A Novel Deep-Q-Network-Based Fine-Tuning Approach for Planar Bandpass Filter Design

Masataka Ohira, *Member, IEEE*, Kohei Takano, *Student Member, IEEE*, and Zhewang Ma, *Member, IEEE*

***Abstract*— This letter proposes a novel fine-tuning approach of microstrip bandpass filters (BPFs) with a deep Q-network (DQN). In conventional works, reinforcement learning using DQN has been investigated for the automation of screw tuning in cavity BPFs. However, cross couplings appearing in planar BPF require a more complicated tuning process. To consider all the cross couplings, the proposed method introduces two neural-network-based surrogate models expressing the relationship between a coupling matrix and structural parameters. The two models also enable to drastically speed up reinforcement learning. As an example, a DQN is constructed for the design of the fifth-order microstrip BPF. The effectiveness of the DQN is numerically demonstrated through successful structural adjustments.**

***Index Terms*— Bandpass filter (BPF), neural networks (NNs), reinforcement learning.**

I. Introduction

In RECENT years, rapid growth of wireless communications requires various customized microwave circuits. However, even if a microwave circuit such as bandpass filters (BPFs) is designed based on its theory, a fine-tuning of structural parameters is indispensable. It highly depends on the designer's expertise. The automation of BPF design and the modeling of design skills are eagerly awaited.

To solve these problems, automatic BPF design approaches using neural networks (NNs) [1] have been vigorously investigated so far. One example is a BPF design using a NN called the forward model [2]–[5], which is employed for calculating electrical parameters such as frequency response from structural parameters of BPF. However, an optimization method is additionally needed, since design skills cannot be modeled by such NNs. On the other hand, a BPF tuning technique using a deep Q-network (DQN), which can model the tuning process with reinforcement learning, is being developed for the automatic adjustment of tuning screws in cavity BPFs [6]–[8]. To the best of our knowledge, however, no DQN-based structural fine-tuning approaches for planar BPFs have been developed yet. This is because undesired cross couplings between planar resonators make it much more complicated to tune structural parameters than cavity ones.

Manuscript received February 20, 2021; accepted February 25, 2021. Date of publication March 1, 2021; date of current version June 7, 2021. This work was supported in part by JSPS KAKENHI under Grant 19K04506. *(Corresponding author: Masataka Ohira.)*

The authors are with the Graduate School of Science and Engineering, Saitama University, Saitama 3388570, Japan (e-mail: mohira@mail.saitama-u.ac.jp).

This article was presented at the IEEE MTT-S International Microwave Symposium (IMS 2021), Atlanta, GA, USA, June 6–11, 2021.

Color versions of one or more figures in this letter are available at https://doi.org/10.1109/LMWC.2021.3062874.

Digital Object Identifier 10.1109/LMWC.2021.3062874

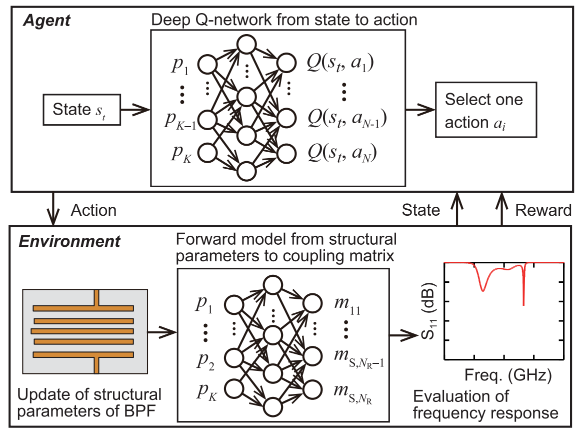

Fig. 1. Framework of the proposed BPF automatic structural adjustment using DQN (reinforcement learning) and forward model (supervised learning).

In this letter, we demonstrate an automatic structural adjustment of microstrip BPFs using DQN for the first time. In order to construct the DQN, a huge amount of computation time is required for electromagnetic (EM) simulations since successes and failures are repeated in reinforcement learning. The proposed method introduces two NNs trained by supervised learning: one is a forward model that considers all cross couplings, and the other is an inverse model that can estimate initial values. The two models also allow us to drastically reduce the learning time. The effectiveness of the constructed DQN is proven by design examples of the fine-tuning of a fifth-order microstrip BPF.

