Research on Missile Intelligent Penetration Based on Deep Reinforcement Learning

Sen Lei, YongLin Lei, Zhi Zhu
College of Systems Engineering, National University of Defense Technology, Changsha 410003, China
944342344@qq.com

Abstract. Penetration capability is one of the most critical measures for assessing the effectiveness of missile systems. In the face of an increasingly sophisticated antimissile system, this article explores the use of artificial intelligence to achieve missile intelligent penetration. The applicability of the deep reinforcement learning method to the missile intelligent penetration problem is elaborated. The Actor-Critic deep reinforcement learning algorithm is studied. The deep reinforcement learning environment model for the missile penetration scenario is developed based on the generic combat effectiveness simulation system, WESS. A multi-sample collaborative training method is designed. On this basis, a missile intelligent penetration training system framework based on WESS and deep reinforcement learning is proposed to guide the missile penetration intelligence learning.

1. Introduction
Since the German V-2 ‘Revenge Weapon’ entered the battlefield in 1944, the long-range and non-contact precise strikes have become the main means of modern air strikes. Missile penetration technology is a technology that uses all detection and interception methods to guide missile to pass through the interception zone of the enemy's antimissile defense system without damage. Missile penetration technology is an important indicator for measuring the tactical and technical performance of missile systems and the level of weapon development [1]. With the maturity of the antimissile system, the multi-level and multi-mode missile defense systems have greatly squeezed the missile's living space, so the difficulty of missile penetration has gradually increased. Therefore, the missile penetration capability has become the most critical index of a missile weapon system.

At present, most of the missile penetration researches are based on the classic control method, by establishing an interpretable differential game model for offensive and defensive sides, they are designing guidance laws to improve the operational effectiveness of missile weapons [2-3]. From a traditional perspective, classic control methods are reliable and effective for missiles and other equipment that require precise control. However, as the weapons and equipment involved in combat become more and more complex, more and more information and control parameters need to be collected by the command and control system, resulting in an increasingly heavy system load. After the command and control system completes the binding of the missile launch instructions, it needs to perform real-time online route planning and decision-making on penetration actions based on battlefield information provided by other platforms (such as satellites, drones, radars and seekers) to avoid enemy reconnaissance and firepower attack. Faced with so many entities and control parameters, the traditional classical control methods seem to be powerless. Therefore, countries are also seeking more intelligent missiles and weapon systems to replace existing design solutions [4].
Considering from the actual situation, the generation of the missile's intelligent penetration capability depends on a large amount of test work, but the organization of the test requires many troops and funds, and because the target of the missile penetration cannot be controlled, the actual experiment cannot be completed directly. This article relies on the generic combat effectiveness simulation system (WESS) [5], builds and develops an extensible simulation model framework and related penetration combat simulation models based on combat simulation technology, and develops a missile penetration learning training system. By formulating scenarios for military operations offline and using high-performance computers to run simulation samples in parallel, the data sets required by intelligent algorithms are obtained. Secondly, the deep reinforcement learning algorithm is studied based on optimized and reasonable algorithms and theories, and a large amount of training and simulation calculations are performed using the data set to obtain an intelligent strategy suitable for missile intelligent penetration decision-making.

2. Missile Intelligent Penetration and Deep Reinforcement Learning

2.1. Missile Intelligent Penetration
The so-called missile intelligent penetration, specific to the missile's combat mission and work process, can be simply summarized as the missile can achieve complete autonomy from detection, tracking, seeking, interception to final destruction of the target [6]. In the course of intelligent missile penetration, missiles can autonomously process various perceived information, analyse, judge, and reason about the external environment, target characteristics, and changes, so that they can make correct decisions and responses. According to the actual situation of the current development of missile weapons, missiles that can fully realize intelligent penetration still do not exist, but the definition of intelligent missiles is also a relative and constantly developing concept. With the development of science and technology, the level of intelligence continues to increase. By upgrading, the intelligence level of missiles will also be improved and improved.

