Research on Application of LSTM-QDN in Intelligent Air Combat Simulation

Dongyuan Hu, Jialiang Zuo*, Wanze Zheng, Ze Zhang, Yu Zhao and Qiang Zhang
Air Force Engineering University

*Corresponding author email: jialnzuo@163.com

Abstract. Aiming at the problem of the lack of intelligence of virtual machine opponents in the human-machine confrontation semi-physical simulation environment, it is proposed to apply the deep reinforcement learning method into tactical making-decision for building an AI virtual pilot with self-confrontation and self-learning ability. First, flight dynamics and kinematics are used to build basic flight models in the simulation environment, and a missile attack area is established for weapon model; Second, inspired by the framework of interaction between the agent and the environment in reinforcement learning, a tactical decision architecture for flight agent based on the one-to-one tactical confrontation process is organized. Finally, the improved DQN method is used to fit the value function in the continuous state space, and the network training is completed by means of agent self-antagonism and human-machine confrontation. the well-trained AI model can undertake the role of virtual opponents in human-machine confrontation environment, and shows a certain degree of intelligence in the confrontation process with pilots.

1. Introduction
With the rapid development of information technology and intelligent technology, the trend of air combat confrontation towards unmanned and intelligent development is becoming more and more crucial [1-3]. In future wars, the intelligent combat mode is a kind of dimensionality reduction combat compared with the traditional combat, and the intelligent air combat will gradually be extended from the simulation mode to actual combat.

The half-physical simulation of human-machine confrontation is the origin of intelligent air combat. Compared with traditional flight training, in a simulation environment, virtual AI opponents can assist pilots to complete various tactical mission training, and can greatly reduce training costs. There are two ways to create AI opponent, Expert system optimization method and reinforcement learning. The result in paper [1] shows that expert system method can meet the requirement of time limitation, while it hard to deal with the state out of traditional sample. Kivelevitch[2] introduces a genetic fuzzy trees and made great process in optimization. In recent years, the successful application of deep reinforcement learning [4] in complex problems such as AlfaGo, AlfaGo Zero, and AlfaStar, makes it possible to use reinforcement learning methods to solve the problem of air combat decision making. Artificial intelligence technology has also become the focus of research in the field of intelligent air combat. Yang[5] and Liu[6] introduce deep reinforcement learning into UCAV combat, while the framework is not suitable for pilot confrontation.

In this paper we will form a confrontation environment and a decision process with the use of reinforcement learning. Also, the virtual agent worked as an AI pilot will be create to realize intelligent
tactical decision. The AI pilot can complete the corresponding tactical actions to avoid the enemy's tracking at a disadvantage and gradually accumulate its own advantages.

2. Design of Simulation Environment for Air Combat

2.1. Flight Motion Model

The flight motion model is the basic model in the air combat environment. It can perform the state transition of the agent at adjacent moments, and it mainly includes the kinematic model and the dynamic model. In an over-the-horizon air combat confrontation, the main focus is on the distance, speed, and angle of entry between enemy and our aircraft. There is no need to consider the attitude angle and flight control issues too much, so the three-degree-of-freedom model can meet the requirements.

The ground coordinate system is an inertial coordinate system, as shown in Figure 1. Based on the ground coordinate system, the mass center motion equations of the aircraft is constructed as followed:

\[
\begin{align*}
\dot{x} &= v \cos \gamma \sin \psi \\
\dot{y} &= v \cos \gamma \cos \psi \\
\dot{z} &= v \sin \gamma
\end{align*}
\]  (1)

The position of the aircraft in the inertial coordinate system, represents the rate of change of the position on each axis, represents the value of the aircraft's speed in the ground coordinate system, and represents the track inclination and track deflection angle of the target aircraft.

\[\text{Figure 1. Three-degree-of-freedom motion model of the aircraft.} \]

2.2. Flight Control Model

Unlike six degrees of freedom, which requires four control variables, the three-degree-of-freedom model can complete various types of aircraft movements with only three control variables. During the flight, the overload formed by the thrust and lift acting on the aircraft is decomposed into tangential overload and normal overload, and the dynamic equation of the aircraft's center of mass is constructed as follows:

\[
\begin{align*}
\dot{v} &= g (n_z - \sin \gamma)  \\
\dot{\gamma} &= \frac{g}{v} (n_z \cos \mu - \cos \gamma)  \\
\dot{\psi} &= \frac{gn_z \sin \mu}{v \cos \gamma}
\end{align*}
\]  (2)

Where \(\dot{v}, \dot{\gamma}, \dot{\psi}\) respectively represent the rate of change of velocity, track pitch angle and track yaw angle over time, \(\mu\) represents the roll angle of aircraft speed, \(n_z\) and \(n_z\) respectively represent the tangential and normal overloads of aircraft.