II. DQN-Based BPF Structural Fine Tuning

A. Framework

The proposed framework of automatic BPF structural adjustment using DQN is described here. An actual structural tuning process by designers needs to be represented in a computing machine. Fig. 1 shows the framework of structural fine-tuning using DQN in the proposed approach. The *agent* can be considered as a BPF designer, whose skills are modeled by the DQN. The *environment* can be regarded as an EM simulator having a BPF structure model and an EM analysis function. The input to the DQN is the *state* $s_t (t = 0, 1, 2, \ldots)$ consisting of structural parameters $(p_1, p_2, \ldots, p_K)$ of BPF. The output from the DQN is *Q value* Q expressing expected values for N predetermined *actions* $a_1, a_2, \ldots, a_N$. The action a_i corresponds to changing one of K structural parameters by $+\Delta p_k$ or $-\Delta p_k$. Therefore, N is set to be 2 K.

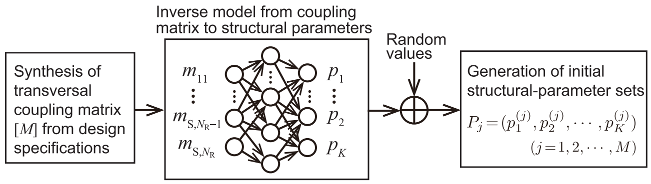

Fig. 2. Automatic generation of initial structural-parameter sets using inverse model (supervised learning).

B. Automated Fine-Tuning Process

The automatic structural adjustment using DQN is to choose actions a_i among the N predetermined actions based on Q values so that a given design specification can be met. First, the initial state s_0 is given. Then, the action with the highest Q value among the N actions for the current state is taken and is sent to the environment. In the environment, the structural parameters of BPF are updated by the action. The S-parameters are instantly calculated using the forward model, which can treat all the couplings between resonators with a transversal coupling matrix [9]. The forward model is built in advance by the NN in a parameter space of the BPF to be designed. The drastic computation-time reduction with the help of the forward model makes reinforcement learning much faster. In our algorithm. the *reward r* expressing the goodness of action is defined by an evaluation value E, which is used for the comparison between the simulated frequency response at the current state and a desired one. The equation E is given in the next section. The DQN maps state s_t to action a_i to maximize the cumulative reward during iterative structural updates. This automatic fine-tuning of BPF is finished when E is less than a target value E_{targ}.

III. Reinforcement Learning for BPF Design

To acquire a BPF design skill with machine, the DQN is constructed with the reinforcement learning. In our approach, M structural-parameter sets $P_j = (p_1^{(j)}, p_2^{(j)}, \ldots, p_K^{(j)})$ $(j = 1, 2, \ldots, M)$ are first prepared for initial states by adding a randomly selected value in $(-\Delta\nu_k, 0, +\Delta\nu_k)$ to each initial structural parameter p_k estimated by the inverse model [5], [10]–[13], as shown in Fig. 2. This is just like a situation when BPF designers fail to obtain a desired result. The reinforcement learning is executed with the following process called *episode*.

1) An initial-parameter set P_j of BPF is randomly selected among M sets, and then its evaluation value E is calculated from the S-parameters.
2) In the agent, the initial state s_t $(t = 0)$ is input into the DQN and the Q values are output from the DQN for N actions a_i $(i = 1, 2, \ldots, N)$.
3) Based on the obtained Q-values, one action is selected and is sent to the environment. Since the action with the highest Q-value is not always optimal in constructing the DQN, ϵ-greedy exploration is used to randomly select an action with a certain probability.
4) In the environment, a structural parameter of BPF is updated based on the action.
5) The value E of BPF with updated structural parameters is calculated. Then the reward r for taking the action and the next state s_{t+1} are returned to the agent.

In each episode, the above structural update is repeated until E is less than E_{targ} or the number of iterations in each episode reaches a maximum value L_{ep}.

