Air Combat Learning from F-16 Flight Information

Mustafa KARLI, Mehmet Önder EFE, Hayri SEVER
Autonomous System Laboratory
Department of Computer Engineering
Hacettepe University
Beytepe 06800, Ankara, Turkey

***Abstract*—Movement sequence of a real air combat flight contains valuable information that can be used to infer artificial air combat learning. There are different ways to control unmanned aerial vehicles for a given flight path. But identifying the best move at the time being relative to an enemy air craft requires learning flight experience from real air combat fighters. This paper shows how to set up learning and control environment with adaptive neuro fuzzy inference system for maneuver decisions using real F-16 flight information. Real flight information is also utilized to justify the test results.**

I. Introduction (*Heading 1*)

Air combat fighting is getting an advantageous position and shooting while enemy is in the effective range of weapon. Artificially deciding a move during combat requires both flight and combat experience. Machine learning methodologies are required to train such a system.

In literature there are rule based systems [2] to make high level decisions. Also modelling a human decision model [3] is used as an alternative. Using artificial intelligence methods [4] and [5] generally assists human pilots. Also there is a prediction model for maneuvers in [6]. Adaptive neuro fuzzy inference systems (ANFIS) are used for autonomous control of UAV in [7] and [8] for static routes. NASA proposed the Trial Maneuver (TM) method [14] but does not utilize machine learning techniques for decision making. This paper assumes that we have enough information about the state of enemy aircraft. There are option like getting enemy status from radar info with latency or using visual techniques as presented in [15]. Also estimating enemy bank angle with visual techniques is presented in [16]. But during real air combat latency in obtaining enemy status is neglected before making high level maneuver decisions. We assume that having velocity with direction derived from radar signals is enough for maneuver prediction.

Our work advances the subject area in terms of designing an ANFIS model for choosing the right move in air combat and generating high level control signals to move the aircraft to the most advantageous position. ANFIS model is trained with F-16 flight information. Some portion of flight information is reserved for test purposes as well. We do not deal with aircraft characteristics or flight control system since it is assumed that aircraft already has a robust control mechanism. It may seem hard to handle non-linear equations of agile aircraft but since data is obtained from real F-16 aircraft, we assume that learning and test data is among acceptable agility ranges.

This paper is organized as follows: In the next section we define the problem of air combat learning. The third section defines the structure of the learning corpus and data source. The forth section proposes the learning and control method. The fifth section includes the test results and last section includes the concluding remarks.

II. Problem Definition

Combat aircrafts can execute agile maneuvers. Control of a combat aircraft differs in terms of speed, robustness and reaction time. Since both aircrafts are moving in three dimensional airspace, there are nearly infinite movement alternatives. Choosing the right move requires more sophisticated learning and control mechanism.

First of all, learning requires good sample data for learning and testing process. Second step should be to set up a control mechanism for both choosing the right maneuver and generating the control signals for UAV model.

III. Corpus

A. Air Combat Data

The sample corpus data is extracted from F-16 aircraft flights and decomposed into sequence of moves. F-16 stores information from various kinds of sensors. The flight data includes more than 200 columns of instant status information. Out of those only positional and angular information is used. The status of the aircraft at a time includes 12 state variables.

$$X = \{n, e, h, v, \varphi, \theta, \psi, \alpha, \beta, P, Q, R\} \quad (1)$$

These are positions in north, east and altitude, speed, roll, pitch and heading angles, angular difference between body and velocity axes (angle of attack and side slip angle) and angular velocity of roll, pitch and heading. More details on flight dynamics can be learned from [18].

In air combat, there are minimum 2 aircrafts involved. So there should be state variables of the two. The combat status of both depends on the relative geometry to each other. Relative geometry and advantage function is discussed in [9] and includes range and antenna train angle of two aircrafts.

