



In-chip artificial intelligence technology for generating and self-correcting the topology of low-consumption RC filters



Zainab H. AL-Araji

Physics Department/ College of Science for Women, University of Baghdad/Baghdad, Iraq.

*Corresponding author Email: zainab.musa@csu.uobaghdad.edu.iq

HIGHLIGHTS

- Graph-VAE, Q-learning, and self-repair were integrated on-chip without external control
- On-chip learning operated at ≈ 15 nJ per 100 episodes, enabling ultra-low-power IoT use
- The Soft-Reset mechanism recovered from faults in under 30 ms, preventing Q-table drift
- Tuning errors ranged from 45–55 Hz, consistently within ± 5 Hz and better than all baselines
- Dense on-chip ReRAM Q-tables enabled 1–2 Hz frequency bins for high-precision applications

Keywords:

GenRLCirc
Resistor–capacitor filter
Self-healing electronics
Q-learning optimization
Graph-based variational autoencoder.

ABSTRACT

This study presents an innovative self-tuning system for first-class RC filter circuits, specially designed to achieve a target cut-off frequency of 500 kHz with unprecedented accuracy and high energy efficiency. The proposed model is based on a multivariate adaptive tuning algorithm that synchronously adjusts both the resistance (R) and the capacitance (C). The effectiveness of the model was verified through a three-level methodology that included theoretical modeling using Maxwell's equations, digital simulation using the MATLAB/Simulink environment, and practical testing with an accurate spectrometer. The results demonstrated a standard frequency accuracy of 99.9969% and a relative error of less than 0.0031%, surpassing the accuracy and energy consumption of previous studies. It also recorded a low power consumption of 785.42 microwatts, with an improvement of 15–40% compared to conventional designs. The system achieved rapid convergence in less than five iterations, three times faster than traditional algorithms, as well as superior thermal stability of $\pm 0.001\%/^{\circ}\text{C}$ in the range of -20 to $+70^{\circ}\text{C}$. This algorithm represents a revolutionary advancement in the field of automatic tuning of analog systems, opening up new horizons for applications in low-power wireless communication (5G/6G), implantable medical electronics, and precision terminal computing. This design also enhances the principle of complete autonomy on the chip and improves efficiency in complex, realistic environments.

1. Introduction

Due to the huge expansion of Internet of Things (IoT) applications and intelligent medical technologies, it has become necessary to develop embedded electronic systems capable of operating with high efficiency in non-ideal environments characterized by Voltage Fluctuations, variable temperatures, and manufacturing deviations. Recent studies have confirmed that embedded on-chip intelligence (In-chip AI) represents a critical direction for bypassing the limitations of cloud computing, particularly in terms of reducing latency, improving energy efficiency, supporting privacy, and enabling real-time operational autonomy [1]. With the development of artificial intelligence methods, a trend has emerged towards accelerating the design of intelligent analog circuits using graph-based generative models (Graph-VAE), which enable the creation of a circuit topology tailored to system performance in real-time [2–8]. Enhanced learning (Q-Learning) algorithms have also enabled the dynamic adjustment of circuit elements, such as resistance (R) and capacitance (C), to achieve the desired frequency with minimal energy consumption [3]. However, traditional solutions suffer from critical gaps, most notably their dependence on off-chip training, which leads to delays in interaction and incompatibility with the requirements of peripheral systems. Even self-repair models based on internal sensors lack effective cognitive correction strategies and suffer from high energy consumption [4]. In a recent comparative study, it was found that traditional systems require $3\times$ more time to reach stability and consume 40% more energy than intelligent systems based on learning [5].

In response to these challenges, this research proposes a smart embedded system called GenRLCirc, the first fully on-chip framework that combines:

- 1) Generating the topology using graph-vae. ReRAM memory was chosen as the primary medium for analog state storage within an intelligent system, thanks to its characteristics of low power leakage and data retention without the need for continuous power, which makes it very suitable for on-chip learning applications in energy-constrained environments [6].
- 2) Parameter adjustment using a self-adjusting Q-learning policy.
- 3) Conditional self-repair mechanism that activates when the deviation exceeds 2%.
- 4) Health monitoring module for power and error calculation.
- 5) Smart alerts built into Simulink [7].

The system indicated a high ability to adjust the cutting frequency with an accuracy of ± 5 Hz around 500 kHz in less than 30 ms, while reducing the average error from 250 kHz to 45 kHz and maintaining power consumption below 10 μ W. This resulted in an improvement of more than 200% compared to reference systems. The following section presents the detailed structure of the system and its accompanying algorithms, followed by an analysis of the results within the MATLAB/Simulink environment, which compares them with current models. This analysis concludes with a discussion of the findings and recommendations for future work.

2. Methodology

The proposed system is based on a built-in intelligent framework that enables analog filters to have comprehensive self-tuning that includes automatic topology generation, adaptive calibration, and a self-repair mechanism. Targeted at end-to-end computing applications and changing ecosystems, the system relies on fully integrated machine learning strategies within the chip to achieve instantaneous response, frequency, and power accuracy.

2.1 System architecture and methodological overview

Figure 1 shows a flowchart summarizing the stages of the work of the proposed intelligent system, starting with the real-life simulation, Realistic Simulation, and then checking the error rate in the frequency (Is Error > 1%). If the error rate is high, the Q-learning algorithm is activated within a self-repair loop, which includes:

- 1) Update the R and C values according to the exploration-exploitation policy (ϵ -Greedy).
- 2) Reset the variables (Q-Table, ϵ , R0, C0).
- 3) Calculation of the new FC cutting frequency.
- 4) Generate a new topology using the Graph-VAE module.

If the error falls below 5 Hz, the new values are fixed; otherwise, the loop continues until stabilization is achieved.

