Extracting Basic Fighter Maneuvers from Actual Flight Data

Mustafa Karli, Mehmet Önder Efe, and Hayri Sever
Department of Computer Engineering, Hacettepe University, Beytepe Ankara, 06800, Turkey
Email: mustafa.karli@hacettepe.edu.tr, {onderefe, hayri.sever}@gmail.com

***Abstract*—Air combat maneuvers are very complex actions performed by agile aircrafts. Extracting critical maneuvers from a combat scenario in a structured format has many advantages like teaching maneuvers to the unmanned systems, evaluating pilot performance or analyzing possible combat scenarios. Basic Fighter Maneuvers are special maneuvers that are building blocks of combat fighting. This article proposes a methodology to identify pre-defined movements, match well-known combat maneuvers in a real flight of agile combat aircraft and build a feasible corpus to use this data for machine learning. The claims of the paper are justified by the simulation results.**

***Index Terms*—flight parsing, basic fighter maneuvers, air combat**

I. Introduction

Basic Fighter Maneuvers are executed by agile aircraft during "Within Visual Range" in defensive or offensive positions or missile evacuation. For training artificial systems like UAVs, besides domain information [1], one of the best learning sources is real flight information of manned air vehicles.

There are auto-pilot designs and combat support systems in the literature like rule based systems [2], influence diagrams [3], human cognitive models [4] and Artificial Intelligence (AI) techniques for air combat maneuvering [5]. There is a comparison of artificial neural networks and rule based system in [6] and maneuver prediction in [7] to support human combat pilots. Autonomous control of UAV is designed using ANFIS in [8]. There is an additional design by ANFIS in [9] including a predefined flight path. Both ANFIS design is for a single UAV without combat fighting. Assuming air combat as a pursuer-evader game and optimizing using approximation and dynamic programming is presented in [10]. Composing a flight trajectory in terms of seven primitive actions and a way point decomposition algorithm is presented in [11]. There is also a sliding mode controller design by the same author proposed in [12] which excludes an arbitrary movement mode.

Our work advances the subject area in terms of representing a maneuver by movement sections instead of many flight parameters and proposes an abstraction stack for flight representation. The real flight data of the agile aircrafts are decomposed into meaningful movement sequences and BFM maneuvers are searched and labeled to be learned by machine learning systems.

This paper is organized as follows: In the next section we introduce the definitive terms of the problem. The third section proposes an abstraction stack for air frame flight representation. Using this abstraction, air operations can be executed from mission planning to physical control layer. The forth section defines the basic fighter maneuvers of close air maneuvers in terms of proposed abstraction. The fifth section defines how real flight information and relative geometry of two fighting aircrafts are decomposed. The sixth section evaluates methods for searching and indexing BFM in flight data and proposes a specific search method. The seventh section discusses the benefits of the proposed approach with simulation results and evaluates the search method. The last section defines the required steps to be performed for machine learning techniques with the concluding remarks.

II. Problem Definition

The objective of an air combat scenario is to move the aircraft into a position where one can shoot the other aircraft or minimize the risk of being shot. This depends on the positional advantage of both aircrafts which depends on the “relative geometry” to each other.

Human pilot control the aircraft using the stick and gas pedal where a series of physical, aerodynamic and atmospheric equations run through propulsion, ailerons, elevators, rudder, wing and platform surface resulting forces and accelerations on 3 dimensions which changes the state of the system. This is a non-linear system control that is also affected by non-deterministic conditions like atmosphere, gravitational changes, varying weight and center of gravity. The air frame has 12 state variables $X = \{n, e, h, v, \varphi, \theta, \psi, \alpha, \beta, P, Q, R\}$ which are north, east, height position, velocity, roll, pitch, heading body axes angles, angular difference of pitch and heading between body and velocity axes and body angular velocity in three dimensions.

Air combat is *using an aircraft as a weapon* and has its own domain rules to learn and practice. In a real air combat, both sides are maneuvering instantly to take advantage. Both sides can be in offensive position while was defensive in the previous action of the engagement. So classical pursuer-evader tactics is not applicable since pursuer only considers pursuing and evader only considers

Manuscript received February 6, 2017; revised June 28, 2017.

doi: 10.18178/lnit.5.1.1-6

evading. Since relative geometry changes instantly, the trajectory cannot be planned for a long period.

