International Journal for Housing Science and Its Applications

Publish September 9, 2025. Volume 47, Issue 1 Pages 462-474

https://doi.org/10.70517/ijhsa47139

Reinforcement learning-based parameter tuning technology for RF circuit design

Yuchen Mu[1,*] and Zhen Tian[2]

[1] School of Materials Science and Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, 030025, China

[2] James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK

Corresponding authors: (e-mail: 18844399319@163.com).

Abstract In circuit design work, parameter optimization is an inevitable issue, especially in analog circuit design, which requires a high level of experience from designers. In traditional parameter optimization processes, designers may also rely on optimization algorithms to find optimal solutions. This paper uses reinforcement learning algorithms to find optimal strategies, exploring two functions under model-free reinforcement learning algorithms: the value function and the policy function. These functions are estimated using recursive forms and policy gradients. Using the Y parameter to extract equivalent circuit parameters in RF circuits, a frequency AI model is established to optimize the parameters of RF circuits. The optimization effect is verified through metrics such as gain and frequency, and the final optimized results of the RF circuit are calculated. The distribution of the receptive field in the value function method model tends toward a Gaussian distribution, exhibiting sparsity, with weight values primarily distributed at both ends of 0, and the frequency approaching 120. This paper proposes three optimization schemes for parameter tuning, with the optimal solution coordinates for Schemes 1 to 3 being [3.671, 0.749], [3.726, 0.834], and [3.847, 0.578], respectively. After optimization, the static power consumption of the RF circuit was reduced by over 54% compared to before optimization, and the circuit cost was reduced by over 40%, indicating that the method proposed in this paper has good optimization effects.

Index Terms reinforcement learning, optimal policy, policy gradient, RF circuit, parameter tuning

I. Introduction

With the advancement of technology, communication methods have been continuously evolving, and the scope of communication has gradually expanded to encompass various aspects of people's lives, including work, study, daily communication, and signal transmission [1]-[3]. Simultaneously, communication methods have become increasingly diverse, with technologies such as RFID, 3G, GPS, Wi-Fi, WLAN, ZigBee, and WiMAX all utilizing wireless communication methods [4]. The wireless communication field primarily relies on radio frequency (RF) circuit systems for implementation. Various wireless communication systems adopt similar structures to accomplish the two processes of wireless communication transmission and reception [5]-[8]. RF communication technology, with its advantages of low cost, low power consumption, and simplicity of development, has become the fastest-growing and most widely applied technology in the communication field in recent years [9], [10]. Its primary application areas include industrial control, environmental monitoring, smart homes, and unmanned operations [11].

Although RF communication technology is already relatively mature, RF circuit design remains a critical technical factor influencing the reliable operation of communication systems, such as impedance matching, filter selection, component arrangement, and PCB routing across different operating frequency bands [12]-[15]. RF circuit design primarily aims to achieve technical specifications such as high transmission reliability, low power consumption, low cost, high communication quality, and high data transmission rates [16]-[18]. By leveraging intelligent optimization techniques to analyze RF circuits, electronic components such as impedance transformers, filters, and DC blocks within the RF circuit can be adjusted to improve RF circuit return loss, reduce insertion loss, increase transmission distance, and decrease packet loss rates [19]-[22].

This paper uses the particle swarm algorithm as an example to explain the operation process of global optimization algorithms and proposes two reinforcement learning algorithms: model-free and model-based. Based on this, Y parameters are used to extract parameters from the equivalent circuits of RF devices in the RF circuit. Electromagnetic results are calculated using full-wave simulation tools and used as training and testing data for generating AI models. The Y parameter data is normalized to construct an RF device equivalent circuit simulation model. Through simulation and testing experiments, the target circuit parameters are verified, the performance parameters of different optimization methods are compared, and the

results of the RF circuit parameter optimization designed in this paper are obtained, verifying the feasibility of the proposed method.

II. Algorithm principles

II. A. Global Optimization Algorithms

Taking the Particle Swarm Optimization (PSO) algorithm as an example, this algorithm is inspired by the study of bird flocks hunting for prey and is a heuristic search algorithm [23]. The core idea is to find the optimal solution through information sharing and collaborative assistance among particles. The implementation of PSO is relatively simple, and the number of hyperparameters is also relatively small.

In the Particle Swarm Optimization algorithm, objects (birds) are represented by abstract particles in an N-dimensional space. A population consists of multiple particles, and the position of the i th particle can be described by the coordinate vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ and its movement is represented by the velocity vector $V_i = (v_{i1}, v_{i2}, ..., v_{iN})$. Each particle has different velocities in each dimension. Particles update their positions using their own known information X_i and p_{best}, as well as the shared information g_{best} within the population. The update formula is given by Equation (1):

$$\begin{aligned} v_i^{(t+1)} &= w \cdot v_i^{(t)} + c1 \cdot rand_{1i}^{(t)} \cdot (p_{besti}^{(t)} - x_i) + c2 \cdot rand_{2i}^{(t)} \cdot (g_{best}^{(t)} - x_i) \\ x_i^{(t+1)} &= x_i^{(t)} + v_i^{(t+1)} \end{aligned} \tag{1}$$

In differential evolution, the mutation operation is implemented using a differential strategy. A common approach is to randomly select two different individuals $X_i^{(t)}$ and $X_j^{(t)}$ from the population, scale them, and then perform the mutation operation on a specific individual $X_k^{(t)}$ to be mutated:

$$V_i^{(t+1)} = X_k^{(t)} + F \cdot (X_i^{(t)} - X_j^{(t)}) \tag{2}$$

Crossing is also a step that reflects random selection, i.e., randomly crossing mutated individuals with unmutated individuals:

$$u_{ij}^{t+1} = \begin{cases} v_{ij}^{(t+1)}, & \text{if } rand() < CR \\ x_{ij}^{(t)}, & \text{else} \end{cases} \tag{3}$$

CR is the survival probability.

The selection process adopts a greedy strategy, i.e., selecting the individual with the best fitness to survive:

$$X_i^{t+1} = \begin{cases} U_i^{(t+1)}, & \text{if } f(U_i^{(t+1)}) > X_i^{(t)} \\ X_i^{(t)}, & \text{else} \end{cases} \tag{4}$$

II. B. Reinforcement Learning Algorithms

The goal of reinforcement learning is to find the optimal strategy [24]. To couple exploration, strategies often use random strategies $\pi(a \mid s)$, which is the distribution of action a in state s. In the process of searching for strategies, reinforcement learning algorithms can be divided into model-free and model-based algorithms based on different interaction methods (direct or indirect) between the intelligent agent and the environment. The relationship between various algorithms in reinforcement learning is shown in Figure 1.

