

# Noise Control for a Moving Evaluation Point Using Neural Networks

**Toshiki Maeda, Toshihiko Shiraishi**

Grad. Sch. of Environment and Information Sciences, Yokohama National University  
79-7 Tokiwadai, Hodogaya-ku, Yokohama 240-8501 Japan

E-mail: shira@ynu.ac.jp

**Abstract.** This paper describes the noise control for a moving evaluation point using neural networks by making the best use of its learning ability. Noise control is a technology which is effective on low-frequency noise. Based on the principle of superposition, a primary sound wave can be cancelled at an evaluation point by emitting a secondary opposite sound wave. To obtain good control performance, it is important to precisely identify the characteristics of all the sound paths. One of the most popular algorithms of noise control is filtered-x LMS algorithm. This algorithm can deliver a good result while all the sound paths do not change. However, the control system becomes uncontrollable while the evaluation point is moving. To solve the problem, the characteristics of all the paths are must be identified at all time. In this paper, we applied neural networks with the learning ability to the noise control system to follow the time-varying paths and verified its control performance by numerical simulations. Then, dropout technique for the networks is also applied. Dropout is a technique that prevent the network from overfitting and enables better control performance. By applying dropout for noise control, it prevents the system from diverging.

## 1. Introduction

Noise must be minimized at any time. The most effective way to reduce noise is eliminating noise sources as many as possible. For further measure, barriers are installed in order to prevent spreading noise. The barriers can suppress only high frequency noise. Noise control is an effective method for low frequency noise. Based on the principle of superposition, noise is reduced by emitting a sound wave with the same amplitude but with inverted phase to the noise from the other sound source called secondary source. There are two approaches to control noise. One is to minimize total sound power over whole space [1-2]. By identifying the characteristics of the sound field, the control is achievable as long as the noise is stationary. The other approach is to minimize noise at a local point [3-5]. Generally, filtered-x LMS algorithm is applied for this approach. LMS algorithm is one of steepest descent methods and coefficients of the filters are updated so as to decrease instantaneous squared errors. In filtered-x LMS algorithm, the transfer characteristics from the secondary source to the evaluation microphone must be identified by an adaptive filter in advance for a controller and the noise can be cancelled. However, in the case that the evaluation microphone moves, the transfer characteristics change and the system becomes uncontrollable when the identification error is not small. Thus, methods following the change of the characteristics are required. One of the methods is the virtual error method which is based on filtered-x LMS algorithm [3]. It requires three adaptive filters and all of them are updated by LMS algorithm simultaneously. Although the identification in advance do not have to be required in the



method, there are serious problem with the converging characteristics. The filters do not converge quickly so that it cannot follow the sudden change of the transfer characteristics [4].

In this paper, artificial neural networks were applied to noise control. Neural network is a model of signal processing circuit in the human brain. It has learning ability and adopts to a controlled system with varying parameters. In the conventional studies, it is clear that the neural network control system is effective on time-variant system [5-6]. When the evaluation microphone moves, the controller can identify the movement and also generate control outputs appropriately. However, higher frequency of noise and traveling speed of an evaluation microphone deteriorate the control performance. For further improvement, “dropout” technique was applied to the training process of the neural networks in this paper. Dropout is one of techniques that prevent the networks from overfitting [7]. In this way, the units of the network were randomly dropped out during the training. This paper also demonstrated the effects of dropout on the performance of the neural network controller.

**2. Numerical simulation method**

In this paper, all simulations were conducted in a free sound field as shown in figure 1. According to the theory of the wave equation, sound pressure denoted by  $P$  is a function of time and position, and determined by

$$\nabla^2 P - \frac{1}{c^2} \frac{\partial^2 P}{\partial t^2} = 0 \tag{1}$$

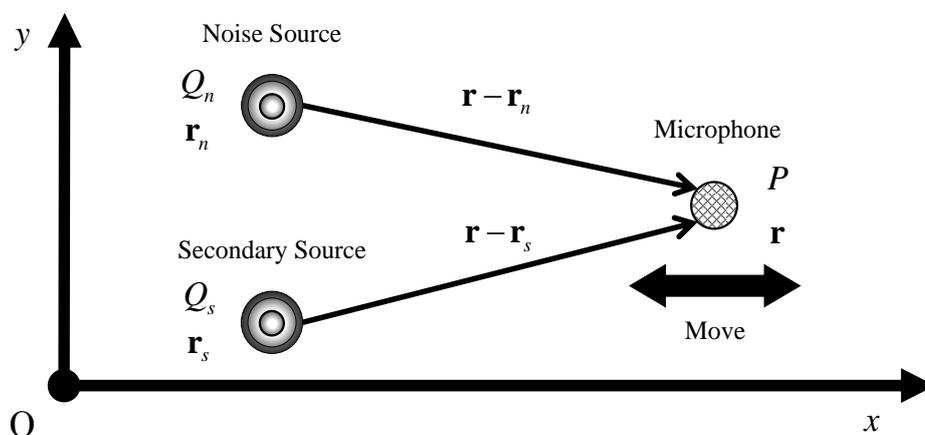
where  $c$  is sound speed. Assuming that the sound is spherical wave,  $P$  is rewritten as a function of distance from an origin position. Then, by using polar coordinate, equation (1) is transformed as

$$\frac{\partial^2(rP)}{\partial r^2} - \frac{1}{c^2} \frac{\partial^2(rP)}{\partial t^2} = 0 \tag{2}$$

where  $r$  is distance from an origin position. Solution of equation (2) is the sum of traveling wave and backward wave. If there is a monopole sound source on the origin and its output is  $Q(t)$ , the solution of equation (2) is given by

$$P = \frac{Q(t - r/c)}{r} \tag{3}$$

where the backward wave is ignored. Equation (3) means that sound pressure varies in inverse proportion to distance and that time lag of sound spread,  $r/c$ , is proportional to the distance between a sound source and a microphone.



