ICAS 2016
30th Congress of the International Council of the Aeronautical Sciences
DCC, Daejeon, Korea ; September 25-30, 2016

ONE-VERSUS-ONE AIR-TO-AIR COMBAT MANEUVER GENERATION BASED ON DIFFERENTIAL GAME

Byung-Yoon Lee*, Seungyeop Han*, Hyun-Ju Park, Dong-Wan Yoo** and Min-Jea Tahk***
***Korea Advanced Institute of Science and Technology, **Agency for Defense Development**

Keywords: *Air-to-Air Combat, Differential Game, Scoring Function Matrix, UCAVs*

Abstract

This paper deals with the one-versus-one air-to-air combat maneuver generation based on differential game. The algorithm is composed of scoring function matrix calculation, and enemy combat maneuver analysis method in order to find the optimal combat maneuvers. This paper will be limited to consideration of the Within Visual Range (WVR) one-versus-one air-to-air combat. Several combat scenario results are analyzed by 3D trajectory, relative angle history and possession rate table to examine performance of the suggested algorithm.

1 Introduction

In recent years, due to the development of guidance, control and navigation technology, unmanned aerial vehicles (UAVs) have begun to expand their abilities to combat area not only for reconnaissance missions [1]. For example, the X-47B Unmanned Combat Aerial Vehicle (UCAV) developed by the Northrop Grumman Company successfully arrested landing on an aircraft carrier at sea in 2013 [2]. Judging from these trends, it is clear that the air-to-air combat between UCAVs will be occurred in the near future. Therefore autonomous air combat algorithm need to be developed.

The previous researchers were developed an air-to-air combat algorithm with various approach. In the references [3], [4], [5], and [6], the researchers are developed the air-to-air combat algorithm by the basic pursuit-evade game models based on the differential game theory, and this procedure includes calculating scores according to the combat situation between friend and enemy aircraft. In a different way, reference [7] has applied an approximate dynamic programming approach to solve the air-to-air combat strategy problem. In addition to this, reference [8] uses a heuristic value-driven system to avoid time and effort consuming characteristic of rule-based method.

In this paper, we developed the one-versus-one air-to-air combat algorithm using the scoring function matrix calculation based on the differential game theory and enemy combat maneuver analysis method. In addition, factors like range, velocity, and terrain are handled by specific methods to consider realistic combat situation. In order to analyze the performance of the air combat algorithm, 3D trajectories, possession rate table, and relative angle history are used.

This paper is composed as follows. In the chapter 2, we introduce dynamic model for the air combat simulation, and the combat algorithm will be described. Scenarios and simulation results are introduced in the chapter 3 in order to show up the performance of the air combat algorithm. Finally, a discussion on performance of the suggested air combat algorithm and conclusion is followed.

Fig. 1. Northrop Grumman X-47B UCAV

2 Air-to-air Combat Maneuver Algorithm

In this chapter, air-to-air combat algorithm will be introduced. This algorithm is based on the author's previous research [9-10]. As like in [9-10], we have used F-16 pseudo 6-DOF model to simulate air-to-air combat. Readers can refer the author's previous research [9-10] for detailed simulation conditions and terminal condition of the combat. In this research, partial procedures of the algorithm has been modified for the improvement of performance. Detailed description will be followed.

2.1 Scoring Function

In order to perform air-to-air combat maneuver, we have adopt the score concept that is calculated by combat geometry and other primary features. For example, score of the friend aircraft will be very low when the enemy aircraft is located right after of the friend aircraft. Additionally, range between enemy and friend aircraft and each aircraft's altitude are also considered by scoring function. Finally, as we mentioned in our previous research [10], the scoring function is composed of three elements; geometrical relationship, velocity error, and range error. These three elements were exploited in order to satisfy following reasons respectively; chasing enemy's tail, maintaining optimal corner speed, and avoiding overshoot from enemy's tail.