B. Maneuver Sequence

Training and controlling 12 state variables is a real hard work. This paper does not focus on low level control of these variables. But we bear in mind that we should know the control logic of low level controllers and generate necessary control

978-1-5090-6034-4/17/$31.00

inputs for these controllers. Decomposing flight information into sequence of movements eases the learning process because;

- Control logic for every movement is designed separately.
- There is less number of control variables for each mode.
- Controlling the 12 state variables are left to the low level controller and we focus on high level angular changes.

There is a higher and lower order MIMO controller design in [17]. There are 6 modes defined in [10]. These are level flight, climb/descend, turn, loop, pitch-yaw transition and roll movements. We decided to decrease number of modes by 3 and add 1 new mode since;

- Level flight and climb descent modes may be important for low level control but they do not differ for learning process so combined into straight path move.
- Roll mode have no effect on resulting path so neglected.
- Pitch-yaw transition modes do not differ from looping or turning so neglected.
- 3D mode is required which was introduced in [11] but neglected later in [10]. 3D mode consists of looping and rolling at the same time but results change in both pitch and heading angle.

As a result, flight data is decomposed into 4 modes as defined in Table 1.

TABLE 1 MOVEMENT MODES

Mode	Name	Inputs
SP	Straight Path	$\Delta t, v, \theta$
LO	Loop	$\Delta t, v, \theta, \dot{\psi}$
TU	Turn	$\Delta t, v, \dot{\theta}, \dot{\psi}'$
3D	3D loop and turn	$\Delta t, v, \dot{\theta}, \dot{\psi}$

IV. METHODS USED

We designed a fuzzy identification of movement mode and neural method for learning. Combining both results an adaptive neuro fuzzy inference system [1] which has state input from both with the relative geometry and results control signals of movement sequence.

A. ANFIS Architecture

ANFIS [1] model integrates the fuzzy control logic and neural learning system into single architecture in 6 layers. A generic sugeno [12] style model is in Figure 1.

If x_1 *is* A_1, .. x_n *is* A_n *then* $y=x_0+k_1x_1+\ldots+k_nx_n$ (2)

The first layer is the input layer which simply inputs x_i and outputs to the second layer. The number of cells is equal to the number of inputs for first layer.

Second layer executes the fuzzy membership functions also known as fuzzification layer. These functions determine degree of membership of an input in a fuzzy set. Total number of neurons in this layer is equal to the total number of fuzzy sets for all inputs.

Third layer is where fuzzy rules are executed. These rules are first order sugeno fuzzy rule [12]. Neurons inputs from fuzzification neurons of related rules. Multiplication with a coefficient computes the truth of that rule.

Forth layer normalizes the firing strength of rules that determines the contribution of the rule to the final result.

Fifth layer is the defuzzification layer where weighted consequent values of rules are calculated. Neurons in this level also receive the input signals as well.

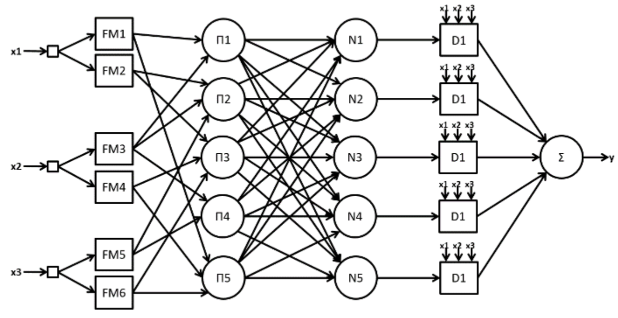

Figure 1 Generic ANFIS Architecture

I. Input Layer

The input layer receives the state and movement information of both aircrafts and relative geometry to each other. Actual velocity and desired velocity in movement sequence are different. Also the angular values may differ in movement mode where some are deltas and some are desired change in the angle.