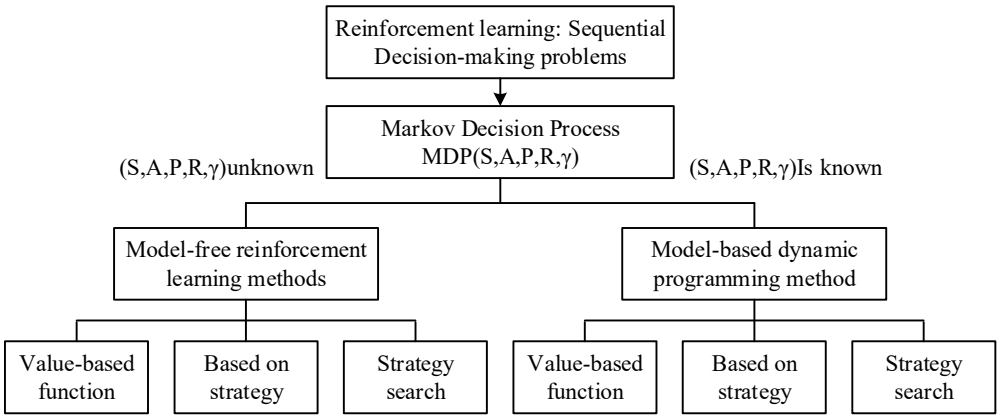


Figure 1: The relationship among various algorithm systems in reinforcement learning

II. B. 1) Model-free reinforcement learning algorithms

(1) Value Function

In model-free reinforcement learning algorithms, since the model of the agent's interaction with the environment is unknown, Monte Carlo methods use empirical averaging to estimate the value function. Obtaining sufficient experience is the core of model-free reinforcement learning, and whether the correct value function can be obtained depends on whether the experience is sufficient [25]. When obtaining experience, it is necessary to conduct multiple experiments based on the current strategy to generate multiple sets of data, which are referred to as experience. As shown in Figure 2.

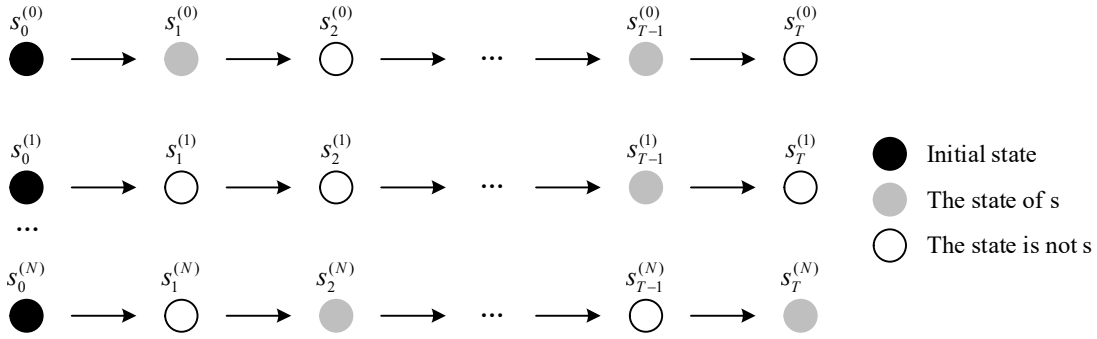


Figure 2: Sampling process of the Monte Carlo algorithm

In reinforcement learning algorithms, the value function is used to evaluate the quality of a strategy. The value function is defined as follows:

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \mid S_t = s\right] \\ Q^{\pi}(s,a) &= E_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \mid S_t = s, A_t = a\right] \end{aligned} \tag{5}$$

The average of the experience is obtained by using the statistics obtained from sampling to estimate the distribution and find the mean, that is:

$$\hat{V}(s) = \frac{\sum_{i}^{N}\sum_{s_t=s} R_i(s_t)}{N(s)} \tag{6}$$

$R_i(s_t) = \sum_{k=t}^{T-1}\gamma^{k-t} r_{k+1}$ is the total discounted return obtained in the i th experiment under state s_t, and $N(s)$ is the number of times state s occurs. According to the law of large numbers, we have:

$$\lim_{N(s)\to\infty}\hat{V}(s) = V^{\pi}(s) \tag{7}$$

After estimating the value function of the state, it is natural to improve the policy. In Monte Carlo, the method for improving the policy is to maximize the action value function, i.e., $\pi(s) = \arg\max_a Q(s,a)$, where the action value function $Q(s,a)$ was introduced in (5). Therefore, the learning speed is slow, and the learning efficiency is not high. Another search strategy method is called the temporal difference (TD) method.

Equation (6) defines the value function. While the Monte Carlo method uses statistical methods for estimation, the TD method converts it into a recursive form for estimation:

$$V^{\pi}(s) = E_{\pi}[r_{t+1} + \gamma V^{\pi}(S_{t+1}) \mid S_t = s] \tag{8}$$

When calculating cumulative returns using a recursive approach, the value function of the next step is used to estimate the current cumulative return, i.e., $R_t^1 = r_{t+1} + \gamma V(S_{t+1})$ represents the use of the value function of the next step to estimate. Of course, the value function of the next n steps can also be used for estimation: $R_t^n = r_{t+1} + \gamma r_{t+2} + \cdot + \gamma^{n-1} V_{S_{t+n}}$. When the value function of the next step is accurate, $R_t^1, R_t^2, \cdots, R_T^n$ should all be equivalent. However, in experiments, this assumption clearly does not hold, so further, $TD(\lambda)$ uses these n next states to estimate the current return:

$$\begin{aligned} R_t^{\lambda} &= (1-\lambda)(R_t^1 + \lambda R_t^2 + \cdot + \lambda^{n-1} R_t^n) \\ &\approx (1-\lambda)(1+\lambda+\cdot+\lambda^{n-1}) V(S_t) \\ &= (1-\lambda^n) V(S_t) \\ &\approx V(S_t) \end{aligned} \tag{9}$$

(2) Policy function

The main methods for optimizing policies are policy gradient methods and confidence interval methods, and the optimization objective $J(\theta)$ is to maximize the return under the policy π_θ. For random policies, it can be defined as [26]:

$$\theta^* = \arg\max_\theta \sum_{s\in \mathrm{S}} d(s) \sum_{a\in \mathrm{A}} \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a) \tag{10}$$

$d(s)$ represents the distribution of states.

