

RESEARCH ON AERODYNAMIC-STEALTH INTEGRATED DESIGN METHODOLOGY BASED ON MPPO ALGORITHM

GAN Keyao[1], ZHAO Huan[2, *], Wang Shekun[1], Gan Yujie[1]

[1]School of Astronautics, Northwestern Polytechnical University, Xian 710072, China

[2]National Key Laboratory of Aircraft Configuration Design, Xian 710072, China

Abstract

Traditional design optimization methods often suffer from low efficiency and a tendency to converge to local optima when applied to the multidisciplinary, high-dimensional, and nonlinear problem of aircraft aerodynamic-stealth design. To address these issues, a Modified Proximal Policy Optimization (MPPO) algorithm is proposed. By optimizing the extraction of the surrogate objective function from the traditional PPO algorithm, the MPPO algorithm effectively enhances the quality of samples selected by the model. This mechanism reduces unnecessary perturbations to the environment while simultaneously improving the algorithm's global search capability. Integrating MPPO with surrogate optimization techniques, this study develops an MPPO-based integrated aerodynamic-stealth optimization design framework.To validate the effectiveness of the proposed MPPO algorithm, aerodynamic-stealth optimization is conducted on both the NACA 65,3-014 airfoil and an X-47B-like configuration. The results are compared against those obtained using the traditional Particle Swarm Optimization (PSO) algorithm. The optimization case studies demonstrate that, compared to the PSO algorithm, the MPPO algorithm significantly accelerates the convergence speed of the aerodynamic-stealth shape optimization. Furthermore, the final optimized designs obtained by MPPO outperform those from PSO in both aerodynamic and stealth performance metrics. These results indicate that the proposed MPPO-based integrated aerodynamic-stealth design methodology possesses higher optimization efficiency and stronger global optimization capability.

Keywords: flying-wing configuration, aerodynamic stealth optimization, deep reinforcement learning, MPPO algorithm, surrogate model

1. Introduction

With the rapid development of modern aerospace technology, increasingly stringent requirements are imposed on the overall performance of aircraft. To meet diverse mission requirements such as long endurance, high maneuverability, high survivability, and efficient penetration, aircraft shape design has shifted from the sole pursuit of superior aerodynamic performance to a comprehensive trade-off and integrated design of multidisciplinary characteristics, including aerodynamics and stealth. However, there exist strong mutual coupling and complex constraint relationships between the aerodynamic and electromagnetic stealth characteristics of aircraft, and the design space exhibits high-dimensional, nonlinear, and multimodal complexity. Meanwhile, with the rapid advancement of computer technology, high-fidelity CFD computation and radar cross section (RCS) analysis have been widely adopted, and the effectiveness of aircraft aerodynamic and stealth shape design increasingly depends on the advancement and effectiveness of the optimization design method employed.

In 2013, DeepMind [1] combined deep neural networks with reinforcement learning algorithms—both subfields of machine learning—giving rise to deep reinforcement learning (DRL), which offers strong learning and planning capabilities as well as high generalization and versatility [2]. Subsequently, algorithms such as Deep Deterministic Policy Gradient (DDPG), Asynchronous Advantage Actor-Critic (A3C) [3], and Proximal Policy Optimization (PPO) emerged. In 2017, DeepMind released

AlphaGo Zero, a reinforcement learning algorithm for Go that learns through self-play, ultimately making AlphaGo the first artificial intelligence to defeat a human player and fully igniting enthusiasm for reinforcement learning [4]. In recent years, reinforcement learning has been introduced into the field of aircraft shape design. Yonekura proposed a framework for applying reinforcement learning to optimization design, laying the foundation for subsequent research [5]. Li Runze subsequently achieved aerodynamic optimization design of a supercritical airfoil using deep reinforcement learning [6], demonstrating the feasibility of this approach. Recently, P. Scavella et al. explored the use of reinforcement learning in airfoil optimization design. By developing a global optimization strategy for single-point airfoil design via DRL, the method can reliably converge to high-performance airfoils under various flow conditions without requiring any prior knowledge [7], exhibiting stronger global optimization capability compared with traditional algorithms.

To overcome the limitations of poor optimization efficiency and insufficient global search capability in traditional surrogate optimization methods, and to explore novel approaches for the integrated aerodynamic-stealth design of wide-speed-range aircraft, this study introduces an aerodynamic-stealth optimization method grounded in a deep reinforcement learning algorithm — Modified Proximal Policy Optimization (MPPO). The method's performance is validated through comparative studies on three benchmark cases—the Rosenbrock function, a NACA 65,3-014 airfoil, and an X-47B-like configuration—against the traditional Particle Swarm Optimization (PSO) algorithm. Results demonstrate that the proposed MPPO-based integrated design method for wide-speed-range aircraft demonstrates superior optimization efficiency and enhanced global search capability, offering valuable reference for future aerodynamic-stealth optimization of aircraft.

