

Reinforcement Learning Algorithm and FDTD-Based Simulation Applied to Schroeder Diffuser Design Optimization

ADAM KUROWSKI¹ AND BOZENA KOSTEK², (Senior Member, IEEE)

¹Multimedia Systems Department, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland

²Audio Acoustics Laboratory, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland

Corresponding author: Bozena Kostek (bokostek@audioakustyka.org)

ABSTRACT This paper aims to propose a novel approach to the algorithmic design of Schroeder acoustic diffusers by employing a deep learning optimization algorithm and a fitness function, which are based on a computer simulation of the propagation of acoustic waves. The deep learning method employed for the research consists of a deep policy gradient algorithm. It is used as a tool for carrying out a sequential optimization process, which seeks to maximize the fitness function based on parameters characterizing the autocorrelation diffusion coefficient of the designed acoustic diffuser. As the autocorrelation acoustic diffusion coefficients are calculated based on the polar response of a diffuser, the finite-difference time-domain (FDTD) simulation method is used to obtain a set of impulse responses, which are necessary to calculate the polar responses of the optimized Schroeder diffusers. The results obtained from the optimization derived from the deep learning algorithm were compared with the outcomes of a similar algorithm by employing a genetic algorithm and based on random selection of the Schroeder diffuser well-depth pattern. We found that the best result was achieved by the deep policy gradient, as it produced outcomes that, in terms of the provided autocorrelation diffusion coefficient, were statistically better than the properties of the designs supplied by two other baseline approaches.

INDEX TERMS Schroeder diffuser optimization, acoustic simulation, finite-difference time-domain (FDTD), reinforcement learning.

I. INTRODUCTION

The main goal of the experiments reported in this paper is to use an acoustic simulation approach for the automated optimization of a Schroeder diffuser based on machine learning. Acoustic simulations are extensively used in acoustics for tasks such as auralization [1], noise prediction [2], simulation-driven optimization of room acoustics, and designing acoustic devices [3]–[7]. Following the development of GPU-accelerated computations, the simulations are also being performed more efficiently with respect to time and resources [8], [9].

In this paper, techniques involving computer simulation based on the finite-difference time-domain (FDTD) method and deep reinforcement-learning-based optimization are applied to Schroeder diffusers, commonly used in rooms undergoing the process of acoustic treatment. There is a

crucial need to obtain a specific set of diffuser acoustical properties that can be used to shape the response of the enclosed space under treatment. Acoustic diffusers utilize the phenomenon of acoustic wave scattering in order to increase the sound diffusion at the place of their application [10], which may be beneficial in several cases. Hence, the research presented in this paper attempts to propose a solution, which creates an acoustic diffuser architecture based on the optimization process.

Various techniques are employed to design diffusers, from classical approaches based on pseudo-random number sequences to methods that utilize fractal geometry [11]. As an operational principle, they may employ not only the phenomena of reflection, deflection, and scattering of the waves but also resonance phenomena [10], [12], [13]. It is necessary to carefully select a methodology that will consider an entire range of factors that affect the behavior of the diffuser. Therefore, optimization algorithms in the form of genetic approaches, for instance, are an attractive way to resolve

The associate editor coordinating the review of this manuscript and approving it for publication was Jinming Wen.

the problem of finding the acoustic diffuser geometry, which optimizes the listening parameters in a given context. Another optimization approach may be the use of machine learning algorithms. Among several machine learning algorithms, one class seems to be particularly beneficial in sequential optimization problems, i.e., reinforcement learning and its modification employing deep neural networks—deep reinforcement learning [14], [15]. This class of algorithms was successfully applied to complex problems involving decision-making and multi-step optimization, such as the design of industrial infrastructure [16], control of modular robots [17], and so on. Accordingly, for the optimization of an acoustic diffuser structure, which is a sequential optimization problem, deep reinforcement learning is to be applied. In summary, we propose a method for the optimization of the properties of a simulated acoustic diffuser with the use of a reinforcement learning algorithm. The algorithm employed for this task is a deep policy gradient.

II. SIMULATION OF ACOUSTIC WAVE PROPAGATION

The use of meta-heuristics such as genetic algorithms or sequential optimization algorithms for the design of acoustics diffusers leads to the need for precise assessment of the acoustic diffuser properties. This can be utilized to calculate the so-called fitness function, which is optimized by the algorithm of choice, employing a computer simulation. Computer modeling and simulation of the sound field also find their application in the design of rooms, their construction, and their modernization. It is possible to estimate the room acoustic properties with high accuracy when measurements are not feasible, or the room is still at the design stage. In the context of the sound treatment of rooms, this significantly reduces the cost and time of construction and avoids the additional step of designing acoustic adaptation.

A. STATE-OF-THE-ART SIMULATION APPROACHES

As mentioned above, the main goal of the experiments reported in this paper is to use the acoustic simulation for automated optimization. Although it can be tempting to employ acoustic computer-aided design (CAD) to evaluate diffuser designs generated by optimization algorithms, they often do not provide an application programming interface (API) that would be suitable for the fast testing of automatically generated measurement scenarios. Moreover, they are usually not compatible with machine learning libraries such as the TensorFlow and Keras libraries [18], [19], whereas these are essential components of the deep reinforcement-learning-based approach implemented by the authors. Moreover, geometrical room acoustic models (GRAMs), such as ray-tracing and virtual source methods, are typically used as the means of simulating room acoustics in CAD software [10]. Still, in some applications, such as the automated design of passive acoustic devices, it is necessary to obtain more accurate solutions by numerical solution of acoustic wave equations. Selected state-of-the-art simulation methods are as follows:

- boundary element method (BEM),
- finite element method (FEM),
- finite-difference time-domain method (FDTD).

Algorithms such as BEM and FEM are capable of obtaining very accurate estimates of acoustic wave field distribution [20]. Both of these methods are based on frequency-domain calculations, and due to this fact, their output does not allow the impulse responses of a room to be obtained in a direct way. An alternative to these two methods is the FDTD method. FDTD is relatively easy to implement in terms of the length of code needed to obtain meaningful results, which helps to easily integrate this simulation algorithm with the program performing optimization of an acoustic Schroeder diffuser. This, for instance, allows the implementation of direct communication between the simulation and design algorithms through the RAM of the computer in a single script written in the Python language. Also, it is desirable to employ a method that directly permits the estimation of impulse responses in both room-related and anechoic conditions if additional lossy acoustic wave equations are solved by the FDTD method. FDTD simulations can also benefit from being performed on GPUs, which significantly increases the speed of calculations [9], [10]. All of these benefits resulted in this method being selected to optimize acoustic diffuser designs.

B. SIMULATIONS BASED ON THE FDTD APPROACH IN AN ANECHOIC ENVIRONMENT

The optimization of Schroeder diffusers described in the latter parts of the paper is associated with acoustic properties measured in anechoic conditions. Therefore, the simulation has to consider and provide results that reflect the interaction of the excitation acoustic wave and the diffuser in an anechoic environment. A perfectly matched layer (PML) is an example of a technique allowing FDTD-based simulations of an acoustic wave in anechoic conditions, which is necessary for the numerical estimation of all parameters of scattering devices requiring assessment of polar response such as the reflection and diffusion coefficients of Schroeder diffusers. The introduction of PML into a numerical domain involves modification of the equations guiding the acoustic wave propagation in the area of the PML. The first step of the derivation of such layer equations is using a variant of the wave equation, taking into account losses in a propagation medium [21], [22]. A starting point for such a problem is the acoustic wave equations for a homogeneous lossy media [23]:

$$\begin{cases} \nabla p + \rho \frac{\partial \mathbf{u}}{\partial t} + \alpha^* \mathbf{u} = 0 \\ \nabla \mathbf{u} + \frac{1}{\rho c^2} \frac{\partial p}{\partial t} + \alpha p = 0 \end{cases} \quad (1)$$

where: p denotes pressure, ρ is the density of air, c is the speed of sound wave propagation in the medium, \mathbf{u} is an acoustic velocity vector, α is an acoustic compressibility-related attenuation factor, which is typically referred to as an attenuation factor if the propagation of acoustic waves in the

air is considered, and α^* is an attenuation factor related to the so-called “mass-proportional” damping, which is typical for sound propagation in solids and is usually zero for propagation in the air. Such a kind of PML is similar to the Berenger PML [24].

