ISSN 2224-087X. Електроніка та інформаційні технології. 2022. Випуск 19. С. 48–57
Electronics and information technologies. 2022. Issue 19. P. 48–57

УДК 004.032.26 DOI: https://doi.org/10.30970/eli.19.4

IMPLEMENTATION OF DIGITAL FILTERS USING NEURAL NETWORKS WHEN THE OBJECT MOVEMENT DYNAMICS ARE NOT KNOWN

Z. Liubun[1], V. Nesterenko[1], V. Mandziy[2], O. Karpin[2]

[1]*Ivan Franko National University of Lviv,*
107 Tarnavsky St., UA–79017 Lviv, Ukraine
zinoviy.lyubun@lnu.edu.ua , *Volodymyr.Nesterenko@lnu.edu.ua*

[2]*Infineon Technologies AG,*
Lviv, Ukraine
Vasyl.Mandziy@infineon.com , *Oleksandr.Karpin@infineon.com*

Considered: Neural network-based algorithms to obtain an optimal digital filter. Recurrent neural networks can be used to implement an adaptive digital filter considering the characteristics of both interference and useful signal. The developed program trains recurrent neural networks and as a result obtains the coefficients of recursive digital filters. The efficiency of the proposed algorithm is confirmed with numerical experiments results.

Keywords: digital filter, recursive filters, recurrent neural networks

1. Introduction

To select an optimal digital filter, profound knowledge in the theory of digital filtering and methods of adaptive digital filtering is required. Often, one needs to implement an optimal filter using the known interference characteristics and technical requirements of the device and without delving into the theory of filtering. For example, such can be the filtering a sensor signal from a human ear to a mobile phone in order to select the optimal level of radiation power or filtering signals for gesture detection [1]. The simplest case of Kalman filter – under an unknown value of the motion level, to iteratively select the value of the filter coefficients. Such a case can be implemented by training a recurrent neural network.

A typical example – obtaining an optimal filter considering the disturbances – a linear Kalman filter. Figure 1 shows a recurrent neural network that makes it possible to obtain optimal values of the coefficients of such a digital filter.

$$Out(k) = W_x \cdot x(k) + W_y \cdot Out(k-1) \quad (1)$$

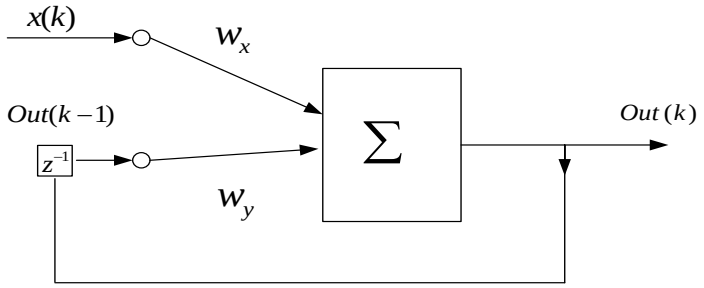

Fig.1. The structure of a recurrent neural network for implementation a linear Kalman filter

To obtain the optimal coefficients, it is enough to train a neural network on a multiple of training pairs that are sets of training pairs – each pair consists of a real noisy signal and a known signal without noise. Moreover, one can avoid researching the noise characteristics and just train a large number of training pairs in different measurement modes for a specific task and a specific device. If adaptation to new operating conditions is necessary, it is enough to retrain the neural network.

In general, the simplest structure of a recurrent neural network (Fig. 2) can be used to implement recursive filters with an arbitrary number of coefficients:

$$\begin{aligned} Out(k) &= W_x(0)\cdot x(k) + W_x(1)\cdot x(k-1) + ... + W_x(m)\cdot x(k-m) + \\ &+ W_y(0)\cdot Out(k-1) - W_y(1)\cdot Out(k-2) - ... - W_y(n-1)\cdot Out(k-1-n) \end{aligned} \quad (2)$$