With the difficulties of controlling air frame, including the combat logic to the control process makes it even worse where the control logic becomes domain specific. A mechanism is needed to distinguish physical, control, maneuver, tactical and strategic layers from each other.

The domain information about air combat is at tactical and maneuver layer. After distinguishing layers from each other, the experience in real air combat flights should be extracted and used by machine learning techniques to train artificial systems like UAV auto-pilots.

III. Abstraction

The air platform is abstracted by using its interfaces without dealing design details. We assume that the system is controlled in a robust, deterministic and well known way. Abstraction should be *sound*, *complete* and *orthogonal*. Abstracting an air platform is performed in five levels as show in Figure 1.

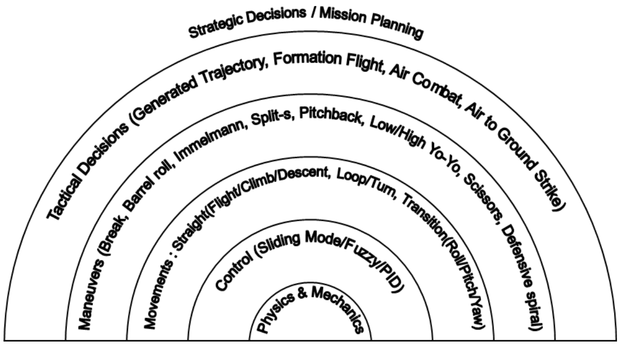

Figure 1. Abstraction stack

A. Control Level

There is a robust air platform that can execute agile maneuvers. The platform is controlled in a stable, robust, optimal way and overcomes unexpected conditions as a non-linear system. The control system accepts reference inputs, generates control signals, handles the dynamics and measures the system output correctly. Control system accepts predefined movement orders with parameters and handles each order separately. Parameters of orders are interpreted as input signals.

B. Movement Level

The movements of an air platform can be states as a set of predefined actions. Every displacement instant of a route is a member of these actions. Movements are building blocks of maneuvers performed by human or artificial pilots.

C. Maneuver Level

Movements are executed sequentially to compose specific maneuvers. Maneuvers are well tested and know action series performed by human to achieve specific objectives. This level covers the domain information that should be learned.

D. Tactical Level

Maneuvers have different objectives at tactical level. In air combat it is not possible to generate a trajectory early, since there are instant decisions to achieve the best advantageous position over the enemy. Formation flights are executed by multiple friendly aircrafts that run formation specific maneuvers. Air to ground strike requires following the path through uncovered radar zones and approaching and releasing the munitions over the target at specific angles and altitude. BVR (Beyond Visual Range) maneuvers are executed to take best advantageous position before close engagement. Tactical decision makers may not understand details of control and movement level, but should make decisions using maneuvers.

E. Strategic Level

This level is where high level objectives are defined to make tactical decisions.

IV. Movement Definition

There are 7 movement definitions in [11] are specific flight or control modes that have three set of variables. First set is mode inputs. These are duration of the mode, velocity and related angular changes. Second set is constraint states that are constant during the mode which have zero deltas. Third set is the driven dynamics which are the set of varying states. Two of the modes do not have duration and velocity inputs since they are transition modes. For the abstraction, second and third set have no meaning. The system is abstracted as a black box which has modal inputs and system states as outputs.

The 3D mode which changes angles in all 3 dimensions is excluded in [12] and pitch yaw transition mode is added instead. But this leads to an undefined mode in real flights. 3D mode may look arbitrary but there is a mode that air platform both rolls and loops at the same time which inputs the angular change in roll and loop and results the change in heading angle like quarter plain maneuvers. In barrel roll attack, pitch changes 360^0 and roll changes 270^0 at the same which results 90^0 of change in heading angle.

Other case occurs during turns. For getting a force on body east and west directions, aircraft rolls and starts turning. Roll is retained back to zero before desired change in the heading achieved. There for the last roll in turn overlaps with the turn. The overlapping time can be quarter of the turn time and leads to miscomputing while checking angle identification matrix.

Level Flight is ignored because climbing with constant or zero pitch angles has no difference in terms of identification. They are grouped in straight path flight.