2.2. Deep Reinforcement Learning
Many control problems can be reduced to reinforcement learning problems, and reinforcement learning has proven to be successful. At present, based on deep learning and reinforcement learning algorithms, many challenging problems have been successfully solved, from AlphaGo to StarCraft’s game AI, Deep reinforcement learning continuously overcomes difficulties, and has achieved good results in a continuous high-dimensional, complex, and incomplete information environment state space. These experiments show that deep reinforcement learning can propose new method ideas for intelligent combat simulation problems in complex combat environments.

Missile penetration tasks often have clear objectives, such as destroying a specific target of the enemy or getting rid of the enemy’s interception, but each specific action in the actual work process does not have the correct answer as a reference. This mission scenario is exactly reinforcement learning aiming at. In front of the dynamics and unknowns of the missile's working process, reinforcement learning conducts interactive online learning with the external environment through continuous "trial and error" methods, then selects the most optimal decision sequence based on the cumulative return value during the learning process [7].

The decision-making process of the missile's working process is confronted with the scenario where the offensive missile and the anti-missile system are confronting each other. Many uncertain factors such as the performance of the weapon system of the attack and defense and the battlefield environment must be comprehensively considered. In front of the controlled object, the standard reinforcement learning search process is random, and the accumulation of returns is slow. It is difficult to obtain a relatively good decision sequence in a short period of time. By combining deep learning’s perception ability and reinforcement learning’s decision-making ability, the use of Deep Reinforcement Learning technology-based control strategies can better help train and develop intelligent strategies for missile decision-making.
3. Missile Penetration Decision Model Based on Actor-Critic

3.1. Missile Intelligent Penetration Problem Description

When the missile engine ignites into the boost phase, the high-temperature flames at the tail may be detected by the defense early warning satellite; the early warning satellite can automatically transmit the detected information to the Strategic Defense Allegation Center. After further positioning, the allegation center quickly transmits the target information to the anti-missile system in the direction of the offensive missiles; the early-warning radar of the anti-missile system will start searching, identifying and tracking. The information is transmitted to the leading-edge guidance radar; the guidance radar intercepts and tracks the offensive missile and guides the interceptor to strike the offensive missile [8]. During the entire penetration process, offensive missiles will take some technical measures to evade the defense's reconnaissance and interception.

Traditional missile penetration research is mostly a model-based method. By establishing a dynamic model, the movement of the missile during the entire process is controlled. The parameters of this method have clear meanings and are supported by corresponding theoretical research, which proves to be effective and reliable. However, from the perspective of the entire missile penetration process, it is relatively difficult to comprehensively consider the reasonableness of the ballistic trajectory and the decision timing of the penetration method. Based on this, it is relatively difficult to design a dynamic model to describe the entire process. The entities and controls included in the entire environmental model Too many parameters to complete the task.

In recent years, the development of deep learning technology has provided us with new ideas. Deep learning uses neural networks for deep characterization and completes the construction and learning of problem knowledge by inputting original information; however, this model-free method lacks a description of problem mechanisms. It is not very explanatory and not stable enough, it is easy to diverge during training, and it is not advisable to learn from the perspective of the entire penetration process. So, we concentrate on the advantages of model-free and model-based methods to complete the construction of the problem model. Based on the dynamic model of the missile penetration process, a deep reinforcement learning algorithm is integrated to make the correct penetration decision for the environmental situation, to realize the intelligent penetration of the missile.

3.2. Reinforcement learning model based on Actor-Critic algorithm

The traditional simulation of the missile attack and defense mainly focuses on the final missed target result, and the interaction between models is only state information. Reinforcement learning-oriented missile intelligent decision-making requires feedback on the operational effects of weapon system decisions. During the simulation process, each penetration decision and mission planning of the weapon system will affect the entire battlefield environment. Within the time step \( T \), the combat entity receives the environmental state \( s_t \), executes a behavioral decision \( a_t \), and then gets decision feedback \( r_t \) from the environment, while the environment enters the next state \( s_{t+1} \). Increase the decision feedback \( r_t \).