First, the geometrical relationship score between friend (blue) and enemy (red) aircraft is shown in the Fig. 2 and Eq. (1). We called it orientation score, S_{Ori}. In Fig. 2, AA_B means blue's aspect angle, and BA_R means red's bearing angle. Therefore, orientation score would have values from 0 to 1. For instance, score of the friend aircraft will be 1 when the friend aircraft is located on the enemy aircraft's tail.

Second, velocity error score S_V is determined by the Eq. (2). Desired corner speed, V_{des} is determined by the V-n diagram of the aircraft's dynamics. V-n diagram example which is used in this research is shown in the Fig. 3. This diagram has plotted in the conditions of altitude of 5000 m and 10G of load factor limit.

Lastly, range error score S_{RE} is determined as Eq. (3). Range error score should be calculated in order to maintain gun range and safe distance between friend and enemy aircraft. In order to archive this purpose, we determined range error R_E as like in Eq. (3). In Eq. (3), a_0, a_1, a_3 and a_d has been adopted for shaping range error score function.

Finally, total score function is determined by the Eq. (4).

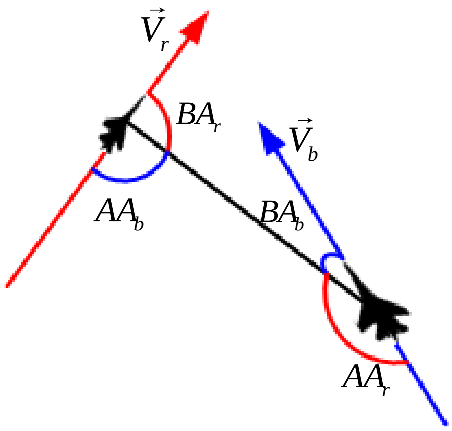

Fig. 2. Relative Angles Relationship between Friend(blue) and Enemy(red) Aircraft

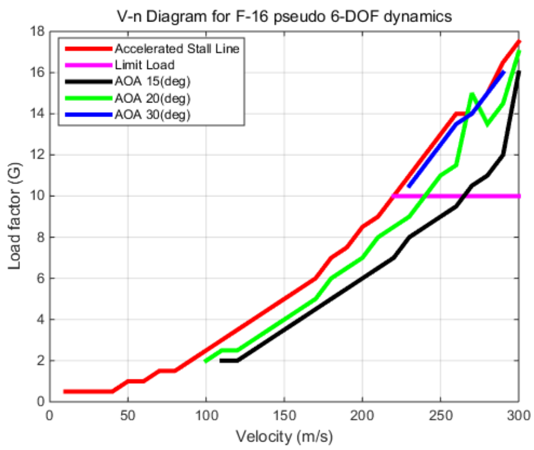

Fig. 3. V-n Diagram for F-16 Pseudo 6-DOF Model

$$S_{Ori} = \frac{2\pi - (BA + AA)}{2\pi} \tag{1}$$

$$S_V = \left| \frac{V_{des} - V_{ref}}{V_{ref}} \right| \tag{2}$$

$$S_{RE} = 1 - \left(\frac{a_1 - a_0}{R_E + a_1} \right)^2 e^{-(R_E - a_d)^2 / a_3^2} \tag{3}$$
$$R_E \triangleq |R| - R_{safety} \;(\text{Range error})$$

$$S_{total} = S_{Ori} \times S_{RE} \times S_V \tag{4}$$

2.2 Scoring Function Matrix (SFM)

In this paper, to determine the optimal combat maneuver, the number of maneuver choices of each aircraft is limited to seven as shown in figure 4. Therefore friend and enemy aircraft have seven maneuver choices each, so there are total 49 possible maneuver combinations as shown in Fig. 5. To determine the final scores of each combination, the end states of each combination are calculated by numerically integrating the equations of motion. Final optimal maneuver command is selected by min-max algorithm that is based on the game theorem. With this algorithm, friend aircraft can avoid the worst case. This min-max procedure is called as the security strategy. On the contrary, max-max procedure is called as the greedy strategy. Detailed procedures are described on our previous research [10].