The evaluation value E is defined in this letter by

$$E = \alpha \frac{1}{N_p} \sum_{i=1}^{N_p} F_p(f_i) + \beta \frac{1}{N_s} \sum_{i=1}^{N_s} F_s(f_i) \tag{1}$$

where

$$F_p = \begin{cases} \left|S_{11}(f_i) - S_{11}^{(\text{RL})}\right|^2, & \text{if } S_{11}(f_i) > S_{11}^{(\text{RL})} \\ 0, & \text{if } S_{11}(f_i) \le S_{11}^{(\text{RL})} \end{cases} \tag{2}$$

$$F_s = \left|S_{11}(f_i) - S_{11}^{(\text{ideal})}(f_i)\right|^2. \tag{3}$$

Here, α and β are weighting coefficients; N_p and N_s denote the number of sampling frequencies f_i at passband and stopband, respectively; S_{11}, $S_{11}^{(\text{RL})}$, $S_{11}^{(\text{ideal})}$ in decibels represent simulated reflection coefficient, specified in-band S_{11}, ideal S_{11} curve, respectively.

If the updated E by the action is smaller than previous one, a positive reward r is given to the agent. In the proposed method, we introduce E_{prog} as a short-term goal and E_{targ} as a final goal to effectively evaluate the state of BPF in the design process. The positive reward is defined by

$$r = \begin{cases} +1, & \text{if } E > E_{\text{prog}} \\ +2, & \text{if } E \le E_{\text{prog}} \end{cases} \tag{4}$$

and

$$r = +10 \text{ if } E \le E_{\text{targ}} \tag{5}$$

where $E_{\text{targ}} < E_{\text{prog}}$. Both values are reset to be gradually smaller according to the progress of reinforcement learning. If the updated E by the action is larger than the previous one, a negative reward $r = -1$ is given to the agent. The goal of reinforcement learning is to select optimal actions by the agent from its current state so that a future reward can be maximized. In the next section, the DQN will be constructed for a typical BPF design.

IV. Design Examples

A. Design Specification

As an example, the automatic structural adjustment using DQN is demonstrated for a fifth-order microstrip BPF shown in Fig. 3. In this well-known BPF configuration, cross couplings between non-adjacent resonators hinder from forming a desired passband. The design specification is given as follows.

1) Center frequency: $f_0 = 3.0$ GHz.
2) Fractional bandwidth: $\Delta f/f_0 = 5\%$.
3) In-band return loss: $RL \ge 14$ dB.
4) Number of resonators: $N_{\text{R}} = 5$.

A dielectric substrate of relative permittivity $\varepsilon_r = 2.6$ and thickness $t = 1.0$ mm is used.

B. Reinforcement Learning

The symmetric BPF structure has the six design parameters $(l_1, l_2, l_3, g_{12}, g_{23}, l_q)$. For the forward and inverse models using NNs [5], the data set is prepared in the parameter space of $33.00 \le l_1, l_2, l_3 \le 35.00$, $1.50 \le g_{12}, g_{23} \le 4.00$, and $11.00 \le l_q \le 13.00$ in mm with 0.50-mm step.

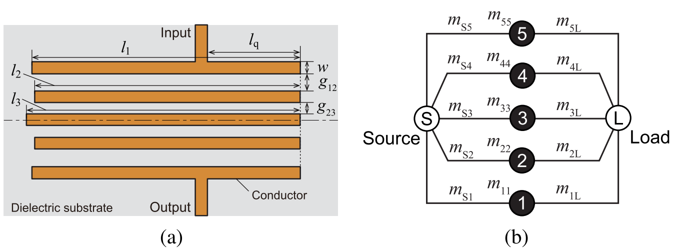

Fig. 3. (a) Layout of the fifth-order microstrip BPF. (b) Its transversal coupling topology used for the forward and inverse models.

