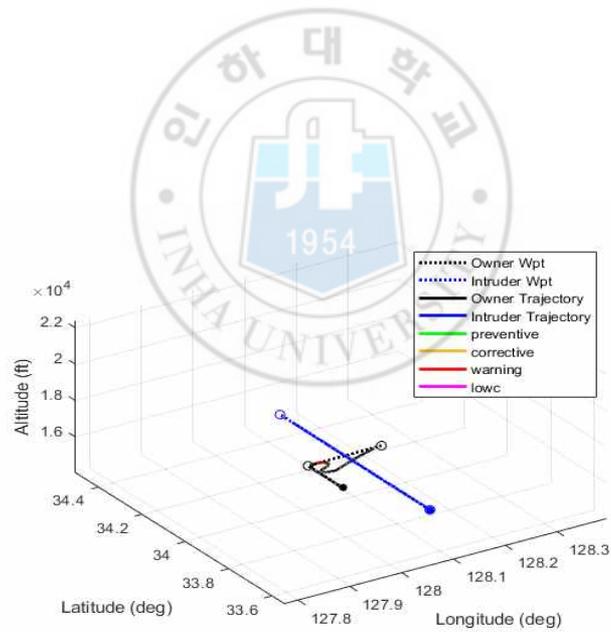


그림 36. AI mode 3-D trajectory for Maneuver 6

4.2.3 2 Intruder 1 결과

그림 37은 DAIDALUS 모드일 경우 시간에 따른 DAIDALUS와 DNN 모델의 alert level이며, 그림 39과 41는 2-D와 3-D로 표현한 항공기 기동 궤적이다. 해당 궤적에서는 위의 경우들과는 다르게 수평 기동만을 이용하여 회피한 것을 볼 수 있다. 회피 하는 도중에 다른 intruder와의 위험도가 더 높아져 해당 intruder와의 회피를 위해 또 다른 수평 기동이 이루어진 것을 확인할 수 있다.

그림 38은 AI 모드일 경우 시간에 따른 DAIDALUS와 DNN 모델의 alert level이며, 그림 40과 42는 2-D와 3-D로 표현한 항공기 기동 궤적이다. 해당 궤적에서는 intruder가 한 대일 경우와는 다르게, corrective alert로 인식하는 시간이 DAIDALUS보다 더 빨리 일어났으며, 두 대의 intruder가 있는 이유로 더 과하게 회피가 진행된 것을 확인할 수 있었다. 앞의 두 상황과 마찬가지로 AI 모드일 경우, DAIDALUS 모드일 때보다 약 65 kg의 연료를 더 사용하여 더 비효율적으로 기동한 것을 확인하였다.