**Figure 1.** Model of free acoustic field with a moving microphone position

Now consider the case where there are two sound sources and a microphone in the acoustic field. One source acts as a noise source and the other acts as a secondary source. In general, linear superposition is assumed. In this case, sound pressure at position  $\mathbf{r}$  is given by

$$P(\mathbf{r}, t) = \frac{Q_n(t - |\mathbf{r} - \mathbf{r}_n|/c)}{|\mathbf{r} - \mathbf{r}_n|} + \frac{Q_s(t - |\mathbf{r} - \mathbf{r}_s|)}{|\mathbf{r} - \mathbf{r}_s|} \tag{4}$$

where  $Q_n$  is sound from the noise source,  $Q_s$  is sound from the secondary source,  $\mathbf{r}_n$  is position vector of the noise source, and the  $\mathbf{r}_s$  is position vector of the secondary source, respectively. By substituting the position vector  $\mathbf{r}$  to equation (4), sound pressure can be determined. In simulation, time history of waveform from the sound source is saved at each time step, and time lag is discretized in order to be processed as integer multiple of time step.

### 3. Noise control algorithm using neural networks

#### 3.1. Neural network control system

A block diagram of the control system using neural networks is shown in figure 2. It is composed of two connected networks; a controller part and an identification part. The controller part produces control output and the identification part identifies the transfer characteristics. These parts are constructed by multilayer networks and connected in series. The identification and the control errors were calculated using the output of the identification part. The identification and the controller parts were trained so as to reduce the identification and control errors by the back propagation method [8], respectively. In this method, the errors determined by the output of the networks are reduced by the steepest decent method. The control error was determined by the difference between the identification part output and the desired value so that it propagates the controller part via the identification part. In the virtual error method, three filters are updated. On the other hand, two filters are trained in the neural network control system.

Figure 3 is a block diagram of the proposed neural network control system in detail. It may be easy for the networks to be trained by using time history of input signals because the controller should identify the time delay of sound transfer. By using the time history of the reference signal, the control output denoted by  $y(k)$  was calculated as [8]

$$y(k) = \sum_{i=0}^{N-1} [W_i^C \cdot P_R(k-i)] \tag{5}$$

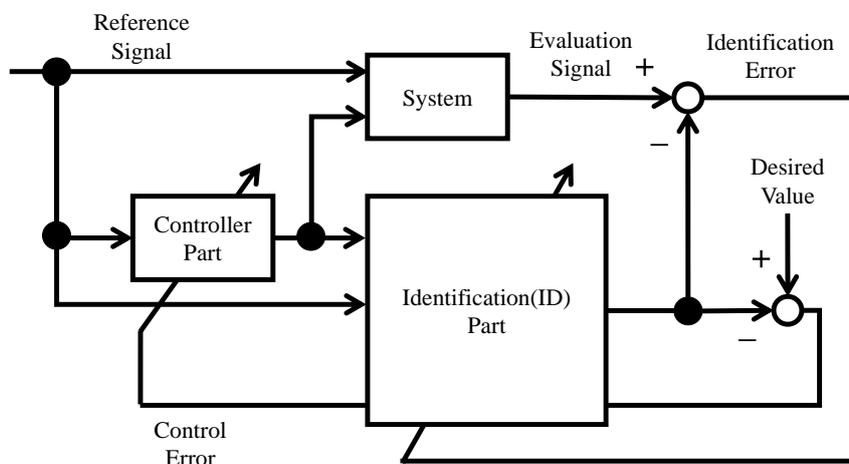


Figure 2. Block diagram of the control system using neural networks

where  $k$  is discrete time step,  $W_i^C$  is weight of the controller part,  $P_R(k)$  is a signal from a reference microphone, and  $N$  is the number of time steps for input signals.  $y(k)$  is equivalent to  $Q_s$  in equation (4). Similarly, the output of the identification part defined by  $\tilde{P}$  was calculated as

$$\tilde{P}(k+1) = \sum_{i=0}^{N-1} [W_i^{S1} \cdot y(k-i)] + \sum_{i=0}^{N-1} [W_i^{S2} \cdot P_R(k-i)] \tag{6}$$

where  $W_i^{S1}$  and  $W_i^{S2}$  are weights of the controller part. Each part acts as an FIR filter.