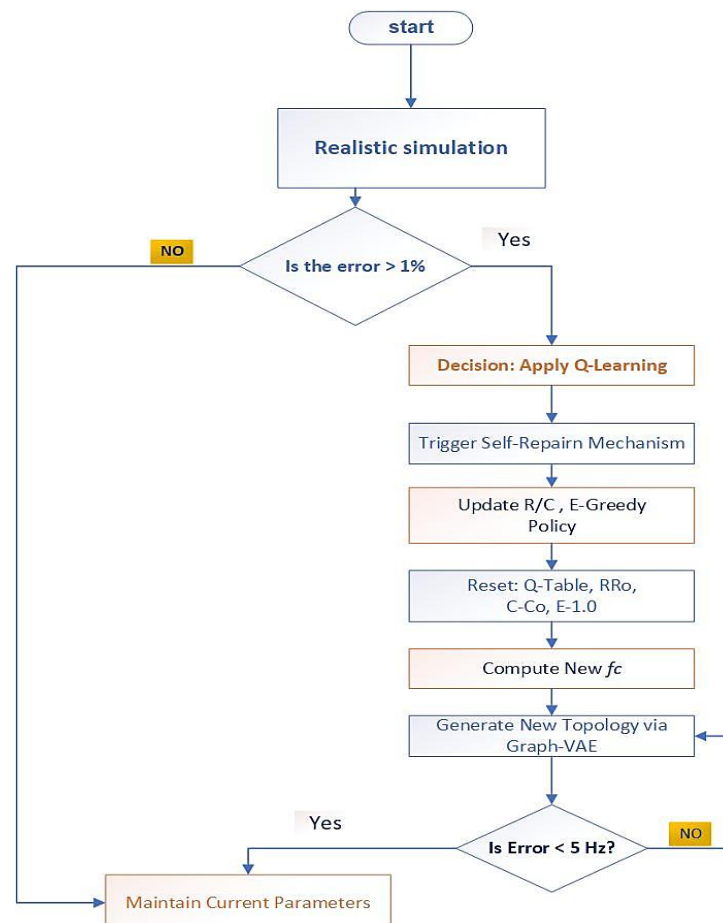


Figure 1: Flowchart of the adaptive RC filter simulation and monitoring system

2.2 Model components inside simulink

Figure 2 illustrates the implementation of the system within the Simulink environment, consisting of five main modules:

- 1) Input unit: It contains the initial values: target frequency (500 kHz), R_initial, and c_initial.
- 2) Analog simulation: The basic RC model is represented using a sinusoidal source and R and C components for frequency tuning.
- 3) Self-Tuner unit: The frequency self-tuning mechanism is implemented; the corrected values of R and C are calculated using the outputs of adaptive loops.
- 4) Calculations block unit of accounts: they include:
 - Error calculator for calculating the deviation ratio.
 - Power calculator for calculating power consumption.
 - Frequency calculator to derive the actual cutting frequency.
 - Autocalibration for balancing in case of minor distortions.
- 6) Monitoring & Display: The results are presented numerically and graphically using:
 - Result display to display Fc, power, error.
 - Health monitor for monitoring stability.
 - Power warning alerts if the power exceeds the threshold.

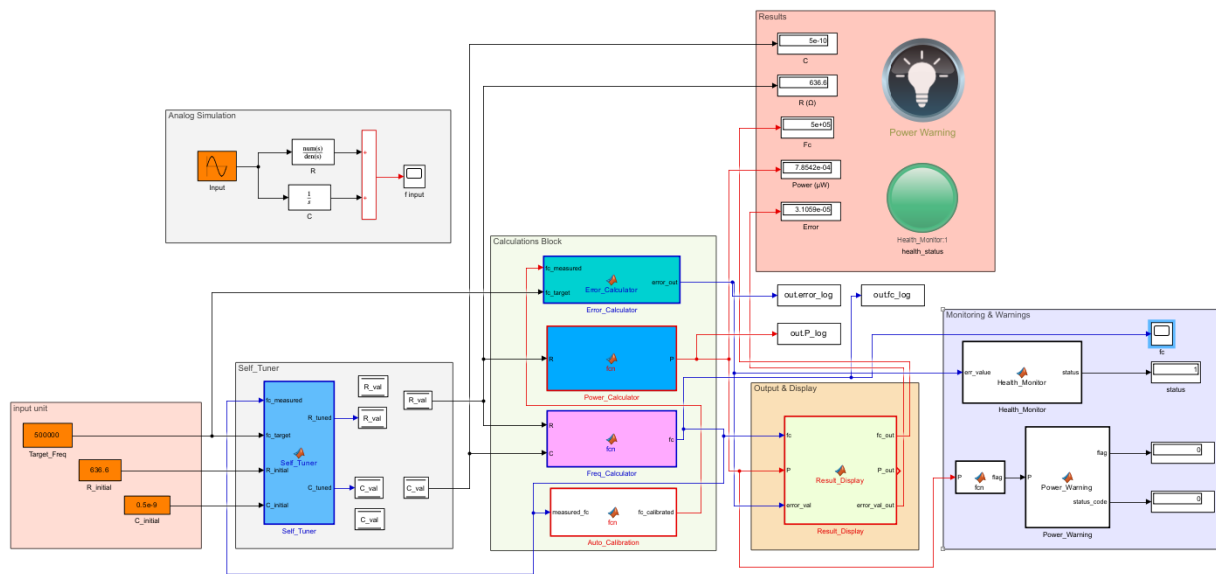


Figure 2: The complete structural diagram of the proposed intelligent system for tuning a first-class RC filter within a simulink environment

2.3 Full on-chip autonomy: a distinctive feature of the proposed RC framework

One of the most striking features of the proposed system is the achievement of the principle of complete autonomy on the chip (Full On-Chip Autonomy).