The related intelligent decision-making algorithms can be divided into two categories: value-based and policy-based. Among them, value-based algorithms, such as the classic algorithm Q-learning, can obtain the optimal solution, but are limited to the case where the output action space is discrete; while strategy-based algorithms often have better convergence properties, The continuous action space is more effective, but the local optimal solution is usually obtained, the evaluation strategy is not efficient, and it has a high deviation. As a result of the combination of value-based and strategy-based, Actor-Critic (AC) algorithm [9] can promote strengths and avoid weaknesses, combining the advantages of the two algorithms.
As shown in Figure 1, the combat entity agent includes two parts: Actor action module and Critic evaluation module. The input of the Actor action module is the environmental state, and the output is the specific action. The parameterized gradient method is used to improve the strategy. The Critic evaluation module uses the value function method to evaluate the strategy. The Critic evaluation module is updated based on historical actions and decision feedback, giving the Actor better gradient estimates, which improves the evaluation of the strategy, solves the local optimality problem of the policy gradient method, and the Actor action module avoids inefficiencies in the value function method. The value estimation process can also deal with the situation of continuous action space.

The missile penetration problem can be regarded as a Markov decision process. At the beginning, the combat entity agent uses a random strategy to perform experiments to obtain a series of sample data of state actions and rewards collection:

\[ s_1, a_1, r_1, s_2, a_2, r_2, \ldots, a_{t-1}, r_{t-1}, s_t \] (1)

Where \( r_t \) represents the reward received by the agent at time \( t \), the total reward value \( G_t \) from the time \( t \) to the end of the action:

\[ G_t = \sum_{k=0}^{n} \gamma^k r_{t+k} \] (2)

\( \gamma \) indicates a discount factor, which indicates the importance of the reward, and generally decreases with time.

In order to evaluate the strategy of the agent, the state value function \( V_\pi(s) \) is defined as the value when the Agent starts from the state \( s \) and then executes the strategy \( \pi \), the value of the state under the strategy. therefore:

\[ V_\pi(s, a) = E_\pi \left\{ \sum_{t=0}^{T} \gamma^t r_{t+1} \mid s_t = s, a_t = a, \pi \right\} \] (3)

In the formula: \( T \) is the final time; \( t \) is the current time; \( r_{t+1} \) is the return obtained at time \( t+1 \).

The state action value function \( Q_\pi(s, a) \) is defined as the expected return obtained when the agent selects a specific behavior \( a \) from the state \( s \) and then executes the strategy \( \pi \). The state action value function \( Q^*(s, a) \) under the optimal strategy can be rewritten as a Bellman optimal equation:

\[ Q^*(s, a) = \sum_{s' \in S} T(s, a, s') (R(s, a, s') + \gamma V^*(s')) \] (4)

In traditional reinforcement learning, the Q-value function is generally solved by iterating the Bellman equation:
\[ Q_{t+1}(s, a) = E_{s' \in S}[r + \gamma \max Q_t(s', a')] \]  

Among them, when \( i \to \infty \), \( Q_t \to Q^* \), that is through continuous iteration, the state action value function \( Q_\pi(s, a) \) will eventually converge, and the optimal policy \( \pi^* = \arg \max Q(s, a) \) is obtained. However, for actual missile penetration scenarios, the state of the environment and the actions of the agents are continuous quantities. Iteratively solving the optimal strategy is obviously not feasible and the calculation cost is too large. Therefore, using a deep neural network constructor approximator, a Q-valued neural network is defined as: \( Q(s, a) \approx Q(s, a, w) \), where \( w \) refers to the parameters of the neural network. Deep neural networks are intelligent, robust, and capable of approximating any non-linear function, so they are very suitable for approximate representation functions or strategies.

The operating flow of Fig.1 is: the state \( s_t \) is input to the Actor and Critic networks, respectively, and outputs the action \( a_t \) and the error function \( E_t \) under the current strategy; the error \( E_t \) is used to update the parameters in the Actor and Critic networks; action \( a_t \) acts on the environment to produce the next state variable \( s_{t+1} \).