$$S_1 = \{h_1, \theta_1, v_1\} \quad (3)$$
$$S_2 = \{h_2, \theta_2, v_2\} \quad (4)$$
$$\sigma_1 = \{\Delta t_1, q_1, v_1, \theta_1, \psi_1\} \quad (5)$$
$$\sigma_2 = \{\Delta t_2, q_2, v_2, \theta_2, \psi_2\} \quad (6)$$
$$R = \{r, h, v, \lambda, \eta_1, \eta_2\} \quad (7)$$

Range (r) also known as line of site vector is computed from the square root of the sum of distance squares in 3 axis. Relative altitude (h) is the difference in Z axis. Closure velocity (v) is change in the range. Angle of tail (λ) is the angle between velocity vectors at the intersection point. Line of site angle or antenna train angle (η) is the angle between velocity vector and range for each aircraft. Computing these variables are explained deeply in [9]. Variables that don't contribute much to the learning process are discarded to reduce the number from 18 to 10.

Altitudes and velocities of each aircraft are neglected since difference in altitude and closure velocity are more meaningful. Only exception is crash avoidance that is not studied here. η_2 can be neglected since λ gives enough sense about how advantageous opponent is. This also eliminates need for θ for learning. Time is also derived from actual velocity and desired action. Excluding these variables from learning process does not limit us using these state variables at the final stage where maneuver logic is implemented. Final input and output variables are;

$$\{r, h, v, \lambda, \eta, q, V, \varphi, \theta, \psi\} \quad (8)$$

II. Fuzzification Layer

This layer includes fuzzy set definitions for each input and membership functions of the whole set members. Same input types of both aircrafts share the same fuzzy sets and membership functions and are summarized in Table 2.

$$y_i^2 = f(x_i^2) \quad (9)$$

TABLE 2 FUZZY SETS AND MEMBERS

Set	Member	Count
A_h	$down\|close\|up$	3
A_θ	$0\|\frac{\pi}{2}\|\pi\|\frac{3\pi}{2}$	4
A_ψ	$0\|\frac{\pi}{2}\|\pi\|\frac{3\pi}{2}$	4
A_v	$cruise\|subsonic\|supersonic$	3
A_r	$close\|wvr\|bvr$	3
A_η	$pure\|lead\|deffensive$	3
A_λ	$0\|\frac{\pi}{3}\|\frac{2\pi}{3}\|\pi$	4

Difference in altitude is divided into 3 whether we are up, down or close to the opponent less than 300 feet in altitude.

$$A_h = \{down|close|up\} \qquad (10)$$

Pitch and heading angles are classified into quarters of *π/2* values.

$$A_\theta = \left\{0|\tfrac{\pi}{2}|\pi|\tfrac{3\pi}{2}\right\} \qquad (11)$$

$$A_\psi = \left\{0|\tfrac{\pi}{2}|\pi|\tfrac{3\pi}{2}\right\} \qquad (12)$$

The speed limits the agility of the aircraft. Minimum value is the stall speed where aircraft should not fall below. Cruise speed is the neutral speed during patrol or formation flights. The critical speed is where the aircraft passes through speed of sound. These are 0.8 to 1.2 mach.

$$A_v = \{cruise|subsonic|supersonic\} \qquad (13)$$

The range between two aircrafts determines the style of combat. Close distance is the effective weapon zone often less than 3500 feet. Within visual range (WVR) results one to one air combat while beyond visual range (BVR) allows the initial setup of multi-aircraft combat. WVR range is often between 3500 and 6500 feet.

$$A_r = \{close|wvr|bvr\} \qquad (14)$$

Antenna train angle determines if the position is offensive or defensive and ranges from 0 to π. Values less than *π/2* mean offensive while larger is defensive. Offensive position should be handled in 2 different ranges as pure and lead positions.

$$A_\eta = \{pure|lead|deffensive\} \qquad (15)$$

Angle off tail is divided into 4 where values less than 60 degrees shows less closure velocity with pure pursuit and increasing λ means high closure velocity. Values more than 120 degrees means a kamikaze position.

$$A_\lambda = \left\{0|\tfrac{\pi}{3}|\tfrac{2\pi}{3}|\pi\right\} \qquad (16)$$

III. Rule Layer

The rule layer combines the membership output of the related inputs into rules.

$$y_i^3 = \Pi_c^m x^3 c_i \qquad (17)$$

There are total seven fuzzy sets which multiplication results 3072 combination. This is the maximum number of rules that can be defined. Although every combination is not feasible for flight, defining this many rules manually is very difficult even using the domain information of experienced pilots. There are certain cases that one can say a rule cannot be defined for a specific condition but also combinations that none can be sure if can be neglected.