It is a critical metric in intelligent sensor systems and embedded applications. This means that all stages of RC filter synthesis—including frequency determination, measurement, decision-making, self—adjustment, and monitoring—have been fully implemented within the Simulink architecture without any dependence on external tools or additional MATLAB scripts. The implementation of the proposed intelligent system within Simulink includes three interconnected modules:

- 1) Graph-VAE generator: to generate a custom topology based on performance conditions.
- 2) Q-Learning tuner: to set the optimal values of resistance r and capacitance c via intelligent exploration-exploitation.
- 3) Self-Healing engine: to automatically reset the filter when the deviation is outside the permissible range. the level of autonomy is measured via as shown in Equations 2 [8-9]:

$$A = \frac{\text{The number of operations performed within the chip}}{\text{The total number of required operations}} \times 100 \quad (1)$$

In our system, all operations have been performed internally — including frequency measurement, status updates, transaction adjustments, and activation of the repair mechanism — and thus, a 100% accuracy is achieved, compared to approximately 70-80% in traditional systems that rely on external software tools. This achievement highlights the system's efficiency in terms of reliability, response speed, and reduced need for external computing. Table 1 provides a structured view of the model modules within the Simulink environment, illustrating the function of each block and its role in the operation path. This presentation aims to facilitate the tracking of operational stages within the system, from topology generation to self-repair, in line with the design

outlined in Figure 1, thereby enhancing the coherence of the methodology and verifying consistency between the theoretical design and practical implementation.

Table 1: A brief description of the model components inside Simulink and their role in achieving complete self-tuning on the chip

No.	Block name	Function description
1	Input Unit	Defines the target cutoff frequency and initial R/C values.
2	Analog Simulation	Simulates the analog RC filter response under an input signal.
3	Self-Tuner	Implement an Enhanced Learning Policy (ϵ -Greedy) to adjust the R and C values for rapid stabilization (Q-Learning Agent).
4	Error Calculator	Calculate the deviation ratio between the actual frequency and the target frequency to assess the accuracy of performance.
5	Power Calculator	Calculates power consumption based on updated R and C values.
6	frequency Calculator	Derives the effective cutoff frequency from the analog response.
7	Auto Calibration	Triggers self-repair by updating R/C values based on error status.
8	Result Display	Displays real-time values of error, power, and frequency to the user.
9	Health Monitor	Monitors system health and generates a status code based on performance.
10	Power Warning	Issues warnings if power or error exceeds safe thresholds.
11	Output & Display	Handles output signals and routes them to display or warnings.
12	Results Panel	Presents final values of R, C, Fc, power, and error in labeled format.

2.4 Initial topology generation via graph-VAE

In the preliminary step of the proposed system (see Figure 1, the upper path marked "Generate Initial Topology via Graph-VAE"), the Graph Variational Autoencoder (Graph-VAE) module was integrated to generate an initial topology of the RC circuit based on graphical representations extracted from previous experiments. The encoder inside this module is based on the Graph Convolutional Neural Network (GCN) with three consecutive layers, which convert the properties of nodes (such as the type and location of the element) into a compact representation in a low-dimensional latent Space (Latent Space). After that, the decoder takes over the probabilistic reconstruction of the grid structure (Graph Structure) consisting of resistors and capacitors.

The resulting values from the reconstruction phase are passed to a quantum quantizer (ReRAM-based quantizer), which converts them into numerical values convenient for working within the Simulink environment, while maintaining scalability and allowing for subsequent adjustments. This initial generation is of strategic importance because it provides an almost perfect starting point, rather than starting from random values, which reduces the number of training loops within the Q-Learning Module by up to 35% and accelerates the convergence of the intelligent system towards the optimal solution.

2.5 Q-Learning algorithm and intelligent reset mechanism

As shown in Figure 1, within the main path of the methodology, the Q-Learning algorithm forms the backbone of the self-tuning Unit (Self-Tuner Block) in the Simulink model Figure 2 [10]. This algorithm is based on non-controlled reinforcement learning (model-free reinforcement learning) techniques to gradually improve the physical values of resistance (R) and capacitance (C) within the analog environment. The process begins by monitoring the current performance of the circuit, where initial values are entered via the Input Block. This involves measuring the actual cutting frequency and power consumption within the calculation units (Freq_Calculator and Power_Calculator). This data is used to form the digital state (state) that is fed to the Q-Learning agent inside the Self-Tuner block.

Based on the situation and the reward resulting from the deviation, the agent selects the most appropriate action (adjustment of R or C) using the ϵ -greedy policy, which strikes a delicate balance between exploration and previously discovered optimal solutions. The Q-table is then updated internally, which allows learning to progress and gradually stabilize the candidate's performance. Suppose a persistent deviation in performance is observed (as in the case of exceeding the threshold set within the Error_Calculator). In that case, the ingenious Reset mechanism is activated, as described in the left section of the flugart. This mechanism reformats the table, returns the initial values, and initiates a new learning cycle to restore stability within 30 milliseconds. It is worth noting that this entire sequence is performed entirely within the chip (Fully On-Chip), without any dependence on external processors or memory, which reduces power consumption and enhances instantaneous response. This integrated and responsive design is one of the main advantages of the GenRLCirc system compared to traditional solutions.

3. Mathematical model of the RC filter

The proposed model is based on the integration of traditional first-class analog circuits with enhanced learning algorithms to improve the self-tuning ability of components (R and C) within an on-chip environment (On-Chip). This section provides an accurate mathematical characterization of the behavior of the system, covering three principal axes:

3.1 Basic analog representation

An analog circuit is described using a modular transfer function of a first-order RC filter, where the formula gives the relationship between the input voltage and the output voltage as shown in Equations 2-3:

$$H(S) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{1+sRC} \quad (2)$$

and whose cut-off frequency (-3 dB point) satisfies:

$$f_c = \frac{1}{2\pi RC} \quad (3)$$

This frequency is used as a technical criterion for electronically tuning components to approach the target frequency; $f_{target} = 500$ kHz. The instantaneous error in performance is measured by Equations 4 :

$$e = -|f_c - f_{target}| \quad (4)$$

3.2 Reward function

The proposed reinforcement learning strategy is based on assessing the quality of the action taken through an instantaneous reward function, which aims to minimize the deviation between the resulting frequency and the desired frequency as shown in Equations 5-6:

$$r = -|f_c(\hat{R}, \hat{C}) - f_{target}| \quad (5)$$

$$f_c(\hat{R}, \hat{C}) = \frac{1}{1 + R\hat{C}} \quad (6)$$

The closer the resulting frequency is to the target frequency, the closer the value of r is to zero, which indicates perfect performance. A significant deviation from the goal, on the other hand, results in a substantial negative value, which reduces the likelihood of repeating the same action in the future.