2. Technical Foundation

In the aerodynamic and stealth optimization design of aircraft, the agent learns through interaction with the environment. The environment includes the design variables and the aerodynamic and stealth characteristics, while the state is represented by the design variables. The agent takes the design variables as the state input and learns a policy for action selection based on the state. The policy is defined as a mapping from states to actions, describing the probability distribution over the actions that the agent executes in a given state. The agent learns and selects a feasible action in a given state; after the action is executed, the agent receives feedback from the environment, including positive and negative feedback, where positive feedback serves as a reward for correct actions and negative feedback serves as a penalty for incorrect actions. The action chosen by the agent in the current state influences the environment and causes it to change. By interacting with this complex environment, the agent learns the optimal control policy. The interaction between the entities within this training loop is formulated as a Markov decision process (MDP), the framework of which is shown in Figure 1.

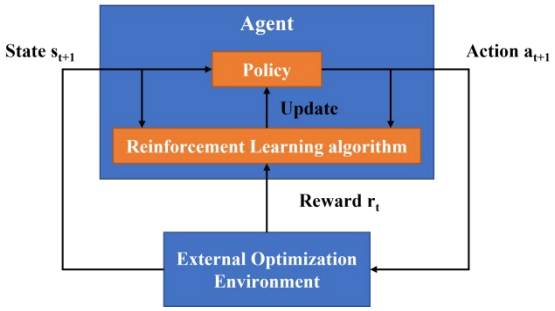


Figure 1 Reinforcement Learning Control Framework.

The objective of reinforcement learning is for the agent to maximize long-term rewards, thereby learning the optimal decision sequence. Unlike value-based methods, which maintain a value function model, policy-based methods do not rely on such a model and instead directly search for the optimal decision sequence by solving the policy function. The policy function π is defined by a set of adjustable parameters θ. By adjusting these parameters and observing the resulting differences in rewards, the parameters are updated in the direction that yields higher rewards. Since the policy is stochastic, its parameters determine the action sampling probabilities and thus influence

the selection of the optimization direction. Accordingly, the objective function of the policy gradient-based reinforcement learning algorithm is defined as:

$$J(\theta) = E_{\tau\sim\pi_\theta} R(\tau) = \sum P(\tau;\theta) R(\tau) \tag{1}$$

Where $R(\tau)$ denotes the cumulative reward over the entire process, and $P(\tau;\theta)$ denotes the probability of this process occurring. The parameters of the policy function are updated in the direction consistent with the gradient of the expected reward, so as to maximize the cumulative expected reward, and are updated via gradient ascent until convergence:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \tag{2}$$

This paper adopts the sample-based modified Proximal Policy Optimization (MPPO) algorithm, combined with conventional aerodynamic and stealth surrogate-based optimization methods. The components of the MPPO algorithm are introduced below.

2.1 State Space

In the optimization design framework, the complex aircraft geometry must first be parameterized. The airfoil is defined using the Class-Shape Transformation (CST) method [8], and the three-dimensional configuration is described with the Free-Form Deformation (FFD) method [9]. Both approaches generate a set of n-dimensional design variables. Together with the corresponding aerodynamic and stealth characteristics of the shape, these variables constitute a multi-dimensional feature vector, which forms the environmental state input. Thus, the state space observed by the agent can be expressed as:

$$S_t = \langle \rho_n, c, RCS \rangle \tag{3}$$

Where ρ_n denotes a set of n-dimensional design variables $\langle \rho_1, \rho_2, \ldots, \rho_n \rangle$, c presents the aerodynamic state of the aircraft under the corresponding design variables (typically selected from parameters such as the lift coefficient C_l,the drag coefficient C_d or lift-to-drag ratio L/D based on optimization objectives and constraints), and RCS indicates the stealth characteristic, usually represented by the mean Radar Cross Section (RCS) over a specified angular range.

2.2 Action Space

Since each design variable is relatively independent while collectively determining the aircraft geometry, the agent must generate a continuous action vector to control the adjustment of these variables. Accordingly, the action space is defined as:

$$A = \{a_1, a_2, \ldots, a_n\} \quad -a \le a_i \le a \quad i = 1,..n \tag{4}$$

Here, n represents the number of design variables. In the actual computation, the new set of design variables $\rho_{i+1} = \rho_i + a_i$, along with the corresponding aerodynamic characteristics c_{i+1} and RCS values, are predicted and updated via the surrogate model.