In [2] it is proposed to apply a layered rule base system. This is a starting point to eliminate certain conditions. Rules are analyzed with domain information according to following objectives.

Physical Surveillance: When physical conditions force a certain movement and there is no other choice only movement is to survive. If both altitude and velocity are very low, then we should increase the speed first and altitude after not to fall into stall status. If altitude is high then we should increase speed or trade speed against altitude. Another condition is dive recovery where there is no other choice except maximum g pullup.

if h is small and θ<0 : diverecovery

BFM in progress: If already running a BFM keep with its logic, else check if current geometry forces a defensive maneuver.

if (120<λ<180 and 60<η<120 and |h|<1000 and r<6000) : break

if (120<λ<180 and 60<η<120 and h1<h2 and r < 3500) : verticalbreak

if (120<λ<180 and 60<η<120 and h1>h2 and r < 3500) : oppositeturn

if (120<λ<180 and η<60) : climb

Pointing algorithm decision: When range is beyond visual range, it is better to cruise to an advantageous point rather than deciding a maneuver or when at a high probability of shooting position in close range, small *λ* and *η* angles.

if (λ<45 and η<30) : pointing

Lead/lag algorithm decision: When there is a possible shooting position or risk of being shot, tune *η* to keep trace of required angle. This is the condition when we are in a very close range and in the turn circle of other aircraft.

if (λ<30 and η<30) : lead pursuit

if (λ>30 and 30<η<90-λ) : pure pursuit

if (λ>30 and 90-λ<η<180-λ) : lag pursuit

if (λ>30 and 180-λ<η) : pursuit

if (λ<60 and η_2*<-5 and |φ|<30 and* $v_1<v_2$ *and φ<φ) : negativeG*

if (λ<60 and η_2*<-5 and |φ|<30 and* $v_1<v_2$ *and φ>φ) : intercept*

Load factor determination: Load factor is how much g will be applied to the aircraft. High load factor causes sharp turns but loose air speed where low load factor preserves air speed but with longer turn radius. For implementing limited number of maneuvers in this paper, decisions based on load factor are not included in the work.

Evasive maneuver decision: During recovery or overshoot conditions, throttle settings on engine are set to idle if speed is over corner velocity or other aircraft is in lead pursuit. Otherwise throttle is set to maximum power.

if (λ>120 and η>120) : evasion

These rules are derived from domain information [13] and a sample rule-based air combat algorithm in [2]. Implementation of maneuvers dictates changes in angles and speeds until certain conditions satisfied and will be covered in next sub-sections. Final list of rules studied in this work is provided in Table 3.

TABLE 3 FINAL RULE LIST

Condition	Resulting Maneuver
h is small and θ<0	diverecovery
120<λ<180 and η<60	climb
120<λ<180 and 60<η<120 and |h|<1000ft and r<6000ft	break
120<λ<180 and 60<η<120 and h1<h2 and r < 3500ft	verticalbreak
120<λ<180 and 60<η<120 and h1>h2 and r < 3500ft	oppositeturn
λ<45 and η<30	pointing
λ<30 and η<30	lead pursuit
λ>30 and 30<η<90-λ	pure pursuit
λ>30 and 90-λ<η<180-λ	lag pursuit
λ>30 and 180-λ<η	pursuit
λ<60 and η_2<-5 and |φ|<30 and v_1<v_2 and φ<φ	negative
λ<60 and η_2<-5 and |φ|<30 and v_1<v_2 and φ>φ	intercept
λ>120 and η>120	evasion