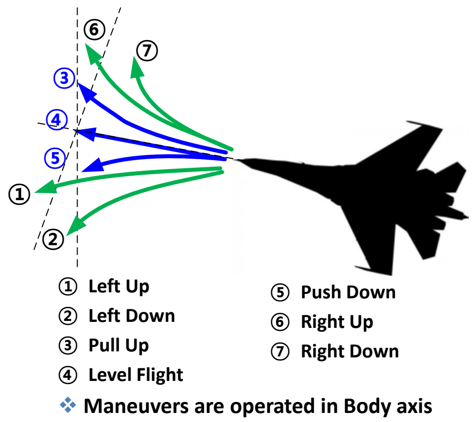

Fig. 4. Seven Maneuver Choices

		RED MNVR Choices						
		1	2	3	4	5	6	7
BLUE MNVR Choices	1	...						
	2		...					
	3			...				
	4				...			
	5					...		
	6						...	
	7							...

Fig. 5. Scoring Function Matrix

2.3 The counter-greedy security strategy

As mentioned in previous chapter, friend aircraft can avoid worst case using the security strategy during the air combat situation. On the other hand, the security strategy can be inefficient combat method according to enemy's strategy. For example, in the case of that enemy does not use the security strategy, then friend aircraft does not have to use the security strategy. In other words, if enemy's combat strategy is equal to the greedy strategy, then friend aircraft have to choose counter maneuver of the enemy's greedy maneuver in order to get more efficiency than the security strategy. In this paper, first, we have suggested that detection method for enemy strategy. Second, compensation method for disadvantage of the security strategy will be suggested.

In order to detect enemy's combat strategy, we assumed that enemy's combat strategy is divided into two parts; the greedy strategy, and the other strategy. Now, to determine enemy's combat strategy, friend aircraft calculate following Eq. (5) in every operation time of the combat algorithm. In Eq. (5), $SFM_{est\_prev}^{enemy}$ means predicted enemy scoring function matrix for current time step that is calculated at previous time step. SF_{cur}^{enemy} means current score that is calculated by current measured state values.

$$SFM_{diff}^{enemy} = \left| SFM_{est\_prev}^{enemy} - SF_{cur}^{enemy} \right| \tag{5}$$

We can predict enemy's greedy strategy combat maneuver solution by the $SFM_{est\_prev}^{enemy}$. In sequence, we can determine enemy combat strategy is the greedy or not, by comparison between smallest value of the SFM_{diff}^{enemy} and predicted enemy's greedy strategy combat maneuver solution. During this procedure, the greedy and the security strategy combat maneuver solutions can be equal to each other solutions. In this case, we have determined that enemy combat strategy is equal to the security strategy for the conservative solution. In summary, friend aircraft tried to predict enemy combat strategy on every operation time. If predicted enemy combat strategy is equal to the greedy strategy, then friend aircraft choose counter-greedy combat maneuver to get more efficiency.

The counter-greedy combat maneuver is selected easily by observing enemy's scoring function matrix. Specifically, first, find row index with the maximum value in the enemy's scoring function matrix. In sequence, column index of the smallest value in the selected row is the counter-greedy combat maneuver. On the other hand, friend aircraft choose the security combat strategy when the predicted enemy combat strategy is not the greedy strategy. Suggested combat algorithm was named as the counter-greedy (CG) security combat strategy.

3 Simulation

3.1 Simulation Setup

F-16 pseudo 6-DOF dynamic model is used in this paper. Total three scenarios were setup to verify the performance of suggested air-to-air combat algorithm. Specific initial conditions are shown in the Table. 1. We have simulated for 60 seconds for all simulation scenario. Each algorithm runs at every 0.6 second and predict 0.6 second at every operation time.