$\tilde{P}$  must be the same value as an evaluation signal denoted by  $P$ . In the back propagation method, all of the errors are calculated by using output of the whole network. The identification error was determined by the difference between the output from the identification part and an evaluation signal written as  $\tilde{P} - P$ . Since all these signals may have both positive and negative values, the evaluation function for identification is defined by using the squared value as

$$E_s(k) = \frac{1}{2} [\tilde{P}(k) - P(k)]^2, \tag{7}$$

and the weights of the identification part are modified so as to decrease  $E_s(k)$  based on the back propagation method. The update rule for identification part is given by

$$W_i^S(k+1) = W_i^S(k) - \eta_s \frac{\partial E_s}{\partial W_i^S}(k) = W_i^S(k) - \eta_s \cdot [\tilde{P}(k) - P(k)] \cdot \frac{\partial \tilde{P}}{\partial W_i^S} \tag{8}$$

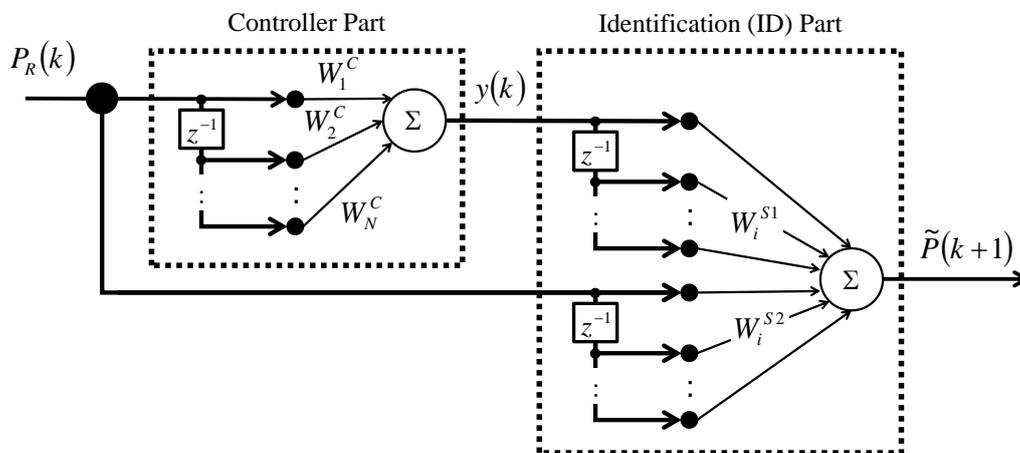
where  $\eta_s$  is learning rate for the identification part. This learning rate has the same mathematical meaning as step size parameter for the steepest gradient method. Similarly, the control error was determined by the difference between the output from the identification part and the target value written as  $\tilde{P} - 0$ , so the evaluation function for the controller is defined by using the squared value as

$$E_c(k) = \frac{1}{2} [\tilde{P}(k) - 0]^2 = \frac{1}{2} \tilde{P}(k)^2 \tag{9}$$

and the weights of the controller part are modified so as to decrease  $E_c(k)$ . The update rule for the controller part is given by

$$W_i^C(k+1) = W_i^C(k) - \eta_c \frac{\partial E_c}{\partial W_i^C}(k) = W_i^C(k) - \eta_c \cdot \tilde{P}(k) \cdot W_0^{S1}(k) \cdot P_R(k-i) \tag{10}$$

where  $\eta_c$  is learning rate for the controller part. This formula includes the particular weight  $W_0^{S1}$ . When this weight becomes zero, the control error is never propagated from the identification part to the



**Figure 3.** Block diagram of the proposed neural network control system

controller part. Thus, the specified weight should not keep zero while controlling. Each learning rate is determined by trial and error method.

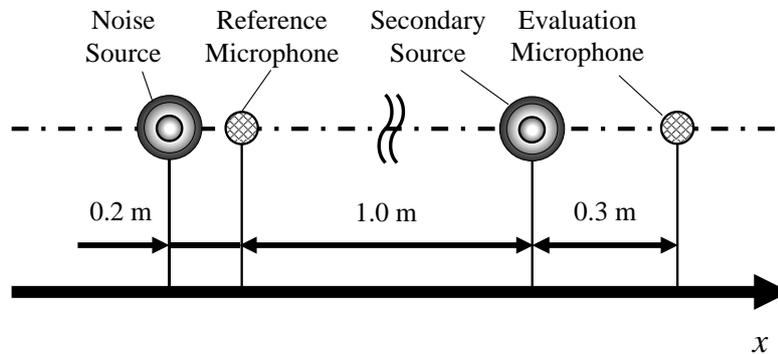
### 3.2. Dropout

We applied “dropout” technique to the training of neural networks to improve the noise control performance for a moving evaluation point. Dropout is an efficient technique to train multi-layered neural networks. In dropout, each unit is temporarily and randomly dropped out during training to realize the effect of averaging the characteristics of all these scale-downed networks. It prevents overfitting to training data and gives neural networks the generalization ability to adapt unknown data. In noise control for a moving evaluation point, it may move at one time and stop at another time. If neural networks overfit to the moving state, they cannot adapt to the stopping state. The problem can be solved by dropout. Another reason to use dropout is to preferentially train an important unit which connects the controller part to the identification part because it is only unit to propagate the control error from the identification part to the control part by the back propagation method.