For transition modes $\Delta t, v$ should not be ignored since there is a significant change of position in high speeds. Six modes have self-explaining names and are listed below with inputs.

Mode is expressed as below using the inputs as seen in Table I.

$$\sigma = \{q, \Delta t, v, \dot{\varphi}, \dot{\theta}, \dot{\psi}\} \quad (1)$$

TABLE I. MOVEMENT IDENTIFICATION MODE

Mode	Name	Inputs
SP	Straight Path	$\Delta t, v, \theta$
TU	Turn	$\Delta t, v, \theta, \dot{\psi}$
LO	Loop	$\Delta t, v, \dot{\theta}, \dot{\psi}'$
PY	Pitching Yaw	$\Delta t, v, \dot{\theta}, \dot{\psi}$
RO	Roll	$\Delta t, v, \dot{\varphi}$
RP	Rolling Pitch	$\Delta t, \dot{\varphi}, \dot{\theta}$

A flight mode consists of mode label, duration in milliseconds, velocity and desired angular change in roll, pitch and heading.

V. BFM REPRESENTATION

BFMs are building block of fighter tactics that are decided based on relative geometry to another aircraft.

A. *Relative Geometry*

Relative geometry *R* is the range and antenna train angles between two aircrafts and calculated as described in [13].

$$\mathrm{R} = \{\mathrm{r}, \eta_1, \eta_2\} \tag{2}$$

r is the magnitude of range vector between each aircraft and η is the angle between range and velocity vectors of aircrafts.

B. *Maneuvers*

Five sample maneuvers in Table II are decomposed into movements using the definitions in [13]. Unspecified parameters $\Delta t, v$ should be calculated depending on conditions of the flight to achieve the required delta changes. The exact parameters are derived from definitions but angular parameters should be fine-tuned based on *R*.

TABLE II. SAMPLE BFM DECOMPOSITION

BFM	*q*	φ	θ	ψ
Break: Turn sharply across the high speed attacker's flight path.	RO TU			 $\pi/2$
Barrel Roll Attack: Offensive maneuver counter to break. It consists of 360^0 loop and 270^0 roll completed at the same time.	RP RO	$3\pi/2$ $\pi/2$	2π	
Immelmann: Decrease speed, increase altitude, change direction by 180^0.	LO RO	 π	π	
Split-S: Decrease altitude, increase speed, change direction by 180^0.	RO LO	π	 π	
Chandelle: Looks like Immelmann but is executed with a lateral loop using constant pitch angle *y*. Roll angle *x* should be calculated for the desired turn radius.	RO PY TU PY RO	*x* -*x*	 *y* -*y*	 π

Besides these sample BFMs, there are also maneuvers used for formation flight and air to ground strikes. These maneuvers are not included in this paper but can be decomposed using the domain information.

VI. FLIGHT DATA

The flight data of agile airframe is recorded in a device. For example F-16 records 232 columns of binary data with ~40 milliseconds time frames. After noise filtering and combining modes of similar instants, flight data *F* is converted to sequence of modes. Every mode is also labeled with its timestamp.

$$F = \{\sigma_1, \sigma_2, \dots, \sigma_n\} \tag{3}$$

A. *Movement Identification*

The angle identification matrix proposed in [11] is modified as below to include roll angle. Pitch and heading angle is removed since their values are not important for identification, but the change is important. The notation is kept same as 0 for zero and T as time varying. Final matrix *I* is presented in Table III.

TABLE III. ANGLE IDENTIFICATION MATRIX

q	$\dot{\varphi}_r$	$\dot{\theta}_w$	$\dot{\psi}_w$
SP	0	0	0
TU	0	0	T
LO	0	T	0
PT	0	T	T
RO	T	0	-
RP	T	T	-

Algorithm 1 is used to convert flight data to modal sequence where $A_i = \{A_{\varphi i} A_{\theta i} A_{\psi i}\}$ is angular change matrix for each record, *Th* is delta threshold and *Td* is data threshold for each angle.