2.3. Relative Motion Model
In simulated confrontation training, pilots often judge the relative situation based on the positions of the enemy and ourselves, and then make tactical decisions. The relative motion model can integrate the independent motion models of the enemy and ourselves to calculate the relative motion parameters. The coordinate position of the red square in the ground coordinate system is $P_R = (x_R, y_R, z_R)$, where $x_R$, $y_R$, and $z_R$ are its coordinates. The coordinate position of the blue square in the same system is $P_B = (x_B, y_B, z_B)$, where $x_B$, $y_B$, and $z_B$ are its coordinates. The relative situation of the two sides is shown in Figure 3.

![Figure 2. The spatial position and situation of the two parties in the air combat.](image)

The distance formula between the red square and the blue square is:

$$
\Delta x = x_B - x_R \\
\Delta y = y_B - y_R \\
\Delta z = z_B - z_R \\
r = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}
$$

Based on the red plane, the velocity vector of the red plane is converted to the ground coordinate system, $\mathbf{v}_R$ represents the velocity vector of the red plane, and $\mathbf{v}_B$ is the velocity vector of the blue plane. The relative distance vector which points from the center of mass of the red plane to the center of mass of the blue plane is:

$$
\mathbf{r}_{RB} = \mathbf{r}_e = (x_R - x_B, y_R - y_B, z_R - z_B)
$$

The azimuth of the red plane is:

$$
\varphi_R = \arccos \left( \frac{\mathbf{r}_{RB} \cdot \mathbf{v}_R}{|\mathbf{r}_{RB}| |\mathbf{v}_R|} \right)
$$

The entry angle of the target (blue side) is:

$$
\psi_B = \arccos \left( \frac{\mathbf{r}_{RB} \cdot \mathbf{v}_B}{|\mathbf{r}_{RB}| |\mathbf{v}_B|} \right)
$$

The angle between the two speed vectors is:

$$
\eta_{RB} = \arccos \left( \frac{\mathbf{v}_R \cdot \mathbf{v}_B}{|\mathbf{v}_R| |\mathbf{v}_B|} \right)
$$

### 2.4. Missile Attack Model

The missile attack area is an important indicator to measure the range and combat capability of air-to-air confrontation, and it is also the basis for situation assessment and decision making. There are two common ways to express the missile attack zone, the attack envelope and the kill envelope. The attack envelope is a closed-loop area centered on the target aircraft. The attack aircraft can hit the target with a certain probability when launching missiles in this area. As shown in Figure 3 (1), it is mainly composed of a far boundary, a near boundary, and an inescapable envelope. The kill envelope is centered on the attack aircraft, describing the attackable range of the air-to-air missiles carried on the attack aircraft. It takes a certain flight time for the missile to hit the target. The kill envelope can be divided according to the type of maneuver that the target performs during this flight time, as shown in Figure 3(2).
3. Intelligent Tactical Decision AI Model

3.1. Reinforcement Learning

Reinforcement learning can be described by the Markov decision process, usually expressed in the form of four tuples: $<S,A,P,R>$. $S$ describes the state space constituted by all the states of the environment; $A$ describes the action space constituted by the actions that the agent can take in the process of interacting with the environment; $P$ is the state transition probability, the agent takes a certain action in the current state, and will use the transition probability $P$ to transfer current state to another state, and the environment will feed back a reward to the agent. The whole process is shown in figure 4.

![Figure 4](image)

Figure 4. Framework for the interaction between the agent and the environment.

At the current moment $t$, the agent can acquire state $s_t$ $(s_t \in S)$ from the environment and carry out an action $a_t$ $(a_t \in A)$; The environment model moves to the next state $s_{t+1}$ according to the current state and action, returns a reward $r_{t+1}$ in the same time. So that, the following trajectories can be produced during multiple interactions between the agent and the environment:

$$s_0,a_0,r_1,s_1,a_1,r_2,s_2,a_2,r_3,...$$

The cumulative income of the agent from the current moment to the end is:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

(8)

The goal of reinforcement learning is to maximize the expectation of cumulative income, that is, to find an optimal strategy:

$$\pi^* = \arg \max_{\pi} G_0 = \arg \max_{\pi} E_{\pi} [\sum_{k=0}^{\infty} \gamma^k r_t]$$

(9)

Where, $\pi^*$ is optimal strategy, $\gamma$ is discount rate, $r_t$ is reward at $t$.

Value function in state $s$ with politic $\pi$

$$v_{\pi}(s) = E_{\pi}[G_t | s_t = s] = E_{\pi} [\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s]$$

(10)

The optimal value function with optimal strategy is $v^*(s) = \max_{\pi} v_{\pi}(s)$

Action-value function in state $s$ and action $a$ with politic $\pi$

$$q_{\pi}(s,a) = E_{\pi}[G_t | s_t = s, a_t = a] = E_{\pi} [\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$$

(11)

The optimal action-value function is $q^*(s) = \max_{\pi} q_{\pi}(s)$
3.2. Maneuver Decision Framework

As with reinforcement learning, the one-to-one air combat decision-making framework mainly implements the interaction process between the agent and the environment. As shown in Figure 5, the agent continuously accumulates air combat experience through interaction with the environment. The core of the agent is the decision model. Two independent agents with the same decision model structure are regarded as the red side and the blue side respectively, and conduct one-to-one confrontation training to independently complete the mapping from state to action. The environment is mainly composed of an agent state transition model, an air combat reward and punishment model, and a termination judgment model. According to the current agent's maneuver strategy, it can realize the transition from the current state to the next state, determine whether the air combat is terminated and give the reward value of the current decision.