		#1	**#2**	**#3**
Situation		**Head-on**	**Head-on**	**Head-on**
Algorithm	**Blue**	Security	Greedy	Security
	Red	Greedy	CG secu.	CG secu.
NED Pos. (m)	**Blue**	[-1000, 0, -7000]	[-1000,0, -7000]	[-1000, 0, -7000]
	Red	[1000, 0, -7000]	[1000, 0, -7000]	[1000, 0, -7000]
Heading (deg)	**Blue**	0	0	0
	Red	180	180	180
Velocity (m/s)	**Blue**	250	250	250
	Red	250	250	250

Table. 1. Simulation Initial Condition

3.2 Simulation Results

Simulation results were mainly analyzed by three features; 3D trajectory, relative angle history and possession rate table. Examples of each graph are shown in the Fig. 6, Fig. 7, Fig. 8 and Table 2. For this example scenario, we have used scenario #1 but combat algorithms are same for both aircraft. At this time, we have used the security strategy for both aircraft. Fig. 6 is 3D trajectory result. Fig. 7 and Fig. 8 are relative angle history for blue and red aircraft respectively. Table 2 is possession rate of each aircraft. We can observe that combat result is draw because initial conditions are fair and same combat algorithm were used.

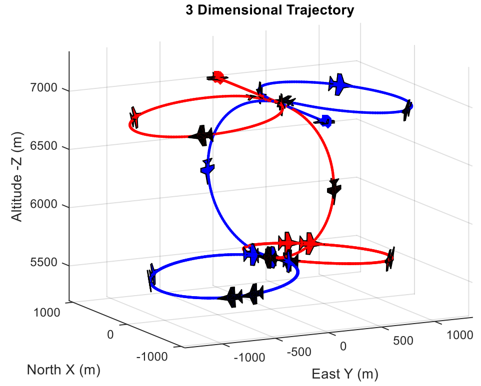

Fig. 6. 3D Trajectory (sample scenario)

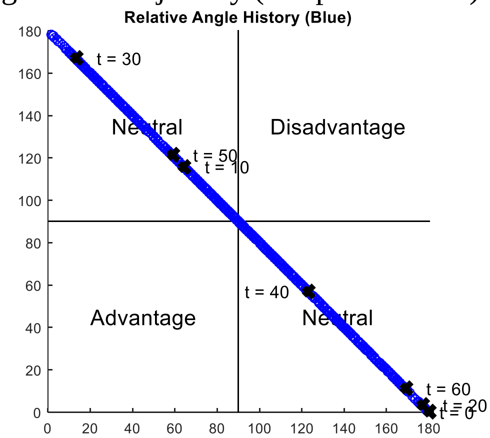

Fig. 7. Relative Angle - Blue (sample scenario)

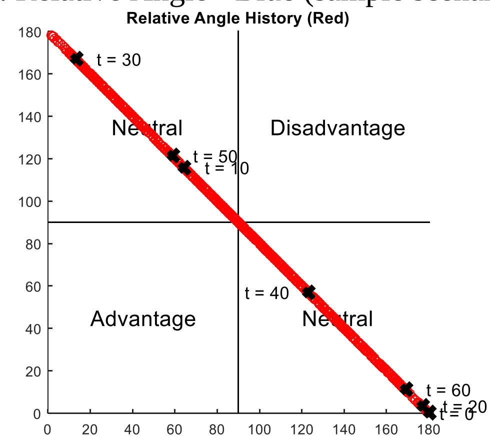

Fig. 8. Relative Angle - Red (sample scenario)

Possession Rate for Sample Scenario	Blue	Red
Advantage (%)	0.00	0.00
Disadvantage (%)	0.00	0.00
Neutral (%)	100.00	100.00

Table. 2. Possession Rate (sample scenario)

3.2.1 Simulation Results for Scenario #1

Simulation results for the scenario #1 are shown below. In this scenario, we want to observe performance difference between the security and the greedy algorithm. As we can observe from the results, the security algorithm has an advantage in possession rate about 3 percent high compared to the greedy algorithm.