Dropout can be applied to every layer of neural networks. For example, with dropout in the input layer of the identification part, the forward operation becomes

$$\tilde{P}(k+1) = \sum_{i=0}^{N-1} [W_i^{S1} \cdot r_i^{S1} \cdot y(k-i)] + \sum_{i=0}^{N-1} [W_i^{S2} \cdot r_i^{S2} \cdot P_R(k-i)] \quad (11)$$

where  $r_i^S$  has the probability of being 1.  $r_i^S$  is zero when the corresponding unit was dropped out. The update formula of these weights is the same as that in the case without dropout, then the weights of dropped-out units were not trained. As mentioned above, the particular weight  $W_0^{S1}$  is important when modifying the control weights based on the back propagation method. In the method, the errors derived from the output of the network are propagated to former units by multiplying weights of related units. Normally, the corresponding unit to  $W_0^{S1}$  is sometimes dropped out, and then the



**Figure 4.** Arrangement of sound sources and fixed microphones

**Table 1** Simulation conditions of noise control for a fixed evaluation point

Temperature [°C]	14.8
Sound speed [m/s]	341
Sampling frequency [Hz]	5000
Learning rate of controller part	0.01
Learning rate of identification part	0.01
Dropout probability of controller part	1.0 (without dropout)
Dropout probability of identification part	1.0 (without dropout)
The number of past inputs [steps]	25
Initial value of weights	random[-0.001~0.001]
Waveform of noise	Sinusoidal
Frequency of noise [Hz]	200, 300, and 400

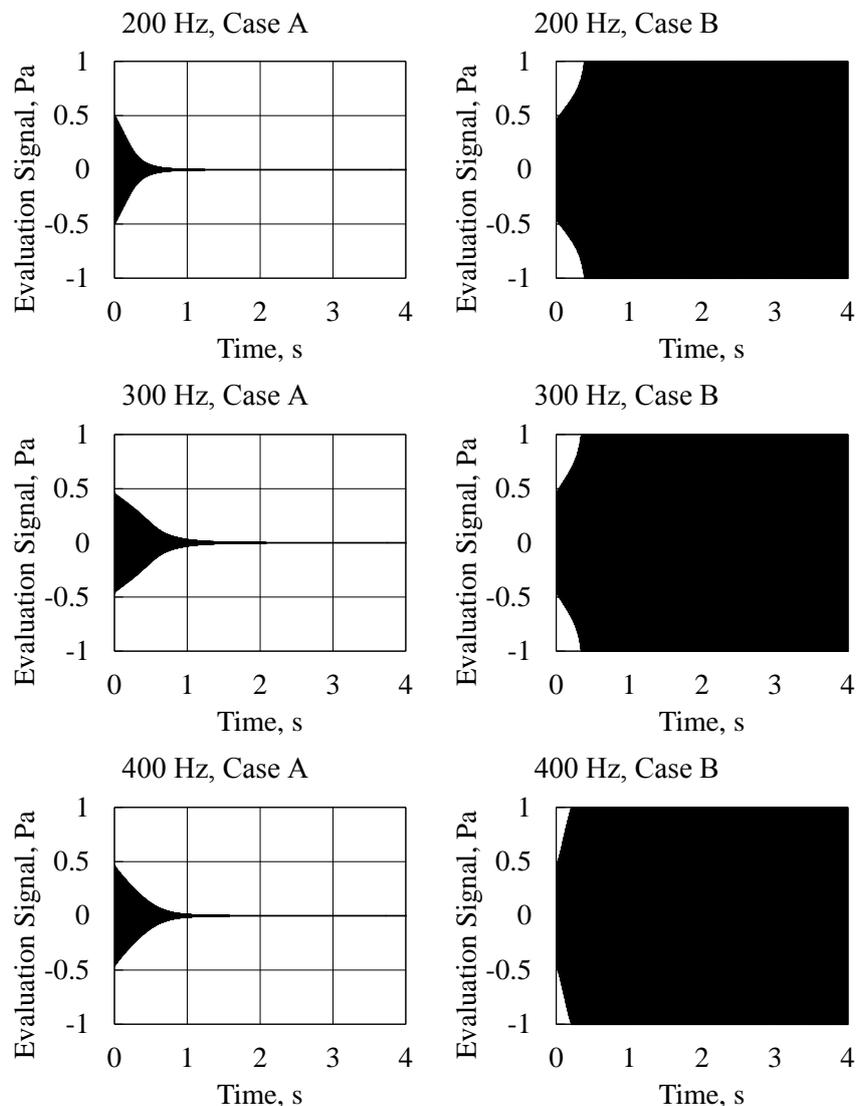
controller cannot learn in this time. To avoid this problem, exceptionally, this special unit must not be dropped out, which means  $r_0^{S1}$  is always 1.

#### 4. Noise control with neural networks for a fixed evaluation point

##### 4.1. Numerical simulations

The arrangement of the experimental setup is shown in figure 4. A noise source, a reference microphone, a secondary source, and an evaluation microphone were lined up on a straight line in a free sound field. In this section, all the objects were set on a fixed point.