A second-order acoustic wave equation derived from Eq. 1 has the following form:

$$c^2 \nabla^2 p = \frac{\delta^2 p}{\delta t^2} + \alpha_A \frac{\delta p}{\delta t} + \alpha_B p \quad (2)$$

where:

$$\alpha_A = \rho c^2 \alpha + \frac{1}{\rho} \alpha^* \quad (3)$$

$$\alpha_B = c^2 \alpha \alpha^* \quad (4)$$

After discretization of Eq. 2, transformations, and some rearranging, the following equation can be obtained:

$$\begin{aligned} p_{i,j,k}^{n+1} = & \left(1 + \frac{\alpha_A}{2} T + \alpha_B T^2\right)^{-1} \left(\lambda^2 \left[p_{i+1,j,k}^n \right. \right. \\ & + p_{i-1,j,k}^n + p_{i,j+1,k}^n + p_{i,j-1,k}^n \\ & + p_{i,j,k+1}^n + p_{i,j,k-1}^n - 6p_{i,j,k}^n \left. \right] \\ & + 2p_{i,j,k}^n - \left[1 + \frac{\alpha_A}{2} T\right] p_{i,j,k}^{n-1} \right), \end{aligned} \quad (5)$$

where: T is the sampling period and λ denotes the Courant number defined as:

$$\lambda = c \frac{T}{X}, \quad (6)$$

where X is the spatial sampling distance.

This form of acoustic wave equation can be the basis for obtaining a version of the equation proposed by Webb and Bilbao in their work on acoustic wave equations tailored to computation on a GPU if $\alpha_A = \alpha_B = 0$ [9]. For a lossy propagation media, one has to consider both of the damping factors, as the original formula proposed in [9] does not permit boundary losses to be introduced. This fact makes it impossible to use PMLs while also employing an original version of the homogeneous acoustic wave equation for calculation on a GPU from [9]. Therefore, we propose and use Eq. 5 in the computations in this study, which is a more general form of the equation derived in [9]. The proposed equation is capable of both employing the MPLs and being performed on a GPU with a homogeneous calculation procedure in a manner similar to that described in [9]. To achieve this goal, the Neumann boundary condition has to be introduced into Eq. 5. The Neumann boundary condition has the following form [25]:

$$\frac{\partial p}{\partial t} + \frac{c}{\beta} \nabla_n p = 0, \quad (7)$$

where β is the specific acoustic admittance of the surface, and ∇_n denotes the gradient calculated along the vector normal to the boundary to which the Neumann condition

is applied. After this substitution, the following equation is obtained:

$$\begin{aligned} p_{i,j,k}^{n+1} = & \left(1 + bk_{i,j,k} + \frac{\alpha_A}{2} T \right. \\ & \left. + \alpha_B T^2\right)^{-1} \left(\lambda^2 \left[p_{i+1,j,k}^n \right. \right. \\ & + p_{i-1,j,k}^n + p_{i,j+1,k}^n + p_{i,j-1,k}^n \\ & + p_{i,j,k+1}^n + p_{i,j,k-1}^n \left. \right] \\ & + \left[2 - k_{i,j,k} \lambda^2\right] p_{i,j,k}^n \\ & + \left[bk_{i,j,k} - 1 - \frac{\alpha_A}{2} \right] p_{i,j,k}^{n-1} \right), \end{aligned} \quad (8)$$

where: $k_{i,j,k}$ is a coefficient, which has a different value for each boundary condition. For free space, it is equal to 6, for the proximity of a single boundary plane, it is equal to 5, for the proximity of two boundary planes, it is equal to 4, and for the proximity of three boundaries – it is equal to 3. Coefficient $bk_{i,j,k}$ is calculated as $(6 - k_{ijk}) \lambda \beta / 2$. The interpretation and calculation of k_{ijk} , and $bk_{i,j,k}$ is the same as in [9].

Eq. 8 is a form of the discretized wave equation for a lossy propagation media that can be employed for FDTD-based computations on a GPU employing PMLs to obtain anechoic conditions. To further simplify the computation procedure, it is possible to assume that $\alpha = \alpha^*$. This means that $\alpha_A = \alpha (\rho c^2 + 1/\rho)$, and $\alpha_B = c^2 \alpha^2$. This allows only one set of damping factor values, which are related to α , to be passed to the GPU processor. This reduces the amount of data needed to be sent during the program initialization phase. Such a form of the equation is employed in the simulations presented further on to obtain the final results.

In practice, introducing a lossy term into a wave equation means the need for an additional matrix used in the computation process. The α parameter characterizes a local loss factor, and if it is subject to rapid change, for instance, from 0 in areas that are not lossy to the value of 0.15 in the area of PML, it will cause a reflection. Due to this fact, it is necessary to introduce a matrix of α coefficient values, which gradually increase starting from the beginning of the PML layer up to a maximum level at the boundary of the domain. An example profile used for this purpose can have the following form [10]:

$$\alpha = \alpha_{max} \left| \frac{x - x_0}{x_{max} - x_0} \right|^n, \quad (9)$$

where: x_0 is the initial point in the PML, x_{max} is the last point in the PML, α_{max} is the maximum value of damping eventually reached in point x_{max} , and n is a number between 2 and 3.

An example of the FDTD-based simulation employing PML to eliminate reflections from the boundaries of the computational domain is shown in Fig. 1. This type of simulation is useful for evaluating such properties of acoustic diffusers as the reflection and diffusion coefficients.

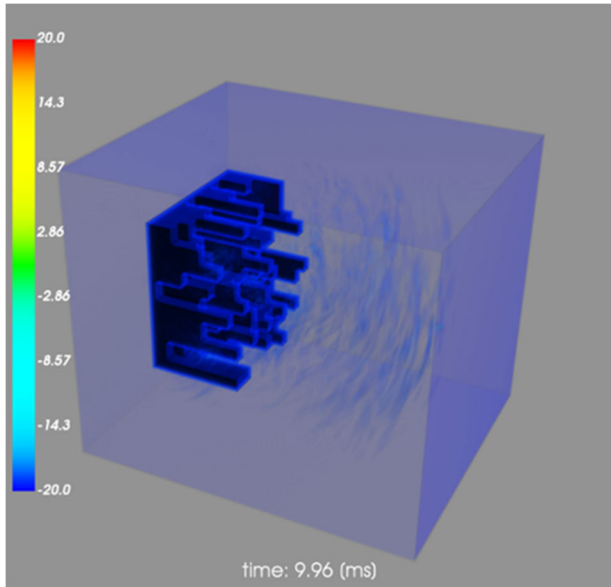


FIGURE 1. Simulation of acoustic pressure wave propagation in a closed room in which PML is used to attenuate reflections from the boundaries of the computational domain.

III. DESIGN OF SCHROEDER DIFFUSER PROPERTIES

A Schroeder diffuser is an acoustic device that can be defined by a matrix of integers greater than or equal to zero and a set of several scalar parameters with real values. Such a system consists of a one-dimensional series or a two-dimensional well grid. In the first case, such a diffuser affects the sound field only in one plane. In the second, in two perpendicular planes. The diffuser wells have different depths and are separated from each other by thin walls [10], [11]. The proportions of the depth of the wells can therefore be presented either as a series of values if the diffuser interacts with the sound field only in one plane or a matrix of values if it interacts with it in two planes. Examples of one-dimensional and two-dimensional geometries of an acoustic diffuser are shown in Fig. 2. The scalar parameters, which are also part of the description of the acoustic diffuser, include a factor for converting the ratio of the depth of the wells to their target depths and the physical dimensions of the wells.

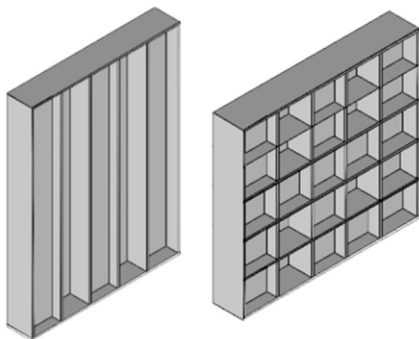


FIGURE 2. Schroeder diffusers designed for one plane (left side) and two planes (right side).

IV. MATH DESIGN OF SCHROEDER DIFFUSER PROPERTIES

A. STATE-OF-THE-ART DESIGN METHODS

The methods that determine the geometries of Schroeder diffusers are based on the application of pseudo-random number generation techniques to create sequences of numbers determining the relative proportions of the diffuser cavity depths. The properties of the scattering system vary depending on what type of pseudo-random sequence is used to design its geometry. Examples of frequently proposed sequences used for this purpose are the MLS (maximum-length sequence), PRD (primitive root diffuser), and QRD (quadratic residue diffuser) sequences [11]. In a diffuser geometry designed through a process based on QRD sequences, the ratio of the proportional depth of the n -th well marked by $W_{d,QRD}$ is given by the formula [11]:

$$W_{d,QRD}(n) = n^2 \bmod \xi, \quad (10)$$

where: n is consecutive integers greater than or equal to zero, and ξ is the selected prime number.