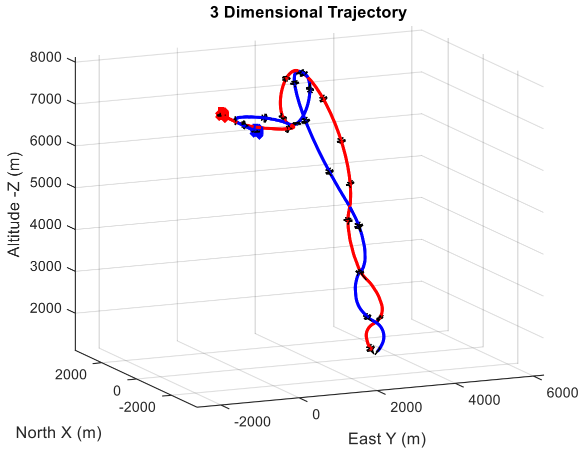

Fig. 9. 3D Trajectory (scenario #1)

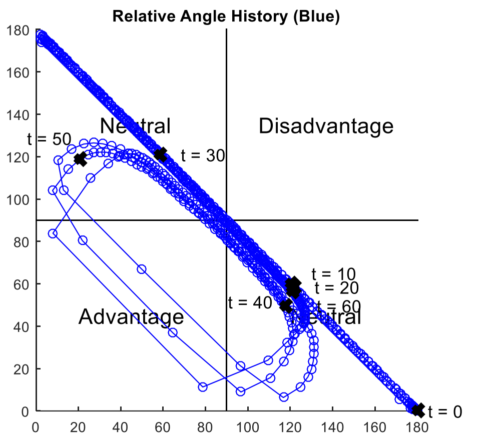

Fig. 10. Relative Angle - Blue (scenario #1)

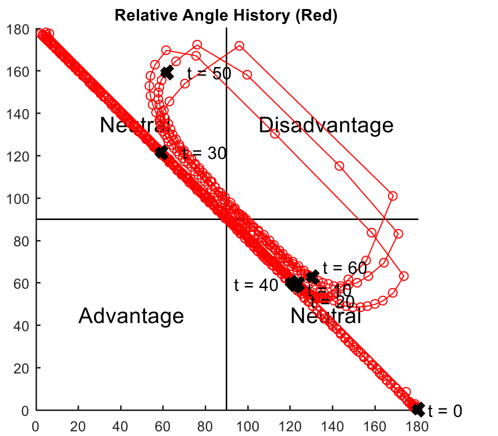

Fig. 11. Relative Angle - Red (scenario #1)

Possession Rate for Scenario #1	Blue	Red
Advantage (%)	3.16	0.00
Disadvantage (%)	0.00	3.16
Neutral (%)	96.84	96.84

Table. 3. Possession Rate (scenario #1)

3.2.2 Simulation Results for Scenario #2

In the scenario #2, combat algorithm of the blue aircraft is fixed in the greedy algorithm. On the other hand, the counter-greedy security combat algorithm is used for the red aircraft. As mentioned in chapter 2.3, the counter-greedy algorithm have designed to take highest performance against the greedy algorithm.

Simulation results for the scenario #2 are shown in Fig. 12, Fig. 13, Fig. 14 and Table. 4. These results has shown that designed counter-greedy algorithm operate correctly. Especially the red aircraft have chased enemy's tail rapidly after 30 seconds. As a results, counter-greedy algorithm has an advantage in 46.26 percent possession rate.

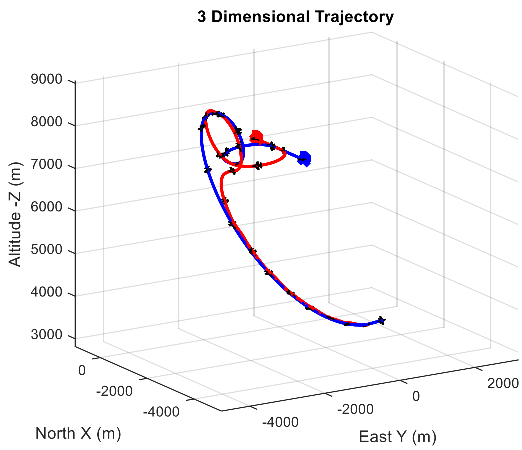

Fig. 12. 3D Trajectory (scenario #2)