The policy gradient method can directly calculate the gradient of the objective function $\nabla_\theta J(\theta)$, and use the gradient ascent method to update the weights $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$. The gradient can be expressed as:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_{s\in \mathrm{S}} d(s) \sum_{a\in \mathrm{A}} \nabla_\theta \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a) \\ &= \sum_{s\in \mathrm{S}} d(s) \sum_{a\in \mathrm{A}} \pi_\theta(a \mid s) \frac{\nabla_\theta \pi_\theta(a \mid s)}{\pi_\theta(a \mid s)} Q^{\pi_\theta}(s,a) \\ &= \sum_{s\in \mathrm{S}} d(s) \sum_{a\in \mathrm{A}} \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a) \\ &= E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a)] \end{aligned} \tag{11}$$

II. B. 2) Model-based reinforcement learning algorithms

The supervised learning module is the learner, which obtains actual trajectory data and optimal control rates from the optimal controller and uses a neural network to fit the relationship between them, thereby learning the optimal control method. Coupling is achieved in GPS using constraint conditions, which take the following form:

$$\begin{aligned} &\min_{q(\tau),\theta} \quad E_q[\ell(\tau)] - \mathrm{H}(q(\tau)) \\ &s.t. q(s_1) = p(s_1) \\ &q(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid s_t, a_t) \\ &D_{KL}(\pi_\theta(a_t \mid s_t) q(s_t) \,\|\, q(s_t, a_t)) = 0 \end{aligned} \tag{12}$$

In this context, ℓ is the meta-loss defined in the environment, which is the opposite of the definition of reward—the smaller the meta-loss, the better. H represents the entropy of the distribution q, and the optimization objective is to minimize the cost function ℓ while maximizing the entropy of q. The first two terms in the constraints ensure that the distribution $q(\tau)$ of the control module is consistent with the initial state of the environment, i.e., the state transition distribution, and are primarily applied in trace optimization. The last term requires the divergence between the two distributions to be zero, meaning that the control strategy $q(a_t \mid s_t)$ and the distribution of the supervised learning module $\pi_\theta(a_t \mid s_t)$ are completely consistent, indicating coupling between the two modules. The role of the optimal controller is to adjust its control strategy to minimize the meta-loss. The constraint conditions can be expressed in the form of Lagrange multipliers as:

$$\begin{aligned} &\mathrm{L}(\theta, q, \lambda) = E_q[\ell(\tau)] - \mathrm{H}(q(\tau)) \\ &+ \sum_{t=1}^{T} \lambda^t D_{KL}(q(s_t)\pi_\theta(a_t \mid s_t) \,\|\, q(s_t, a_t)) \end{aligned} \tag{13}$$

Therefore, the constrained objective function is transformed into an optimization Lagrange function: alternately optimize the control system q, the supervised learning module weights θ, and the Lagrange multiplier λ_t.

Therefore, the search process of the GPS algorithm is to alternately optimize the track and the supervised learning module. The process of optimizing the control strategy q of the above regulator is called track optimization. In general optimal controllers, assuming the dynamic system is linear and the cost function is quadratic, a local linear Gaussian dynamic system is used to approximate $q(s_{t+1} \mid s_t, a_t)$, and the control strategy is $q(a_t \mid s_t) = \mathrm{N}(a_t + Ks_t, A_t)$ is defined, where N is the Gaussian distribution. With the definition of q, the entropy of q can be calculated:

$$\mathrm{H}(q) = -\frac{1}{2}\ln|A_t| \tag{14}$$

Substituting entropy into equation (14) and using the Laplace approximation, i.e., modeling the policy $\pi_\theta(a_t \mid s_t)$ with a local linear Gaussian distribution, whose mean is $\mu_t^\pi(s_t)$ and variance is Σ_t^π while the dynamic system is $q(s_t+1 \mid s_t, a_t) = \mathrm{N}(f_{st}' s_t + f_{at}' a_t, F_t)$ Then, the gradient of the cost function is represented as ℓ_{sat}', i.e., the derivative of the cost function with respect to the state-action $(s_t, a_t)^T$, and the Hessian matrix is represented as $\ell_{sa,sat}''$. Thus, the objective function of the control module can be written in the following form:

$$\begin{aligned}
\mathrm{L}(q) \approx & \sum_{t=1}^{T} \frac{1}{2}(s_t, a_t)\ell''_{sa,sat}(s_t, a_t)^T + (s_t, a_t)\ell'_{sat} \\
& + \frac{1}{2}tr(\Sigma_t \ell''_{sa,sat}) - \frac{1}{2}\ln|A_t| + \frac{\lambda_t}{2}\ln|A_t| \\
& + \frac{\lambda_t}{2}(\hat{a}_t - \mu_t^\pi(\hat{s}_t))^T A_t^{-1}(\hat{a}_t - \mu_t^\pi(\hat{s}_t)) + \frac{\lambda_t}{2}tr(A_t^{-1}\Sigma_t^\pi) \\
& + \frac{\lambda_t}{2}tr(S_t(K_t - \mu_t^\pi(\hat{s}_t))^T A_t^{-1}(K_t - \mu_t^\pi(\hat{s}_t)))
\end{aligned} \tag{15}$$

S_t denotes the covariance matrix of the state distribution $q(s_t)$ and A_t denotes the covariance matrix of the conditional distribution $q(a_t \mid s_t)$

In the policy optimization module, if the control policy remains unchanged, it becomes a supervised learning optimization problem. The training samples s_{ti} are sampled from the distribution $q(s_t)$ of the dynamical system at time t, and the objective function can be written as:

$$\begin{aligned}
\mathrm{L}(\theta) &= \sum_{t=1}^{T}\lambda_t \sum_{i=1}^{N} D_{KL}(\pi_\theta(a_t \mid s_{ti}) \| q(a_t \mid s_t)) \\
&= \sum_{t=1}^{T}\lambda_t \sum_{i=1}^{N} \frac{1}{2}[tr(\Sigma_t^\pi(s_{ti})A_t^{-1}) - \ln\Sigma^\pi(s_{ti}) \\
&+ (K_t s_{ti} + k_t - \mu^\pi(s_{ti}))^T A_t^{-1}(K_t s_{ti} + k_t - \mu^\pi(s_{ti}))]
\end{aligned} \tag{16}$$

Among them, matrix K_t and vector k_t are used to fit the local linear relationship of the strategy.