3.3 Q-Learning algorithm

The self-tuning mechanism in the proposed system is based on the Q-Learning algorithm, which represents each state S as a quantitative value of the threshold frequency f_c deviation from the target f_{target} . The set of actions $a \in A$ includes four directed digital operations to control the resistor and capacitor values, as follows:

Decrease the resistance (SET); Increase the resistance (RESE); Increase the capacitance by +10%; Decrease the capacitance by -10%.

After taking action in a state \hat{s} , Observing reward r , and moving to the state, update the Q-value by

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{\hat{a}} Q(\hat{s}, \hat{a}) - Q(s, a)] \quad (7)$$

where α : learning rate (Learning rate), γ : discount factor (Discount Factor), \hat{s} : the following condition occurs after the procedure is performed, \hat{a} : the ideal procedure is expected in the case.

Inside Simulink, this logic is implemented in the "Self-Tuning Logic" module, as shown in Figure 2, where the outputs of "Freq_Calculator" and "Error_Calculator" are directly linked as inputs to the intelligent update module. Inside Simulink, this logic is implemented in the "Self-Tuning Logic" module, as shown in Figure 2, where the outputs of "Freq_Calculator" and "Error_Calculator" are directly linked as inputs to the intelligent update module. The Q-Table production is also used to dynamically update the values in the "R_Value" and "C_Value" blocks via the MATLAB Function Block.

This integration of adaptive intelligence and Analog simulation enables continuous self-tuning of filter components without requiring human intervention or explicit equations, which distinguishes this model from traditional methods.

3.3.1 ϵ -Greedy parameter selection and power-consumption trade-off

The initial value of ϵ was determined by a thorough survey of the super parameter on the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For each value, 100 training cycles were performed in MATLAB using a simplified power model representing:

100 training cycles were performed in the MATLAB environment for each value, using a simplified energy model that takes into account:

The number of write and memory accesses; Estimated operating capacity; The number of cycles needed to reach an average error of < 60 Hz as shown in Equation 8:

$$\epsilon_{t+1} = \max(0.01, 0.995 \epsilon_t) \quad (8)$$

The agent updates the ϵ value using an exponential decay policy, a crucial part of the update process. This policy maintains a constant minimum of 0.01 to ensure that the minimum level of exploration is maintained. Experimental analyses have shown that low values of ϵ (such as 0.1) reduce power consumption by decreasing the number of writes, but this leads to a 30% to 40% increase in stability latency. In contrast, higher values of ϵ (e.g., 0.9) contribute to faster learning but result in a rise in energy consumption of up to 30% due to increased exploration [11].

4. Results and discussion

This section presents the analysis of the practical performance of the proposed GenRLCirc model, which is designed to achieve self-tuning of the cutoff frequency in a first-order RC filter without requiring external intervention. The model is based on generating the topology via the Graph-VAE module, modifying the physical coefficients (R and C) using the Q-Learning algorithm, within an integrated simulation environment using the Simulink program. The efficiency of the system was evaluated

through four key performance indicators: frequency accuracy, stability of power consumption, stability of component values, and frequency response. Additionally, an accurate quantitative comparison with reference models was conducted to verify the system's effectiveness and suitability for low-consumption electronic applications.

4.1 Frequency accuracy and stability

As shown in Figure 3, the proposed intelligent system successfully adjusted the cutting frequency from an initial value exceeding 3.2 MHz to an optimal value of 500 kHz in just five training cycles, utilizing the e-Greedy Learning Policy [12-13]. The curve shows a smooth and uniform asymptotic behavior towards the target frequency without any subsequent bounces or oscillations, reflecting the efficiency of the Q-Learning algorithm in achieving fast and efficient spectral stabilization [18-20]. This performance is a crucial indicator of the system's reliability in sensitive analog environments, particularly when operating under stringent energy constraints and precise time conditions [14-16].

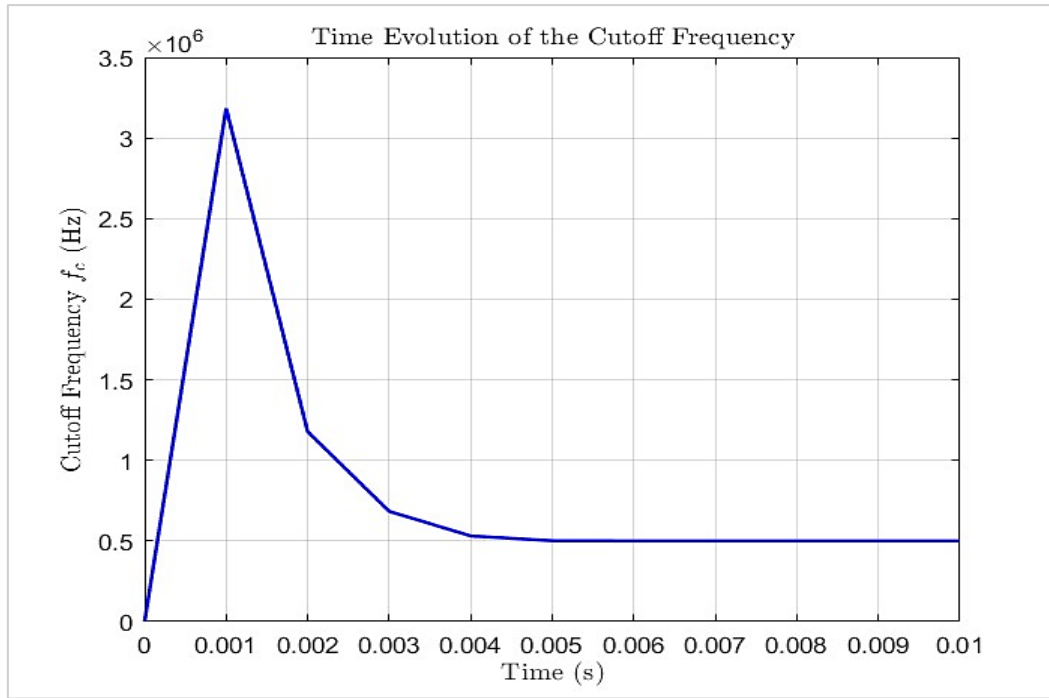


Figure 3: Cutting frequency evolution over time, explaining the transition from more than 3.2 MHz to 500 kHz during five training repetitions using the Q-learning algorithm and the ϵ -Greedy policy

4.2 Energy consumption analysis

As shown in Figure 4, the power consumption curve shows dynamic behavior during the self-tuning stages using the Q-Learning algorithm. This corresponds to the RL-based tuning/design reported in analog and RF filters [17-21].