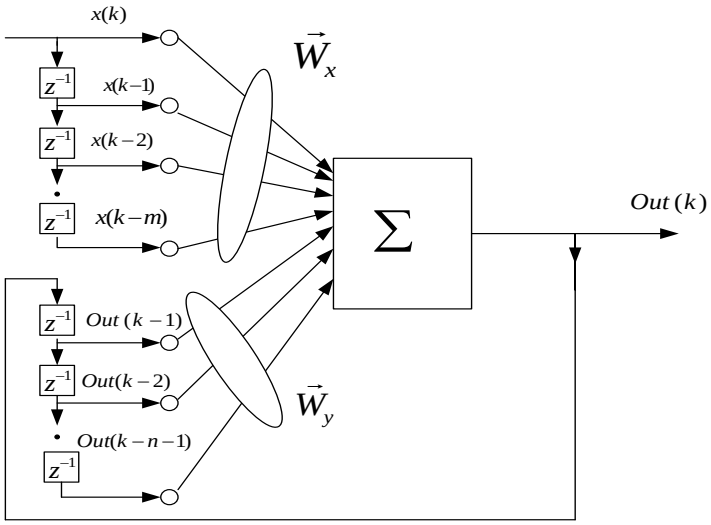

Fig. 2. The structure of a neural network for implementing a recursive filter with an arbitrary number of coefficients

To test this approach, model input signals were generated, to which noise with a Gaussian distribution with the zero mathematical expectation was added. The network was trained from random initial values of the weights using the gradient method with a variable learning rate.

2. Results

Model signal (3) was chosen to test the efficiency of this approach

$$x(t) = \sin(0.2 \cdot t) + \varepsilon(t) \tag{3}$$

where $\varepsilon(t)$ is a disturbance with a Gaussian distribution and zero mathematical expectation and dispersion equal to a unit.

The appearance of the signal with and without noise is shown in Fig. 3 with a sampling step $\Delta t = 0.04$.

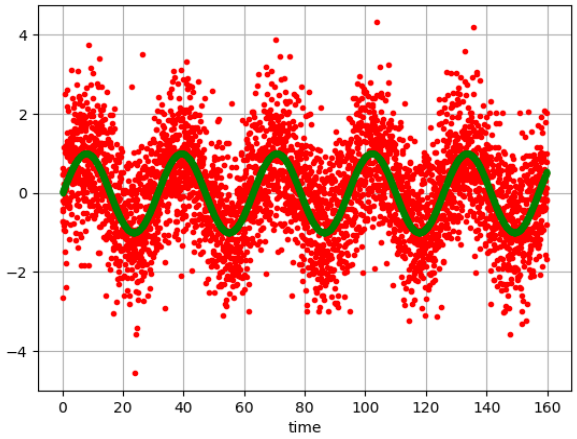

Fig. 3. The appearance of the signal with and without noise. Green – input signal; red – signal with noise

The change in the values of the weights during the training process is shown in Figure 4.

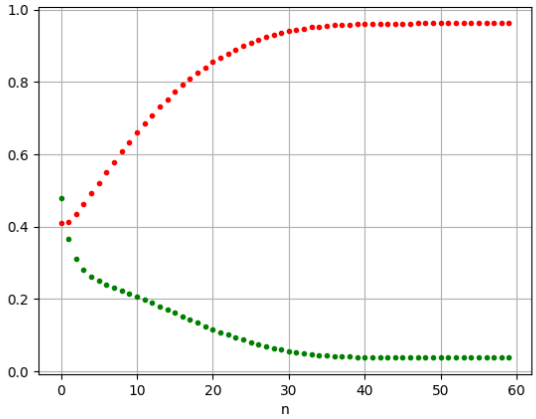

Fig.4. The change in the values of the weights during the training process. Green – W_x ; red – W_y

The final values of the weights after training: $W_x = 0.038$, $W_y = 0.962$. Note that the total value of the weights is equal to a unit, as in the standard filtering formulas for the simplest linear Kalman filter [2-4]. As a result of training, the resulting digital filter reduces the mean square value of the noise component by five $K_f \approx 5$. The appearance of the signal after filtering and the signal without noise is shown in Figure 5.

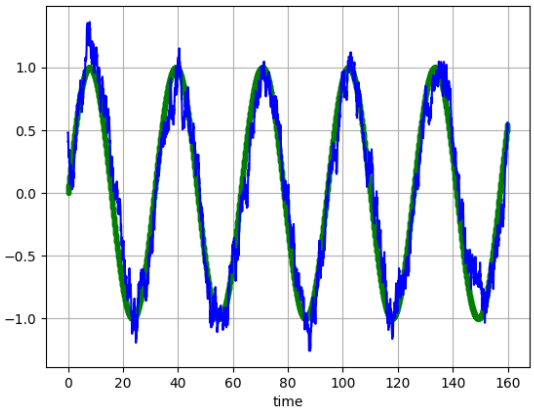

Fig. 5. The appearance of the signal after filtering. Green – input signal; blue – signal after filtering

Note that during the training process, the optimal values of the weights are established quite quickly that is in several tens of training epochs. (Fig. 4).