II. C. RF device parameter extraction based on reinforcement learning

II. C. 1) Establishing a frequency AI model

When extracting parameters from the equivalent circuit of RF devices using Y parameters, relatively dense frequency points are required to supplement the Y parameter information corresponding to the frequency points needed for equivalent circuit parameter extraction.

The electromagnetic results calculated by the efficient full-wave simulation tool IC (UltraEM®) at certain frequencies will be used as training and testing data for generating AI models. Once the AI model is established, electromagnetic results at any frequency can be directly obtained from the model. Then, the equivalent circuit parameters are extracted using Y-parameters via the innovative hybrid genetic algorithm proposed in this paper.

The Y-parameters at certain frequency points are calculated using UltraEM®, and the Y-parameter data is normalized. The normalized data is divided into a training dataset and a testing dataset. 70% of the data is used as training data, and 30% is used as testing data. The Adam optimization algorithm is used to minimize the loss function MSE, which can be expressed as:

$$MSE_{(y,y')} = \frac{\sum_{i-1}^{n}(y_i - y_i')^2}{n} \tag{17}$$

where i is the i th sample, y_i is the true value corresponding to the input item x_i, y_i' is the predicted value corresponding to the input item x_i calculated by the neural network, and n is the number of samples. Once the loss function meets the requirements, an AI model is established. This model can be used to obtain the Y parameter at any frequency.

II. C. 2) Equivalent circuit modeling of RF devices

The full-wave simulation results of the RF device layout are obtained through an AI model, and the corresponding Y parameters are marked as Y_{EM}. By using a Spice simulator, the Y parameters of the equivalent circuit can be obtained from the calculated port voltages and currents, namely:

$$I = YV \tag{18}$$

where Y is an $N \times N$ matrix, V and I are the port voltage vector and port current vector, respectively. The elements of the Y matrix are defined as:

$$Y_{nm} = \frac{I_n}{V_m}|_{V_k=0, k\neq m} \tag{19}$$

If the voltage of the excitation port is set to 1 volt, then the Y parameter is equal to the port current, i.e.:

$$Y_{nm} = I_n \tag{20}$$

Mark the simulated Y parameters in the equivalent circuit as Y_{spice}. To make the results of the equivalent circuit fit the accurate EM simulation, the objective function to be optimized can be defined as:

$$obj(Y_{spice}, Y_{EM}) = \sum_{i=fs}^{fe}\sum_{m=1}^{N}\sum_{n=1}^{N}|(Y_{EMmn})_i - (Y_{spicemn})_i| \tag{21}$$

In this context, N is the slogan, and fs and fe represent the start and end frequencies, respectively.

II. D. RF circuit parameter tuning based on reinforcement learning

This paper uses the blktrace tool to capture I/O information under different data loads. Based on the differences in I/O characteristics under different loads, it identifies changes in the data load of the power Internet of Things edge. Figure 3 shows the I/O information capture, illustrating the process of the blktrace tool collecting disk I/O information at the block device layer.

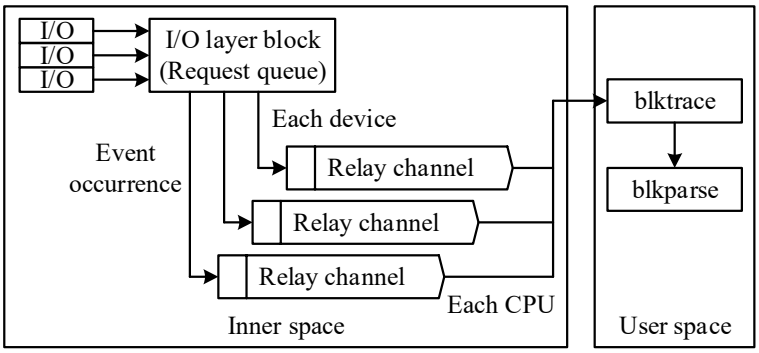


Figure 3: I/O information capture

First, the workload analyzer identifies new data workload feature types, which are combined with predefined configuration parameters to form a scoring data matrix. Singular value decomposition (SVD) is then used to handle missing values in the matrix. Next, the similarity between workload types in the power data storage system is calculated to generate a set of nearest neighbor workload types. Finally, a combination of a performance cost function and workload similarity estimation is used to generate recommended configuration parameters. Based on the actual runtime of the workload and the cluster state, it is determined whether to update the power data storage system configuration parameter database. The specific implementation steps are as follows.

The workload analyzer identifies and collects different workload feature types, testing the resource utilization of each workload feature under different configuration parameter samples within the cluster. To facilitate the recommendation process calculations, resource utilization is treated as the score value of the data workload for different configuration parameters.

Use the SVD method to decompose matrix A and obtain the product of two matrices P and Q to represent the rating matrix A:

$$A = PQ \tag{22}$$

Using the known ratings in the rating matrix A to train the matrices P and Q, such that the product of P and Q best fits the known ratings. Let the missing value rating for load type u and configuration parameter i in the rating matrix A be denoted as R'_{ui}. Multiplying the u th row P_u of matrix P by the i th column Q_i of matrix Q yields equation (23):

$$R'_{ui} = P_u^{\acute{u}} Q_i \tag{23}$$

Assuming that the score R_{ui} is known, for each missing value error in matrix A, let $E_{ui} = R_{ui} - R'_{ui}$, and calculate the total sum of squared errors as follows:

$$E_{SSE} = \sum_{ui} (R_{ui} - \sum_{k=1}^{K} P_{uk} Q_{ki})^2 \tag{24}$$

In the equation, P_{uk} and Q_{ki} are the corresponding elements in matrices P and Q, respectively.