Table 1 shows the simulation conditions. Sampling frequency of the control system was determined according to that of noise. In general, the sampling frequency needs ten times as much as the frequency of noise. The learning rate plays an important role in training. Smaller the value is, slower the convergence is. However, the system may be uncontrollable when the value is too big. Unfortunately, the learning rate of each part cannot be determined automatically. The value was given by a trial and error method in the simulation. In this part, units were never dropped out, therefore the dropout probability of each part was set to 1. The effects of dropout are discussed in the section 6. In the control,



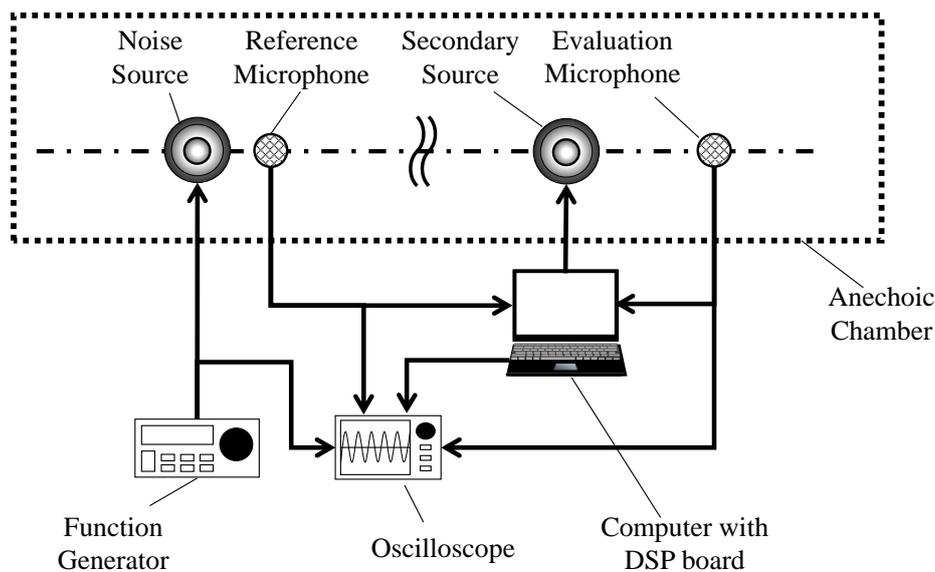
**Figure 5.** Simulation results of noise control for a fixed evaluation point. Case A indicates the better results than Case B.

each weight was randomly set to a small value in the initial state in order to reduce the dependency of initial values. For the sake of simplicity, a sinusoidal wave with constant frequency was used as target noise. The noise frequency was much lower than the sampling frequency. Any external disturbances were ignored in the simulations.

Figure 5 shows the simulation results of noise control for a fixed evaluation point. The control performances, which means the evaluation signals, depends on the initial value of the weights of the networks. There were two major cases observed in the simulation, one of which was that the signal converged (Case A) and the other of which was that the signal immediately diverged (Case B). In Case A, the convergence speed did not depend on the target frequency so much. There were 20 dB attenuations of the noise in all the frequencies. In Case B, the signals were soon diverges and good control performance could never be obtained.

#### 4.2. Experiments

In order to verify the validation of the simulation, the corresponding experiments for noise control were conducted. A schematic diagram of the experimental setup is shown in figure 6. In order to reduce the effect of external noise and any reflections of sound, speakers and microphones were set in an anechoic chamber. A function generator gave a noise signal which was sinusoidal wave with constant amplitude



**Figure 6.** Schematic diagram of the experimental setup for noise control

**Table 2.** Experimental conditions of noise control for a fixed evaluation point

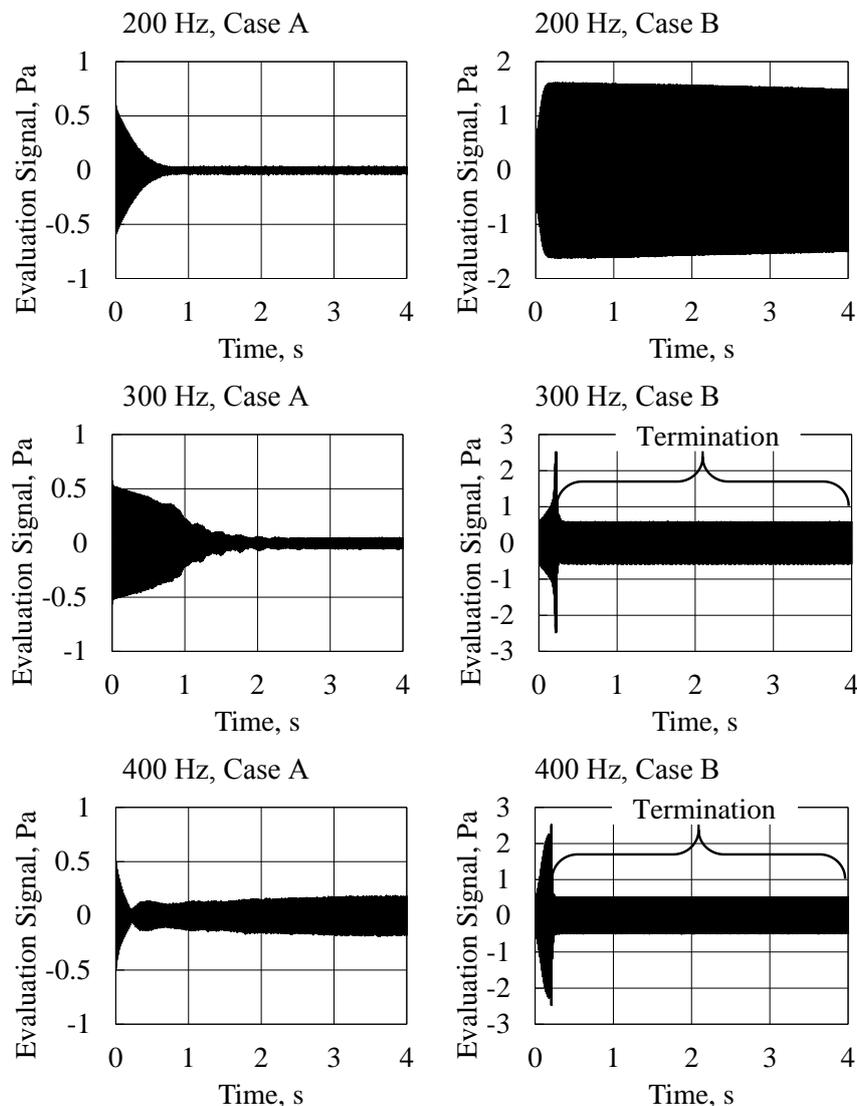
Temperature [°C]	14.8
Sound speed [m/s]	341
Sampling frequency [Hz]	5000
Learning rate of controller part	1.0
Learning rate of identification part	1.0
Dropout probability of controller part	1.0 (without dropout)
Dropout probability of identification part	1.0 (without dropout)
The number of past inputs [steps]	25
Initial value of weights	random[-0.001~0.001]
Waveform of noise	Sinusoidal
Frequency of noise [Hz]	200, 300, and 400