For PRD diffusers, the ratio of the proportional depth of the n -th well is given by the formula:

$$W_{d,PRD}(n) = g^n \bmod \xi, \quad (11)$$

where: g is the smallest primary root of the prime number ξ .

In the final step, the relative well depth factor is converted into the physical depth proportionally to the consecutive values of the given pseudo-random sequence.

It is important to note that Schroeder diffusers can be implemented in two manners – with or without physical barriers (walls) between the wells. In this paper, the variant without the wells is considered, as it is easier to manufacture. The following parameters can fully define any 2D Schroeder diffuser:

- matrix of integer numbers denoting the height of each Schroeder diffuser well,
- physical dimensions of a single segment making a well (width, length, and height),
- maximum number of segments per well,
- dimensions of the diffuser (in segments/wells, i.e., 10 segments by 10 segments),
- the thickness of the back part of the diffuser on which the wells are positioned.

Visualization of such a description method applied to an example of a borderless 2D Schroeder diffuser is shown in Fig. 3. It was employed in the course of all numerical and physical experiments. It can be applied to bordered and borderless designs. In the case of designs with walls, additional information about the thickness of the walls has to be included in the description. The definition of the diffuser structure presented in this Section is especially useful for algorithms optimizing diffuser designs, especially if the goal is the maximization of the diffusion and scattering coefficients. A detailed method for computer-based optimization of the aforementioned acoustic diffuser parameters is presented in the next subsection.

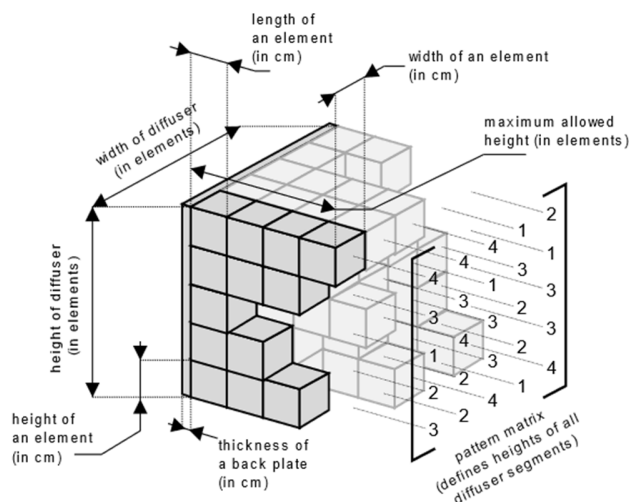


FIGURE 3. Depiction of all dimensions and other data crucial for an unambiguous definition of the Schroeder-type diffuser simulated in the experiments.

B. OPTIMIZATION-BASED METHODS

1) GENETIC ALGORITHM

In addition to design methods based on pseudo-random sequences, there are also more advanced methods for designing acoustic diffusers. For some problems, an optimization process may be the most viable way of obtaining satisfactory results in fields of knowledge such as acoustics and fluid mechanics. An example of such a task may be a design of an acoustic horn with specific properties [26], a structure of rotating turbines [27], or even a biomedical problem of designing an aortic valve [28]. Most of those approaches employ some form of computer simulation; many of them are based on the finite element method, which is used to calculate the optimized solution quality metric. This approach may also be applied to the problem of automatically designing an acoustic diffuser. The metric is only one part of the optimization-based approach to automated acoustic diffuser design. The second element is an optimization algorithm. An example of a popular optimization algorithm employed for automated design tasks is a genetic algorithm [29], [30]. When designing acoustic diffusers using a genetic algorithm, it is necessary to define the so-called genome. For diffusers, it may be defined as a matrix containing proportional cavity depth coefficients and a range of values that can be a number representing the acceptable cavity depth coefficient. First, a population of N_i random individuals consistent with the given genome is generated. Individuals may be subjected to crossing-over and mutation operations. In the algorithm examined in the experiment conducted, crossing-over involves randomly swapping columns or rows of two diffuser designs. Whenever the swap involving columns or rows is random, the probability of choice is 0.2. Then the algorithm iterates through rows or columns and converts them between projects with a probability of 0.2. The mutation operation

also involves iterating the algorithm randomly through design rows or columns and changing the cavity depth coefficient value by a random, non-zero integer from -1 to 1 with the exclusion of zero. The value of N_i used in the successive parts of this work is assumed to be equal to 36. The genetic algorithm is an iterative process. Within each iteration, a so-called generation of solutions is tested. The result of the test is the value of the fitness function, which is a parameter optimized by the genetic algorithm. It is assumed that the value of this parameter is maximized; however, in the case of the minimization task, it is enough to multiply the given fitness function by -1, which results in the task of minimizing the value of the function changing into the task of maximizing its value. The exact definition of the fitness function depends on the definition of the problem to be solved by the genetic algorithm. After each iteration, new diffuser designs are evaluated by calculating a fitness function that determines how much they improve the acoustics of the room in which they are applied.

In the case of acoustic diffusers, the task may relate to maximizing the diffusion coefficient, or minimizing the standard deviation of the frequency response at a given point of the tested room. The list of best diffusers is updated based on the final ranking list, and the process of crossing-over, mutation, and calculation of the ranking is performed again. Interruption of this process can occur either automatically, e.g., after reaching the desired value of the optimized coefficient characterizing the diffuser design (e.g., autocorrelation diffusion coefficient), or, for example, after a predetermined number of algorithm epochs has been carried out.

2) DEEP POLICY GRADIENT

Reinforcement learning algorithms are a group of machine learning methods for tackling problems involving interaction with the environment and managing the process of acquiring knowledge from such an interaction. A reinforcement learning algorithm often takes the form of a so-called agent that interacts with a given, often unknown environment by taking actions. The number of actions available can be limited but can also be infinite. Sometimes the action space can even be continuous. The outcomes of the actions undertaken by the agent are evaluated, and feedback information about the quality of a given action is returned to the agent in the form of a so-called reward. An approach based on reinforcement learning may be employed for the unsupervised design of neural network structures [31] or playing computer games [32]–[35]. A reward is the main signal used for training the algorithm to choose the best actions. It is often a numeric value and is maximized by the agent in the process of learning and interacting with the environment. The definition of such a reinforcement learning process is depicted in Fig. 4 [15].

In reinforcement learning, a common problem is taking into account the value of future states, which may not necessarily happen but may yield a future reward. A common

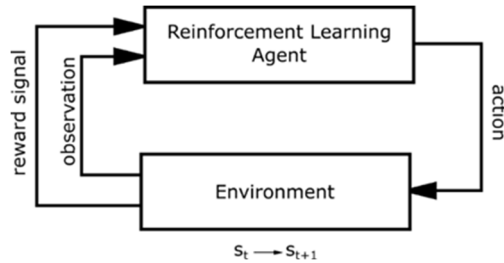


FIGURE 4. A definition of reinforcement learning problem – the agent interacts with the environment by taking actions. In return, it obtains feedback in the form of the reward signal and observation about the next state after the current action.

practice is introducing a so-called discount factor to estimate future rewards that may be obtained after taking a particular action. The expected reward in such a case is equal to:

$$E(r) = r_t + \sum_{i=1}^N \gamma^i \cdot r_{t+i} \quad (12)$$

where: $E(r)$ is expected reward to be obtained by the agent, r_t is a reward achieved by the agent in the current step of the interaction with the environment, γ is a discount factor, $\gamma \in (0, 1)$, and r_{t+1} is a reward obtained by the agent in the next steps of interaction.

An example of a deep reinforcement learning method that may be used to optimize the properties of acoustic diffusers is a deep policy-gradient (DPG) algorithm, which employs a deep neural network as part of the agent. In this case, the output of the network is a probability distribution of actions to be taken by the agent. In the case of a policy-gradient neural network, the network approximates this so-called policy function, which maps the current state of the environment to the choice of action undertaken by the agent. The policy gradient ascent algorithm, the objective of which is maximizing the reward gained by the policy gradient algorithm, is employed in such a case. Weights of the network (denoted by w) are updated by a gradient ascent algorithm. The gradient is calculated with the use of the following formula:

$$\nabla_w \pi_w(s, a) = \pi_w(s, a) \nabla_w \log(\pi_w(s, a)), \quad (13)$$

where: ∇_w denotes calculation of gradient with respect to the weights w of a neural network used to approximate the policy function denoted as $\pi_w(s, a)$.

