Air Combat Strategies of CGF Based on Q-Learning and Behavior Tree

Jun FANG¹,a,* Wen-Jun YAN¹,b and Wei FANG¹,c

¹Institute of Information Fusion, Naval Aeronautical and Astronautically University, Yantai, China
ᵐmaidifj@163.com, ‼wj_yan@foxmail.com, ‡bluehawkfw@163.com

Keywords: air combat strategies, artificial intelligence, behavior tree, Q-learning.

Abstract. The intelligence of CGF is one of the important problems in the air combat simulation. A new method for air combat strategies of CGF was proposed based on Q-Learning and Behavior Tree. The intelligence of CGF was formed through establishing behavior tree. And through Q-learning on behavior tree, the evolutionary ability was gained for CGF. Simulation shows that the method performs better and with a stronger learning ability when combat with traditional algorithm.

Introduction

In the development work of flight simulator, method for air combat strategies of CGF has been one of the most important problems of concern, which affects the facticity of flight simulator, as well as the training effect of pilot. Research for air combat strategies of CGF has always been the focus of many countries [1]. Traditional method for air combat strategies, such as neural network [2], genetic algorithm [3], achieved good results. A good method for air combat strategies of CGF is supposed to have two characteristics: the intelligence, and the learning ability. Traditional methods for air combat strategies of CGF has strong ability of independent confrontation, but it cannot update the model data automatically according to the outcome of the training.

R. Dey et al. combined Q-learning and behavior tree to study the effect of reinforcement learning on the behavior tree [4], which is the base of the research. However, there are many problems with the method when applied to air combat strategies of CGF.

1) The establishment of air combat model based on behavior tree;
2) How to express the deep learning using Q value;
3) The Q learning algorithm is easy to converge to the local optimal solution.

We propose the algorithm for air combat strategies of CGF based on the combination of Q-learning and behavior tree. The proposed model makes aircrafts of CGF have some intelligence behavior through building the behavior tree, and it can improve the intelligence and operational capability of CGF model by reinforcement learning based on Q-learning in initial stage and training stage. Meanwhile, in this paper, the Q-learning algorithm is able to converge to the global optimal solution utilizing ε-greedy strategy.

Behavior Tree Settings

Behavior Tree

The behavior tree is the improvement of the hierarchical finite state machine, and it is like a tree in the form [5]. The behavior tree has 4 types of nodes; Sequence node, Selection node, Condition node, Action node.

Sequence node: Executing the child nodes in order;
Selection node: Selecting one of the child nodes to execute according to the rule;
Condition node: Determining whether the condition is true;
Action node: Executing child nodes;
The general structure of the behavior tree is shown in Figure 1, where ‘?’ is selection node, ‘S’ is sequence node, ‘A’ is action node, and ‘C’ is condition node.

**CGF State Space**

Taking the need of air combat into consideration, the state space of CGF is set with four environment and state indexes: IsAttack, IsConfront, Ammo, Health, among which Ammo and Health are continuous. The four indexes are discretized into the following state space description:

- Whether to attack the enemy: IsAttack(Yes, No);
- Whether to face the enemy: IsConfront(Yes, No);
- Number of ammunition: Ammo (None, Low, Medium, High);
- The state of health: Health (None, Low, Medium, High);

For number of ammunition, none indicates no ammunition, Low stands for the lack of ammunition, and High indicates the ammunition is plentiful. With regard to the state of health, none shows the plane is destroyed, Low indicates serious damage, and Medium represents slight damage, High expresses the aircraft is in good condition.

**CGF Behavior Space**

The aircrafts in CGF have the following four independent acts:
- Patrol: The aircrafts patrolled the scheduled route;
- Attack: The aircrafts attack;
- Turn Around: If the aircraft meets the enemy that is not in sight, he would turn around to attack the enemy;
- Flee: The aircraft would flee when the state of ammo and health is not ideal.

**CGF Behavior Tree Settings**

Considering the state space and the behavior space, the CGF behavior can be set to the behavior tree shown in Fig. Among them, the execution conditions of CGF behavior is shown in Table 1.

<table>
<thead>
<tr>
<th>action</th>
<th>condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patrol</td>
<td>IsAttack(No)</td>
</tr>
<tr>
<td></td>
<td>Health(High)</td>
</tr>
<tr>
<td>Attack</td>
<td>IsAttack(Yes)</td>
</tr>
<tr>
<td></td>
<td>IsConfront(Yes)</td>
</tr>
<tr>
<td></td>
<td>Ammo(High)</td>
</tr>
<tr>
<td></td>
<td>Health(High)</td>
</tr>
<tr>
<td>Turn Around</td>
<td>IsAttack(Yes)</td>
</tr>
<tr>
<td></td>
<td>IsConfront(No)</td>
</tr>
<tr>
<td></td>
<td>Ammo(High)</td>
</tr>
<tr>
<td></td>
<td>Health(High)</td>
</tr>
<tr>
<td>Flee</td>
<td>IsAttack(Yes)</td>
</tr>
<tr>
<td></td>
<td>Ammo(None)</td>
</tr>
<tr>
<td></td>
<td>Health(High)</td>
</tr>
</tbody>
</table>

Table 1. CGF Behavioral Conditions.
Learning Process Based On Q-Learning

Q-Learning

![Diagram of Q-Learning process]

Figure 3. An Agent that interacts with the environment Agent.