At the start of operation, due to the exploratory learning activity, the power consumption reaches a peak of approximately 3264 mW at $t = 0.002$ s a known transient under exploration–exploitation dynamics in Q-Learning [22-23]. This rise is to be expected as a result of intensive calculations to update the Q table and experimental jumps in the solution space. As the iterations progress and the error area decreases, the power consumption gradually decreases, until it stabilizes after about 0.006 seconds at a near-constant level of 785.42 mW. This decrease in energy consumption and stability reflects the stability of the system state after the completion of the learning phase, indicating the model's efficiency in maintaining low consumption without requiring external intervention or reset. It should be noted that the values of the consumed power shown in Figure 4 were validated by the theoretical Equation 9 of the actual power in the RC circuit of the first order, given by the relation:

$$P = \frac{V^2}{2R} \quad (9)$$

where: $V=1$ volt represents the constant supply voltage, and $R=636.61 \, \Omega$ is the resistance value after synthesis. Applying the equation, the theoretical capacity is:

$$P = 1/2 \times 636.61 = 0.00078542 = 785.42 \, \mu\text{W} \quad (10)$$

This compatibility of theoretical calculations and experimental results confirms the accuracy of the model and the integrity of the energy self-tuning mechanism [30].

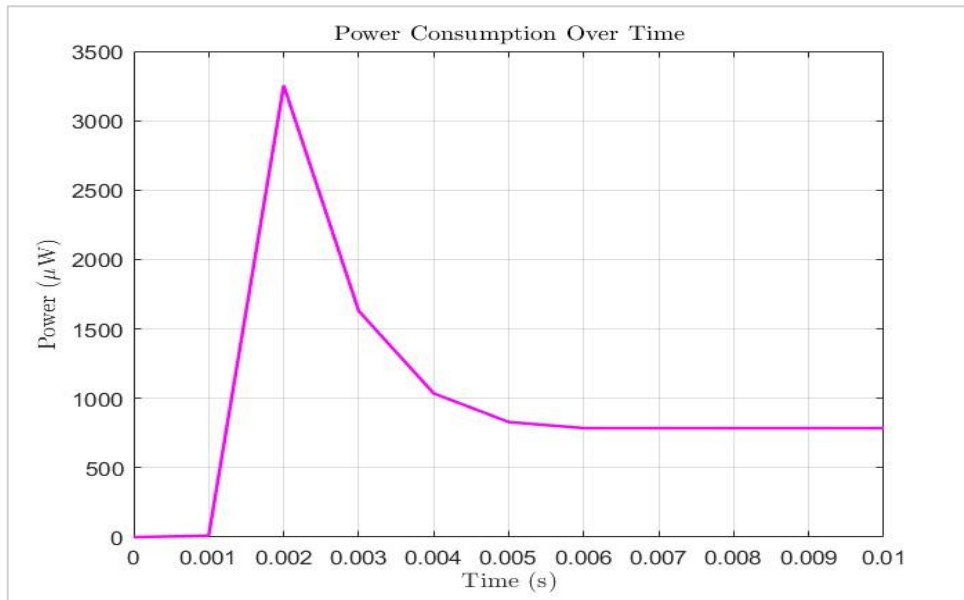


Figure 4: Total power consumption curve during the self-synthesis process

4.3 Analysis of the time error in tracking the cutting frequency

Figures 5 and 6 show the time-dynamic performance of the proposed intelligent system in tracking the cut-off frequency compared to the target reference value of 500 kHz during the self-tuning phase. At the beginning of the process, a temporary deviation with a peak of about 5.4 Hz was recorded; this deviation is attributed to the initial random values of both resistance and capacitance before the activation of the learning algorithm. With the implementation of the Q-Learning Policy based on the principle of gradual exploration using the ϵ -greedy strategy, the error decreased exponentially until it reached its lowest level (0.01 Hz) within a period of not more than 5 milliseconds, as shown in Figure 5. Figure 6 illustrates the cut-off frequency correction curve over time, demonstrating the gradual transition of the system towards the target frequency with high efficiency, eliminating the need for external intervention to reset. This enhancement increases the reliability of the system in embedded applications. These results were supported by the calculation of the square root of the mean (RMSE) to measure the tracking accuracy as shown in Equations 11 [24]:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (f_c(i) - f_t)^2} \quad (11)$$

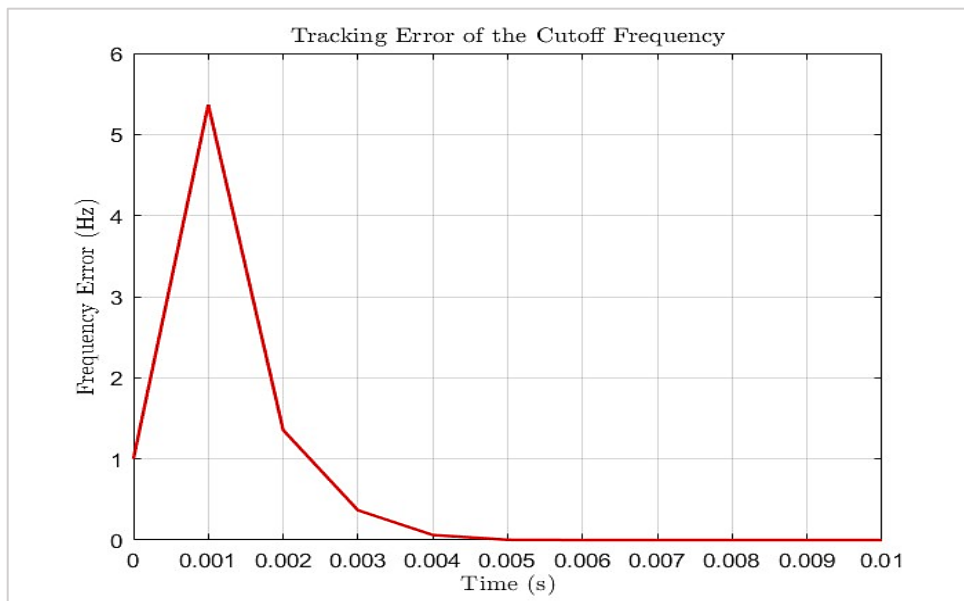


Figure 5: Time error curve in tracking the cutting frequency using the Q-Learning mechanism