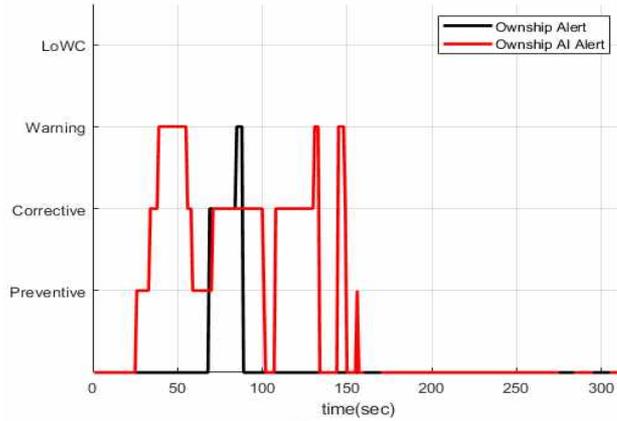


그림 37. DAIDALUS mode Alert level for 2 Intruder 1

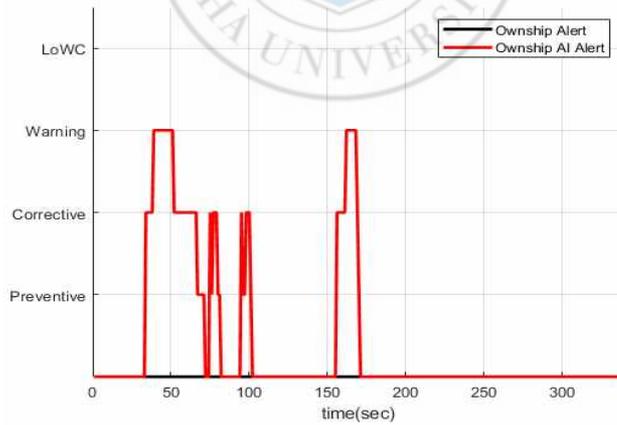


그림 38. AI mode Alert level for 2 Intruder 1

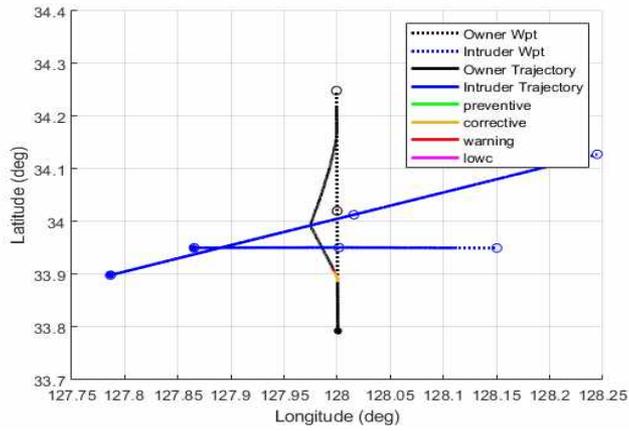


그림 39. DAIDALUS mode 2-D trajectory for 2 Intruder 1

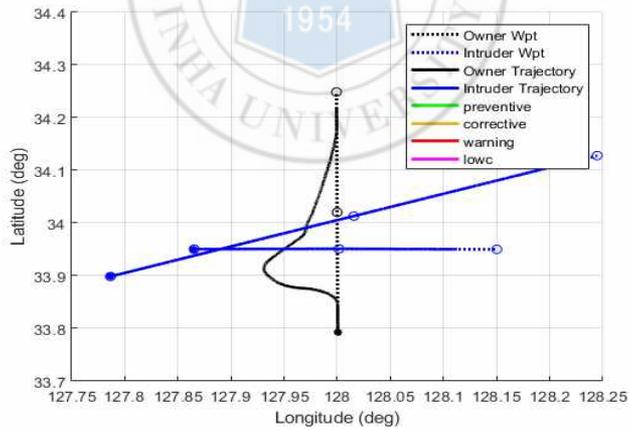


그림 40. AI mode 2-D trajectory for 2 Intruder 1

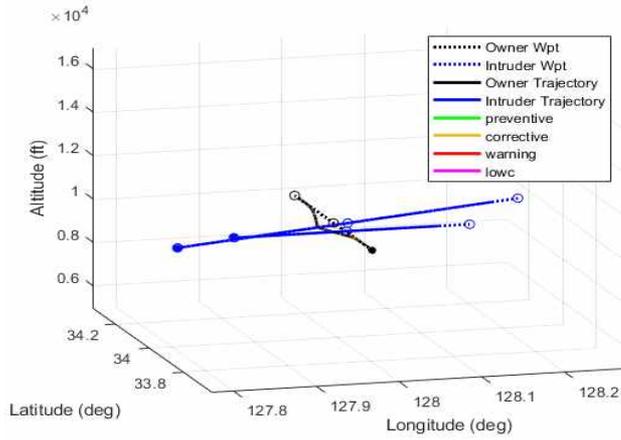


그림 41. DAIDALUS mode 3-D trajectory for 2 Intruder 1

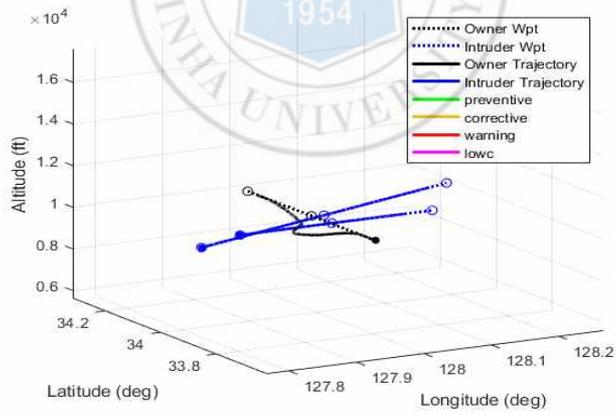


그림 42. AI mode 3-D trajectory for 2 Intruder 1

4.2.4 그 외의 시나리오 결과

AI 모드일 경우, 전체적으로 corrective alert로 인식하는 시간이 늦어지고, alert level이 낮아짐에도 높다고 인식하여 DAIDALUS 모드보다 비효율적으로 기동하였으나 모든 시나리오에서, LoWC가 일어나지 않은 것을 확인할 수 있다. 몇몇 시나리오는 오히려 DAIDALUS 모드보다 corrective alert로 인식하는 시간이 빨라 위험도가 더 낮게 기동을 한 것을 확인하였다.

표 8. 시나리오별 추가 연료소모량 및 DWC 위반이 일어난 시간

Scenario	Fuel Consumption(kg)		Warning Alert time (sec)	
	DAIDALUS	AI mode	DAIDALUS	AI mode
Converge 1	5.95 kg	156.9 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
Over taking 12	0.67 kg	175.01 kg	12 sec	0 sec
Over taking 15	19.75 kg	189.76 kg	10 sec	4 sec
2 Intruder 1	7.15 kg	73.60 kg	4 sec	0 sec
2 Intruder 2	5.96 kg	319.77 kg	15 sec	0 sec

제 5 장. 결론

본 연구에서는 심층 신경망 [5]의 지도 학습 기법을 이용하여 충돌 회피 알고리즘을 구성하였다. 광범위한 항공기들의 조우 상황을 이용하여 dataset을 구성하기 위해 두 가지 타입의 dataset을 구성하였다. 2019년 인천 FIR을 통과한 항공기들 중 DWC 위반이 일어난 항공기의 데이터와 MOPS [3]에서 제시한 test vector를 이용하여 input을 구성하였으며, 해당 데이터에 대한 DAIDALUS 결과를 이용하여 output을 구성하였다.