2.3 Reward Function

In aerodynamic-stealth shape optimization for aircraft, a reward function is formulated to guide the agent toward designs with superior performance. This function is defined directly from the aerodynamic and stealth characteristics corresponding to a given set of design variables, as expressed below:

$$r_t = f(c, RCS, \psi) \tag{5}$$

2.4 MPPO Aerodynamic-Stealth Optimization Model

In the traditional Proximal Policy Optimization (PPO) algorithm [10], the surrogate objective function form of Contrastive Language-Image Pre-training (CLIP) is adopted to limit the policy update step as

follows:

$$L^{CLIP}(\theta)=\hat{E}\left[\min\left(r_t(\theta)A_t, CLIP\left(r_t(\theta),1-\varepsilon,1+\varepsilon\right)A_t\right)\right] \tag{6}$$

Where

$$\begin{aligned} A_t(s_t,a_t)&=r_t+\gamma V_\omega(s_{t+1})-V_\omega(s_t)\\ r_t(\theta)&=\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)} \end{aligned} \tag{7}$$

The role of the advantage function is to reduce variance and enhance the robustness of model updates, $r_t(\theta)$ measures the magnitude of change between the current policy and the previous one, aiming to correct the distributional shift between successive policies, thereby enabling more stable policy optimization. The second term, the CLIP function, clips this ratio to the range $(1\text{-}\varepsilon,1\text{+}\varepsilon)$, eliminating the possibility of the current policy deviating too far from the old policy. Finally, the minimum of the clipped and unclipped objectives is taken. In this environment, the influence of low-quality sample information on model updates is highly detrimental. To address this, the CLIP surrogate objective function in the PPO algorithm is modified as follows:

$$L^{MAXCLIP}(\theta)=\max\left[\min\left(r_t(\theta)A_t, CLIP\left(r_t(\theta),1-\varepsilon,1+\varepsilon\right)A_t\right)\right] \tag{8}$$

Where

$$\begin{aligned} A_t(s_t,a_t)&=r_t+\gamma r_t(\theta)V_\omega(s_{t+1})-V_\omega(s_t)\\ r_t(\theta)&=\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)} \end{aligned} \tag{9}$$

The MPPO algorithm conducts a secondary, maximization-driven selection from the pool of samples pre-filtered by minimization and clipping operations. This process prioritizes the most informative experience for model updates, thereby shielding the model from low-quality samples and minimizing unnecessary environmental interaction. Furthermore, as MPPO allows multiple policy updates per iteration, successive states may stem from different policy distributions, introducing bias into the objective function. To mitigate this bias and ensure algorithmic unbiasedness, an importance sampling ratio is incorporated into the advantage estimation. The objective function for the Critic network is accordingly defined as:

$$L_{critic}(\omega)=\left(r_t+\gamma r_t(\theta)V_\omega(s_{t+1})-V_\omega(s_t)\right)^2 \tag{10}$$

In the MPPO algorithm, the objective function of the Actor network is given by:

$$L^{MPPO}(\theta)=L^{MAXCLIP}(\theta)-\beta KL(\pi_{\theta old},\pi_\theta) \tag{11}$$

where β is the KL penalty coefficient, which serves to keep the KL divergence within a specified range. The purpose of introducing this KL penalty term is to ensure that the distributions of the old and new policies remain sufficiently close, thereby enabling stable policy improvement across multiple updates. The pseudocode of the MPPO algorithm is presented in Table 1.

Table 1 MPPO Algorithm Process

MPPO algorithm
1.Initialization: Experience replay buffer M, *Actor* parameters θ, *Critic* parameters ω; set hyperparameters: batch size B,

learning rate γ.

2. Repeat:

1) Reset the environment and initialize the state S.

2) Repeat for t=1 to T:

a. Select an action at a_t according to the output distribution of the policy network ;

b. Store the transition (s,a,r,s') in the replay buffer M ;

c. Update the policy network ($Actor$) B times via gradient ascent and the value network ($Critic$) B times via gradient descent.

3) Clear the experience replay buffer M .

2.5 MPPO-Based Aerodynamic-Stealth Optimization Method

Building upon the philosophy of integrated aerodynamic-stealth design, this work advances traditional surrogate optimization by synergizing core aerodynamic-stealth techniques with the deep reinforcement learning MPPO algorithm. To manage the high complexity of full-aircraft shape optimization, a cohesive design methodology is established through the integration of a Kriging surrogate model. The complete workflow of this proposed framework is depicted in Figure 2.