TABLE I

COMPARISON OF STURUCUTRAL PARAMETERS IN MILLIMETERS BTWEEN THE DQN-TUNED BPF AND INITIAL ONE. (A) EXAMPLE A (26 STEPS IN DQN), (B) B (29 STEPS), AND (C) C (18 STEPS)

(A)

State	l_1	l_2	l_3	g_{12}	g_{23}	l_q	E
Initial	33.90	33.80	33.35	2.90	3.50	13.20	37.11
Designed	34.05	33.45	33.40	2.50	3.35	13.10	0.12

(B)

State	l_1	l_2	l_3	g_{12}	g_{23}	l_q	E
Initial	34.20	33.50	33.20	2.90	3.80	13.20	16.75
Designed	34.05	33.45	33.40	2.50	3.35	13.10	0.12

(C)

State	l_1	l_2	l_3	g_{12}	g_{23}	l_q	E
Initial	34.05	33.50	33.50	2.60	3.80	13.20	6.44
Designed	34.05	33.45	33.45	2.50	3.35	13.05	0.29

The width w of resonators is fixed to be 2.00 mm. In the DQN, the input is $(l_1, l_2, l_3, g_{12}, g_{23}, l_q)$ and the output is $(Q(s_t, a_1), Q(s_t, a_2), \ldots, Q(s_t, a_{12}))$. The change of one structural parameter in each action is set to be $\Delta p_k = 0.05$ mm ($k = 1, 2, \ldots, 6$). The DQN has six hidden layers with 100 nodes in each layer. The activation function is ReLU. These hyperparameters of the DQN are empirically selected in reference to several tests of supervised learnings, where it is confirmed that a NN having the same number of inputs and outputs as DQN can be accurately trained without overtraining. The initial structural-parameter sets P_j ($j = 1, 2, \ldots, 243$) are generated from estimated values with the inverse model and randomly selected values from $(-0.10, 0, +0.10)$ in mm. We perform the reinforcement learning with 30000 episodes ($L_{\rm ep} = 30$ per one episode), which takes three hours. If the same number of episodes is performed without the forward model, the estimated computation time is 625 days since it takes at least one minute to obtain frequency characteristics with a commercially available EM simulator.

C. Design Results

As three examples of the fine-tuning using the constructed DQN, the changes of the structural parameters and the evaluation values E between initial and designed BPFs are compared in Table I. Their frequency responses of S_{11} are shown in Fig. 4, where the EM simulation results are obtained by Sonnet *em*. Once the DQN is constructed, the DQN can find optimum structural parameters in less than 30 steps, of which the tuning time is within about 0.4 seconds with an Intel Xeon 3.4-GHz processor. As shown in Fig. 4, the DQN succeeds both in achieving the specified in-band return loss and in adjusting the center frequency by automatically tuning lengths and gaps, even when different initial-values sets are given to the same DQN. If a genetic algorithm (GA), which is one of the well-known global optimizers, is employed for this filter design, several hundreds of iterations (generations in GA) are needed to fully explore solutions. The effectiveness of the proposed approach can also be confirmed from five reflection zeros generated in the passband of the DQN-tuned BPFs. Finally, Table II summarizes the features of the proposed method in comparison with recently developed NN-assisted BPF design methods.

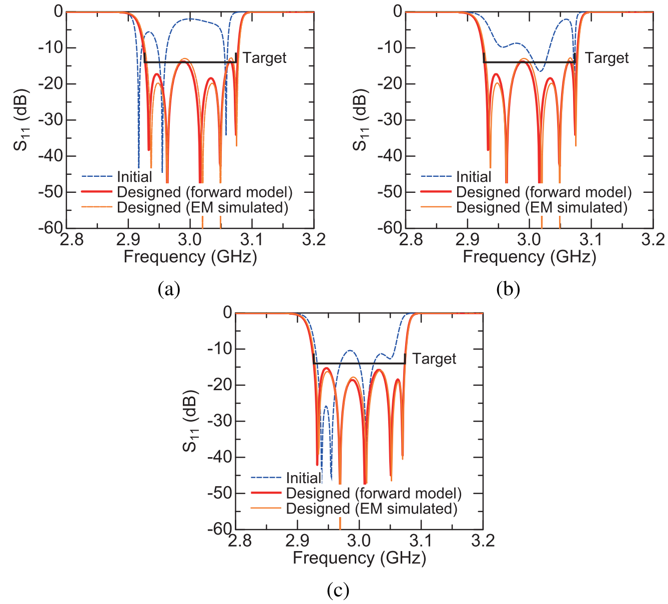

Fig. 4. Comparison of frequency responses between the DQN-tuned BPF and initial one. (a) Examples A, (b) B, and (c) C.