3.3. Multi-sample parallel processing architecture

In order to achieve the goal of reinforcement learning, it is necessary to optimize the action strategy. Traditional strategy gradient methods are prone to instability when combined with neural networks. Most deep strategy gradient methods use empirical playback mechanisms to eliminate the correlation of training data, but this method leads to the interaction between the Agent and the external environment consumes more computing and memory resources and can only be updated based on the data generated by the old strategy.

Asynchronous Advantage Actor-Critic (A3C) \([10]\) is a combination of multiple threads on the same machine. Each thread corresponds to a different exploration strategy, by creating an environment parallel to the main thread, it has a sub-model with the same structure as the main thread model. Each sub-model runs and interacts in a parallel environment at the same time, updates the parameters learned by itself to the global model, and copies the network parameters of the global model next time. The A3C method abstract architecture is shown in Figure 2.

![Fig.2 A3C method abstract architecture](image)

Parallel Agents are independent of each other. The parameter update of the public model is affected by the discontinuity of the sub-model submission update, which reduces the correlation of the update and improves the convergence, which is beneficial to the training of the model neural network.
Compared with the traditional method, A3C consumes less time and resources and achieves better results, making a single machine capable of processing more complex reinforcement learning problems. The algorithm flow is as follows:

Input: the neural network structure of the public part; the neural network structure of current thread; hyperparameters
Output: neural network parameters $\theta$, $\omega$ of the public part.

Step 1: Update time series $t = 1$.
Step 2: Reset network gradient update: $d\theta \leftarrow 0$, $d\omega \leftarrow 0$.
Step 3: From the neural network synchronization parameters of the public part to the neural network of this thread: $\theta' = \theta$, $\omega' = \omega$.
Step 4: $t_{\text{start}} = t$, and initialize state $s_t$.
Step 5: Select action $a_t$ based on strategy $\pi(a_t \mid s_t; \theta)$.
Step 6: Perform action $a_t$ to get reward $r_t$ and new state $s_{t+1}$.
Step 7: $t \leftarrow t + 1$, $T \leftarrow T + 1$.
Step 8: If $s_t$ is terminated, or $t - t_{\text{start}} = t_{\text{local}}$, go to step 9, otherwise go back to step 5.
Step 9: Calculate $Q_{(s_t)}$ for the last position $s_t$.
Step 10: for $i \in (t - 1, t - 2, ..., t_{\text{start}})$:
   (a) Calculate $Q(s, i)$ at each moment: $Q(s, i) = r_i + \gamma Q(s, i + 1)$
   (b) Cumulative local gradient update of Actor network:
       
       $d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi_{\theta'}(s_i, a_i) \left(Q(s, i) - V(s_i, \omega')\right) + c \nabla_{\theta'} H (\pi(s_i, \theta'))$
   (c) Cumulative local gradient update of Critic network:

       $d\omega \leftarrow d\omega + \frac{\partial (Q(s, i) - V(s_i, \omega'))^2}{\partial \omega'}$

Step 11: Update the model parameters of the global neural network: $\theta = \theta - \alpha d\theta$, $\omega = \omega - \beta d\omega$.
Step 12: If $T > T_{\text{max}}$, the process ends, and the neural network parameters $\theta$, $\omega$ of the common part are output, otherwise, the process proceeds to step 3.

4. Missile Intelligent Penetration Learning and Training Framework Based on WESS and DRL algorithm

4.1. WESS-based missile penetration simulation engine

According to the definition and description of intelligent missiles, the key to using the reinforcement learning technology to solve the missile's intelligent penetration problem is to realize the interactive learning of the combat agent and the environment, and obtain knowledge in continuous iterative learning, thereby improving the action plan to adapt to the environment and achieve the expectations. The goal is to build a training environment. At present, the basic platform Gym developed by foreign OpenAI companies provides a lot of built-in training environments for developing and comparing various reinforcement learning algorithms. In the field of combat simulation, there are few related platforms.