IV. Normalization Layer

Normalization layer runs after firing rules in previous layer. When more than one rule is fired, this layer detects the effect of each firing rule in the resulting output. For the simplicity, one node is defined for each rule and weight of that rule is set to 1. Weight for the other rules is set to 0.001.

$$y_i^4 = \frac{x_d^4}{\sum x_d^4} \quad (18)$$

V. Defuzzification Layer

This layer finalizes the output of the system. In air combat case, one node is defined for implementation of each selected maneuver. Since movements have 6 variables for duration, mode, velocity, pitch change, heading change, roll change, the ANFIS should have 1 output for each variable. Duration is neglected. Coefficients not applicable to the movement are set to 0.001. For constant states like velocity, it is set to 1.

$$y_i^5 = x_i^5(k_{i0} + k_{i1}x_1 + \cdots + k_{in}x_n) \quad (19)$$

Implementations of each maneuver are explained below.

Dive recovery: If dive recovery is selected, than θ is already less than zero and should be recovered to zero. Pitch Yaw Transition is selected as the movement mode.

σ=4, θ'=-θ, ψ'=ψ

Climb: Climbing as a movement of a maneuver is decided as quarters. Another objective is not to lose energy, thus using low g or loop radius. Loop is selected as movement mode.

σ=3, θ'=π/2, v=v

Break: Break is forcing the second aircraft to saturate η shooting. Maneuver starts with a roll towards second aircraft. General application is sin(φ)=|h|/range.

σ=5, φ=asin($|h_1$-$h_2|$/r)

Vertical break: Vertical break is a defensive maneuver to climb up when opponent is diving into you. It has the same logic as climb while range is less than 3500 feet.

σ=3, θ'=π/2, v=v

TABLE 4 DE-FUZZIFICATION EQUATIONS

Maneuver	De-fuzzification output				
	σ	*φ*	*θ'*	*ψ'*	*V*
Dive recovery	4		-θ	ψ	
Climb	3		π/2		V_1
Break	5	asin(|h|/r)			
Vertical break	3	π/2			V_1
Opposite turn	2			π	V_1
Pointing	4		asin((h)/r)	η_1	
Lead Pursuit	2			ψ'_2	0.9*V_2
Pure Pursuit	2			η_2-η_1	V_2
Lag Pursuit	2			η_1	V_1
Pursuit	2			η_2-η_1	V_1
Negative G	5	φ-π*sign(φ)			
Intercept	4		asin((h)/r)	η_1	
Evasion	5	φ-π*sign(φ)/2			

Opposite turn: Turn nose to opponent aircraft at a lower altitude until η<60.

σ=2, ψ'=π, v=v

Pointing: When there is a strong probability to shoot turn the nose directly to the opponent position

σ=4, θ'=asin((h)/r), ψ'=η

Lead Pursuit: Follow opponent keeping nose direction in front of opponent.

σ=2, ψ'= ψ'_2, V=V_2 * 0.9

Pure Pursuit: Follow opponent keeping nose direction in front of opponent.

σ=2, ψ'= η_2-η, V=V_2

Lag Pursuit: Decrease η to keep track of opponent.

σ=2, ψ'= -η, V=V

Pursuit: Pursuit has the same algorithm with the pure pursuit without change in velocity.

σ=2, ψ'= η_2-η, V=V

Negative G: Turn nose to the opponent by making 180 degrees of roll for loosing less energy when φ > 30 degrees.

σ=5, φ= φ-π*sign(φ)

Intercept trajectory: Has the same logic with pointing unless negative G is required.

σ=4, θ'=asin((h)/r), ψ'=η

Evasion: Roll with 90 degrees from current position and break through opponent aircraft.