충돌 회피 알고리즘을 평가하기 위해 항공기 운동 모델 [10]과 항공기 의사 결정 모델 [11]을 기반으로 한 충돌 회피 시뮬레이션을 진행하였다. 충돌 회피 시나리오는 test vector들의 속도와 고도를 변경하여 생성하였으며, intruder를 추가하여 intruder가 두 대인 시나리오를 생성하기도 하였다. 시뮬레이션 결과, DNN 모델 기반의 AI 모드일 경우 전체적으로 위험도를 인식하는 시간이 늦고, 위험도가 낮아짐에도 높다고 인식하는 경우가 많아 비효율적인 기동을 유발하였다. 그러나 모든 시나리오에 대하여 LoWC는 일어나지 않았으며 intruder가 두 대인 경우는 충돌 위험도가 더 적게 기동하는 것을 볼 수 있었다.

추후, 현재 track resolution과 alert level만 구현하는 신경망 모델에서 altitude resolution, t_{cpa} 등의 파라미터를 구현할 수 있도록 하여 DNN 기반의 항공기 의사 결정 모델의 성능을 향상 시킬 예정이다. 추가로 현재 DAIDALUS 결과를 이용하여 dataset을 구성하였지만, 향후 조종사나 관제사의 지시가 반영된 dataset을 구성하고 학습을 진행할 예정이다.

제 6 장. 참고 문헌

- [1] ICAO, Manual on Remotely Piloted Aircraft Systems (RPAS), Doc 10019, 2015.
- [2] 유병선, 신동진, 장재호, 박정민, 강자영 (2015). ADS-B 수신기가 결합된 항공관제시물레이터 활용방안 연구. 한국항공우주학회 학술발표회 논문집, 2132-2135.
- [3] DO-365: Minimum operational performance standards (mops) for detect and avoid (daa) systems, RTCA Special Committee 228, 2017
- [4] C. Munoz, 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
- [5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, May 2015.
- [6] 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.
- [7] 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.
- [8] DO-365A: Minimum operational performance standards (mops) for detect and Avoid (daa) systems, RTCA Special Committee 228, 2020.
- [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.
- [11] 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