To make the predicted values better fit the matrix A, it is necessary to minimize the value of E_{sse} through training. Using the gradient descent method, the gradient of the value of E_{sse} at the variable P_{uk} is obtained as:

$$\frac{\partial}{\partial P_{uk}} E_{SSE} = \frac{\partial}{\partial P_{uk}} E_{ui}^2 = -E_{ui} Q_{ki} + \lambda P_{uk} \tag{25}$$

In the equation: λ is the regularization factor. To avoid overfitting, the objective function $E_{ss\varepsilon}$ at P_{uk}, P_{uk} is updated in the direction of the negative gradient. Let the parameter update rate be β. The update equations for P_{uk} and Q_{ki} are:

$$\begin{cases} P_{uk} \leftarrow P_{uk} + \beta (E_{ui} Q_{ki} - \lambda P_{uk}) \\ Q_{ki} \leftarrow Q_{ki} + \beta (E_{ui} P_{uk} - \lambda Q_{ki}) \end{cases} \tag{26}$$

The stochastic gradient descent (SGD) algorithm is used to update P_{uk} and Q_{ki}. In each iteration, the parameters are updated using a single training data point until the algorithm converges.

Compared to the batch gradient descent (BGD) algorithm, SGD does not require the use of all data to calculate the objective function. Instead, it randomly optimizes the SSE value on a single training data point in each iteration, significantly accelerating the update speed of P_{uk} and Q_{ki} and helping to avoid local optima.

III. Testing and verification

III. A. Target circuit parameter verification

III. A. 1) Verification of gain

Use the reinforcement learning algorithm obtained from the search to perform forward propagation prediction of the output waveform for 100 sine waves with constant power and gradually increasing frequency. Select 5,000 consecutive points from the input signal of each curve and perform FFT transformation:

$$X(f) = FFT(x(t)) \tag{27}$$

Identify the frequency with the maximum amplitude in the frequency spectrum of the signal:

$$f_{\max} = \arg\max |X(f)| \tag{28}$$

Convert the amplitude of this frequency into signal power:

$$V_{\max} = |X(f_{\max})| \tag{29}$$

$$P_{in} = 10 \log_{10} (\frac{2V_{\max}^2}{10^{-3} Z}) \tag{30}$$

In the equation, Z = 50 Ω, $V_{\max}$ is the signal amplitude, and P_{in} is the input signal power.

Performing the same calculation process on the output signal yields the value of the output signal power Pout. Then, the model-predicted gain is:

$$G_{model} = P_{out,model} - P_{in} \tag{31}$$

The actual gain of the target circuit is:

$$G = P_{out,measure} - P_{in} \tag{32}$$

For the gain-frequency curve, the gain value and frequency value at the point of maximum spectral amplitude were recorded. Based on 100 sine waves of different frequencies in the test set, 100 pairs of actual values and model-predicted values for the gain-frequency points were calculated, as shown in Figure 4.

In the figure, the gain-frequency curve Predicted_Gain is inferred by reinforcement learning, while the gain-frequency curve VNA_Gain is the actual test result from the vector network analyzer. The OSC_Gain curve in the figure is the actual test result gain-frequency curve obtained by calculating the actual test data from the oscilloscope using FFT and a formula.

By comparing the three curves, it can be seen that the error in the model's estimation of the target circuit parameters primarily stems from the waveform data measurement errors of the oscilloscope. The curve corresponding to the test waveform data differs from the curve directly measured by the vector network analyzer, which may be due to the insufficient precision of the oscilloscope used in the experimental testing. However, the parameter curve inferred by the model has a high degree of overlap with the curve corresponding to the waveform data measured by the oscilloscope, which serves as the data source for the model, indicating the accuracy of the modeling method. The parameter curves inferred by the model exhibit high matching accuracy with the waveform data curves from oscilloscope testing below 2.5 GHz. Above 2 GHz, the error between the predicted results and the oscilloscope data is also less than 0.724 dB.

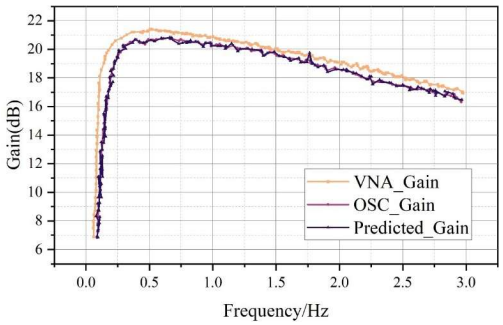
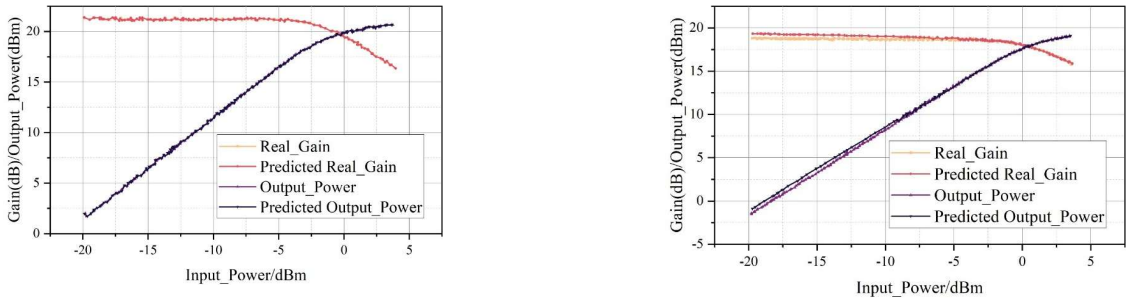


Figure 4: The gain-frequency curve is compared

Using an additional two sets of 100 sine waves with fixed frequencies and gradually increasing power from the test set, the gain-input power curves at input frequencies of 0.9 GHz and 1.9 GHz were calculated and compared with the results obtained from the actual input and output waveforms of the target circuit.

Due to experimental environment constraints, the actual gain-input power curves of the devices were not obtained through testing but were derived from actual input and output waveform data collected by an oscilloscope and subsequently calculated. Figure 5 shows the gain-input power curves predicted using reinforcement learning, with Figure (a) representing 0.9 GHz and Figure (b) representing 1.9 GHz. At an input frequency of 0.9 GHz, the model's prediction results are relatively accurate. However, at an input frequency of 1.9 GHz, the prediction at low amplitudes has an error of approximately 0.237 dB. This error primarily originates from the high-frequency components in the training set. Due to the limitations of the experimental testing equipment, the oscilloscope's signal acquisition has a truncation error at 7 GHz, resulting in insufficiently accurate training data in the high-frequency region and consequently causing errors in high-frequency predictions.