This approach can be implemented with standard libraries allowing the implementation of gradient-descent training libraries. A drawback of the policy gradient algorithm is the fact that an update for the policy neural network may only be applied after a full episode of interactions with the environments. Therefore, they often converge at a lower speed. On the other hand, policy-gradient methods make it easier to design solutions that rely on a probability distribution of actions to be taken and thus do not require explicit complication such as ϵ -greedy strategies to encourage off-policy exploration of the environment. They also consist of less complex neural networks and do not require additional data structures such as replay memory, which further simplifies their practical implementation.

V. DESIGN OF THE NUMERICAL BENCHMARK EXPERIMENT

A standard method of evaluating the performance of a diffuser is a measurement of an autocorrelation diffusion coefficient. Therefore, the numerical experiment was designed to compare two investigated optimization-based approaches (namely, one based on a deep policy gradient and one based on a genetic algorithm) to designing Schroeder diffusers in terms of the average autocorrelation diffusion coefficient. We decided not to include classic design methods based on pseudo-random sequences. Those methods would not yield designs of diffusers of the same dimensions as the two optimization-based methods. Instead, we introduced another baseline method based on the generation of diffuser designs in a random manner and the choice of the best design obtained in such a way. The autocorrelation diffusion coefficient can be calculated based on a polar response of a given acoustic diffuser. This is also a common way of characterizing the scattering properties of diffusers for the purpose of selecting appropriate diffusers for the given application. Therefore, it is crucial to determine if genetic and reinforcement learning algorithms can optimize the diffusion coefficient and design diffusers better than those obtained by simple randomization of the lengths of Schroeder diffuser segments.

Additionally, we wanted to check if the deep policy gradient would work better if the starting design improved by the deep learning algorithm is random or if it is the best historical design achieved by the iterative process so far. A simulation employing PML (perfectly matched layers) layers should be used to achieve this goal, as anechoic conditions are necessary for both simulation-based prediction and measurement of the diffusion coefficient. The experiment meant to investigate the performance of the aforementioned design principles, employing the four aforementioned algorithms, is schematically depicted in Fig. 5.

Four methods of acoustic diffusers design were investigated:

- random selection of the length of diffuser pattern segments,
- optimization of the length of diffuser pattern segments employing genetic algorithms,
- optimization of the length of diffuser pattern segments employing a deep policy gradient algorithm which optimizes patterns generated randomly,
- optimization of the length of diffuser pattern segments employing a deep policy gradient algorithm that optimizes patterns sampled from the 10 best designs generated by the algorithm at the time of the creation of the initial pattern.

To speed up the computations, the action space was designed to make it possible for the agent to change multiple elements at once. Visualization of an example action after such a modification is depicted in Fig. 6. The structure of the neural network used to predict actions in such an action space is shown in Fig. 7.

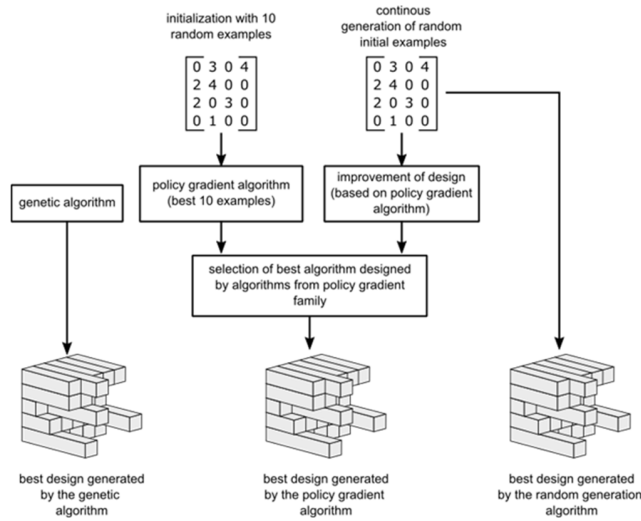


FIGURE 5. Diagram of the experiment conducted to test the simulation-based optimization and compare the outcomes of this method with a process of acoustic diffuser design based on a random selection of the diffuser segment height.

The designs in the experiment have equal width and height patterns, being 10 segments, and for each segment, three decisions can be made. The decisions are:

1. decreasing the element height by 1,
2. leaving the element unchanged,
3. increasing the element height by 1.

As only heights from 0 up to 10 are accepted, the values are trimmed after modifications introduced to the pattern by the neural network to avoid negative segment heights and heights greater than 10. This means that there are $3^{10 \cdot 10} = 3^{100}$ possible actions in the action space. As denoted in Fig. 7, the input to the DPG neural network was also modified; additional channels were added. Instead of providing only the information about the diffuser pattern, the neural network also takes:

1. 2-dimensional FFT of the pattern,
2. 2-dimensional cepstrum of the pattern,
3. autocorrelation of the pattern interpolated from the shape of (19, 19) to the shape of (10, 10).

In the case of autocorrelation, the spline-based interpolation implemented in the *RectBivariateSpline* class from the Python *scipy.interpolate* package was used. The version number of the SciPy library employed in this experiment was 1.5.4. The cepstrum of the Schroeder diffuser pattern was calculated according to the formula below:

$$X_{cepstrum} = FFT2D (|FFT2D (X)|) \quad (14)$$

where: \mathbf{X} is a matrix containing the pattern of the Schroeder diffuser, $FFT2D$ is a 2-dimensional Fast Fourier transformation, and $\mathbf{X}_{cepstrum}$ is a spatial cepstrum of \mathbf{X} .

The order of the spline used for interpolation in both the x and y axes was 5. Such an approach allowed the neural network to estimate the properties of the diffusers by analyzing

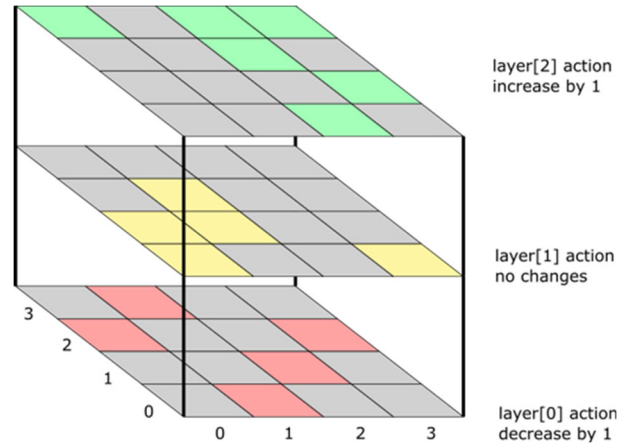


FIGURE 6. Modified structure of an action being the effect of inference carried out by the DPG neural network.

their shape in terms of spatial frequencies and autocorrelation. The use of cepstrum allowed for easier sensitization of the neural network to take into account the harmonicity of the spatial frequencies present in the design.

Two approaches to DPG-based optimization were investigated:

4. an approach in which a neural network tries to improve the wideband autocorrelation diffusion coefficient of randomly generated diffuser patterns,
5. an approach, in which, through the whole process of training, the 10 best historical designs are retained and updated. A neural network tries to improve a design randomly chosen from the aforementioned set of best diffuser designs.

Neural networks are trained in a loop of interaction between the network and the environment. The algorithm of training is presented in Fig. 8. In this case, the environment is an FDTD simulation of the diffuser interaction with a band-limited Gaussian impulse obtained in a manner described in [10]. Due to the necessity of limiting the numerical dispersion, the bandwidth of the simulation was limited to 4 kHz – this is the maximum frequency of the exciting impulse employed in the simulation.

Each training step was performed for 10 epochs. The Adam algorithm was employed for learning rate optimization; the initial learning rate of the Adam algorithm was set to 0.001. The momentum parameter of batch normalization layers was set to 0.95. The discount factor was set to 0.99.