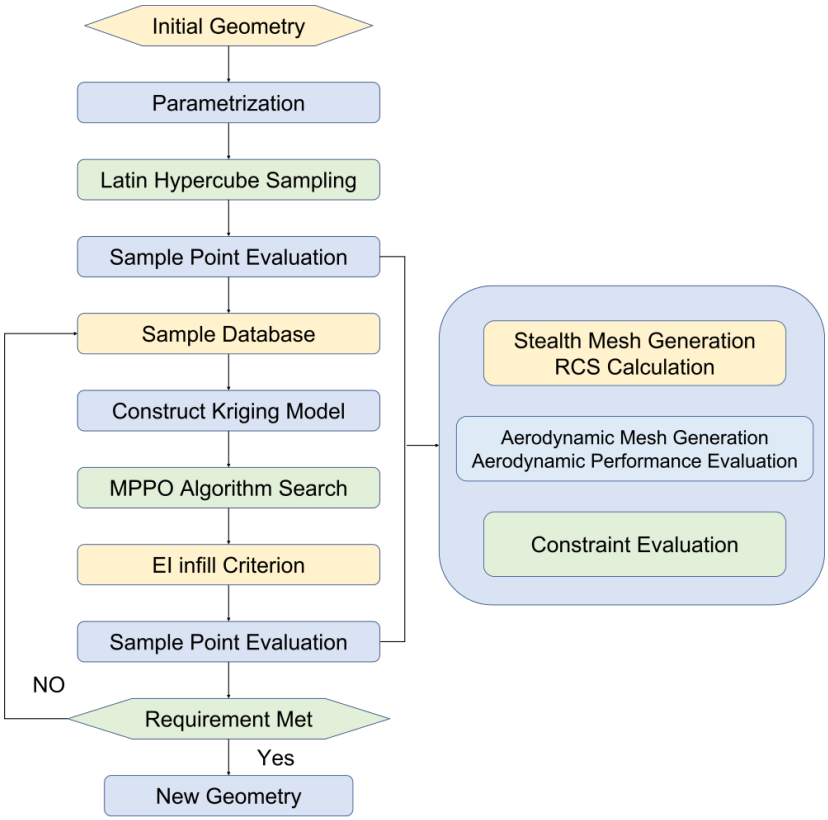


Figure 2 Aerodynamic Stealth Optimization Method Based on Deep Reinforcement Learning MPPO Algorithm.

3. Aerodynamic-Stealth Optimization Example

3.1 Function Test Case

This section selects the 10-dimensional Rosenbrock function to test the performance of the MPPO algorithm within the surrogate optimization framework, with a comparative analysis against the PSO algorithm. The optimal solution of this function is located within a narrow, smooth, parabolic-shaped valley. The optimization objective, defined by Equation (12), is considered fully achieved upon

convergence. During the optimization process, the values of the Rosenbrock function are predicted by a Kriging model. The corresponding results are presented in Figure 3 below.

$$f(\mathbf{X}) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \tag{12}$$

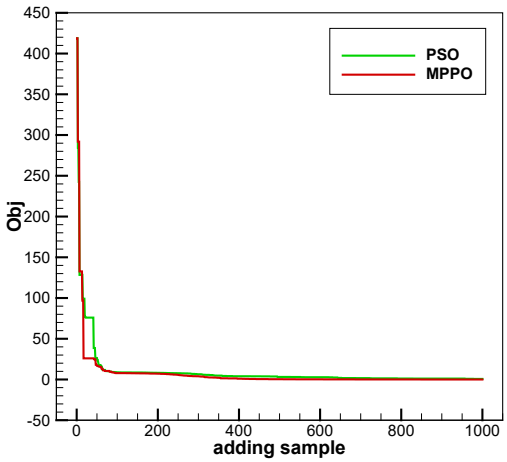


Figure 3 Convergence process of the Rosenbrock function values for two algorithms

Table 2 Number of extra points when fully converged

Algorithm	Obj	Adding sample
PSO	0.8192	958
MPPO	0.9797	396

The results indicate that for the Rosenbrock function, which features a unimodal and smooth design space, the MPPO algorithm converges after adding approximately 400 sample points. In contrast, the PSO algorithm requires nearly 960 sample points to reach convergence. This demonstrates the superior search speed of the MPPO algorithm.

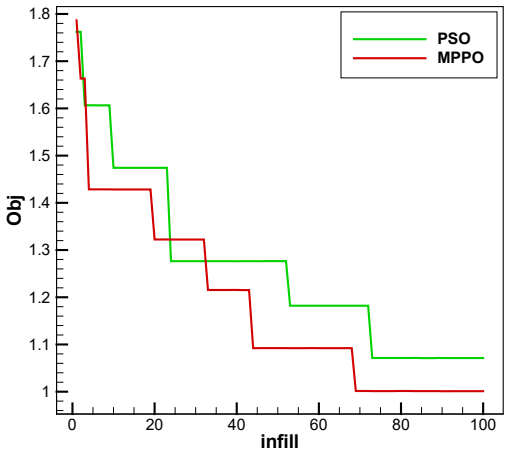
3.2 Aerodynamic-Stealth Airfoil Optimization

To validate the optimization performance of the MPPO-based method for aerodynamic-stealth airfoil design, this section employs the NACA 65,3-014 airfoil as the test case. The design condition is set at $Ma = 0.65, C_l = 0.25, Re = 6e7$. The constraints require that the relative thickness be no less than that of the original airfoil and the pitching moment coefficient be greater than 0.03. The optimization objectives are to reduce drag and, simultaneously, lower the average forward-sector RCS (within $\pm$ 30°) at an incident frequency of 9 GHz. The radar cross section is computed using the Method of Moments (MoM). The optimization process utilizes the developed MPPO-based surrogate optimization framework. Airfoil deformation is parameterized via the Class-Shape Transformation (CST) method, with an initial sample of 200 points generated by Latin Hypercube Sampling (LHS). A comparison with the traditional Particle Swarm Optimization (PSO) algorithm is conducted to verify the effectiveness of the proposed method.