(a)0.9GHz (b)1.9GHz

Figure 5: Comparison of gain-input power curve results

III. A. 2) Comparison of parameter optimization effects

From the experimental results, it can be seen that reinforcement learning achieves better fitting performance for RF integrated circuits compared to RNN and LSTM models. The possible reason is that RF signals typically contain local periodic or short-

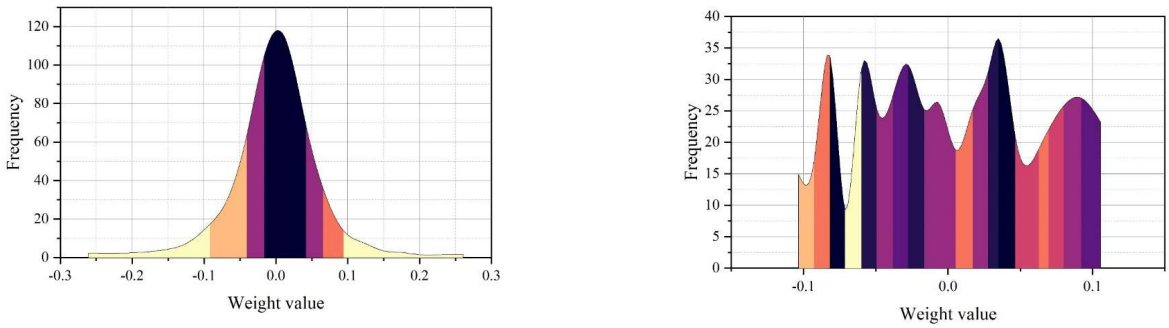
term patterns (such as pulses or modulation characteristics). Convolutional networks efficiently capture these local features by sliding the convolution kernel along the time dimension, whereas RNN and LSTM models, which rely on sequential processing, exhibit weaker sensitivity to local patterns. Reinforcement learning's residual connections mitigate the vanishing gradient problem, allowing deeper networks to be constructed to learn complex features. Even with gating mechanisms, deep recurrent networks still face gradient decay challenges during training, making it difficult to train the model to the required accuracy.

Figure 6 shows the impact of the two search methods on the receptive field parameter distribution of the residual model. Figure (a) shows the value function search results, and Figure (b) shows the policy function search results. Figure 7 shows the impact of the two search methods on the prediction accuracy of signals at different frequencies. The experimental results indicate:

(1) The policy function search method, which divides the size of each convolutional kernel layer, can search for more flexible structures and achieve model structures with the same level of performance as those obtained by the value function search method using fewer parameters. Except for a few frequency points where the accuracy is inferior to that of the value function method, the policy function method generally achieves higher modeling accuracy than the value function method, especially in the high-frequency range. The reason for the insufficient accuracy in the high-frequency range is that the cutoff frequency of the oscilloscope used in the experiment is not high enough.

(2) Models searched using the policy function method significantly reduce search time while maintaining the same level of performance as models searched using the value function method. For the search space defined in this paper, on the server equipped with an NVIDIA GeForce GTX 1080 Ti, the value function method required 6.315 hours, while the policy function method only required 1.348 hours, saving 78.654% of the time compared to the value function method.

(3) The policy function method and the value function method identified receptive fields of the same size, but the distribution patterns of the receptive field parameters for these 8 channels were different. The receptive field distribution of the model trained by the value function method tended toward a Gaussian distribution, exhibiting sparsity, with weight values centered around 0 and a frequency close to 120. In contrast, the receptive field distribution of the model identified by the policy function method tended toward a uniform distribution, with higher dependency on each parameter.



(a)Value function (b)Policy function

Figure 6: The effect of the residual model on the distribution of field parameters

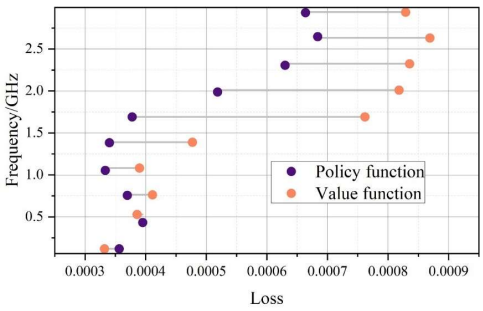


Figure 7: The effect of the prediction accuracy of different frequency signals

III. B. RF circuit design parameter optimization testing

III. B. 1) Comparison of optimization method performance tests

This section compares the software-based optimization method with the reinforcement learning-based optimization method proposed in this paper through experimental verification to validate the superiority of the proposed method in terms of optimization efficiency. Since different decision variables have varying weights in the objective function, this paper assigns

different weight coefficients to each decision variable when constructing the objective function. For the objective variable C_T, it represents the impact of transistor cost on the overall circuit cost. Based on engineering experience, it is believed that the high-frequency performance metric, i.e., the product of CJC and r_{bb}, has a significant impact on the cost of RF circuits. Therefore, its coefficient is set to c=0.4, while the weights of the remaining metrics are distributed evenly. For the objective variable C_O, it is primarily composed of passive components. Compared to the other two objective variables, its impact on the overall circuit cost is relatively small. Therefore, its coefficient is set to k = 0.1. For the objective variables C_T and C_Q, three weighting schemes were designed in this paper. Scheme 1 represents equal weights for both variables, while the other two schemes prioritize one of the objective variables. The corresponding coefficients are shown in Table 1. The population size of the PSO algorithm is set to 30, and the number of iterations is set to 10. Both simulation-based and reinforcement learning-based circuit component parameter optimization methods are used to solve the three schemes. The results are shown in Tables 2 and 3.

From the comparison of the two tables, it can be seen that although the optimization effects of the two methods are not entirely the same, the simulation-based method does indeed have a higher time cost. At the same time, the comparison results also confirm that the reinforcement learning-based method has a significant advantage in terms of optimization time cost. In terms of solution time, the reinforcement learning solution time is stable between 8.6 and 8.9 seconds, far exceeding the optimization solution time of the simulation method. At the same time, the optimization rate has also been slightly improved.