Algorithm 1. Movement Decomposition Algorithm

CALCULATE missing state variables and $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ for each record in the wind axis.
SMOOTH $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ for noise filter.
    REPEAT 2 times
        $\dot{\alpha}_{2i} \leftarrow \dfrac{\dot{\alpha}_{2i} + \dot{\alpha}_{2i+1}}{2}$
        $\dot{\alpha}_{2i+1} \leftarrow \dfrac{\dot{\alpha}_{2i} + \dot{\alpha}_{2i+1}}{2}$
    REPEAT 1 times
        $\dot{\alpha}_i \leftarrow \dfrac{\dot{\alpha}_{i-4} + \dot{\alpha}_{i-3} + \dot{\alpha}_{i-2} + \dot{\alpha}_{i-1} + \dot{\alpha}_i}{5}$
REPEAT for every record *i*.
    REPEAT for α in $\dot{\varphi}, \dot{\theta}, \dot{\psi}$
      IF $\dot{\alpha} > Th_\alpha$
          $m \leftarrow I$
          IF $\sum_m^i \dot{\alpha} > Td_\alpha$
              $A_{\alpha\, m..i} \leftarrow T$
          ELSE
              $A_{\alpha\, m..i} \leftarrow 0$
      ELSE
          $A_{\alpha\, m..i} \leftarrow 0$
FOR each record *i* $q_i \leftarrow \mathrm{match}(A_i, i)$
FOR each sequential group q at [*s*..*e*] create mode σ_i
    $v_i \leftarrow \sum_s^e v$/(e-s), $\Delta t_i \leftarrow$ e-s, $q_i \leftarrow q_s$
    CASE q_i
        SP: $\theta_i \leftarrow \sum_s^e \theta$/(e-s)
        TU: $\theta_i \leftarrow \sum_s^e \theta$/(e-s), $\psi_i \leftarrow \sum_s^e \dot{\psi}$
        LO: $\theta_i \leftarrow \sum_s^e \dot{\theta}$, $\psi_i \leftarrow \sum_s^e \psi$/(e-s)
        PT: $\theta_i \leftarrow \sum_s^e \dot{\theta}$, $\psi_i \leftarrow \sum_s^e \dot{\psi}$
        RO: $\varphi_i \leftarrow \sum_s^e \dot{\varphi}$
        RP: $\varphi_i \leftarrow \sum_s^e \dot{\varphi}$, $\theta_i \leftarrow \sum_s^e \dot{\theta}$

After creating movements, below corrections are made to unify repeating modes or eliminate noisy movements;

1) Sequential roll and turn movements are combined into one roll and turn couple if roll and turn directions do not change.

2) Turn or loop between two RP modes are ignored and merged into the RP mode if RP mode is rolling and looping at the same direction.

3) Movements with total angle below data threshold are ignored and merged into the previous movements.

This method guaranties that every instant is labeled for a movement mode and there is no repeating movement sequentially.

B. BFM Extraction

To extract BFM information, there should be two flights to calculate relative geometry using time stamp to synchronize with each other. BFM are executed in WVR (Within Visual Range) conditions. So the conditions in BVR (Beyond Visual Range) are ignored. For learning purposes, some values in the state vector of both sides are also included. These are altitude, pitch angle and velocity.

$$S=[h_1\theta_1 v_1, h_2\theta_2 v_2, \ldots,] \tag{4}$$

Input data for BFM learning contains; states S_1 and S_2 with relative geometry R, mode σ is executed.

$$L = \{S_{11}S_{21}R_1\sigma_1, \ldots, S_{1i}S_{2i}R_i\sigma_i\} \tag{5}$$

This learning set is based on movements only. Instant decisions for simple movement selection can be learned with this data. But executing a maneuver cannot be based on simple movement selections. Indeed movements should be executed based on selected maneuver.

VII. Pattern Matching

Searching a BFM in a flight can be simplified as searching a sequence of movements in a longer sequence of movements. There are different pattern search algorithms [14] that can be used. Three of search methods are examined.

A. Sequential Search

The movements of BFM are serially compared with the flight movements beginning from the first movement. When there is a mismatch search is assumed to be not matching and restarted from last position. This method has draw-backs. First there may be repeating sequences in the BFM and when there is a mismatch repeating sequence has to be looked back. Other draw-back occurs for other BFMs. Searching cost of one BFM is multiplied for each BFM to search. Sequentially searching the series of BFM modes is the worst method to use.