Agent could obtain the best strategy through reinforcement learning [6-8]. The general flow of reinforcement learning is shown in Figure 3. An Agent is in the environment, which do effect on the environment through performing an action. Environment changes after acquiring the action of Agent, meanwhile it gives the reinforcement learning system a feedback value (reward or punishment). Then Agent chooses the next action based on the state of the environment and the reinforcement learning system. In this process, Agent tries to maximize the value of reward.

Q-learning is one of the reinforcement learning. The execution of each action in reinforcement learning corresponds to the corresponding Q value. In the process of Q-learning, the value of Q updates to convergence based on the feedback of the environment. The Q-learning is defined as follows:

\[ Q(s, a) = R(s, a) + \gamma \max(A) \]

where \( Q(s, a) \) is evaluation function, the value of which is the maximum conversion cumulative return from the beginning of the state \( s \) and the use of \( a \) as the implementation of the action, \( s \) is the current state, \( s' \) is the next state, \( a \) is the current action, \( A \) is the sets of actions that can be performed for the next state; \( R(s, a) \) is immediate return sequence, \( \gamma \) is the return factor, where \( 0 \leq \gamma < 1 \) is constant. Therefore, the value of \( Q \) is the immediate return of the action \( a \) for state \( s \) and the value of the optimal policy (using the return factor \( \gamma \) in calculation).

As can be seen, Agent only need to consider the current state of the \( s \) under the action of \( a \), and select action that maximizing the \( Q \) value.

Dynamic Greedy Strategy

The disadvantage of Q-learning is that it is easy to converge to the local optimal solution, thus we utilize \( \epsilon \)-greedy strategy to solve the problem [9, 10]. \( \epsilon \) is the ratio between search and application, the value of \( \epsilon \) adjusted adaptively with the search. We computed \( \epsilon \) as:

\[ \epsilon(k) = 1 - \frac{k}{M} \times 0.99 \]  

where \( k \) was the learning times, \( M \) was the total learning times. When \( k = 1, \epsilon \approx 1 \) indicates the model only searches but does not apply; when \( k = M, \epsilon \approx 0 \), indicates the model only apply but does not search; 0.99 is to avoid \( \epsilon \) getting the value of 0 or 1 which is not meaningful.

Learning Principles

After using the Q-learning method to obtain the converged Q-table, we map the Q-table into the behavior tree, and introduce Q value to each node in the behavior tree, as shown in Figure 4.
For action 1, there are three states corresponding to A, B, and C, respectively. At the sequence node S, the three states are arranged according to the value of $Q$, so the judgment of the behavior condition will be made according to the value of $Q$. At the same time, the $Q$ value of the sequence node S is given the maximum state, that is, the $Q$ value of the sequence node S in Figure 4 should correspond to the $Q$ value of state C9.

After the child nodes are arranged, the parent nodes need to be arranged. The value of the parent node $Q$ is the maximum value of the child node $Q$. The nodes of the behavior tree are arranged in descending order of $Q$, and the process of $Q$ value arrangement is shown in Fig.5.

**The Proposed Algorithm**

1) Establish the initialized model for air combat of CGF based on $Q$-learning and behavior tree.
2) Renew the reward function and the $Q$ value table is converged;
3) The $Q$ value of the behavior tree node is updated according to the $Q$ value table. If the behavior tree node is updated, the learning is continued and the learning is terminated if there is no update. The CGF air combat decision model based on $Q$-learning and behavior tree is completed.
Experiment and Results

To evaluate the model an experiment was performed using an aircraft simulation system. Our model is trained by the pilots’ training data in aircraft simulation system, then the model would confront with model based on hierarchical finite state machine. In this paper, the simulation aircraft using the proposed model is defined as the red, the aircraft utilizing air combat strategy model based on hierarchical finite state machine is called the blue.

In order to reduce the testing time, in the confrontation, the two sides were set to make three against three simulations placed within the scope over the 50 square kilometers. Where $\alpha = 0.8$, $\gamma = 0.5$. The results of 100 confrontations are shown in Fig.6. As shown, most of the time the red remains one or two aircrafts. Therefore, the proposed method of air combat strategies of CGF based on the combination of $Q$-learning and behavior tree performs better than the air combat model of CGF based on hierarchical finite state machine. In addition, with the number of rounds increasing, the survival plane of the red increases gradually. It indicates that through the combination of $Q$-learning and behavior tree, CGF air combat model obtains good learning ability and intelligence.

Summary

This paper introduces a method for air combat of CGF. Our method combines $Q$-learning and behavior tree for the intelligence and the learning ability, which are significant in air combat strategy simulation. Experiments indicate the superiority of our algorithm with initial intelligent behavior gained by initial training and actual combat training. We will apply the proposed method to the development of simulator.

References