σ=5, φ= φ-π*sign(φ)/2

Total list of coefficients for equations of defuzzification layer is listed in Table 4 that includes the first movement of the related maneuver and initial angular changes. Defuzzification layer has 5 times more nodes for mode, velocity, pitch, yaw and roll. The generic view of the multiple output ANFIS is show in Figure 2.

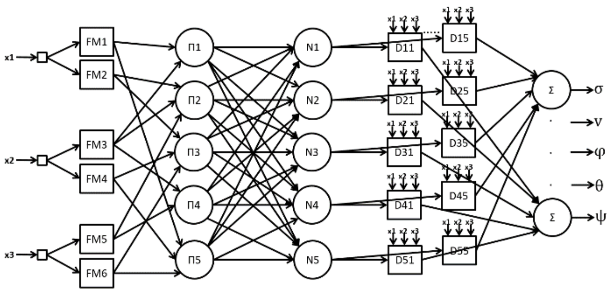

Figure 2 Multiple Output ANFIS

B. Training

I. Activation Functions

General form of a bell-shaped function of training is below. Choosing s and r values depends on the number and quality of the samples. Initially s is chosen 1 and r is chosen 10.

$$y = \frac{1}{1+\left[\left(x-\frac{s}{r}\right)^2\right]^t} \tag{20}$$

II. Corpus Content

A sample of the training data source is listed in Table 5. The table includes all data. Some columns are not used during training but included in final level for maneuver implementation. Learning corpus consists of 10 F-16 combat sorties. These sorties are decomposed into approximately 700 moves per hour. Only ~25 minutes of the flight contains combat maneuvers, rest is cruising to combat area. A clear combat scenario contains about 30 moves for two aircrafts. Of those 150 combat moves, 120 of them are used for learning and 30 for testing.

TABLE 5 SAMPLE CORPUS DATA

h_1	θ_1	v_1	h_2	θ_2	v_2	q_1	v_1	θ_1	ψ_1	q_2	v_2	θ_2	ψ_2	r	λ	η_1	η_2
1000	-50	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	135	50	..
5000	..	..	5500	..	..	..	..	..	..	..	..	..	..	5000	135	90	..
5000	..	..	5500	..	..	..	..	..	..	..	..	..	..	3000	25	25	..
5500	..	..	5000	..	..	..	..	..	..	..	..	..	..	3000	45	90	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	40	25	..

C. Test

Focusing deeply into the flight data, we find out that majority of combat maneuvers include 3D moves. 3D move is performed by both pitching and rolling at the same time and results a helical path changing direction in all 3 axis. The limited number of rules in the ANFIS architecture seems compliant with the sample data. Visually observing the result, a sample and simple maneuver execution is printed in Figure 3 and change in the angles, reference values and control signals for pitching are presented in Figure 4.

For simulation purposes, a high fidelity F-16 model is used as defined in [20] which implements all the constraints and aerodynamic coefficients based on experimental result of [19].