WESS has formed a relatively complete offensive and defensive simulation engine framework, which can provide relevant missile kinematics and dynamics models, and can use the simulation deduction platform to get the battlefield environment information, such as weapon systems parameter, situation assessment results, and evolutionary process. Those elements are used as state inputs for reinforcement learning. Based on the existing missile penetration technology, design the agent's action output, and design intelligent decision-making algorithms to be integrated into the offensive and defensive combat environment. Realize interaction and learning training with combat environment.
4.2. Overall framework of missile intelligent penetration training environment

The main goal of missile intelligent penetration based on WESS and DRL is to take the missile defense system as the penetration target, and by constructing an intelligent reinforcement learning training environment, it can provide an operational scenario data generation environment for the offensive missile penetration decision intelligent algorithm learning and training. The overall framework of the deep reinforcement learning training environment is shown in Fig.3.

**Overall System Architecture**

![Diagram of Overall System Architecture](image)

**Fig.3 The overall framework of missile intelligent penetration training**

The running process of the framework is as follows:

Step1: Exploit and integrate the corresponding offensive missile model and defense system model based on the WESS platform;

Step2: Manage the set of combat scenarios by scenario management tools and distribute multiple scenario samples to the simulation engine. The simulation engine runs in multiple threads in parallel, each sample will create a reinforcement learning database to cache the data of each round.

Step3: The penetration decision model integrates the decision neural network. The decision neural network receives the battlefield situation information transmitted by the WESS simulation engine through the interface, and the intelligent decision output receives the decision instruction and returns it to the simulation engine to complete the simulation.

Step4: The deep reinforcement learning training module is decoupled from the overall framework. The sample round database is input to the deep reinforcement learning training module, and the decision neural network is updated according to the DRL algorithm. After the training update is completed, the latest network information is broadcast to the decision model of each thread. To complete the update of the decision network for the decision of the next simulation step.

Step5: When a certain number of trainings are reached or the missile penetration success rate is relatively increased to a certain threshold, the decision network model stabilizes and exits the scenario; otherwise returns to Step2.

4.3. Scenario Management Module

Modern military combat scenarios are complex and changeable. The intelligent decision networks trained by different scenarios may not be compatible with each other. Even with the same scenario, the results obtained by intelligent decision networks may be different. Secondly, due to the continuity of the training process, reinforcement learning algorithms It is unstable with the deep neural network, and it is difficult to converge to the optimal solution. Therefore, it is necessary to traverse the proposed set
of defensive operations and run repeatedly. Based on the specific combat scenario, use the WESS scenario editor to build the set of defensive operations. The experimental operation management module automatically completes the scenario’s running and training process, to achieve the requirements of decision network learning training convergence.

4.4. Intelligent Penetration Decision-making Module

Connected with the WESS, the intelligent penetration decision-making module should include the combat real-time situation interface of the simulation engine, the decision instruction interface, the reinforcement learning sample database, the integration and up-date of penetration decision network, and the analysis of decision instructions.

The combat real-time situational interface design is based on the state quantities that are focused on during the offensive missile intelligent penetration process and is used as the input to the decision-making network. It is also a state space design for reinforcement learning; missile’s penetration action is the action space design of reinforcement learning; the decision instruction analysis is to convert the output of the decision neural network into the executable instruction of the simulation model; in the penetration decision behavior model, by setting the combat rule base and tactics knowledge base, adding empirical knowledge to make agents' behavioral decisions more reasonable. The reinforcement learning sample database should cache the state, action, and reward data at each moment in the entire round as a database set for the training module.

Due to the complexity of the missile penetration process and the difficulty of convergence in reinforcement learning training, the action output of the penetration decision network is layered into discrete control instructions and continuous control instructions. Discrete control instructions are responsible for the switch of the penetration defense technology. Continuous quantity control instructions call the instruction calculation model. The traditional classical control method is used to solve the control quantity and provide it to the relevant penetration simulation model.