and single frequency to one of the speakers. For real-time control, a digital signal processor (DSP) was used. The DSP received voltage signals from these microphones and then modified the weights of the neural network controller. Output of the DSP was fed to the other speaker to cancel the noise. All of the wave signals were observed with an oscilloscope. Table 2 shows the experimental conditions. Almost all of the experimental conditions were the same as those of the simulations.

Figure 7 shows the experimental results for the fixed evaluation point. Similarly to the simulations, the control performance depended on the initial values of the weights. These were classified into two cases; convergence one (Case A) and divergence one (Case B). The results of each case were obtained at least 3 times.

The tendency of the experimental results were same as that of the simulation results except for 400 Hz in Case A, the reason of which is under consideration. In Case A, the evaluation signals in all the results were reduced by at least 6 dB attenuations. The control performance of other frequencies such as 100 Hz and 500 Hz after the convergence was similar to that of 200 Hz and 300 Hz. Another problem is the variation of convergence speed depending on the frequency of noise.

In Case B, the signal got bigger and diverged immediately. Not to damage the DSP by the high voltage signal from microphone amplifiers, it executed forced termination command after the

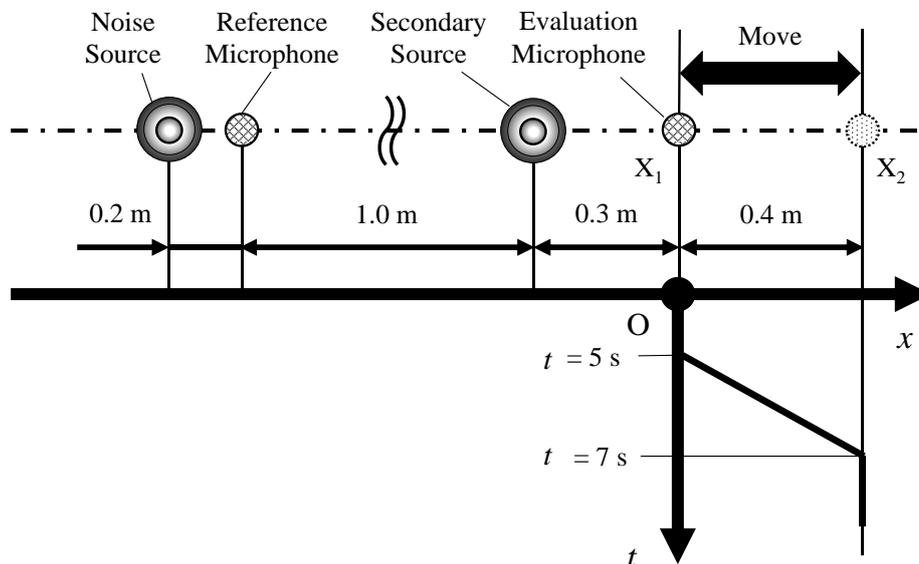


**Figure 7.** Experimental results of noise control for a fixed evaluation point Case A indicates the better results than Case B.

divergence and kept stopping. For this reason, some results looked as if controller did not work after the divergence in some failure cases. When the frequency of noise was 200 Hz, the divergence stopped and the evaluation signal kept almost constant and high amplitude. The cause of this may be convergence of the weight  $W_0^{S1}$  around zero. This weight is a key to train the controller part of the neural network control system. If the weight and the identification error are zero, the control error is not propagated well to the controller part and then weights of the controller parts are hardly modified. In this case, the output of the DSP does not changes unless input to the DSP changes.

**5. Noise control simulation with neural networks for a moving evaluation point**

In this section, the control performance for a moving evaluation point was verified by simulations. Figure 8 shows the arrangement of sound sources and microphones, and the movement path of the evaluation microphone. The initial arrangement of these objects was the same as the simulatins above. The evaluation microphone moved on the straight line with a constant speed. At first, the evaluation microphone was set at  $X_1$  point and then started to move 5 seconds after the control start. When it arrived at  $X_2$  point, it stopped at the  $X_2$  point and kept the position.