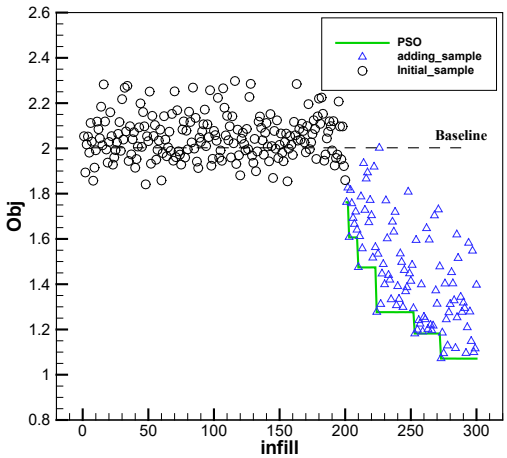
$$\begin{aligned} Min \quad & \frac{C_d}{C_{d_initial}} + \frac{RCS}{RCS_{initial}} \\ s.t. \quad & C_l = 0.25 \\ & C_m \geq 0.03 \\ & thick_{max} \geq thick_{max_initial} \end{aligned} \tag{13}$$

The convergence histories of the two methods are shown in Figure 4. The red curve, representing the MPPO algorithm's objective (Obj), converges faster than the green curve of the PSO algorithm.

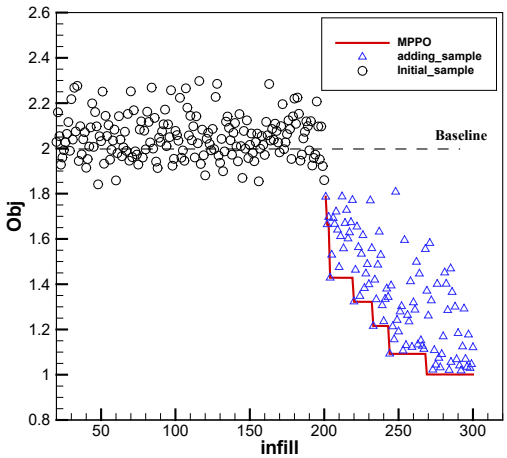
A comparison of the designed airfoils is presented in Table 3 ("Comparison of Airfoil Optimization Results"). Compared to the baseline NACA 65,3-014 airfoil (denoted as Ori_foil), the MPPO-designed airfoil (MPPO_foil) achieves a drag reduction of 5 counts. Its pitching moment coefficient is trimmed to a neutral level, and its forward-sector RCS is reduced by approximately 91%. The PSO-designed airfoil (PSO_foil) achieves a drag reduction of 4 counts and an RCS reduction of about 86%. As evident from Figure 4 and Table 3, the MPPO algorithm not only converges faster but also yields superior optimization results compared to PSO. Figure 5 illustrates that the optimized airfoils, MPPO_foil and PSO_foil, exhibit the following geometric characteristics relative to the NACA 65,3-014, while satisfying the thickness constraint: a reduced leading-edge radius and increased positive camber, giving a "beak-like" shape; a slight aft shift of the maximum thickness location; and an increased trailing-edge radius with enhanced negative camber. As shown in Figure 6, for the MPPO_foil, the upper-surface low-pressure region moves forward, the leading-edge suction peak is significantly reduced, and pressure recovery occurs more rapidly. Concurrently, to meet the pitch-up moment constraint despite its reduced leading-edge radius, the positive pressure coefficient on the lower surface near the leading edge increases. Figure 7 presents a comparison of the forward-sector RCS characteristics. Compared to the initial Ori_foil, the peak RCS values across all angles are substantially lower for both MPPO_foil and PSO_foil, with MPPO_foil achieving a greater reduction magnitude.



(a)



(b)



(c)

Figure 4 Convergence process of two algorithms

Table 3 Comparison of Airfoil Optimization Results

	C_l	C_d	C_m	RCS (m^2)
Ori_foil	0.25	0.0069	-0.0066	0.0344
MPPO_foil	0.25	0.0064	0.0309	0.0029
PSO_foil	0.25	0.0065	0.0307	0.0049