TABLE II

COMPARISON OF NN-ASSISTED BPF DESIGNS BETWEEN RECENT WORKS AND THE PROPOSED METHOD

	Ref. [4]	Ref. [13]	This work
Design target	Waveguide BPF	SIW BPF	Microstrip BPF
Undesired couplings	Negligible small	Negligible small	Not negligible
EM analysis	Forward models for substructures	EM simulation for whole structure	Forward model for whole structure
Initial-values guess	Filter designer	Inverse models for substructures	Inverse model for whole structure
BPF design method and design time	Homotopy optimization in minutes	Updates of coarse model with EM simulations in hours	DQN in seconds
Modeling of design process	Out of scope	Out of scope	Modeled in DQN (3 hours)

V. CONCLUSION

In this letter, a novel automatic fine-tuning approach using DQN has been proposed for planar BPFs with undesired cross couplings. The proposed method incorporates forward and inverse models into reinforcement learning, thereby enabling a huge number of trials and errors in reinforcement learning. We constructed the DQN for the design of a fifth-order microstrip BPF and demonstrated that the DQN successfully adjusted the structural parameters even if a different initial-values set is given to the DQN. This illustrative example may open up new horizons to future microwave circuit design.

REFERENCES

[1] Q. Zhang, K. C. Gupta, and V. K. Devabhaktuni, “Artificial neural network for RF and microwave design—From theory to practice,” *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1339–1350, Apr. 2003.

[2] G. Fedi, A. Gaggelli, S. Manetti, and G. Pelosi, “Direct-coupled cavity filters design using a hybrid feedforward neural network-finite elements procedure,” *Int. J. RF Microw. Comp.-Aided Eng.*, vol. 9, no. 3, pp. 287–296, May 1999.

[3] F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, “Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions,” *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60–77, Jan. 2016.

[4] P. Zhao and K. Wu, “Homotopy optimization of microwave and millimeter-wave filters based on neural network model,” *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 4, pp. 1390–1400, Apr. 2020.

[5] M. Ohira, A. Yamashita, Z. Ma, and X. Wang, “Automated microstrip bandpass filter design using feedforward and inverse models of neural network,” in *Proc. Asia–Pacific Microw. Conf.*, Kyoto, Japan, Nov. 2018, pp. 1292–1294.

[6] Z. Wang, J. Yang, J. Hu, W. Feng, and Y. Ou, “Reinforcement learning approach to learning human experience in tuning cavity filters,” in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Zhuhai, China, Dec. 2015, pp. 2145–2150.

[7] Z. Wang, Y. Ou, X. Wu, and W. Feng, “Continuous reinforcement learning with knowledge-inspired reward shaping for autonomous cavity filter tuning,” in *Proc. IEEE Int. Conf. Cyborg Bionic Syst.*, Shenzhen, China, Oct. 2018, pp. 53–58.

[8] E. Sekhri, R. Kapoor, and M. Tamre, “Double deep Q-learning approach for tuning microwave cavity filters using locally linear embedding technique,” in *Proc. Int. Conf. Mech. Syst. Mater.*, Bialystok, Poland, Jul. 2020, pp. 1–6.

[9] R. J. Cameron, C. M. Kudsia, and R. R. Mansour, *Microwave Filters for Communication Systems: Fundamentals, Design, and Applications*. Hoboken, NJ, USA: Wiley, 2007.

[10] M. Ohira, A. Yamashita, Z. Ma, and X. Wang, “A novel eigenmode-based neural network for fully automated microstrip bandpass filter design,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, Honolulu, HI, USA, Jun. 2017, pp. 1628–1631.

[11] Y. Wang, M. Yu, H. Kabir, and Q. Zhang, “Effective design of cross-coupled filter using neural networks and coupling matrix,” in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, USA, Jun. 2006, pp. 1431–1434.

[12] H. Kabir, Y. Wang, M. Yu, and Q.-J. Zhang, “Neural network inverse modeling and applications to microwave filter design,” *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 4, pp. 867–879, Apr. 2008.

[13] G.-Y. Du and L. Jin, “Neural network of calibrated coarse model and application to substrate integrated waveguide filter design,” *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 30, no. 10, pp. 1–17, Oct. 2020.