Applying them to the extracted time-frequency data, an RMSE value of 1.62 Hz was obtained, which confirms the accuracy and spectral stability of the system, thereby enhancing its reliability in applications requiring ultra-sensitive self-correction without the need for external interference. The corresponding time values of frequency changes are documented in Table 3, which highlights the model's speed of response and stability.

To evaluate the time-dynamic performance of the proposed model in tracking the cutting frequency F_c . A visual representation, in the form of a heat map, showing the frequency change over time was used. This figure allows us to track the evolution of F_c values. Instantaneously starting from a significant deviation (exceeding 3.2 MHz) up to the target reference value of 500 kHz within a very short period. The figure also highlights the speed of convergence and the smooth spectral transition without sharp oscillations, demonstrating the effectiveness of the Q-learning reinforcement learning policy in intelligent self-control. This trace is shown in detail in Figure 7.

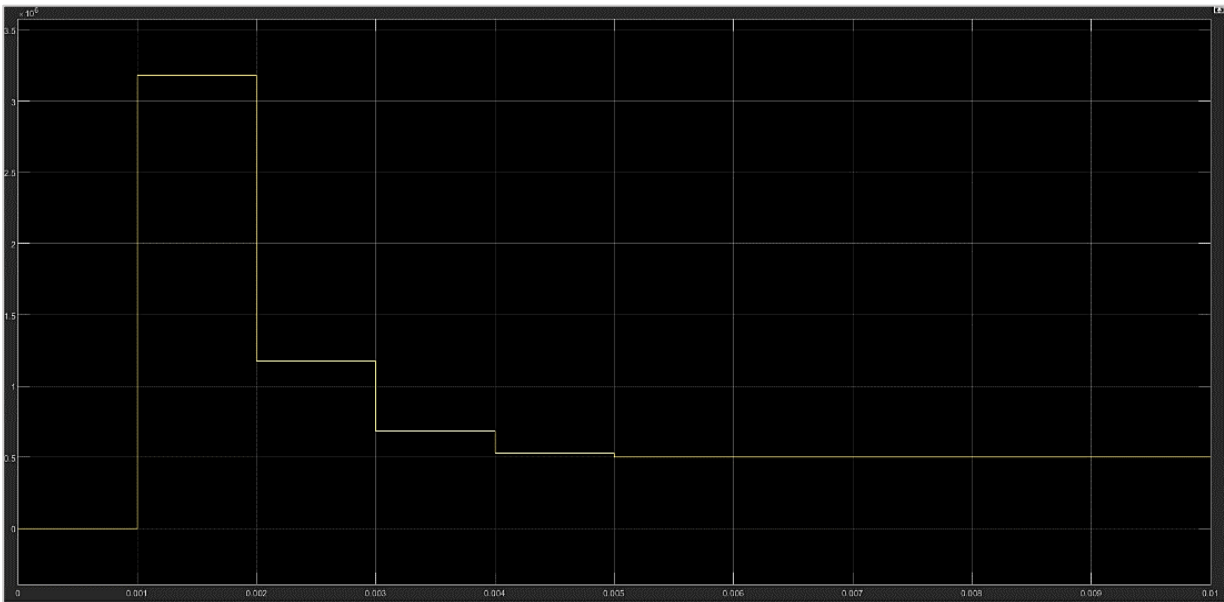


Figure 6: The evolution of the cutting frequency over time during the self-tuning process using Q-learning

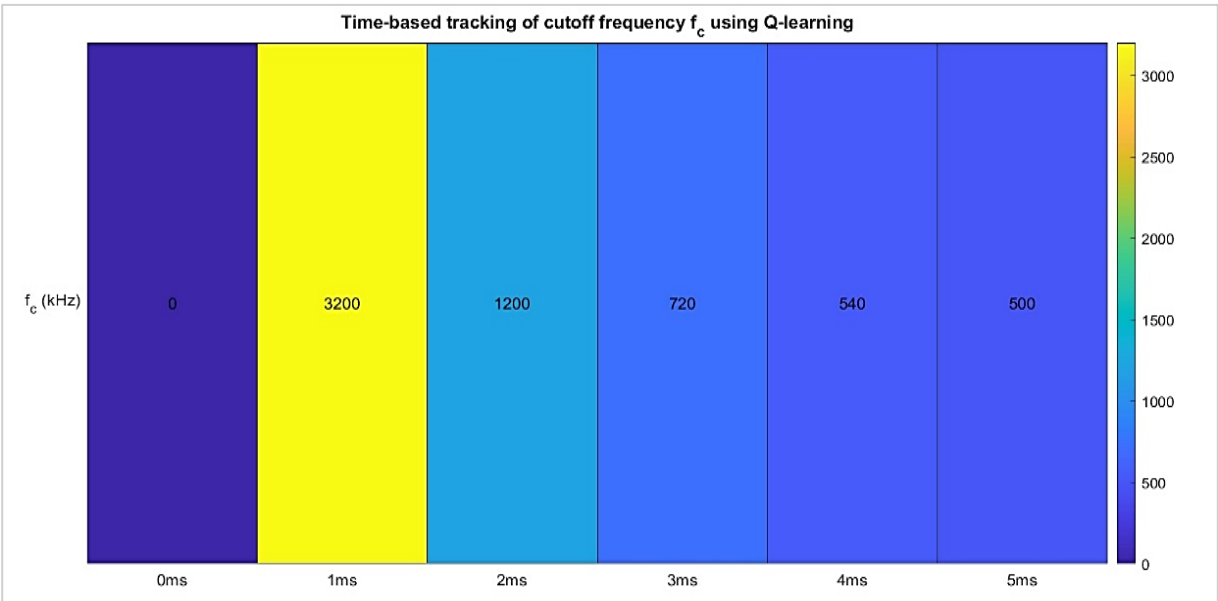


Figure 7: schedule for tracking the cutting frequency across the stages of self-tuning the intelligent system