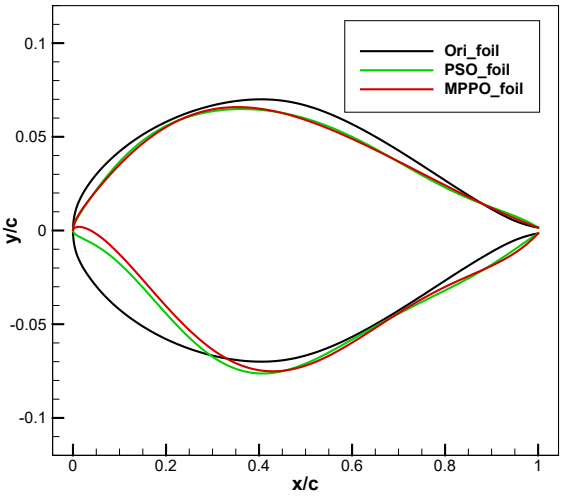


Figure 5 Comparison of Airfoil Before and After Optimization

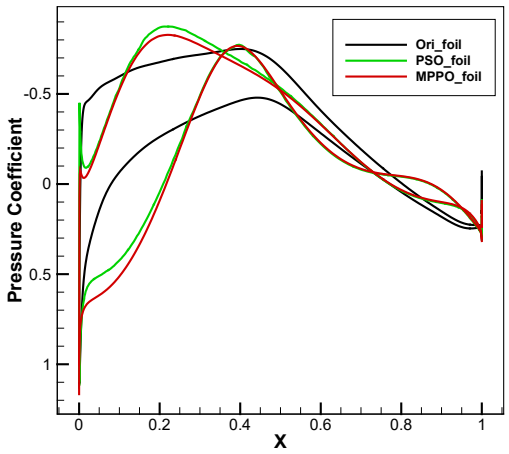


Figure 6 Comparison of Airfoil Pressure Coefficient Distribution

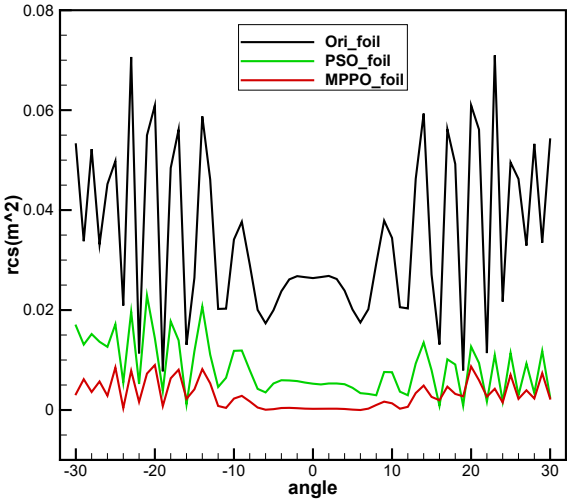


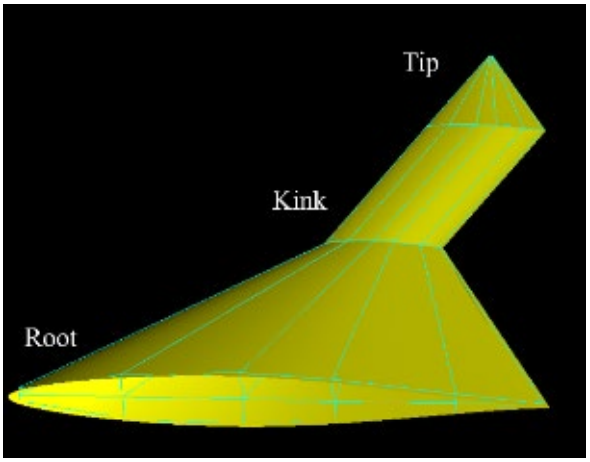
Figure 7 Comparison of Forward RCS Characteristics of Airfoils at 9 GHz

3.3 Flying Wing Configuration Optimization

Based on the proposed aerodynamic-stealth surrogate optimization design method, this section conducts an aerodynamic-stealth optimization design for an X-47B-like configuration. A comparison with the traditional Particle Swarm Optimization (PSO) algorithm is performed to validate the effectiveness of the proposed method for different types of aircraft. The design condition is set at $Ma = 0.8, C_l = 0.25, Re = 6.8e6$. To ensure flight control efficiency, the X-47B must maintain a pitch-up moment while reducing drag. Concurrently, it is required to reduce the average forward-sector Radar Cross Section (RCS) within $\pm 60^\circ$ at an incident frequency of 9 GHz. Consequently, the design objectives are to minimize the wing drag coefficient and the average RCS in the yaw direction. The design constraints are that the maximum thickness at three specified sections of the X-47B must not decrease, and the pitching moment coefficient must not be lower than its initial value. As the

objectives involve drag reduction at a fixed lift and reduction of the average forward $\pm 60^\circ$ RCS, these two targets are combined into a single objective function via linear normalization with an equal weighting of 1:1. The mathematical model of the optimization problem is formulated as follows:

$$\begin{aligned} Min \quad & obj = \frac{C_D}{C_{D_initial}} + \frac{RCS}{RCS_{initial}} \\ s.t.: \quad & C_L = 0.25 \\ & \left| C_{\mathrm{M}_initial} \right| \geq \left| C_M \right| \\ & thick_{1,2,3max} \geq thick_{1,2,3\,max_initial} \end{aligned} \tag{14}$$