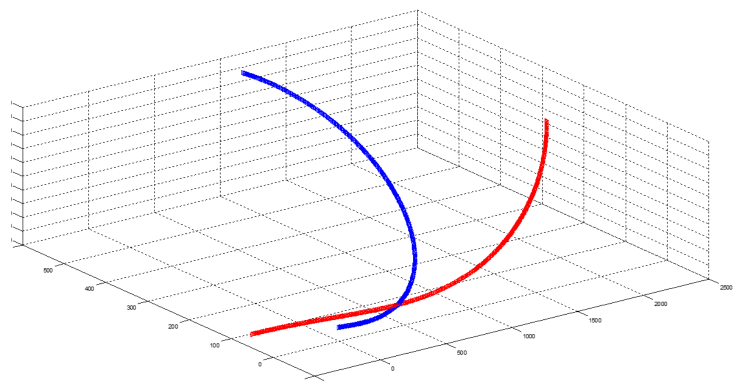

Figure 3 Breaking route

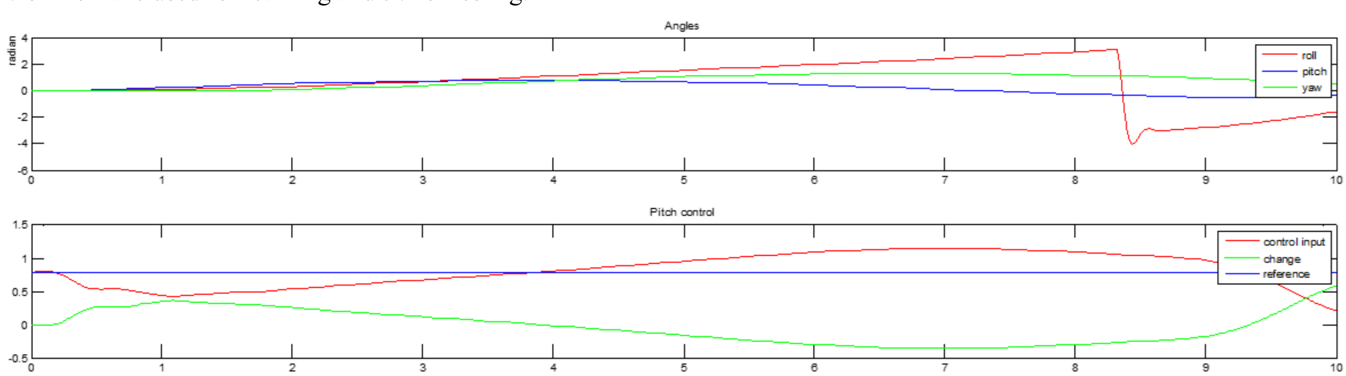

Figure 4 Angles and signals

As a simple maneuver, the break rule is sampled. Rule for break is: if λ is pursuit and η is defensive than mode is roll, angle is asin(h/r) following mode is loop, angle is π/2 for vertical break and mode is turn, angle is π for break resulting a helical path upwards. The F-16 model trimmed at 6000 meters and velocity at 0.6 mach results an evasive route against a high speed attacker in Figure 3. The simulation run for 10 seconds. The aircraft completes the maneuver at 8 seconds. After completion, aircraft recovers its roll position back and corrects nose direction resulting π/2 heading change in total.

V. Results

This paper does not focus on low level control of aircraft. The control mechanism at low level effects on the success of the solution. There are two matured solutions in [10] and [17] for this purpose. Also we should bear in mind that the proposed action in high level control and actual action at low level is different. For example making a 90 degrees of turn at high speeds is performed by rolling slightly and performing a turn with some looping. The lift vector contributes to move in horizontal plane with $sin(\varphi)$ and some looping is required to preserve altitude and change in direction as well with the turn. So low level controller acts different than high level decisions.

The test results justify that climb and vertical break maneuvers have confusing decisions since maneuver logic is similar even though decision arguments vary. Same occurs in evasion and break maneuvers since both maneuvers include 180 degrees of move towards the following aircraft. These two maneuver sets should be unified for decision. Lead, pure and lag pursuit maneuvers are decided perfectly on close altitudes. Defensive maneuvers break or evasion is also found successful.

The only offensive maneuver utilized is the pointing algorithm where nose is simply pointed to the opponent. But in the cover of this paper it is only aimed to show the usability of the technique, so advanced moves like barrel roll attack, other quarter plane or high/low yoyo maneuvers are not be studied.

Other technique of using 5 outputs for the ANFIS structure also performed well. First 3 layers of the network is combined in 1 decision model. After the normalization layer 5 nodes are designed for each maneuver. Outputs of same kind of nodes are summed at the final layer resulting 5 outputs.

VI. Conclusion and Future Work

This paper examines rules for 13 maneuvers and starts the initial movement of related maneuver. There are more advanced fighter maneuvers that should be learned from air combat domain. This paper shows whether the maneuver logic can be defined based on the angular and relative geometry and learned by proposed technique. A method for deriving more rules from combat can be designed using decision tree models.