4.5. Reinforcement learning training Module

The design of the reward function is very important in reinforcement learning. Without a well-designed feedback signal, it will have a poor effect on the reinforcement learning algorithm. Considering the purpose of intelligent penetration, we hope that the offensive missile can complete the combat mission as much as possible, and the value of the reward function at each moment is related to the amount of penetration status at the current moment, so the reward function design should be based on domain knowledge, and combat The method of effectiveness index system builds the effectiveness index system of missile penetration. Therefore, a design scheme of the reward function is as follows:

![Fig.4 Reward function calculate process](image)
First, the intermediate process effectiveness measure system is designed based on the action output of each penetration decision network that is beneficial to its own side and adverse to the enemy. It is also combined with state quantities such as detection, tracking, and interception that the model sensor can collect to design the measure system algorithm, to get the reward function. Reinforcement learning reward calculation process diagram as fig.4 shows.

The reinforcement learning training module uses the saved sample database for training and updating of decision neural networks. Decision network training is based on A3C algorithm and adopts multi-sample collaborative training method. Each sample has a separate training sample database, which is called by the training module to support the gradient update of the parameters of the main network. Multi-sample synchronization is achieved by locking. After all the samples have been trained, the next scene is trained. In this way, in any time step, the parallel Agent will go through different environmental states, thereby cutting off the correlation of the data, making it easier for the deep neural network to converge to the optimal solution. Both are superior to traditional methods.

5. Conclusion
In recent years, with the breakthrough and development of artificial intelligence technology with deep learning and reinforcement learning as the core, new technological approaches have been opened for the future development of missile weapon systems, and missiles based on artificial intelligence technology are the future development of missile weapons. Important direction. This article combines the equipment combat effectiveness simulation system to build a reinforcement learning training environment, integrates deep reinforcement learning algorithms into the offensive and defensive combat environment, and conducts multi-scene collaborative training to form the overall framework of missile intelligent penetration learning training. A mature combat agent can judge the battlefield situation through various types of data received by sensors, quickly form effective decision-making instructions, and combine commander experience with future battle rules to help guide the study of new tactical tactics.

Acknowledgments
This work was supported by the project of "Intelligent XX Decision Battlefield Training Environment Generation System".

Reference

[1] Cheng Chuanhao, Wang Ruichen, Lutherin. Penetration and Interception of Ballistic Missiles[J]. Modern Defense Technology, 2002, 30(4):20-23(in Chinese).
[2] Zhao Xiuna, Yuan Quan, Ma Hongxu, et al. Research on Attitude Search Algorithm of Maneuvering Warhead Penetration[J]. Aerospace Control. 2007,25(4):13-16 (in Chinese).
[3] Li Longyue, Liu Fuxian, Shi Xiangfeng, et al. A Model for Chasing and Escape in Missile Attack and Defense and Its Solution Method[J]. System Engineering and Electronics. 2016, 38(5):1067-1073(in Chinese).
[4] Douglas D Diehl, LT USN. How to Optimize joint Theater Ballistic Missile[EB/OL]. http://www.go.com/, 2004,3
[5] Lei Yonglin, Yao Jian, Zhu Ning, et al. Weapon Effectiveness Simulation System (WESS) [J]. Journal of System Simulation. 2017,29(6):1244-1252(in Chinese).
[6] Wen Suli, Chen Qi, Su Xinxin, et al. Research on Intelligent Missile and Missile Intelligence [J].Tactical Missile Technology. 2016,42(5): 685-695
[7] CHEN X G, YU Y. Reinforcement learning and its application to game of go[J]. Acta Automatica Sinica, 2016,42(5):21-26 (in Chinese).
[8] Luo He et al. An Evaluation Model of UAV Reconnaissance Capability[J]. Firepower and Command Control. 2016,4(2):7-12(in Chinese).
[9] Lillicrap T P,Hunt J J,Pritzel A,et a1. Continuous control with deep reinforcement learning[J]. Computer Science. 2016,8(6):A187.
[10] Volodymyr Mnih et al. Asynchronous methods for deep reinforcement learning[J]. Computer Science. arXiv:1602.01783 (2016).