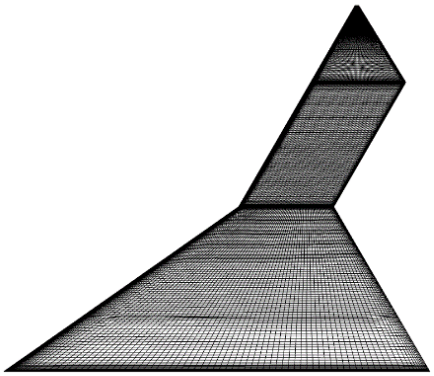
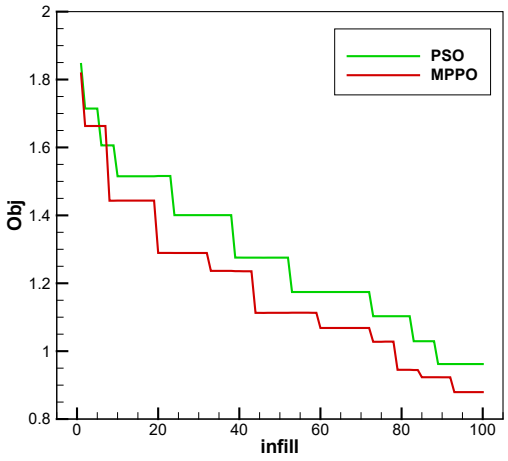
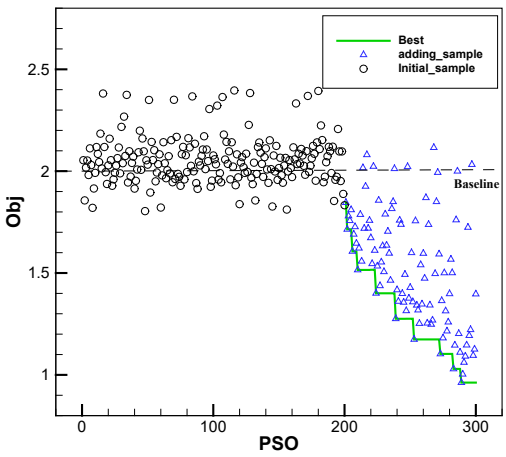


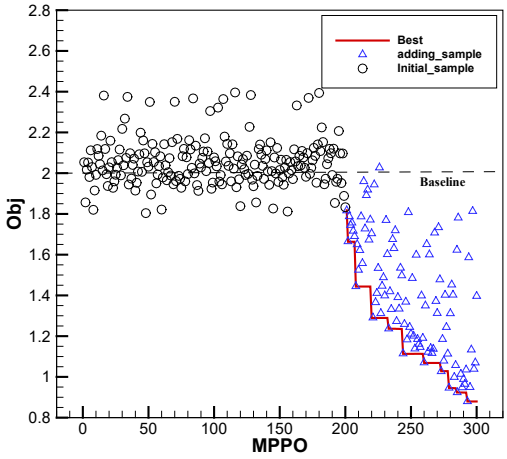
Figure 8 FFD Control Box and Mesh

The optimization results of the two algorithms are denoted as Obj_MPPO and Obj_PSO, respectively. Figure 9 shows that both methods have essentially converged after 90 sample infill points. A comparison of the optimization results is presented in Table 4, Figure 10 and Figure 11, confirming that all optimized designs satisfy the thickness constraints. Under these satisfied constraints, both algorithms achieve significant improvements in aerodynamic performance at the design lift-drag condition compared to the BaseModel. The drag coefficients for Obj_MPPO and Obj_PSO are reduced from the baseline value of 0.019665 to 0.014488 and 0.014982, corresponding to reductions of 51.7 counts and 46.8 counts, respectively. In terms of stealth characteristics, the average forward $\pm 60^\circ$ RCS for Obj_MPPO and Obj_PSO decreases from the original 3.7951 m^2 to 0.5404 m^2 and 0.7565 m^2, respectively, representing substantial reductions. Furthermore, the converged result of the MPPO algorithm is notably superior to that of PSO. Specifically, MPPO achieves a 4.9-count lower drag coefficient and a 0.2161 m^2 lower average forward $\pm 60^\circ$ RCS compared to PSO. These results demonstrate that the MPPO algorithm possesses a significantly enhanced global optimization capability over the traditional PSO algorithm.



(a)





(c) (d)

Figure 9 The convergence process of two algorithms

Table 4 Comparison of Optimized Wing Layout Results

	C_l	C_d	C_m	RCS (m^2)	Object
X47B	0.25	0.019665	-0.026167	3.7951	/
Obj_MPPO	0.25	0.014488	-0.025369	0.5404	0.8791
Obj_PSO	0.25	0.014982	-0.024593	0.7565	0.9612