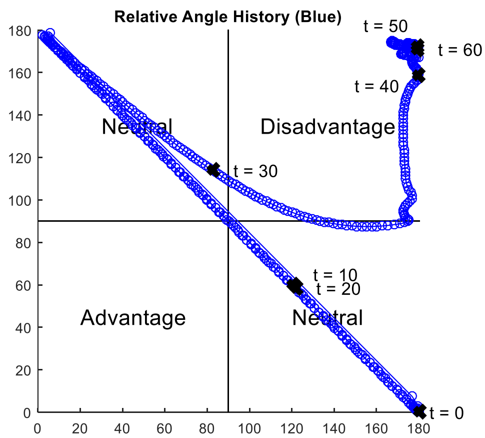

Fig. 13. Relative Angle - Blue (scenario #2)

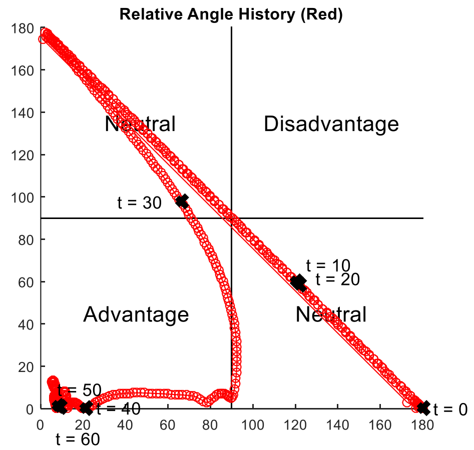

Fig. 14. Relative Angle - Red (scenario #2)

Possession Rate for Scenario #2	Blue	Red
Advantage (%)	0.00	46.26
Disadvantage (%)	46.26	0.00
Neutral (%)	53.74	53.74

Table. 4. Possession Rate (scenario #2)

3.2.3 Simulation Results for Scenario #3

In the previous section, we can observe that the counter-greedy combat algorithm works well against the greedy algorithm. In the scenario #3, we want to observe performance difference between the security and the counter-greedy security algorithm. Simulation results for the scenario #3 are shown below in Fig. 15, Fig. 16, Fig. 17 and Table. 5. Observed results from the Table 5, the counter-greedy algorithm has an advantage in 58.07 percent possession rate. This result involves that large portion of the security combat algorithm solution is equal to the greedy combat algorithm solution.

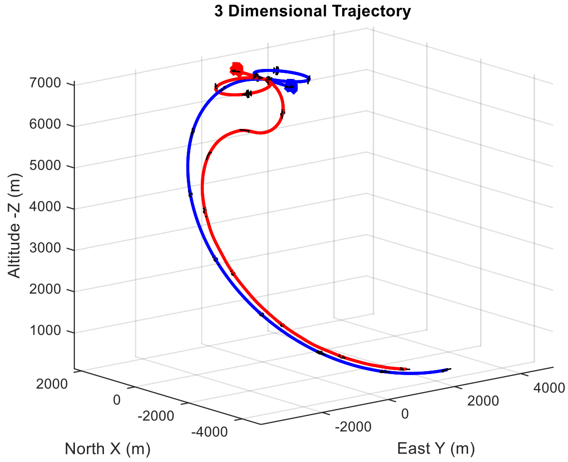

Fig. 15. 3D Trajectory (scenario #3)