**Figure 8.** Arrangement of sound sources and microphones, and the movement path of an evaluation microphone

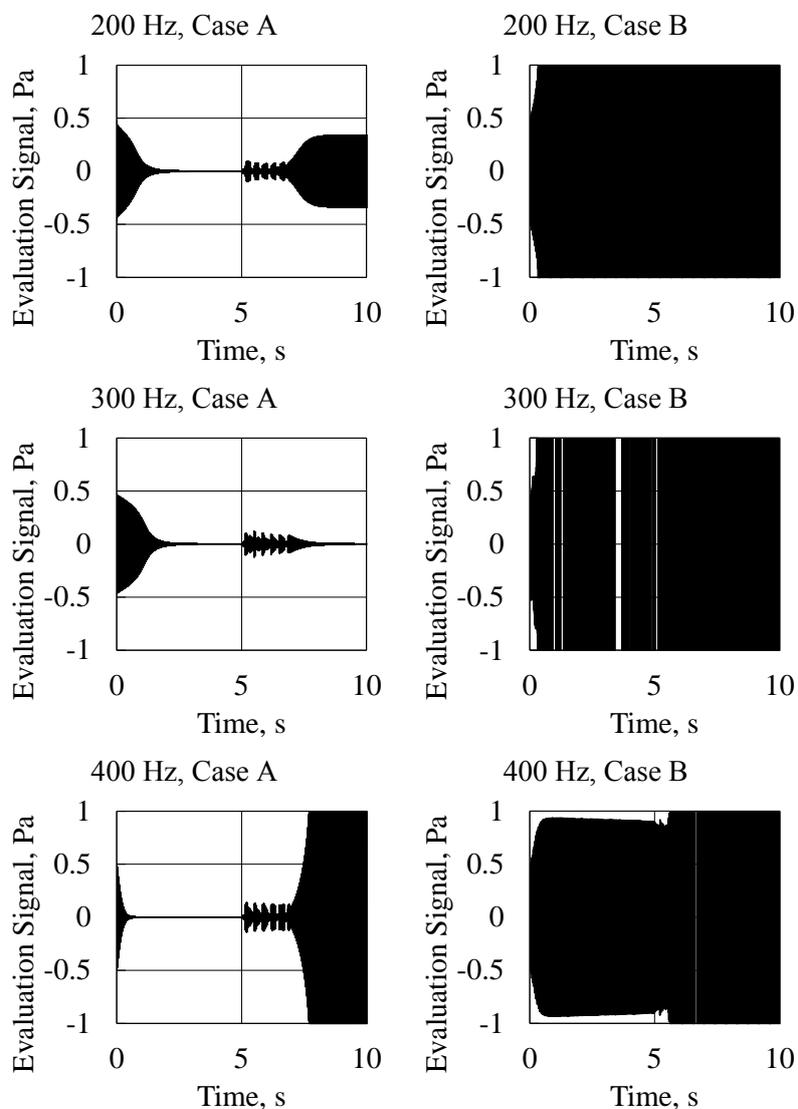
**Table 3.** Simulation conditions of noise control for a moving evaluation point

Temperature [°C]	14.8
Sound speed [m/s]	341
Sampling frequency [Hz]	5000
Learning rate of controller part	0.01
Learning rate of identification part	0.01
Dropout probability of controller part	1.0 (without dropout)
Dropout probability of identification part	1.0 or 0.9
The number of past inputs [steps]	25
Initial value of weights	random[-0.001~0.001]
Waveform of noise	Sinusoidal
Frequency of noise [Hz]	200, 300, and 400
Movement speed of an evaluation microphone [m/s]	0.2 m/s

The other conditions are shown in table 3. To verify the effects of dropout, the dropout probability was set to the networks. Since the controller part was required to connect to the identification part as mentioned in the section 2, dropout was not applied to the connecting point. The dropout probability of each part was also given by a trial and error method.

The control performance of the system without dropout is shown in figure 9. Similarly to the simulations above, the evaluation signal may decrease or increase depending on the initial values of the weight. In Case A, good control performance was obtained while the evaluation microphone was moving at each frequency. There were approximately 10 dB attenuations during the movement. Especially, at 300 Hz, the signal kept good control performance even after the evaluation microphone stopped. However, at the other frequencies, the signal started to diverge immediately after the state transited from the moving to the stopping. In Case B, the signal diverged immediately after the state transited from the initial to the control.

The performance of the control system with dropout for the identification part is shown in figure 10. In Case A, the divergence of the signals which appeared in the control without dropout immediately after the state transited from the moving to the stopping was prevented. However, the convergence speed with dropout was slower than that without dropout. That may be why the number