A reward signal calculated for each step of interaction between a particular implementation of the neural network architecture from Fig. 7 was calculated as follows:

$$\begin{aligned} reward [n] = & \frac{1}{2} (d_{\psi,n,xy} [n] - d_{\psi,n,xy} [n - 1]) \\ & + \frac{1}{2} (d_{\psi,n,yz} [n] \\ & - d_{\psi,n,yz} [n - 1]) \end{aligned} \quad (15)$$

where: $reward [n]$ is a reward signal value for step n $d_{\psi,n,xy} [n], d_{\psi,n,yz} [n]$ are the normal incidence autocorrelation

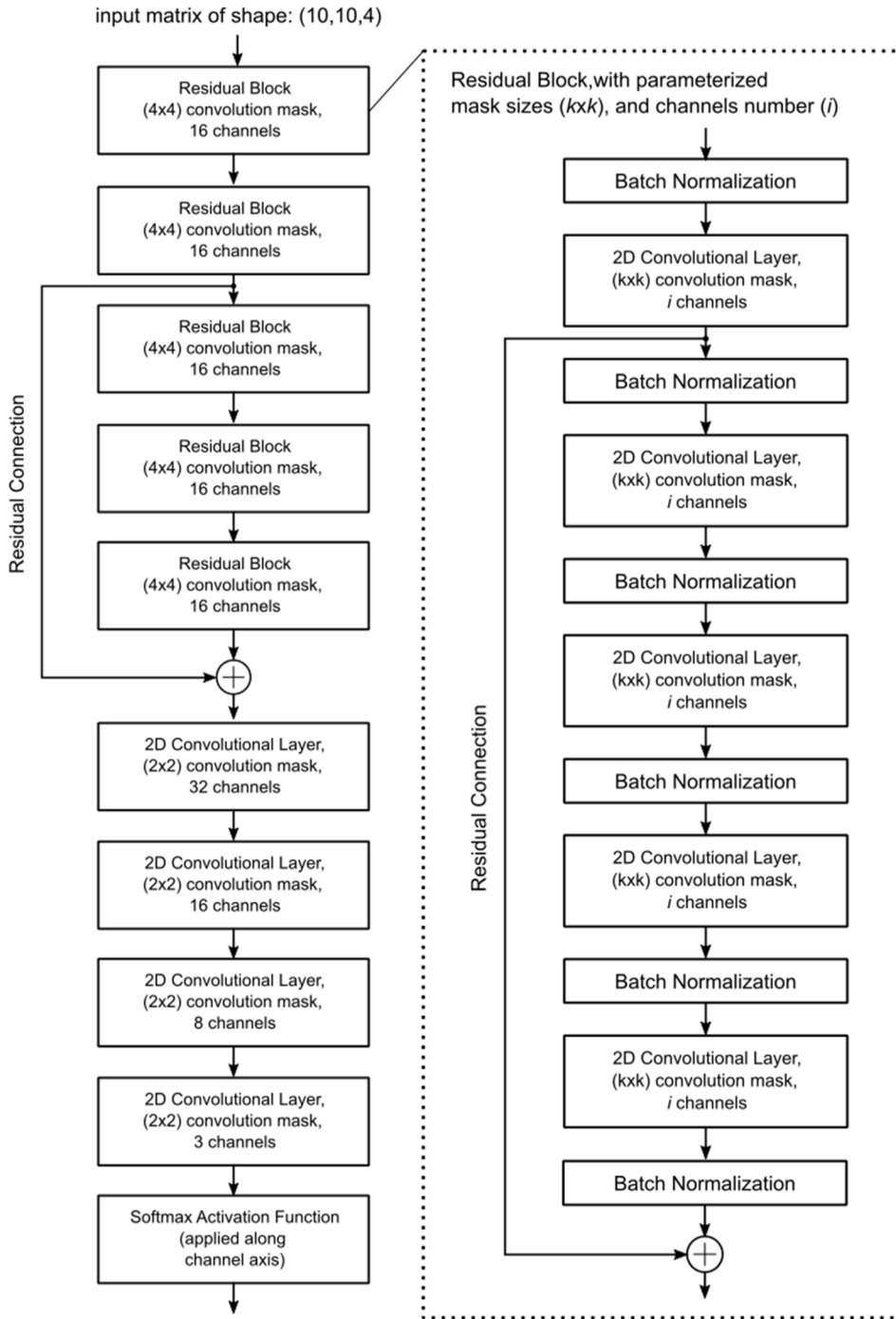


FIGURE 7. Structure of the DPG neural network that learns and performs a strategy of a reinforcement learning agent optimizing designs in the experiment. The activation function of all layers but the last, softmax one, is a parameterized rectified linear unit (PReLU).

diffusion coefficients obtained in the current step of optimization calculated after application of changes obtained from a neural network for the xy, and yz planes of the diffuser, respectively, and $d_{\psi,n,xy} [n - 1]$ and $d_{\psi,n,yz} [n - 1]$ are the normal incidence autocorrelation diffusion coefficients obtained for a diffuser design before applying the changes

obtained from the neural network for the xy, and yz planes of the diffuser, respectively.

VI. RESULTS

The designed and simulated diffusers consisted of 100 segments and had the shape of 10×10 segments.



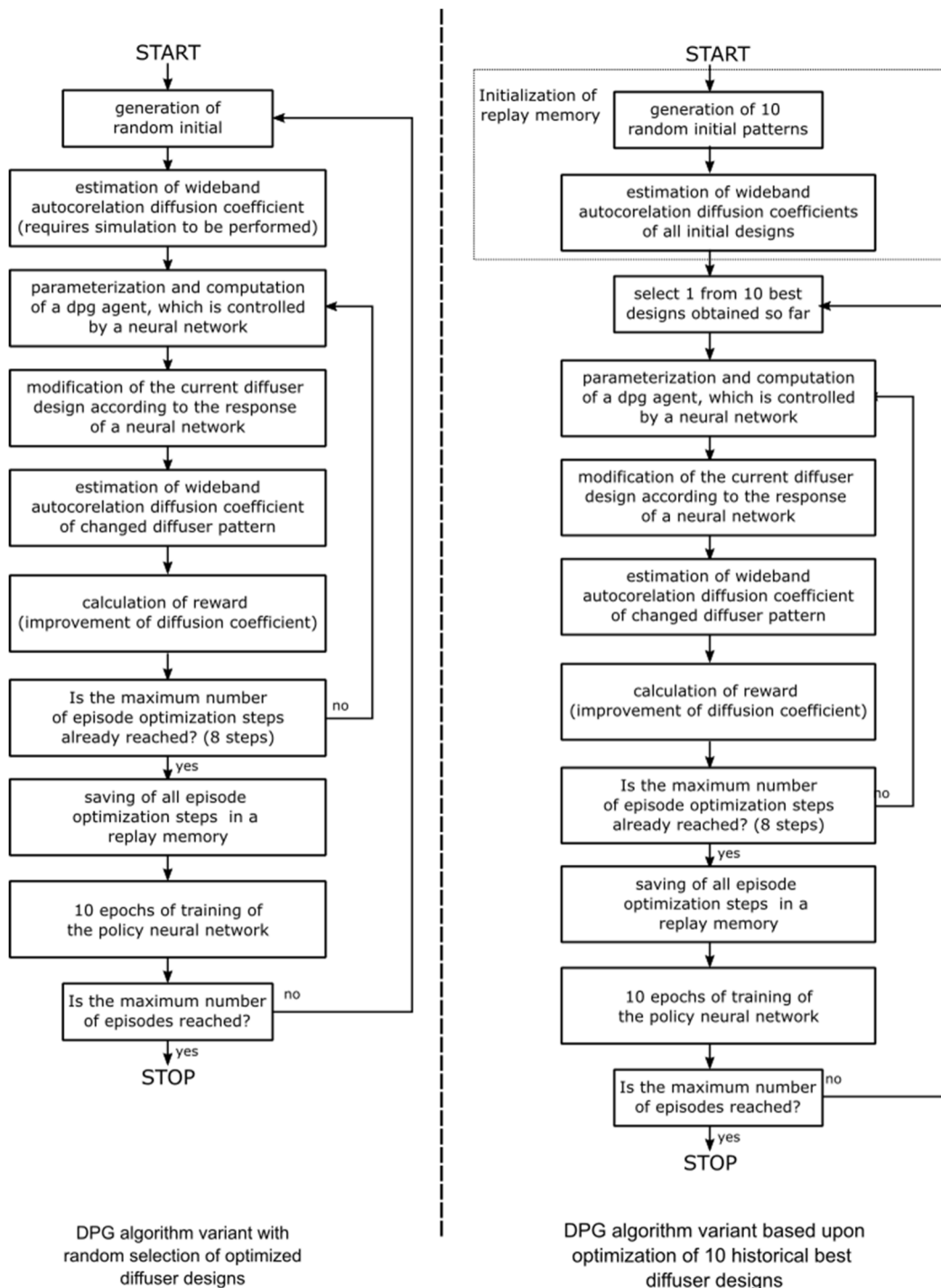


FIGURE 8. Training procedures for reinforcement learning agents using the neural network architecture described in this section.