Important: a neural network can adapt its parameters during operation but, which is natural, only under the desired input signal. Fig. 6 shows how a neural network adapts from random initial values of the weights to improve the filtering quality.

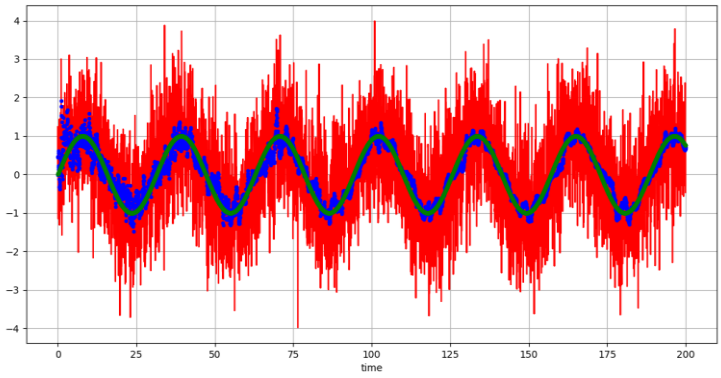

Fig. 6. Dynamics of neural network tuning. Red – signal with noise; green – input signal; blue – signal after filtering

The filter obtained in this way can be used in a certain frequency range of the harmonic signal. Set a multiple of training pairs from the desired working area and train the network, after which it behaves like a low-pass filter. A transmission coefficient constant and close to a

unit is desirable for the operating range of high-frequency interference filtering for a useful harmonic signal.

Fig. 7 shows the frequency response of the filter obtained as a result of network training with the following initial values: sampling frequency 1000 (Hz), signal frequency for training pairs - 10 and 100 (Hz), root mean square noise value of training pairs $sign = 1.0$ ($SNR = \frac{A}{6\sigma_{input}} = \frac{1}{6}$, where A is the amplitude of the harmonic signal, σ_{input} - root mean square deviation of the noise of the input signal.)

The process of training yields an uneven frequency response (Fig. 7), but the network filtering properties remain unchanged and they reduce the noise level by approximately three times for all frequencies that satisfy the Nyquist theorem at a given sampling frequency (in this case, for frequencies no higher than 500 Hz).

To accomplish the research, specify the desired SNR value for the training and the results of filtering of the trained network under the other SNR values of the signals to be filtered. Table 1 shows the results of the research.

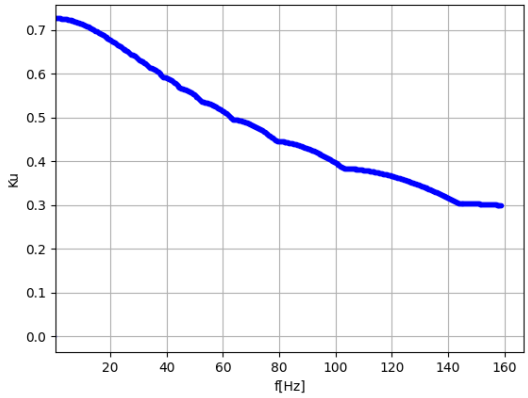

Fig. 7. The frequency response of the trained neural network

Table 1. Filtering degree

№	SNR Input	SNR Output	SNR Input/ SNR Output
1	0.67	1.0	1.5
2	0.33	0.67	2
3	0.17	0.6	3.5
4	0.083	0.54	6.5
5	0.056	0.67	12

Table 1 shows that increasing the noise of the multiple of training signals (reducing the SNR for the multiple of training signals) improves the filtering degree. At the same time, the frequency response of the received filter changes significantly in the operating range. Figure 8 shows the frequency response for cases 1(a) and 4(b) and 5(c) from Table 1.