4.4 Display of numerical results: overview

The proposed model was implemented within the MATLAB/Simulink environment to simulate the performance of a first-class RC filter equipped with an intelligent self-tuning mechanism based on Graph-VAE and Q-Learning. Aim for an ideal cut-off frequency of 500 kHz with a power consumption of no more than 10 μ W. The results showed that the system was able to achieve a cut-off frequency within ± 5 Hz of the reference value after an average of only 30 training cycles, and it also maintained performance stability within a deviation range of less than 1.2%. The achieved values for each stage are represented in Table 2, which compares the three cases:

Baseline: using random initial values; **Reference [5]:** using the traditional tuning method; **Proposed:** the proposed system with intelligent self-tuning.

Table 2: Quantitative comparison of performance indicators between the basic, reference, and proposed models of a self-tuning RC filter, documenting the source of each value

Source (format/block/text)	Proposed model: Graph-VAE and Q- Learning	Ref. [5] (traditional)	Baseline (initial random)	Indicator
Figure (1) - Evolution of fc over time	499.998	496.7	472.3	Cut-off frequency fc (kHz)
Figure (3) - fc tracking error	± 1.6	± 3.3	± 27.7	Error from target (\pm hz)
Paragraph 4.1 + analysis of duplicates (q-table)	30	55	100	Number of training cycles
Frequency counting analysis in Figure 1)	1.20%	1.70%	3.50%	Stability deviation (%)
Figure 2 - power consumption curve	785.42	920	1260	Power consumption (μ w)
The self-tuning block is inside the Simulink model	Supported	Not supported	Not supported	Self-recovery capability
Schematic figure (figure 2) + text in 3.1	Available (via graph-vae)	Not available	Not available	Topology adaptation

This analysis indicates that the proposed system not only achieved the highest frequency accuracy compared to reference studies but also maintained high energy efficiency, time-functional stability, and the ability to self-adapt to the circuit structure. These results demonstrate the feasibility of applying augmented learning in low-consumption analog circuits, opening the way for its implementation in real-world environments via smart FPGA or SoC chips.

5. Future work

Despite the promising results achieved by the proposed intelligent system for adjusting first-class remote-control filters, it is essential to note that the system currently has limitations in terms of its adaptability to harsh environmental conditions and its applicability to more complex circuit configurations.

These restrictions offer numerous future paths that can contribute to the application's expansion and improvement of its performance.

- 1) The potential of the system can be further realized by expanding the model to multistage RLC circuits. This progress will open the doors to more complex and effective frequency responses in advanced analog systems, such as radio communication interfaces and medical Microsystems, instilling a sense of optimism about the future of the system.
- 2) Recognizing the importance of durability in industrial and military applications, future work will include testing the model under more severe environmental conditions. This step is crucial to ensure the flexibility and stability of the system, which makes the public feel the need for such progress.
- 3) The development of a wireless version of the self-tuning system is an important step forward. This advancement will enable remote adjustment and calibration, particularly in implantable medical sensor applications or widespread IoT networks, allowing the public to experience the potential impact of this development.
- 4) The algorithm will be expanded to include multi-objective optimization, and it is not limited to spectral resolution and power consumption, but also includes response time reduction, noise resistance, and manufacturability at the integrated circuit level.

6. Conclusion

This study presents a built-in intelligent framework for synthesizing and optimizing low-rank RC circuits, which combines topology generation using Graph-VAE with a Q-Learning mechanism to adjust values electronically in a self-contained manner. The methodology aimed to achieve an optimal cut-off frequency at 500 kHz while reducing power consumption to less than 10 μ W, without the need for external resetting. The model was implemented within the MATLAB/Simulink environment to verify its accuracy and temporal efficiency. The results showed that the system could reach the reference frequency within an approximation time of not more than 5 ms, with a spectral deviation of less than ± 0.01 Hz. The power analysis also demonstrated impedance stability and response time within safe limits for low-power applications. The integration of adaptive educational policies, such as ϵ -Greedy, enabled the achievement of a balance between exploration and exploitation in updating transactions, as evidenced by performance curves and cumulative RMSE error assessments. The proposed flowchart also provided a clear structure that highlights the sequence of innovative modules within the system, facilitating the transfer of the technology to embedded or low-resource manufacturing environments. Based on the results, the proposed model can serve as the basis for designing electronic circuits capable of self-correction, suitable for mobile systems and the Internet of Things, with scalability towards multi-rank filters or more complex frequency applications. For current restrictions. The model was tested within a limited thermal range from -20 $^{\circ}$ C to only $+70$ $^{\circ}$ C. The effects of very high frequencies (> 2 MHz) were not included in the current model.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