It was assumed that behind the matrix of segments, there is a plate having the same reflective properties as segments in front of it. Therefore, it is possible for a single segment to have a length of 0 cm, which practically means that in its place, there is no segment, just the surface of the plate. Each segment can have a length within the range from 0 to 10 elements. The maximum length of a diffuser was set to

30 cm; therefore, the height of a single element is 3 cm which translates to allowed segment heights in the range from 0 cm to 30 cm.

To speed up computations and limit the numerical dispersion, the bandwidth of the simulation was limited to 4 kHz by using a band-limited Gaussian pulse. The sampling rate of the simulation was 50.2 kHz. This is the minimum sampling

rate that provides sufficient spatial resolution for simulating lengths as small as 3 cm. The spatial resolution was equal to 1.1 cm and was calculated with an assumption that the Courant number is equal to $\sqrt{1/3}$. The temporal duration of the simulation, and thus the duration of the obtained impulse responses used to calculate the diffusion coefficient, was 15 ms.

In terms of the properties of the propagation medium, the temperature of the air was assumed to be equal to 25 °C, atmospheric pressure was considered to be equal to 1000 hPa, and relative humidity of the air was assumed to be equal to 50%. Therefore, the speed of sound in the propagation medium was 347.43 m/s, and the characteristic impedance of the air was equal to 403.52 Rayls. As the simulations were intended to replicate outcomes obtained in anechoic conditions, PML was applied to the borders of the computational domain. The PML had a thickness of 30 elements, and the maximum damping factor was 25000. Admittance of rigid materials present in the computational domain was assumed to be equal to 10^{-10} . The semicircular pattern of pressure measurement points used to calculate the diffusion coefficient had a radius of 0.95 m. The source was positioned at a 2 m distance from the diffuser. The incidence angle of the acoustic wave on the acoustic diffuser was 90° (perpendicular incidence). The dimensions of the domain (with PML and object margins) were 2.9m × 3.4 m × 2.9 m.

Example graphical depictions of reward signals and achieved diffusion coefficients obtained from the policy gradient optimization process are shown in Figs. 9 and 10. Each of the graphs is related to a different implementation of a neural network (for instance, they are initialized with different sets of random weights). Therefore, each implementation of a neural network is referred to as an “agent” to emphasize this fact. For the algorithm using the random pattern as input for the optimization, there were active 20 agents; for the algorithm performing optimization of 1 of the 10 best patterns, there were 23 agents. The training lasted for approximately 15 hours. The number of episodes of training for each agent varied and was in the range between 60 up to 110. This variability resulted from the parallel training of agents on multiple workstations equipped with different GPU cards

characterized by different performances. For this experiment, 2x RTX Titan, 3x RTX 2080 Ti, and 2x RTX 2060 graphic cards were used. The aforementioned graphic cards were used to carry out both the training of the neural networks and the simulation-based prediction of the diffusion coefficient of the diffusers generated by the algorithms.

To compare the neural network performance in terms of optimization capabilities with other heuristic optimization methods, a genetic algorithm was also employed to design diffusers with the same design specification as those generated by the deep policy gradient algorithm. This algorithm was used to carry out 3 instances of genetic algorithm design processes. Each of them employed a population of 30 diffusers. For each of those groups, 18 generations of diffusers were generated.

Histograms of diffusion associated with all of the designs generated by the two optimization processes based on the policy gradient and the genetic algorithm are presented in Fig. 11. To investigate if the optimization outcomes are better than a random selection of the diffuser segment length, a histogram of diffusions provided by randomly generated diffusers also is shown. These random designs are obtained from the DPG algorithm with a random initial design. These initial, purely random designs from the beginning of all episodes can be utilized as a baseline for all designs optimized by the DPG and genetic algorithm.

Fig. 11 shows that the best designs in terms of diffusion coefficients calculated based on the FDTD simulation were those generated by the DPG algorithm with input patterns selected from the 10 best historical designs. The second best algorithm was the genetic algorithm. If the DPG method was fed with inputs obtained in a random manner, a histogram of the obtained diffusion coefficients was barely different from a purely random choice of diffuser segment lengths.

To find out if the median differences visible in Fig. 11 are statistically significant, statistical tests were performed. For all tests, a significance level of 0.05 was assumed. First, Levene’s test for equality of variance of all algorithm-related distributions was carried out. The test statistic was equal

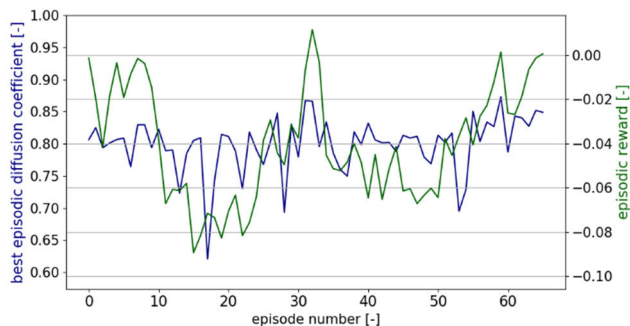


FIGURE 9. Results obtained from agent no. 10, which was fed with random input diffuser designs to be improved over each episode.

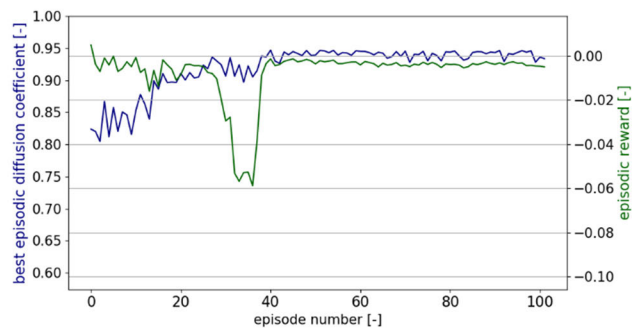


FIGURE 10. Results obtained from agent no. 18, for which a diffuser chosen from the 10 best designs encountered by a group of agents in the optimization process was selected as a starting design.

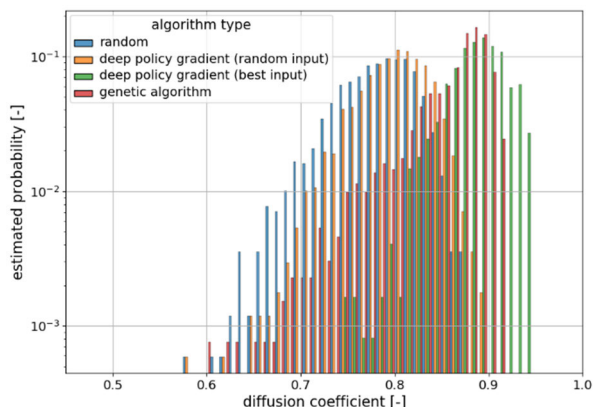


FIGURE 11. Histograms depicting probability distributions of creating the design associated with a specified diffusion coefficient. Four algorithms are shown: DPG with randomly generated input diffusers designs, DPG with input designs selected from the 10 best historic designs, genetic algorithm, and random selection of diffuser segment lengths.

to 41.9, thus the p -value is less than 10^{-3} . Therefore, one must conclude that the variances of the estimated probability distributions shown in Fig. 11 are not equal. Therefore, the ANOVA test for equality of means of distributions could not be carried out. Instead, a nonparametric alternative to ANOVA had to be used. In this case, the Kruskal-Wallis test for equality of medians can be employed. The statistic of this test for the data investigated was equal to 4370.24, and therefore the p -value, in this case, was also less than 10^{-3} . This means that at least one pair of medians of distributions visible in Fig. 11 is not equal. To find out which one this concerns, a post-hoc test had to be carried out. In the case of the Kruskal-Wallis test, a Dunn's test can be employed. In the case of the data in question, all p -values of Dunn's test were also less than 10^{-3} . Therefore, the p -value matrix is not shown here, and one can conclude that all differences visible in Fig. 11 are statistically significant. The medians calculated for the data from Fig. 11 are contained in Table 1.

The DPG algorithm with the 10 best input designs was found to be the best in terms of optimizing diffuser designs on the basis of FDTD simulation. This confirms that reinforcement learning can be used to design acoustic diffusers through simulation-based optimization. The DPG algorithm provided designs with a diffusion coefficient median significantly higher than any other design method tested. In the course of numerical experiments, the following numbers of repetitions for each calculation scenario were carried out:

- best design from a pool of random designs – 1692 patterns were generated; therefore, the number of designs generated by the DPG with a random initial design is also 1692,
- genetic algorithm generated a pool of 1620 diffusers designs,
- deep policy gradient with input design selected from the 10 best historic designs resulted in the creation of 1816 designs.