B. Automaton

BFM is defined as an automaton system. The regular expression of the deterministic finite automaton is composed of six modes as states and transitions between them. Every BFM has a modal sequence and DFA is the mixture of all BFM definitions. The alphabet contains the token $\{q_1 .. q_6\}$ and lexemes are formed from the alphabet. Search problem is solved by recognizing the tokens of BFM language.

Automaton system solves the problem of searching multiple BFMs in a single run. But if other maneuvers are added to BFM set, in other words if BFM language is modified, the search should be repeated over the whole set of flights. It seems very likely to happen because besides air combat maneuvers, there should always be new maneuver sets for other purposes. Also newly created BFM may occur as well. Only one change modifies the whole language definition. This leads to impractical conditions when there is a huge set of flights.

C. 3 Bit Indexing

We propose a specific identification of each mode sequence. There are 6 modes. All modes can be expressed by 3 bits of data. 32 bits of integer can hold 10 maneuver sequence with 30 bits. Every sequence of flight also holds additional 32 bit integer value that stores previous 10 modes (PTM). Modes are stored from least significant bit. Left 2 bits are not used and set to zero. PTM data format for 32 bit value is in Figure 2.

		q_i	q_{i-1}	q_{i-2}	q_{i-3}	q_{i-4}	q_{i-5}	q_{i-6}	q_{i-7}	q_{i-8}	q_{i-9}

Figure 2. 32 bit PTM data format

Flight PTM labeling algorithm is in Algorithm 2.

Algorithm 2. Flight PTM Labeling

$ptm_0 = 0$
FOR each mode in flight
    $ptm_i = (ptm_{i-1}$ shr 3) + (q_i shl 27)

PTM is also calculated for each BFM. When there are less than 10 moves, PTM is shifted left 3 bits for each missing moves. If BFM has more than 10 moves, only first 10 are considered. This is assumed to be a rare condition and if occurs searching the remaining modes sequentially would not have much cost after matching the first 10 moves. The pseudo algorithm of labeling BFM *b* is in Algorithm 3.

Algorithm 3. BFM PTM Labeling

$ptm_b = 0$
$mask_b = 0$
FOR each of first 10 modes in *b*
    $ptm_b = (ptm_b$ shr 3) + (q_i shl 27)
    $mask_b = (mask_b$ shr 3) + 0x3C000000

Both algorithms are straight forward and executed in O(n) time only once for each flight. This method does not search BFM on flight sequence but labels each mode with the previous 10 sequence. The label is integer value where it can easily be indexed, hashed or used in mathematical or bitwise operations. Previous maneuvers are stored starting from left (first is at left most bits) so BFM can be searched with value indexing in logarithmic (like B tree in data access mechanisms) times, hashing in single access or whole flight can be indexed in O(n) time with radix sort [14] using 3 bit bucket size. Instants of specific BFM can be listed and stored separately for later use as well. A simple match command is as below.

$$\text{boolean match}_{ib} = \text{ptm}_b \text{ equals } (\text{ptm}_i \text{ and mask}_b) \quad (6)$$

After indexing air combat scenario, the learning data about BFM executions includes following information. While states of both air frames are S_1 and S_2 and relative geometry is R, B is selected as BFM.

$$C = \{S_{11}S_{21}R_1B_1, \dots, S_{1i}S_{2i}R_iB_i\} \quad (7)$$

The final learning corpus is based on maneuvers instead of movements. The corpus includes maneuver selections based on states of both aircrafts and relative geometry. Both states include 3 variables for each and with additional 3 variables of relative geometry, 9 values are used to identify the selected maneuver.

VIII. RESULTS

The extraction run on 71 minutes 1x1 BFM training flight which was about 104,000 lines of 40 MB binary data, 25 rows per second for each aircraft. This binary data is extracted from recording device of flight computer. During extraction only 12 out of 256 variables are read. Also records before take-off and after landing are neglected. Aircrafts flying together in the same exercise are supposed to have synchronized time.

Two flights are decomposed into average 750 modal sequences consuming about 40 KB in size for each. There were 22 engagements between two aircrafts. Sample engagement labeled as "Barrel Roll Attack" is examined below (see Fig. 3).