Table 1: objective function weight coefficients

Weighting	m	n	k	a	b	c	d
Solution 1	0.4	0.45	0.15	0.25	0.25	0.45	0.2
Solution 2	0.55	0.35	0.2	0.2	0.2	0.4	0.25
Solution 3	0.35	0.65	0.2	0.25	0.15	0.35	0.2

Table 2: The comparison of the optimized results is based on the simulation

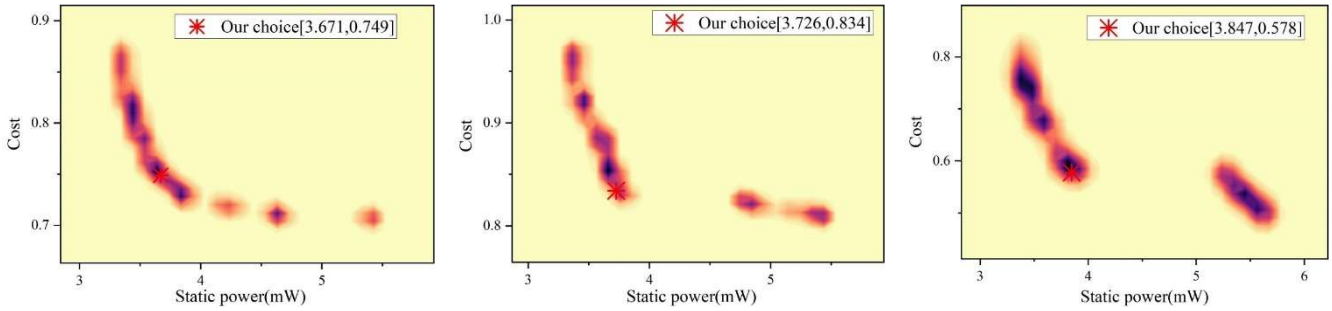
Scheme	Index	Preoptimize	After optimization	Optimization rate	Solution time
Solution 1	Static power(mW)	8.625	4.248	50.75%	2h3min
	Circuit cost	1.345	0.915	31.97%	
Solution 2	Static power(mW)	8.469	4.398	48.07%	2h10min
	Circuit cost	1.598	1.125	29.60%	
Solution 3	Static power(mW)	8.469	4.498	46.89%	2h5min
	Circuit cost	1.045	0.698	33.21%	

Table 3 The comparison of the optimized results of intensive learning was compared

Scheme	Index	Preoptimize	After optimization	Optimization rate	Solution time
Solution 1	Static power(mW)	8.625	4.125	52.17%	8.675s
	Circuit cost	1.345	0.869	35.39%	
Solution 2	Static power(mW)	8.469	4.162	50.86%	8.836s
	Circuit cost	1.598	1.328	16.90%	
Solution 3	Static power(mW)	8.469	4.093	51.67%	8.715s
	Circuit cost	1.045	0.523	49.95%	

III. B. 2) Final tuning results of the radio frequency circuit

In the previous experiment, due to the time-consuming nature of simulation-based optimization methods, only minor adjustments were made to the population size and iteration count parameters of the PSO algorithm. To ensure that the circuit is fully optimized, this experiment employs a deep learning-based circuit optimization method. The initial population size for the PSO algorithm is set to 500, and the number of iterations is set to 400, with the remaining parameters using the default settings from the pymoo framework. The optimal solution sets for the three solutions obtained are shown in Figure 8, with Figures (a) to (c) representing Solutions 1, 2, and 3, respectively. In these three figures, the final selected optimal solutions are marked with "※." The optimal solution coordinates for Schemes 1 to 3 are [3.671, 0.749], [3.726, 0.834], and [3.847, 0.578], respectively.



(a)Solution 1 (b)Solution 2 (c)Solution 3

Figure 8: solution

The values of the decision variables at the optimal solutions for each of the three schemes are shown in Table 4. After rounding the decision variables obtained in the table, they are substituted into the RF transmission circuit. Using the Pspice A/D simulation software to perform time-domain simulation on the circuit will yield the optimized current signal. Analyzing this current signal will yield the corresponding constraint variables under the given decision variable inputs. The constraint variable values for the three schemes are shown in Table 5.

A brief explanation of the transistor's DC gain is provided here. The values given in Table 4 refer to the transistor's values under the current DC bias conditions. The DC gain of a transistor exhibits significant variability under different DC bias conditions. Taking the common transistor 2N3903 as an example, although its data sheet specifies a typical DC gain value of 150–300, when the collector current is 1 mA and the voltage drop is 1 V, its DC gain is 35.

Table 6 shows a comparison of the performance metrics before and after optimization. Within the specified constraints, the circuit was optimized, resulting in a reduction of over 54% in static power consumption and over 40% in circuit cost compared to the pre-optimization state, demonstrating the feasibility of the proposed method.

Table 4: Decision variables at optimal solution

Decision variable	Parameter	Solution 1	Solution 2	Solution 3
x_1	C_{ajt1}	443.849p	394.948p	486.966p
x_2	C_{ajt2}	50.164p	50.648p	50.469p
x_3	β	24	35.269	23
x_4	CJC	36.487p	38.948p	35.536p
x_5	CJE	38.966p	35.648p	38.934p
x_6	r_{bb}	115.648p	146.696	101.469
x_7	f_r	200.469M	200.291M	234M
x_8	R_{bias}	79.069K	76.269K	75.985K
x_9	R_Q	11.596K	10.396K	9.866K
x_{10}	C_1	53.497p	57.395p	54.469p

Table 5: Values of constraint variables at optimal solution

Decision variable	Parameter	Solution 1	Solution 2	Solution 3
y_1	f_{CDM}	33.748KHz	35.425KHz	34.345KHz
y_2	f_{DDM}	19.569KHz	19.648KHz	19.625KHz
y_3	V_D	13.348Kbit/s	13.749Kbit/s	13.525Kbit/s
y_4	B_W	-15.654dB	-15.248dB	-16.166dB
y_5	A_P	-1.348dB	-1.249dB	-1.348dB
y_6	I_c	60.526mA	64.396mA	61.248mA

Table 6: Comparison table of indicators before and after optimization

Scheme	Index	Preoptimize	After optimization	Optimization rate	Solution time
Solution 1	Static power (mW)	8.625	3.648	57.70%	2m21s
	Circuit cost	1.345	0.748	44.39%	
Solution 2	Static power (mW)	8.469	3.618	57.28%	2m35s
	Circuit cost	1.598	0.869	45.62%	
Solution 3	Static power (mW)	8.469	3.869	54.32%	2m31s
	Circuit cost	1.045	0.618	40.86%	

IV. Conclusion

This paper introduces the principles of global optimization algorithms and reinforcement learning algorithms, and derives two types of reinforcement learning algorithms: model-free and model-based. Using this algorithm, a frequency AI model was established to model the equivalent circuit of RF devices in RF circuits and extract parameters related to RF devices. The SVD method was used to decompose the parameter matrix, avoiding overfitting. Test experiments were designed to verify the optimization results of the target circuit parameters and RF circuit parameters. The gain-frequency curve was verified, and the waveform data from the waveform tester showed a high degree of overlap with the curve, with an error of less than 0.724 dB between the two, indicating the accuracy of the modeling method. Comparing the parameter optimization results, the search time for the value function and policy function were 6.315 hours and 1.348 hours, respectively. In terms of time cost, the policy function method saved 78.654% of the time compared to the value function method. In terms of solution time, the reinforcement learning solution time is stable between 8.6 and 8.9 seconds. Additionally, the static power consumption of the optimized circuit is reduced by over 54% compared to the pre-optimization state, and the circuit cost is reduced by over 40%, indicating that the method proposed in this paper is feasible.