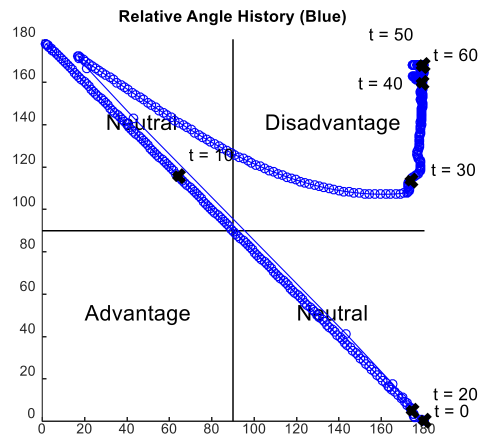

Fig. 16. Relative Angle - Blue(scenario #3)

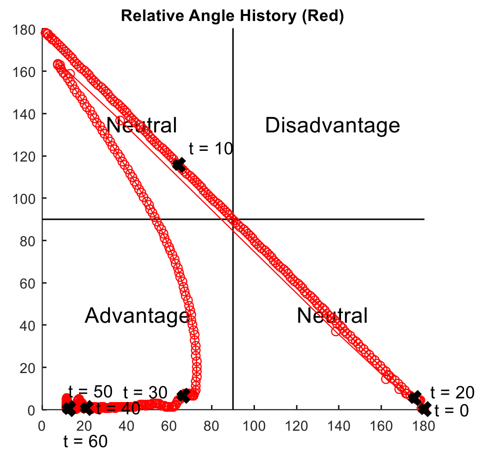

Fig. 17. Relative Angle - Red (scenario #3)

Possession Rate for Scenario #3	Blue	Red
Advantage (%)	0.00	58.07
Disadvantage (%)	58.07	0.00
Neutral (%)	41.93	41.93

Table. 5. Possession Rate (scenario #3)

4 Conclusion

In this paper, autonomous one-versus-one air-to-air combat algorithm have been developed by the security strategy algorithm that is based on the game theory. To improve the performance of the security combat algorithm, several improvements were applied to the algorithm. Specifically, the security combat algorithm was improved by the counter-greedy security algorithm. Performance of the developed combat algorithm was analyzed by several scenarios and their simulation results.

5 Contact Author Email Address

mailto:bylee@fdcl.kaist.ac.kr

Acknowledgements

This research was supported by Agency for Defense Development (Contract UD130041JD).

References

[1] Austin R. *Unmanned Air Systems UAV Design, Development and Deployment*, John Wiley & Sons Ltd, 2010.

[2] The Northrop Grumman homepage, X-47B Makes First Arrested Landing at Sea, http://www.northropgrumman.com/Capabilities/x47bucas/.

[3] Austin F, Carbone G, Falco M, Hinz H and Lewis M. Game Theory for Automated Maneuvering During Air-to-Air Combat, *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 6, pp. 1143-1149, 1990.

[4] He F, Yao Y. Maneuver Decision-making on Air-to-Air Combat Via Hybrid Control, *Aerospace conference*, IEEE, pp.1-6, 2010.

[5] Jang S, Study of Intelligent Pilot Model Based on Basic Fighter Maneuvering for Air Combat Simulation, *KDMT1201222126*, Department of AE, Inha Univ., 2012.

[6] Park H, Yoo D, Tahk M. Generation of the Optimal Air-to-Air Combat Maneuver using Scoring Function Matrix, *2014 KSAS Spring Conference*, Wonju, Korea, 2014.

[7] McGrew James S, et al. Air-Combat Strategy Using Approximate Dynamic Programming, *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, pp. 1641-1654, 2010.

[8] Lazarus E. The Application of Value-Driven Decision-Making in Air Combat Simulation, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Piscataway, Vol. 3, NJ, pp. 2302-2307, 1997.

[9] Park H, Yoo D, Tahk M. Optimal WVR Air-to-Air Combat Maneuver Generation Algorithm using Scoring Function based on Security Strategy, *2014 Asia-Pacific International Symposium on Aerospace Technology*, Shanghai, China, 2014.

[10] Park H, Lee B, Tahk M, Differential Game Based Air Combat Maneuver Generation Using Scoring Function Matrix, *International Journal of Aeronautical and Space Sciences*, Vol. 17, No. 2, 2016. (accepted)

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.