TABLE 1. Medians of diffusion coefficients averaged over the xy , and yz planes for designs generated by four algorithms investigated in the second experiment.

Algorithm	Median of the averaged diffusion coefficient
deep policy gradient (best of 10 input)	0.896
genetic algorithm	0.875
deep policy gradient (random input)	0.803
random pattern selection	0.786

VII. DISCUSSION

The scope of our work focused on employing a numerical acoustic wave propagation model and machine learning algorithms. The aim was to create an autonomous program capable of designing acoustic diffusers with optimized values of given metrics related to their acoustic properties. For optimization, an autocorrelation diffusion coefficient was chosen. However, it should be stressed that any other metric that can be predicted by a numerical simulation can be selected for such kind of an optimization. The use of computer simulation for the prediction of acoustic wave propagation makes it possible to also calculate the fitness function based on the scattering coefficient or the uniformity of the frequency response of a hypothetical shoebox-type room in which the diffuser undergoing the optimization process is placed.

The investigation involving both the genetic and the reinforcement learning algorithms showed that they could optimize the acoustic properties of the designed diffusers (in terms of an autocorrelation diffusion coefficient) based on feedback generated by the simulation. Additionally, all optimization methods achieved better statistically significant results than the ones employing classical approaches, which were based on the use of pseudo-random sequences and on the random selection of the well-depths of Schroeder diffusers. It was observed that the deep policy gradient algorithm could achieve better statistically significant performance by optimizing the design of an acoustic diffuser, even if the action space was as large as 3^{100} possible actions. Moreover, it was found that especially the design obtained from the deep policy gradient method provided a more stable diffusion coefficient across the bandwidth in which the diffuser has a non-negligible effect on the simulated acoustic field.

An interesting observation made in our research is that in case of the deep policy gradient, it is necessary to provide relatively good starting points for the optimization to achieve satisfactory sequential optimization results. Even if the interaction between the algorithm and simulation is relatively long and takes hundreds of episodes, starting with just random diffuser designs does not yield better results than those provided by the genetic algorithm. Our hypothesis regarding this observation is that the improvement of already high-scoring designs involves a very different type of change than just the improvement of relatively poor designs. We further hypothesize that the differences between those kinds of changes are sufficiently pronounced to impose a

TABLE 2. A brief comparison of key advantages and disadvantages of acoustic simulation and optimization algorithms recalled in our study for designing acoustic diffusers.

Optimization algorithm	Simulation algorithm	advantages	disadvantages
genetic algorithm	FDTD without reflections suppression (Webb & Bilbao) [9]	the most simple approach requiring the least amount of computational power; the genetic algorithm is also usually a significantly faster approach than one based on deep reinforcement learning	requires taking additional measures to ensure that the reflections from the computational boundary will not affect the optimization outcome; genetic algorithm yields worse-performing designs than the proposed reinforcement learning-based method
reinforcement learning	FDTD without reflections suppression (Webb & Bilbao) [9]	simple implementation and reduced computational power is required to carry out the simulation; the optimization algorithm can interact with designs obtained from the simulation utilizing the deep reinforcement learning paradigm, which typically yields output diffuser designs of better quality in terms of the optimized metric	requires taking additional measures to ensure that the reflections from the computational boundary will not affect the optimization outcome; the reinforcement learning algorithm is usually more complex to be implemented and requires noticeable longer training than a single run of a genetic algorithm
genetic algorithm	FDTD with a PML layer (the method proposed)	no additional measures are needed to limit the influence of the reflections from the computation boundaries on the final computation outcome - a smaller computation grid can be used due to reduction of the reflections from the computational boundary, the optimization employing genetic algorithm requires less computational effort and is faster than deep reinforcement learning-based approach	more complicated and computation-heavy implementation of the simulation algorithm; genetic algorithm yields worse-performing designs than the proposed reinforcement learning-based method
reinforcement learning	FDTD with a PML layer (the method proposed)	no additional measures are needed to limit the influence of the reflections from the computation boundaries on the final computation outcome - a smaller computation grid can be used due to reduction of the reflections from the computational boundary; the reinforcement learning-based algorithm provides better solutions than the genetic algorithm	the most computationally expensive method, single run of the algorithm is the longest from the all presented in the table due to the increased complexity of both the simulation and optimization method

requirement that the input of the DPG algorithm has to be the best historical design, and not just a random starting design. This has a significant practical consequence, as it implies the non-feasibility of employing an already-trained neural network to carry out the DPG algorithm for improving a given acoustic diffuser design over the course of a single episode. It is more likely that a new training process would be required, as the neural network has to most likely significantly change its behavior during the diffuser optimization process.

It is also noteworthy that the novel set of equations derived in the course of our research, to simulate the propagation of acoustic waves in anechoic conditions, can also be utilized in other cases. The reduction of reflections present in simulation, based on the original set of the FDTD equations proposed by Webb and Bilbao [9], makes it possible to use our paradigm of simulation for predicting experimental outcomes

obtained in anechoic conditions. It requires a smaller computational grid, as acoustic wave reflections from the borders of the computational grid are attenuated. In the case of original equations proposed in [9], possibilities for elimination of reflections are limited. They can probably impose additional requirements in terms of the simulation structure such as increasing the size of the computational domain. This would allow obtaining temporal separation of the useful parts of the simulated impulse response containing signals reflected from the acoustic diffuser and of the one containing unwanted reflections from the boundary of the computational grid. Such an approach is associated with some apparent disadvantages such as the unnecessary use of computational power. Another drawback is the necessity of additional processing in order to eliminate unwanted reflections by trimming the impulse response obtained from such a simulation.

Finally, the choice of both the simulation and the optimization algorithms has its practical consequences in terms of the computational power needed for carrying out the simulation-driven optimization process. It also determines the complexity of the algorithm implementation and the computation time, which is significantly longer in the case of reinforcement learning optimization (tens of hours) if compared to the genetic algorithm (single hours required by the algorithm to find one of the top-performing solutions). Therefore, before deciding which approach to choose in a particular case, it may be advisable to consider all the advantages and disadvantages of all approaches to a simulation-driven optimization discussed in our paper and choose the best-fitted one for a given task. An example of such a comparison is presented in Table 2.

VIII. CONCLUSION

There are two key conclusions that can be drawn from our research—first, we succeeded in using the reinforcement learning algorithm to design the acoustic diffuser. It was possible to fine-tune this design method in order to achieve results with better statistical significance compared to genetic algorithms in terms of the diffusion coefficient. Results obtained from both the reinforcement learning- and genetic algorithm-based algorithms surpassed the design method involving the random choice of the Schroeder diffuser well-depth pattern.

The second achievement is a derivation of equations capable of simulating anechoic conditions employing the FDTD method, which allows fast computation of fitness and reward signals by using graphics unit processing (GPU). Moreover, a Python-based implementation of the numerical model was developed and tested for such a purpose. Simulation-based evaluation of diffuser properties in anechoic conditions may be a potentially useful tool, as several standard parameters that are used to characterize acoustic diffuser designs are derived from the frequency response of the diffusers.