References

[1] Zeng, Y., Clerckx, B., & Zhang, R. (2017). Communications and signals design for wireless power transmission. IEEE Transactions on Communications, 65(5), 2264-2290.

[2] Zhong, K., Zhou, X., Huo, J., Yu, C., Lu, C., & Lau, A. P. T. (2018). Digital signal processing for short-reach optical communications: A review of current technologies and future trends. Journal of Lightwave Technology, 36(2), 377-400.

[3] Basar, E. (2019, June). Transmission through large intelligent surfaces: A new frontier in wireless communications. In 2019 European conference on networks and communications (EuCNC) (pp. 112-117). IEEE.

[4] Kaddoum, G. (2016). Wireless chaos-based communication systems: A comprehensive survey. IEEE access, 4, 2621-2648.

[5] Lanza, M., Sebastian, A., Lu, W. D., Le Gallo, M., Chang, M. F., Akinwande, D., ... & Roldan, J. B. (2022). Memristive technologies for data storage, computation, encryption, and radio-frequency communication. Science, 376(6597), eabj9979.

[6] Paul, B., Chiriyath, A. R., & Bliss, D. W. (2016). Survey of RF communications and sensing convergence research. IEEE Access, 5, 252-270.

[7] Elayan, H., Amin, O., Shihada, B., Shubair, R. M., & Alouini, M. S. (2019). Terahertz band: The last piece of RF spectrum puzzle for communication systems. IEEE Open Journal of the Communications Society, 1, 1-32.

[8] Wang, G., Gao, F., Fan, R., & Tellambura, C. (2016). Ambient backscatter communication systems: Detection and performance analysis. IEEE Transactions on Communications, 64(11), 4836-4846.

[9] Baballe, M. A., & Nababa, F. A. (2021). A comparative study on radio frequency identification system and its various applications. International Journal of Advances in Applied Sciences (IJAAS), 10(4), 392-398.

[10] Olaleye, D. S., Oloye, A. C., Akinloye, A. O., & Akinwande, O. T. (2024). Advancing green communications: the role of radio frequency engineering in sustainable infrastructure design. International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), 13(5), 113.

[11] Lubna, L., Hameed, H., Ansari, S., Zahid, A., Sharif, A., Abbas, H. T., ... & Abbasi, Q. H. (2022). Radio frequency sensing and its innovative applications in diverse sectors: A comprehensive study. Frontiers in Communications and Networks, 3, 1010228.

[12] Miranda, R. F., Barriquello, C. H., Reguera, V. A., Denardin, G. W., Thomas, D. H., Loose, F., & Amaral, L. S. (2023). A review of cognitive hybrid radio frequency/visible light communication systems for wireless sensor networks. Sensors, 23(18), 7815.

[13] Bello, O., Zeadally, S., & Badra, M. (2017). Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). Ad Hoc Networks, 57, 52-62.

[14] Kasemsap, K. (2017). Radio frequency identification and mobile ad-hoc network: Theories and applications. In Handbook of research on recent developments in intelligent communication application (pp. 63-95). IGI Global.

[15] Kalhori, A. H., & Kim, W. S. (2022). Printed wireless sensing devices using radio frequency communication. ACS Applied Electronic Materials, 5(1), 1-10.

[16] Nintanavongsa, P., Muncuk, U., Lewis, D. R., & Chowdhury, K. R. (2012). Design optimization and implementation for RF energy harvesting circuits. IEEE Journal on emerging and selected topics in circuits and systems, 2(1), 24-33.

[17] Zi, R., Ge, X., Thompson, J., Wang, C. X., Wang, H., & Han, T. (2016). Energy efficiency optimization of 5G radio frequency chain systems. IEEE Journal on Selected Areas in Communications, 34(4), 758-771.

[18] Muncuk, U., Alemdar, K., Sarode, J. D., & Chowdhury, K. R. (2018). Multiband ambient RF energy harvesting circuit design for enabling batteryless sensors and IoT. IEEE Internet of Things Journal, 5(4), 2700-2714.

[19] Chen, Y. S., & Chiu, C. W. (2017). Maximum achievable power conversion efficiency obtained through an optimized rectenna structure for RF energy harvesting. IEEE Transactions on Antennas and Propagation, 65(5), 2305-2317.

[20] Afacan, E., Lourenço, N., Martins, R., & Dündar, G. (2021). Machine learning techniques in analog/RF integrated circuit design, synthesis, layout, and test. Integration, 77, 113-130.
[21] Zhuang, L., Roeloffzen, C. G., Hoekman, M., Boller, K. J., & Lowery, A. J. (2015). Programmable photonic signal processor chip for radiofrequency applications. Optica, 2(10), 854-859.
[22] Shahabuddin, A. A., Shalu, P. D., & Akter, N. (2018). Optimized process design of rf energy harvesting circuit for low power devices. International Journal of Applied Engineering Research, 13(2), 849-854.
[23] Nyaradzo Alice Tsedura,Ernest Bhero & Colin Chibaya. (2025). Towards the design of a particle swarm optimization ontology for object classification. Array,27,100449-100449.
[24] Seung Chan Choi,Yohan Lee & Sung Won Cho. (2025). Reinforcement learning-integrated evolutionary algorithm for enhanced unmanned aerial vehicle coverage path planning. Swarm and Evolutionary Computation,97,102051-102051.
[25] Roshan Golmohammadi,Saeed Parsa & Morteza Zakeri Nasrabadi. (2024). Dynamic domain testing with multi-agent Markov chain Monte Carlo method. Soft Computing,28(13-14),8293-8317.
[26] Wei Huang,Xiang Li & Baoxu Shi. (2025). Fast resonant frequency tracking strategy of linear oscillatory motor based on gradient descent method. Journal of Physics: Conference Series,3011(1),012014-012014.