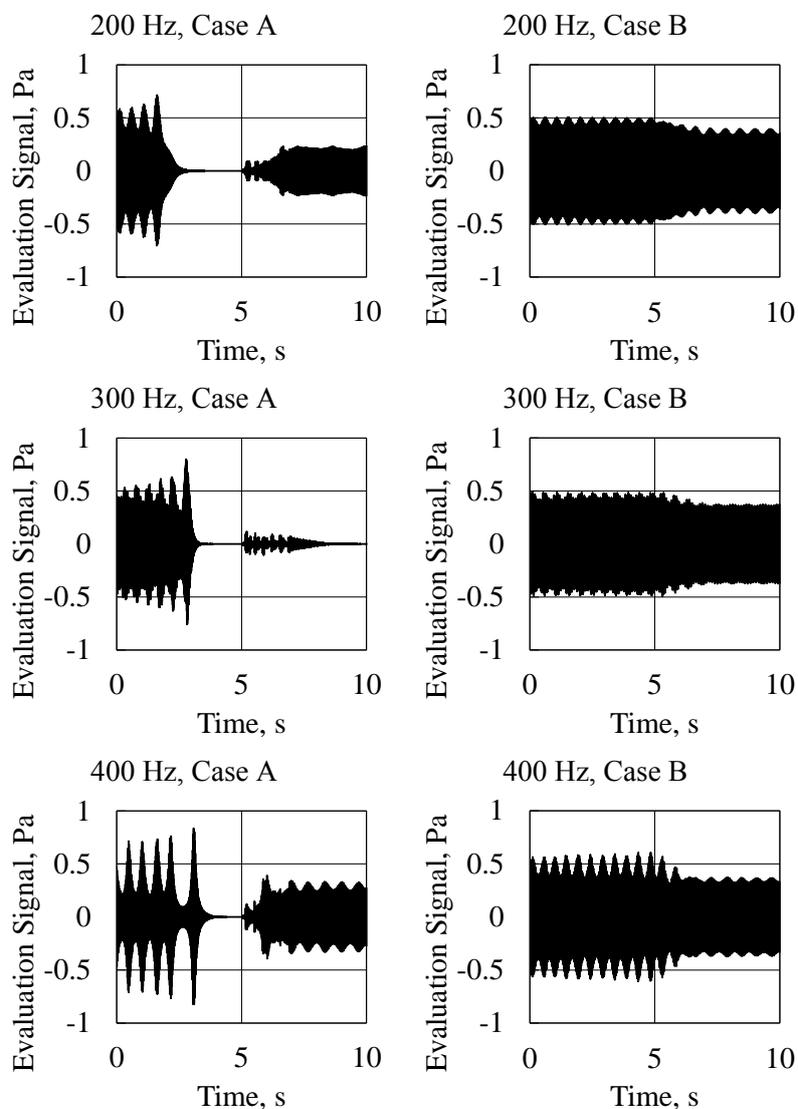
**Figure 9.** Simulation results for a moving evaluation point without dropout.

Case A indicates the better results than Case B.

The evaluation point started to move at 5 s and stopped at 7s.

of times of modifying the weights decreased due to the temporal and random dropout of the units during the training. In Case B, the divergence of the signals which appeared in the control without dropout immediately after the state transitioned from the initial to the control was prevented. These indicate that the use of dropout in noise control systems using neural networks can prevent the signal divergence, which is one of the most important factors for noise control, even if an evaluation point moves. That may be because the overfitting was suppressed by temporarily and randomly dropping units out in the learning steps.

One of the advantages of neural network is that the identification in advance is not required. In the filtered-x LMS algorithm and the virtual error method, the identification must be completed before the control. Therefore, the evaluation signal diverges when the evaluation moves in the filtered-x LMS algorithm [6]. Neural networks can not only require no preliminary identification but also prevent the signal divergence by using the dropout technique.



**Figure 10.** Simulation results for a moving evaluation point with dropout.

Case A indicates the better results than Case B.

The evaluation point started to move at 5 s and stopped at 7s.

## 6. Conclusions

In this paper, a neural network control system was proposed for noise control. The performance of the control without dropout for a fixed evaluation point was verified in simulations and experiments. Also, the performance of the control with and without dropout for a moving evaluation point was verified in simulations. The results to be obtained are as follows:

- (1) For a fixed evaluation point, the good performance of the control without dropout can be obtained although the evaluation signal divergently increases in some cases.
- (2) For a moving evaluation point, the good performance of the control without dropout can be obtained although the evaluation signal divergently increases in some cases.
- (3) The use of dropout in the noise control system using neural networks is very effective on preventing the divergence of the evaluation signal, which is one of the most important factors for noise control, even if the evaluation point moves.

## References

- [1] S J Elliott et al. 1987 The Active Minimization of Harmonic Enclosed Sound Fields, part III: Experimental Verification *Journal of Sound and Vibration* vol 117 No 1 pp35-58
- [2] J Guo and J Pan 2002 Active Control of a Moving Noise Source – Effect of Off Axis Source Position *Journal of Sound and Vibration* vol 251 No.3 457-475
- [3] Khono et al. 2003 Adaptive Active Noise Control Algorithm without Explicit Identification of Secondary Path Dynamics *The Institute of Electronics, Information and Communication Engineers Series A* vol J-86 No.1 pp9-18
- [4] Ohno et al. 2012 Active Noise Control for a Moving Evaluation Point Using Step Size Vector *Transaction of the Japan Society of Mechanical Engineers Series C* vol 78 No.789 pp1670-1679
- [5] Fujii et al. 1990 Development of Self-Organizing Neural-Net-Controller System and Its Application to Underwater Vehicles, *Transactions of the Society of Naval Architects of Japan* vol. 168 pp275 281
- [6] Komatsuzaki et al. 2006 Free-Space Active Noise Control of Moving Evaluation Point *Transaction of the Japan Society of Mechanical Engineers Series C* vol 72 No.718 pp1723-1729
- [7] Nitish et al. 2014 Dropout: A Simple Way to Prevent Neural Networks from Overfitting *Journal of Machine Learning Research* 15 pp1929-1958
- [8] D E Rumelhalt et.al 1986 Learning representations by back-propagating errors *Nature* 323 pp318-362