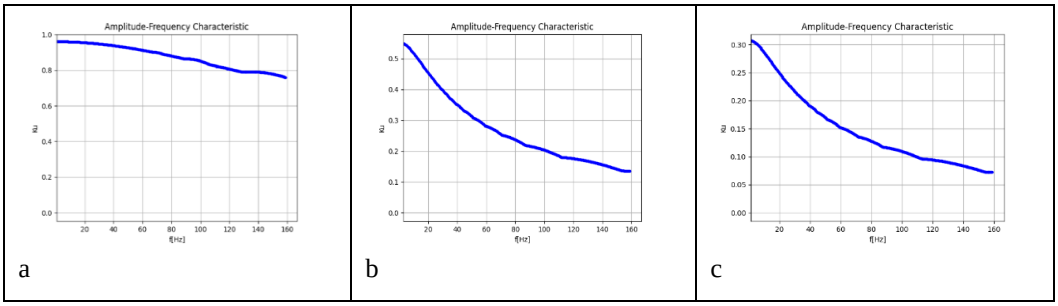

Fig. 8. The frequency response of the filter at different SNR values from Table 1

Fig. 8.c shows that along with good filtering, the level of the useful signal at the output of the filter is small and the practical usage of such a filter is problematic. Moreover, the phase shift between the input and output signals of the filter increases. One more positive property of the obtained filter deserves mentioning – if the noise level of the signal during the operation of the filter is smaller than for signals from the training multiple, then the filtering quality is higher. For example, for Case 4 from Table 1, the training was carried out with a root mean square value of noise of 2.0, then in the case of reducing the root mean square value of noise by half, the increase in the SNR reaches 10.

It is obvious that the shape of the useful signal is different for different tasks. Therefore, it is important to understand how the resulting filter will behave with different types of useful signal that differs from the signal on which the network was trained. In particular, often during the operation of the devices, signal changes due to temperature drift. [11, p.18] Therefore, it is logical to investigate whether the received filter will provide the required level of filtering or whether it must be adapted to such a change in the signal.

In the case of filtering the sensor signal of the distance of a human ear to a mobile phone in order to select the optimal level of radiation power, it is logical to check the ability of the received filter to work in the case of a thermal drift of zero or a change in the distance to the user. To do this, a triangular signal (Figure 9) that simulates a linear change of the signal and a change in the constant level of the signal was sent onto the network trained on the harmonic signal.

The filter obtained by training the network on a harmonic signal (Fig. 9) provides noise filtering only qualitatively repeating the linear change of the useful signal. Therefore, it is logical to train a neural network basing on a linearly variable noisy signal and then check the quality of the received filter for other types of signals including harmonic signals.

Figure 10 shows the test results for this case – the test signal from Fig. 9 becomes the training signal, and the test signal from Figure 10 serves for the testing.

The filter obtained in this way works successfully both with a triangular and harmonic signal at the same level of filtering. Let's set a step signal as a variant of the test signal. Fig. 11 shows the results of the research.

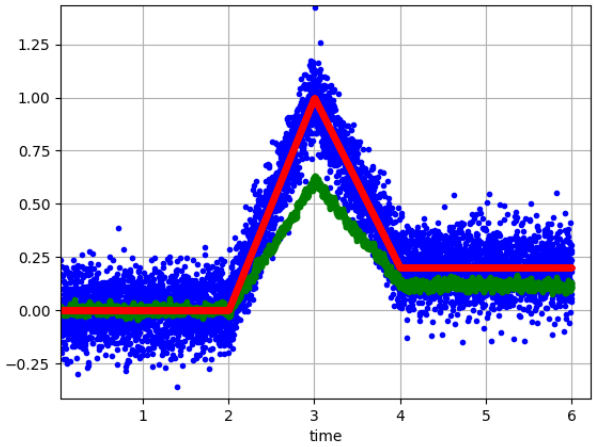

Fig. 9. The results of testing the trained neural network.
Red – test signal; blue – test signal with noise; green – signal after filtering

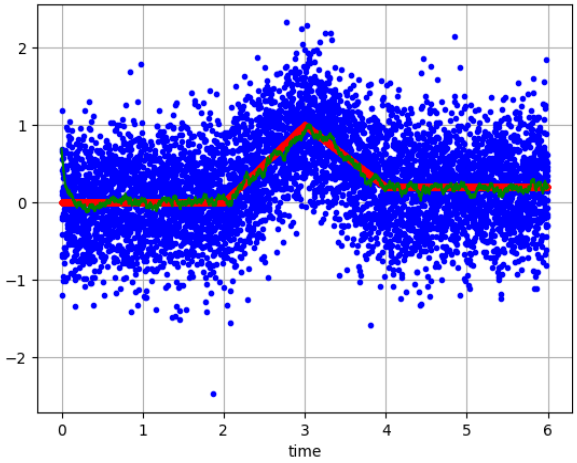

Fig. 10. The result of the network training for the triangular signal case.
Blue – signal with noise; red – input signal; green – signal after filtering

Fig. 11 shows that the filter successfully reproduces the step signal, significantly reducing the root mean square value of the noise.