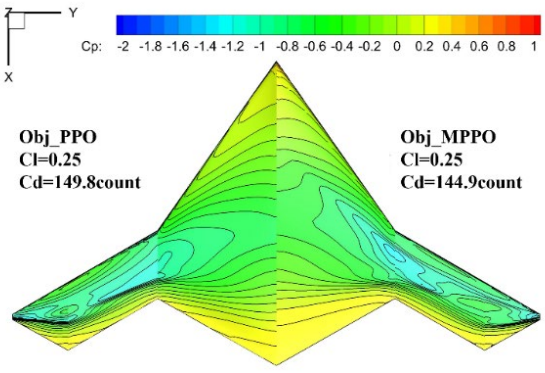


Figure 10 Comparison of 3D pressure cloud maps before and after optimization

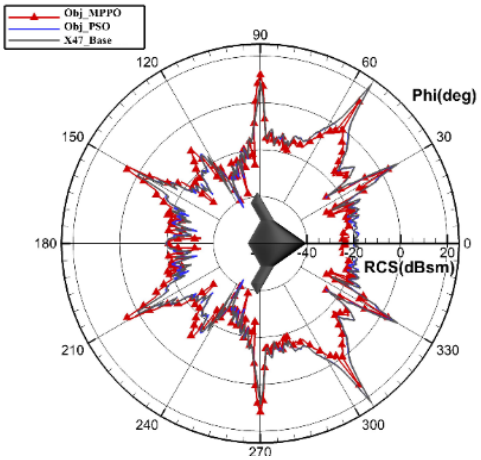


Figure 11 Comparison of yaw-direction RCS before and after 9GHz optimization

4. Conclusion

This paper combines the sample-based modified Proximal Policy Optimization (MPPO) algorithm with surrogate-based optimization to formulate an aerodynamic and stealth optimization method based on the MPPO algorithm, offering a new direction for the development of aircraft shape design methods. The proposed method is applied to the integrated aerodynamic and stealth design of a flying-wing configuration, improving the overall aerodynamic and stealth performance of this flying-wing configuration. The specific conclusions are as follows:

1) This study investigates the integration of the sample-based modified Proximal Policy Optimization algorithm with aerodynamic/stealth surrogate-based optimization. In accordance with the optimization requirements for aircraft aerodynamics and stealth, the state space, action space, and reward function of the MPPO algorithm model under the aerodynamic/stealth optimization environment are established. An MPPO optimization model and an integrated aerodynamic/stealth optimization design method that meet the optimization design requirements are successfully constructed.

2) Aerodynamic and stealth optimization design is performed on the NACA 65,3-014 airfoil and an X-47B-like configuration. The MPPO algorithm achieves superior results in both aerodynamic and stealth characteristics. The outcomes demonstrate that, compared with the conventional particle swarm optimization (PSO) algorithm, the deep reinforcement learning MPPO algorithm exhibits higher optimization efficiency and search capability, along with better global optimization performance. This validates the effectiveness of the MPPO-based aerodynamic/stealth surrogate-based optimization framework.

5. Contact Author Email Address

Corresponding author, Huan Zhao, Email: huanzhao@nwpu.edu.cn

6. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.

[2] Van Hasselt H, Guez A, Silver D. Deep Reinforcement Learning with Double Qlearning[J]. Computer Science, 2015.

[3] Mnih V, Badia, Adrià Puigdomènech, et al. Asynchronous Methods for Deep Reinforcement Learning[J]. 2016.

[4] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge[J]. nature, 2017, 550(7676): 354-359.

[5] Yonekura K, Hattori H. Framework for design optimization using deep reinforcement learning[J]. Structural and Multidisciplinary Optimization, 2019, 60(4): 1709-1713.

[6] Li, Runze,Zhang, Yufei,Chen, Haixin.Learning the Aerodynamic Design of Supercritical Airfoils Through Deep Reinforcement Learning[J].AIAA JOURNAL,2021,59,(10):3988-4001.DOI:10.2514/1.J060189.

[7] Scavella, P.Paolillo, G.Greco, C. S.Deep reinforcement learning-based airfoil design and optimization: An aerodynamic analysis[J].AEROSPACE SCIENCE AND TECHNOLOGY,2025,167,DOI:10.1016/j.ast.2025.110638.

[8] Kulfan B M. Universal Parametric Geometry Representation Method[J]. Journal of Aircraft, 2008, 45(1): 142-158.

[9] Braibant, V., and Fleury, C. "Shape optimal design using B-splines," Computer Methods in Applied Mechanics & Engineering Vol. 44, No. 3, 1984, pp. 247-267.

[10]Barto A G, Sutton R S, Anderson C W. Neuronlike adaptive elements that can solve difficult learning control problems[J]. IEEE Trans on Systems, Man and Cybernetics, 1983, 13 (5): 834-846.

[11]Mnih V, Badia, Adrià Puigdomènech, et al. Asynchronous Methods for Deep Reinforcement Learning[J]. 2016.

[12] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge[J]. nature, 2017, 550(7676): 354-359.

[13] Yonekura K, Hattori H. Framework for design optimization using deep reinforcement learning[J]. Structural and Multidisciplinary Optimization, 2019, 60(4): 1709-1713.

[14] Li, Runze,Zhang, Yufei,Chen, Haixin.Learning the Aerodynamic Design of Supercritical Airfoils Through Deep Reinforcement Learning[J].AIAA JOURNAL,2021,59,(10):3988-4001.DOI:10.2514/1.J060189.

[15] Scavella, P.Paolillo, G.Greco, C. S.Deep reinforcement learning-based airfoil design and optimization: An aerodynamic analysis[J].AEROSPACE SCIENCE AND TECHNOLOGY,2025,167, DOI:10.1016/j.ast.2025.110638.