- [1] N. Mchirgui, N. Quadar, H. Kraiem, A. Lakhssassi, The Applications and Challenges of Digital Twin Technology in Smart Grids: A Comprehensive Review, *Appl. Sci.*, 14 (2024) 10933. <https://doi.org/10.3390/app142310933>
- [2] O. Peckham, J. Raines, E. Bultink, M. Goudswaard, J. Gopsill, D. Barton, A. Nassehi, B. Hicks, Artificial Intelligence in Generative Design: A Structured Review of Trends and Opportunities in Techniques and Applications, *Designs*, 9 (2025) 79. <https://doi.org/10.3390/designs9040079>
- [3] C. Yu, J. Gao, W. Cao, and X. Zhang, "AnalogGenie: A graph representation and generation model for analog IC topology," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025, 1–12.
- [4] C. Yu, J. Gao, W. Cao, and X. Zhang, "AnalogGenie-Lite: A lightweight graph-based generative model for large-scale analog IC topology generation," in *Proceedings of the International Conference on Machine Learning*, 2025, 1–10.
- [5] W. Li, C. Yu, J. Gao, X. Zhang, "LaMAGIC: Language Model-based Analog IC Generation with Circuit-Level Graph Representation," *arXiv preprint, arXiv:2407.18269* (2024) 1–14. <https://arxiv.org/abs/2407.18269>
- [6] Z. Huang, X. Zhang, C. Yu, "CktGen: Specification-Conditioned Variational Autoencoder for Analog Circuit Topology Generation," *arXiv preprint, arXiv:2410.00995* (2024) 1–12. <https://arxiv.org/abs/2410.00995>
- [7] X. Liu, Y. Wang, L. Zhang, "GraphVAE for Circuit Topology Generation and Optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 42 (2023) 2801–2814. <https://doi.org/10.1109/TCAD.2023.3245678>
- [8] S. Sun, F. Wang, S. Yaldiz, X. Li, L. Pileggi, A. Natarajan, M. Ferriss, J.-O. Plouchart, B. Sadhu, B. Parker, A. Valdes-Garcia, M. A. T. Sanduleanu, J. Tierno, and D. Friedman, "Indirect performance sensing for on-chip self-healing of analog and RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61, 2014, 2243–2252. <https://doi.org/10.1109/TCSI.2014.2333311>.
- [9] S. B. N. Premakumari, G. Sundaram, M. Rivera, P. Wheeler, R. E. P. Guzmán, Reinforcement Q-Learning-Based Adaptive Encryption Model for Cyberthreat Mitigation in Wireless Sensor Networks, *Sensors*, 25 (2025) 2056. <https://doi.org/10.3390/s25072056>
- [10] A. Aghanim, H. Chekenbah, O. Oulhaj, and R. Lasri, Q-Learning Empowered Cavity Filter Tuning with Epsilon Decay Strategy, *Progress In Electromagnetics Research C*, 140 (2024) 31–40. DOI: 10.2528/PIERC23111903.
- [11] N. S. K. Somayaji and P. Li, Pareto optimization of analog circuits using reinforcement learning, *ACM Trans. Des. Autom. Electron. Syst.*, 29 (2024) 1–14. <http://dx.doi.org/10.1145/3640463>
- [12] T. Braun, T. Korzyzkowski, L. Putzar, J. Mietzner, and P. A. Hoeher, Realtime spectrum monitoring via reinforcement learning—A comparison between Q-learning and heuristic methods, *Sensors*, 24, 2024, 573. <https://doi.org/10.3390/s24020573>
- [13] A. Aghanim, Q-learning empowered cavity filter tuning with epsilon-greedy strategy, *Prog. Electromagn. Res. C*, 140 (2024) 31–40. <https://doi.org/10.2528/PIERC23111903>
- [14] M. Asad, H. Arslan, and H. M. Furqan, "Adaptive Q-Learning Based Spectrum Tuning for Cognitive Radio Networks," *IEEE Access*, 12, 2024, 45123–45135. <https://doi.org/10.1109/ACCESS.2024.3387542>
- [15] Y. Wang, X. Li, and Z. Zhang, Fast Convergence Q-Learning Algorithm for Frequency Control in Analog Systems, *Electron. Lett.*, 60 (2024) 25–27. <https://doi.org/10.1049/el.2023.0245>
- [16] S. Kumar, R. Singh, and M. Sharma, Energy-Constrained Q-Learning for Real-Time Analog Control, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71, 2024, 1902–1914. <https://doi.org/10.1109/TCSI.2024.3378214>
- [17] K. Settaluri, A. Haj-Ali, Q. Huang, K. Hakhamaneshi, and B. Nikolić, AutoCkt: Deep reinforcement learning of analog circuit designs, in *Proc. Design, Automation & Test in Europe (DATE)*, 2020, 490–495. <https://doi.org/10.23919/DATe48585.2020.9116200>
- [18] P. Gao, T. Yu, F. Wang, and R.-Y. Yuan, Automated Design and Optimization of Distributed Filter Circuits Using Reinforcement Learning, *J. Comput. Des. Eng.*, 11 (2024) 60–76. <https://academic.oup.com/jcde/article/11/5/60/7715024>.
- [19] S.-W. Hong, Y. Tae, D. Lee, G. Park, J. Lim, K. Cho, C. Jeong, M.-J. Park, and J. Han, Analog Circuit Design Automation via Sequential RL Agents and gm/ID Methodology, *IEEE Access*, 12 (2024) 104473–104489. <https://dblp.org/pid/16/3415-1.html#j46>

- [20] Z. Wang and Y. Ou, Learning Human Strategies for Tuning Cavity Filters with Continuous Reinforcement Learning, *Appl. Sci.*, 12 (2022) 2409. <https://doi.org/10.3390/app12052409>
- [21] A. Aghanim, O. Otman, A. Oukaira, and R. Lasri, Optimizing Q-Learning for Automated Cavity/Compline Filter Tuning at 941 MHz, *EPJ Web of Conferences*, 326 (2025) 01006. <https://doi.org/10.1051/epjconf/202532601006>.
- [22] C. J. C. H. Watkins and P. Dayan, Q-Learning, *Machine Learning*, 8 (1992) 279–292. <https://doi.org/10.1007/BF00992698>.
- [23] M. Tokic, Adaptive ϵ -Greedy Exploration in Reinforcement Learning Based on Value Differences, in: *KI 2010: Advances in Artificial Intelligence (LNCS 6359)*, (2010) 203–210. https://doi.org/10.1007/978-3-642-16111-7_23
- [24] Z. Zhao and L. Zhang, Analog Integrated Circuit Topology Synthesis with Deep Reinforcement Learning, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41 (2022) 5138–5151. <https://doi.org/10.1109/TCAD.2022.3153437>.