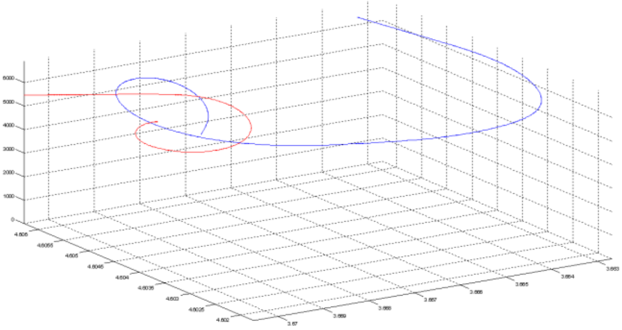

Figure 3. Barrel roll attack (altitude, latitude, longitude)

Start and end points are 30 seconds before and after maneuver for a better look at the initial and final conditions.

Barrel Roll Attack is a difficult offensive quarter plain maneuver. While defensive aircraft makes a sharp break against a high speed enemy, offensive aircraft should not slide, loose energy or stop tracking. So offensive aircraft makes a complete loop with 3/4 roll and additional 1/4 roll after finishing the loop. This is the formal definition of the maneuver in [1]. But in real world defensive pilot can break in any degrees. Offensive pilot should change the total roll during the loop to turn the wind axis to the exact enemy path. This decision is made while looping is at $\pi/2$. The pilot has milliseconds to make this decision. Generally it is a reflexive action. Parsing the movements gives the opportunity to focus on these instants. For artificial decisions after identifying the right maneuver to execute, modal inputs of the individual movements should be calculated, learned or purified using the relative geometry during the movement. (See Fig. 4)

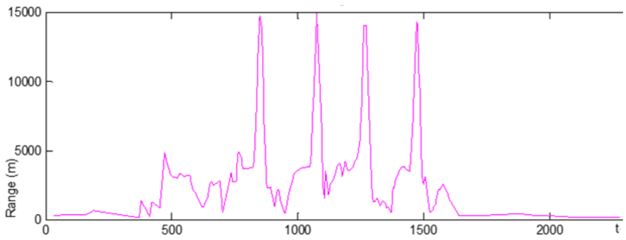

Figure 4. Range (m) vs time (s)

The range between blue and red aircrafts keeps neutral during the cruise mode. Before starting BFM exercises both aircraft depart from each other to initial positions of the combat. Increase in the range shows the BFM start instants. (See Fig. 5)

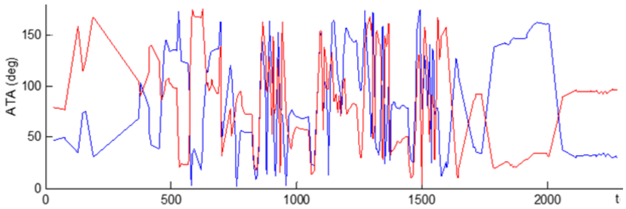

Figure 5. Antenna train angles (degree) vs time (s)

Lower antenna train angle has more advantage on the other. 0^0 angle is the ultimate shooting position where the enemy is absolutely in your flight path. 180^0 degree of ATA means that enemy is absolutely behind you. ATA in the cruise mode is neglected.

During exercise ATA switches according to the success of BFM execution. It can also be observed from the graphic how positional advantage changes in air combat timeline after execution of a movement (see Fig. 6).

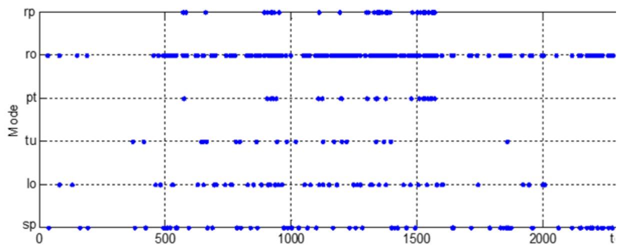

Figure 6. Movement modes vs time (s)

Analysis on the number of movements during the combat flight timeline shows that, while both aircrafts are on cruise mode flying to the training airfield; straight path, turn and roll maneuvers are executed to follow the pre-defined path and looping is not preferred. Rolling pitch movement is almost never used in cruise mode.