Learning corpus of the data is also another challenge for this work. Only 10 sorties of sample F-16 data are utilized for the whole corpus. But for learning advance maneuvers, more data from real combat flights is required. Generating training data in the simulation environment is also possible but the rare resource is advanced combat pilots.

Preferences

[1] J.S.R. Jang, "Adaptive Network Based Fuzzy Inference System", *IEEE Trans. On Systems Man. and Cybemetics*, Vol.23, No.3, (1993), p 665-685.

[2] Burgin, G., Sidor, L. "Rule-based air combat simulation", Tech. rep., NASA, CR-4160, 1988.

[3] Andriambololona M., Lefeuvre P. "Implementing a dogfight artificial pilot", PhD Thesis, Delft University, 2003.

[4] McManus J., Goodrich K. "Application of artificial intelligence (AI) programming techniques to tactical guidance for fighter aircraft", AIAA Guidance, Navigation and Control Conference, August 14-16 1989.

[5] McMahon D.C. "A neural network trained to select aircraft maneuvers during air combat: a comparison of network and rule based performance", University of Mexico, Department of Psychology, 1990.

[6] Rodin E.Y., Amin S.M. "Maneuver prediction in air combat via artificial neural networks, Computers & Mathematics with Applications", vol. 24,1992, p.95-112.

[7] Konakoğlu E., Kaynak O. "Autonomous flight control of unmanned air vehicles by use of ANFIS", MSc Thesis, ASTIN, 2006.

[8] Çetin Ö., Kaynak O. "Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles", Expert Systems with Applications vol 37, p 1229-1234, 2010.

[9] Karli M., Efe M.Ö., Sever H. "Close range one to one air combat maneuvering for autonomous UAV", 8th Ankara International Aerospace Conference, 10-12 September 2015, Ankara, Turkey (AIAC2015-046).

[10] Üre N.K., İnalhan G. "Autonomous control of unmanned combat air vehicles", *IEEE Control System Magazine*, 74-95, October 2012.

[11] Üre N.K., İnalhan G., "Feasible agile maneuver identification and generation algorithms on multi modal control framework", *AIAA Guidance, Navigation, and Control Conference* 10-13 August 2009, Chicago, IL.

[12] Sugeno, M., "Industrial applications of fuzzy control", Elsevier Science Pub. Co., 1985.

[13] Shaw, R., "Fighter Combat Tactics and Maneuvering", Naval Institute Press, Annapolis, Maryland, 1985.

[14] Yigun Dong and Jianliang Ai. "Trial Input Method and Own-Aircraft State Prediction in Autonomous Air Combat", Journal of Aircraft, Vol. 49, No. 3 (2012), pp. 947-954.

[15] Yiqun Dong, Jiong Huang, and Jianliang Ai. "Visual Perception-Based Target Aircraft Movement Prediction for Autonomous Air Combat", Journal of Aircraft, Vol. 52, No. 2 (2015), pp. 538-552.

[16] Yiqun Dong, Xizhong Yang, Jianliang AI. "Research on Enemy Aircraft Bank Angle Estimation for Autonomous Air Combat", 29th Congress of the International Council of the Aeronautical Sciences. St. Petersburg, Russia. Sep. 7-12 2014.

[17] Kasnakoglu C. "Scheduled smooth MIMO robust control of aircraft verified through blade element SIL testing", Transactions of the Institute of Measurement and Control, (2016), DOI: 10.1177/0142331216661760.

[18] Cook M.V. "Flight dynamics principles", Butterworth Heinemann, 1997.

[19] Nguyen L.T., Ogburn M.E., Gilbert W.P., Kibler K.S., Brown P.W. and Deal P.L. "Simulator study of stall post-stall characteristics of a fighter airplane with relaxed longitudinal static stability". Technical report, NASA, Langley Research Center, Dec 1979.

[20] Russel R.S. "Non-linear F-16 simulations using Simulink and matlab". Technical report, University of Minnesota, 2003.