![Figure 5. Reinforcement learning air combat decision-making framework.](image)

At \( t \) time, the environment gives the basic motion parameters \((s_t)\) of red and blue flight. The agent uses the basic flight motion parameters to calculate the relative motion parameters of the two sides. The central function of an agent is to use the state parameters of the environment to build perceptive parameters of state and complete the mapping from the state to maneuver. The red and blue agents respectively use \( s^R_t \) and \( s^B_t \) to calculate the respective decision variables \( a^R_t \) and \( a^B_t \), which will be passed to the environment. The state transition model in environment calculates next state \( s_{t+1} \) based on the current state of the agent and its decision-making action. The judgment will work to estimate termination conditions. After that, rewards and punishment model output a value \( r_{t+1} \) to assess the current decision according to the current state, action, next state and the result of judgment. \( s_{t+1} \) and \( r_{t+1} \) will be passed to agent for the next interaction.

According to the decision-making process of the red and blue sides, it can be simplified as follows:

![Figure 6. Decision-making process for both sides in air combat.](image)

\( \mu_b \) and \( \mu_r \) are the strategies of the blue and red sides respectively, and \( k \) is the decision point of both parties. When \( \mu_b \) and \( \mu_r \) are both given by the AI decision model, the process can be regarded as an agent self-antagonism process. When one of policies is controlled by the pilot and the other is AI's decision-making, the process is a human-machine confrontation process. AI model can learn tactical experience from the confrontation data with the pilot and optimize the decision-making effect. Human-machine confrontation can obtain more valuable samples for AI, and machine self-confrontation is the optimization and test of the training process.

4. LSTM-DQN Algorithm

4.1. Value Network Structure

Due to the advantage of recurrent neural network in processing time series data, we use it as the basic unit of value function fitting. can express the time correlation between serial data. When we construct the
value network and target network, the five adjacent states are selected as the input, the Long Short-Term Memory (LSTM) network works for pre-processing, and the output result of the LSTM is used as the input of the deep network. The network structure is shown in the figure 7.

![Network structure](image)

**Figure 7.** Agent training and testing under decision-making framework

### 4.2. Experience Replay and Training

The experience replay is a sample library for network training, and each sample records the current state, actions of both sides, reward value, and the next state. Such as \( s, a_g, a_s, r, s' \). In model training, the memory size is set in advance and initialization is necessary. When the memory pool is full, the training begins. After \( N \) steps, the network parameters will be used to update experience replay pool, and the next training based on the new sample is followed. The training process is show in figure 7(b). After training, we use the network parameters to test the capability of agent. The tactical decision of an agent can be calculated through the value network and the result of two side is shown in figure 7(c). After training, the red agent with the trained network can evade enemy missiles in an initial disadvantaged state, and gradually accumulate its own advantages to meet the conditions for launching missiles.

### 5. Conclusions

This paper first builds a basic model for man-machine confrontation simulation, and introduces deep reinforcement learning algorithms to create a tactical decision making framework for virtual AI opponents, which can avoid single tactical strategy in traditional opponent training. We focus on the decision making process of AI opponents, as well as the training and network fitting problems of confrontation data. The experimental results show that the man-machine confrontation and agent self-confrontation platform is effective and our work is valid.

### References

[1] Gacovski Z and Deskovski S C 2001 *Modelling of Combat Actions via Fuzzy Expert System* (TO NMSG Conference in Breda, Netherlands)

[2] Kivelevitch E and Cohen K J 2015 *Genetic Fuzzy Trees and their Application Towards Autonomous Training and Control of a Squadron of Unmanned Combat Aerial Vehicles* (Unmanned Systems)

[3] Ma Y F and Ma X L J 2014 *A Case Study on Air Combat Decision Using Approximated Dynamic Programming* (Mathematical Problems in Engineering)

[4] Mnih V and Kavukcuoglu K J 2015 *Human-level control through deep reinforcement learning* vol 518(7540) (Nature) pp 529–533.

[5] Yang Q and M Zhang J 2020 *Maneuver Decision of UAV in Short-Range Air Combat Based on Deep Reinforcement Learning* (IEEE Access) 8 pp 363–378.

[6] Liu P and Ma Y C 2017 *A Deep Reinforcement Learning Based Intelligent Decision Method for UCAV Air Combat* (Asian Simulation Conference) pp 274–286