Thus, it can be argued that after training a recurrent neural network on a triangular signal, a filter is obtained that provides effective filtering for the other types of input signal.

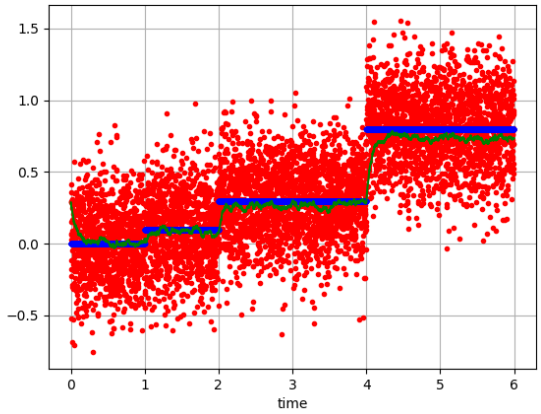

Fig. 11. The result of testing with a step signal.
Blue – signal with noise; red – input signal; green – signal after filtering

The next stage of research shows how the order of the filter affects the level of filtration, which will make it possible to obtain an optimal filter.

A noisy triangular signal was used for training, and a stepped signal for testing. The value of the reduction in the rms value of the noise obtained due to the filtering and rounded to a whole value is entered in the cells.

Table 2. RMS noise reduction

n	*m*				
	1	2	3	4	5
1	10/4	6/4	10/5	12/5	11/4
2	12/5	12/5	14/4	9/5	10/4
3	11/4	9/4	13/4	10/5	11/5
4	16/4	13/5	16/4	10/5	11/4
5	15/3	14/4	11/5	12/5	11/5

Table 2 shows the A/B values for each filter option, where A is the reduction of the variance of the input signal noise without considering the error of determining the values of the constant level and phase shift of the signal, and B is the total error of reducing the variance of the input signal noise, considering the error of determining the constant level and the shift phases.

Increasing "*m*" and "*n*" leads to an improvement in noise filtering but slows down the transition between fixed levels and the accuracy of their determination, in other words, the filter begins to filter out the useful part of the signal. Moreover, often, with large values of "m"

and "n", the values of the weights of the neural network corresponding to the filter coefficients are very small compared to the values of m<3 and n<3. For example, the value of weights at m=n=3:

$W_m[0.012, 0.006, 0.001]$

$W_n[0.492, 0.438, 0.049]$

3. Conclusions

The following conclusions can be drawn on the basis of the conducted research:

1. It is possible to create an optimal filter with the help of the simplest recurrent network based on the features of the useful signal (an analogue of the known dynamics of movement) with given interference parameters.

2. A neural network trained on one dynamic is a filter for other types of signals as well.

3. The SNR ratio of the training multiple affects the properties of the resulting filter and makes it possible to select the best option for a specific task.

4. With the same initial data, as a result of training, for different initial values of the weights, a whole multiple of neural networks is obtained that satisfy the requirements of accuracy in the training process.

REFERENCES

[1] *Z. Liubun, V. Mandziy, O. Karpin, V. Rabyk*, "Neural-network-based Gesture Detection for Capacitive Sensing," 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2022, pp. 362-365, doi: 10.1109/TCSET55632.2022.9767083.

[2] *Grewal M.S., Andrews A.P.* Kalman Filtering: theory and practice. Prentice-Hall, Englewood Cliffs (1993)

[3] *S. Haykin.* Adaptive Filter Theory, 5th Edition, Prentice Hall, 2013.