When range increases or both sides depart from each other to take initial positions of air combat, rolling pitch and loop movements are used a lot. This contradicts with the claims in [12], that it is rarely preferred to control all three angles (roll pitch heading) simultaneously in air combat.

IX. Application Areas

A. Pilot Performance Assesment

Evaluating a combat flight in de-briefing process is time consuming and difficult. Besides trainer pilots are rare resources. One hour flight requires two hours of DVR replay and evaluation.

First contribution of this paper is determination of the exact point in time of maneuver executions. This helps to focus on combat instants and minimize evaluation time.

Second contribution is to score how successful a trainee has flown a maneuver. This can be done comparing the flight data with the perfect maneuver definition or checking the positional advantage during the maneuver.

Third contribution is at statistical level. The improvement of pilot during the training process can be observed and trainers can decide which type of maneuvers to pay more attention.

B. IT Based Air Combat Assistance

Machine learning mechanisms can be utilized using the real flight data and decisions can be made in real time to propose the correct maneuver to the pilot. This proposal not only consists of movements of maneuver, but also modal inputs should be calculated as well.

There are different types of applicable machine learning techniques but a common requirement is data sets. Data sets should be collected from real air combat scenarios and divided into two as training and test set. Original state variables of an aircraft are very complicated and difficult to use in machine learning. It is transferred to a more understandable and purified format.

Forth contribution is to propose a method to build corpus data for air combat training from actual flight data.

C. UAV Combat Training

Deciding the correct maneuver in real time also offers the opportunity to execute it by UAV. The inputs of the modal sequences in the proposed maneuver can be fine-tuned and integrated into the UAV control system and fully autonomous air combat without human pilot can be achieved.

X. Conclusion and Future Work

Real air combat flight includes valuable and expensive data to learn how to execute a successful air combat. This article proposes a methodology to extract meaningful data for learning process. The learning process should be designed to result two decisions during air combat. First decision is to choose the right maneuver using the state and relative geometry. Second decision is to propose appropriate inputs to the movement sequences of chosen maneuver.

The learning data includes information from two enemy flights. But other conditions exists where there are multiple friendly flights or a single flight. The methodology should be implemented for these cases as well. The format of learning data may be extended or minimized accordingly.

Another future work is designing a data management environment to easily access, query and analyze when huge amounts of flight data is gathered from various kinds of resources like different types of aircraft flight computers or multiple radar track fusion.

References

[1] R. Shaw, *Fighter Combat Tactics and Maneuvering*, Annapolis, Maryland: Naval Institute Press, 1985.

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

[3] K. Virtanen, J. Karelahti, and T. Raivio, "Modeling air combat by a moving horizon influence diagram game," *Journal of Guidance, Control and Dynamics*, vol. 29, no. 5, Sep.-Oct. 2006.

[4] M. Andriambololona and P. Lefeuvre, "Implementing a dogfight artificial pilot," PhD thesis, Delft University, 2003.

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

[6] D. C. McMahon, "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.

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

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

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

[10] J. McGrew, J. How, L. Bush, B. Williams, and N. Roy, "Air combat strategy using approximate dynamic programming," *Journal of Guidance, Control and Dynamics*, vol. 33, no. 5, pp. 1641-1654, Sept.-Oct. 2010.

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

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

[13] M. Karli, M. Ö. Efe, and H. Sever, "Close range one to one air combat maneuvering for autonomous UAV," in *Proc. 8th Ankara International Aerospace Conference*, Ankara, Turkey, 10-12 September 2015.

[14] D. E. Knuth, *Sorting and Searching. The Art of Computer Programming*, Reading MA USA: Addison-Wesley, 1973.

Mustafa Karli was born in Turkey in 1973 and completed BcS in Hacettepe University Department of Computer Science in 1996, MsD in Aeronautics and Space Technologies Institute in 2010. He is still PhD student in Hacettepe University Department of Computer Science. His main area of interest is software technologies for military.

He has been working as Turkish Air Forces officer since his first graduate. He has experience on different levels of information system management like software developer, system analyze, design, architect and administration. Lt. Col. Karli is current working as the commanding officer of a military software project. His current research area is close range air combat for UAV systems.