REFERENCES

- [1] B. Hamilton and S. Bilbao, "FDTD methods for 3-D room acoustics simulation with high-order accuracy in space and time," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 11, pp. 2112–2124, Nov. 2017, doi: [10.1109/TASLP.2017.2744799](https://doi.org/10.1109/TASLP.2017.2744799).
- [2] I. Ibrahim, R. Silva, M. H. Mohammadi, V. Ghorbanian, and D. A. Lowther, "Surrogate-based acoustic noise prediction of electric motors," *IEEE Trans. Magn.*, vol. 56, no. 2, pp. 1–4, Feb. 2020, doi: [10.1109/TMAG.2019.2945407](https://doi.org/10.1109/TMAG.2019.2945407).
- [3] D. Li, D. I. W. Levin, W. Matusik, and C. Zheng, "Acoustic voxels: Computational optimization of modular acoustic filters," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016, doi: [10.1145/2897824.2925960](https://doi.org/10.1145/2897824.2925960).
- [4] A. Pilch, "Optimization-based method for the calibration of geometrical acoustic models," *Appl. Acoust.*, vol. 170, Dec. 2020, Art. no. 107495, doi: [10.1016/j.apacoust.2020.107495](https://doi.org/10.1016/j.apacoust.2020.107495).
- [5] J. Su, L. Zheng, and J. Lou, "Simulation and optimization of acoustic package of dash panel based on SEA," *Shock Vibrat.*, vol. 2020, pp. 1–14, Oct. 2020, doi: [10.1155/2020/8855280](https://doi.org/10.1155/2020/8855280).
- [6] H. Sun, S. Wang, S. Huang, L. Peng, Q. Wang, W. Zhao, and J. Zou, "3D focusing acoustic lens optimization method using multi-factor and multi-level orthogonal test design theory," *Appl. Acoust.*, vol. 170, Dec. 2020, Art. no. 107538, doi: [10.1016/j.apacoust.2020.107538](https://doi.org/10.1016/j.apacoust.2020.107538).
- [7] K. Kowalczyk and M. van Walstijn, "Room acoustics simulation using 3-D compact explicit FDTD schemes," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 1, pp. 34–46, Jan. 2011, doi: [10.1109/TASL.2010.2045179](https://doi.org/10.1109/TASL.2010.2045179).
- [8] X. Cao, Y. Cai, and X. Cui, "A parallel numerical acoustic simulation on a GPU using an edge-based smoothed finite element method," *Adv. Eng. Softw.*, vol. 148, Oct. 2020, Art. no. 102835, doi: [10.1016/j.advengsoft.2020.102835](https://doi.org/10.1016/j.advengsoft.2020.102835).
- [9] C. J. Webb and S. Bilbao, "Computing room acoustics with CUDA-3D FDTD schemes with boundary losses and viscosity," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 317–320, doi: [10.1109/ICASSP.2011.5946404](https://doi.org/10.1109/ICASSP.2011.5946404).
- [10] T. J. Cox and P. D'Antonio, *Acoustic Absorbers and Diffusers*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2017.
- [11] F. A. Everest and K. C. Pohlmann, *Master Handbook of Acoustics*, 5th ed. New York, NY, USA: McGraw-Hill, 2009.
- [12] N. Jiménez, T. J. Cox, V. Romero-García, and J.-P. Groby, "Metadiffusers: Deep-subwavelength sound diffusers," *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, Dec. 2017, doi: [10.1038/s41598-017-05710-5](https://doi.org/10.1038/s41598-017-05710-5).
- [13] M. A. Pogson, D. M. Whittaker, G. A. Gehring, T. J. Cox, R. J. Hughes, and J. A. S. Angus, "Diffusive benefits of cylinders in front of a schroeder diffuser," *J. Acoust. Soc. Amer.*, vol. 128, no. 3, p. 1149, 2010, doi: [10.1121/1.3463817](https://doi.org/10.1121/1.3463817).
- [14] N. Buduma and N. Locascio, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. Sebastopol, CA, USA: O'Reilly Media, 2017.
- [15] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
- [16] Y. Kobayashi and E. Aiyoshi, "Automatic core design using reinforcement learning," in *Proc. Amer. Control Conf.*, Jun. 2004, pp. 5784–5789, doi: [10.23919/acc.2004.1384779](https://doi.org/10.23919/acc.2004.1384779).
- [17] P. Varshavskaya, L. P. Kaelbling, and D. Rus, "Automated design of adaptive controllers for modular robots using reinforcement learning," *Int. J. Robot. Res.*, vol. 27, nos. 3–4, pp. 505–526, Mar. 2008, doi: [10.1177/0278364907084983](https://doi.org/10.1177/0278364907084983).
- [18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: <https://arxiv.org/abs/1603.04467>
- [19] F. Chollet. (2015). *Keras Deep Learning Software*. [Online]. Available: <https://keras.io>
- [20] D. R. Bergman, *Computational Acoustics: Theory and Implementation* (Wiley Series in Acoustics Noise and Vibration). Hoboken, NJ, USA, 2018, pp. 1–279, doi: [10.1002/9781119277323](https://doi.org/10.1002/9781119277323).
- [21] D. T. Blackstock, *Fundamentals of Physical Acoustics*. Hoboken, NJ, USA: Wiley, 2000.
- [22] S. Vandekerckhove, G. N. Wells, H. De Gerssem, and K. Van Den Abeele, "Automatic calibration of damping layers in finite element time domain simulations," 2016, *arXiv:1601.07941*. [Online]. Available: <https://arxiv.org/abs/1601.07941>
- [23] X. Yuan, D. Borup, J. W. Wiskin, M. Berggren, R. Eidsens, and S. A. Johnson, "Formulation and validation of Berenger's PML absorbing boundary for the FDTD simulation of acoustic scattering," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 44, no. 4, pp. 816–822, Jul. 1997, doi: [10.1109/58.655197](https://doi.org/10.1109/58.655197).
- [24] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, no. 2, pp. 185–200, 1994, doi: [10.1006/jcph.1994.1159](https://doi.org/10.1006/jcph.1994.1159).
- [25] J. Botts and L. Savioja, "Spectral and pseudospectral properties of finite difference models used in audio and room acoustics," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 9, pp. 1403–1412, Sep. 2014, doi: [10.1109/TASLP.2014.2332045](https://doi.org/10.1109/TASLP.2014.2332045).
- [26] S. Schmidt, E. Wadbro, and M. Berggren, "Large-scale three-dimensional acoustic horn optimization," *SIAM J. Sci. Comput.*, vol. 38, no. 6, pp. B917–B940, Jan. 2016, doi: [10.1137/15M1021131](https://doi.org/10.1137/15M1021131).
- [27] N. Van Dang, J. Jansson, M. Leoni, B. Janssen, A. Goude, and J. Hoffman, "Modelling of rotating vertical axis turbines using a multiphase finite element method," in *Proc. 7th Int. Conf. Comput. Methods Mar. Eng.*, Mar. 2017, pp. 950–959.
- [28] J. H. Spühler, J. Jansson, N. Jansson, and J. Hoffman, "3D fluid-structure interaction simulation of aortic valves using a unified continuum ALE FEM model," *Frontiers Physiol.*, vol. 9, p. 363, Apr. 2018, doi: [10.3389/fphys.2018.00363](https://doi.org/10.3389/fphys.2018.00363).

- [29] S. M. Thede, "An introduction to genetic algorithms," *J. Comput. Sci. Colleges*, vol. 20, no. 1, pp. 115–123, 2004.
- [30] R. Patraquim, L. Godinho, and P. A. Mendes, "Design and optimization of the sound diffusers using radial basis functions-based shapes and genetic algorithms," *J. Acoust. Soc. Amer.*, vol. 141, no. 5, p. 3664, 2017, doi: 10.1121/1.4987939.
- [31] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–18.
- [32] J. Choi, B. J. Lee, and B. T. Zhang, "Multi-focus attention network for efficient deep reinforcement learning," in *Proc. AAAI Workshop*, 2017, pp. 952–958.
- [33] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, "ViZDoom: A Doom-based AI research platform for visual reinforcement learning," in *Proc. IEEE Conf. Comput. Intell. Games (CIG)*, Sep. 2016, pp. 1–8, doi: 10.1109/CIG.2016.7860433.
- [34] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.
- [35] N. Usunier, G. Synnaeve, Z. Lin, and S. Chintala, "Episodic exploration for deep deterministic policies for Starcraft micromanagement," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–16.



BOŻENA KOSTEK (Senior Member, IEEE) is currently a Professor with the Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology (GUT), Poland. She has presented more than 600 scientific papers for journals and at international conferences. She also published three books related to multimedia applications. She has led a number of research projects. She has supervised more than 280 masters and engineering theses and 16 doctoral theses. Her main scientific interests include acoustics, psychoacoustics, multimedia, music information retrieval, cognitive and behavioral processing, and applications of machine learning to the mentioned domains. She is a Corresponding Member of the Polish Academy of Sciences. She is a fellow of the Audio Engineering Society and the Acoustical Society of America. She was a recipient of many prestigious awards for research, including those of the Prime Minister of Poland (twice), the Ministry of Science, and the Polish Academy of Sciences. She was the Editor-In-Chief of the *Journal of the Audio Engineering Society*, an Associate Editor of *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, and a Guest Editor of *JASA*.

• • •



ADAM KUROWSKI was born in Gdansk, in 1991. He received the M.Sc. degree from the Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, in 2015. The subject of his B.Sc. thesis was the implementation and evaluation of the performance of a virtual bass synthesis algorithm for small electroacoustic transducers. His M.Sc. thesis was associated with sound intensity measurement of an acoustic field distribution in the proximity of the human head and the numerical simulation of the observed phenomena. His scientific interests include the issues of design and implementation of digital acoustic signal processing, modern methods of the numerical simulation in acoustics, and the use of machine learning methods for acoustics-related implementations.