[4] *S. Haykin*. Kalman Filtering and Neural Networks. First published:1 October 2001.Print ISBN:9780471369981 Online ISBN:9780471221548 DOI:10.1002/0471221546

[5] Recurrent Neural Filters: Learning Independent Bayesian Filtering Steps for Time Series Prediction Bryan Lim, Stefan Zohren and Stephen Roberts Oxford-Man Institute of Quantitative Finance Department of Engineering Science University of Oxford Oxford, UK {blim,zohren,sjrob}@robots.ox.ac.u.

[6] Oxford-Man. Recurrent Neural Filters: Learning Independent Bayesian Filtering Steps for Time Series Prediction. Retrieved from https://www.oxford-man.ox.ac.uk/wp-content/uploads/2020/03/Recurrent-Neural-Filters-Learning-Independent-Bayesian-Filtering-Steps-for-Time-Series-Prediction.pdf

[7] IOPSCIENCE. Recurrent neural networks as approximators of non-linear filters operators. Retrieved from https://iopscience.iop.org/article/10.1088/1742-6596/1141/1/012115/pdf

[8] STOWERS INSTITUTE RESEARCH WEBSITES. A fast noise filtering algorithm for time series prediction using recurrent neural networks. Retrieved from https://research.stowers.org/bru/RNN_Filter2_V3.pdf

[9] *S. Haykin.* Kalman filtering and neural networks, John Wiley, 2001.

[10] *Parlos A. G., Menon S. K. and Atiya A. F*. (2001). An algorithmic approach to adaptive state filtering using recurrent neural networks. “IEEE Transactions on Neural Networks”, 12-6:1411--1432.

[11] AN64846, Getting Started with CapSense®, Document No. 001-64846 Rev. *T, Retrieved from https://www.mouser.com/pdfdocs/AN64846.pdf

РЕАЛІЗАЦІЯ ЦИФРОВИХ ФІЛЬТРІВ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ, КОЛИ ДИНАМІКА РУХУ ОБ’ЄКТА НЕВІДОМА

З. Любунь[1], В. Нестеренко[1], В. Мандзій[2], О. Карпін[2]

[1]Львівський національний університет імені Івана Франка,
вул. Ген. Тарнавського, 107, 79017 Львів, Україна
zinoviy.lyubun@lnu.edu.ua , Volodymyr.Nesterenko@lnu.edu.ua

[2]Infineon Technologies AG
Львів, Україна
Vasyl.Mandziy@infineon.com , Oleksandr.Karpin@infineon.com

У статті представлено новий підхід до розробки цифрових фільтрів з використанням нейронних мереж. Для налаштування фільтра використовується математична модель сигналу, тобто певний чистий сигнал з шумами та без шумів, які відіграють роль тренувального прикладу для нейронної мережі. Таким чином вдається максимально спростити інтерфейс системи, переклавши відповідальність за вибір налаштувань фільтра на сам цифровий фільтр.

Побудований цифровий фільтр є рекурентною нейронною мережею з одним нейроном. Як функцію активації було використано лінійну функцію. На ваги нейрона подаються значення вхідного сигналу та відфільтрованого сигналу. Наведена у статті схема фільтру дає можливість побудови різних його варіацій з різною кількістю вхідних та вихідних ваг. Залежно від набору ваг, фільтр по-різному реагує на складові вхідного сигналу.

Для тренування та тестування фільтра було обрано синусоїдальний сигнал, до якого було додано шумовий сигнал з нормальним розподілом і нульовим математичним сподіванням. На такому сигналі було перевірено застосовність цифрового фільтра, а також досліджено хід навчання з відповідним аналізом зміни ваг нейронної мережі. Застосовність фільтра в умовах наближених до реальних було перевірено з використанням трикутного та сходинкового сигналів. Візуалізація вхідних сигналів та результатів фільтрації для кожного з випадків дає змогу наочно оцінити якість роботи фільтра.

В якості об’єктивної оцінки фільтрації було обрано коефіцієнт SNR, що характеризує відношення амплітуди сигналу до величини її стандартного відхилення. В ході аналізу отриманих результатів було отримано амплітудно-частотну характеристику фільтра за різних умов навчання. Таким чином було показано вплив потужності шуму у вхідному сигналі на результати фільтрації.

Для реалізації цифрового фільтра з елементами нейронних мереж було використано мову програмування Python.

Ключові слова: цифровий фільтр, рекурсивні фільтри, рекурентні нейронні мережі.

Стаття надійшла до редакції 10.10.2022.
Прийнята до друку 21.10.2022.